Tivoli® Tivoli Provisioning Manager for OS Deployment
Version 7.1.1.3

User's Guide

IBM

Tivoli® Tivoli Provisioning Manager for OS Deployment
Version 7.1.1.3

User's Guide

IBM

Contents

Chapter 1. Getting started 1

Product overview 1
Components 1
Setting up a system profile by unattended setup . . 4
Setting up a system profile by cloning 4
Network boot 4
The deployment process 6
Universal images 7
Shared repository 7

Chapter 2. Working with targets 9

Adding targets 10
Detecting targets automatically 10
Adding targets to the Target Monitor manually 11
Importing and exporting targets lists 11
Configuring new targets 13
Changing the default administrative group . . . 13
Requirements for VMware targets 14
Injecting drivers on WinPE 3.0 to deploy Windows XP guests 15
Booting pSeries targets on the OS deployment server 16
Booting CellBlades targets on the OS deployment server 17
Booting SPARC targets on the OS deployment server 17
Organizing targets 18
Configuring targets 20
Configuring multiple targets. 20
Configuring targets for fully unattended OS deployments 20
Setting partition sizes on the targets 21

Chapter 3. Working with Windows operating systems on x86 and x86-64 targets 23

Overview of WinPE deployment engines 23
Windows Automated Installation Kit 24
Creating a WinPE 3.0 deployment engine . . . 25
Editing the information of a WinPE deployment engine 27
Adding matching target models to a WinPE deployment engine 27
Binding drivers to a WinPE deployment engine 28
System profiles 31
BitLocker compatibility 32
Creating system profiles 32
Organizing and editing system profiles 41
Updating a system profile with a Language Pack or a HotFix 41
Browsing partition files 42
Changing the partition layout 42
Updating device mapping 43
OS configurations and fixed common parameters 44
Binding drivers to a Windows system profile . . 46
Restoring a system profile manually 49
Software modules for Windows operating systems 50
WinPE and its uses 51
Creating software modules 52
Editing software modules 64
Keeping command lines confidential 64
Keyword substitution 65
Customizing the software page 66
OS configuration and software bindings 67
Scheduling the application of software modules for Windows operating systems 70
Working with hardware configurations 71
Setting up your environment 72
Hardware configuration objects and tasks . . . 73
RAID and Fiber Channel hardware capture. . . 74
Creating a hardware environment 74
Creating a hardware configuration object . . . 81
Creating a hardware capture configuration . . . 81
Capturing hardware information using templates 82
Capturing hardware information once 82
Task templates 83
Customizing a screen layout. 84
Creating and editing deployment schemes . . . 84
Creating media for deployment for Windows operating systems 89
Creating an OS deployment USB drive with the wizard 89
Creating an OS deployment USB drive with command lines 90
Creating OS deployment CD and DVD 92
Deploying Windows operating systems 94
The deployment process 94
Deployment requirements 95
Starting a one-time deployment 96
Deploying a hardware configuration 98
Redeploying 99
Monitoring deployments 106
Bindings created during deployment 110

Chapter 4. Working with Linux operating systems on x86 and x86-64 targets 111

System profiles 111
Creating system profiles 111
Organizing and editing system profiles 113
Browsing partition files 114
Changing the partition layout 114
Updating device mapping 116
OS configurations and fixed common parameters 116
Software modules for Linux operating systems . . 118
Creating software modules 118
Editing software modules 123
Keeping command lines confidential 123
Keyword substitution 124

© Copyright IBM Corp. 2006, 2010

Customizing the software page 125
OS configuration and software bindings . . . 126
Scheduling the application of software modules for Linux operating systems 129
Working with hardware configurations 130
Setting up your environment 131
Hardware configuration objects and tasks . . . 132
RAID and Fiber Channel hardware capture . . 133
Creating a hardware environment 133
Creating a hardware configuration object . . . 140
Creating a hardware capture configuration . . 140
Capturing hardware information using templates 141
Capturing hardware information once 141
Task templates 142
Customizing a screen layout 143
Creating and editing deployment schemes. . . 143
Creating media for deployment for Linux operating systems 148
Creating an OS deployment USB drive with the wizard 148
Creating an OS deployment USB drive with command lines 150
Creating OS deployment CD and DVD 151
Deploying Linux 153
The deployment process 153
Deployment requirements 154
Starting a one-time deployment 155
Deploying a hardware configuration 157
Redeploying 157
Monitoring deployments 165
Bindings created during deployment 168

Chapter 5. Working with Linux on PowerPC and Cell targets 169
System profiles 169
Creating an unattended setup system profile for Linux on PowerPC 170
Organizing and editing system profiles 170
Changing the partition layout 171
Updating device mapping 172
OS configurations and fixed common parameters 172
Software modules for Linux operating systems on PowerPC 174
Creating software modules 175
Editing software modules 177
Keeping command lines confidential 178
Keyword substitution 178
Customizing the software page 180
OS configuration and software bindings . . . 181
Scheduling the application of software modules 184
Task templates 184
Creating and editing deployment schemes. . . 185
Deploying Linux on PowerPC 190
Deployment requirements 190
Starting a one-time deployment 191
Monitoring deployments 192

Chapter 6. Working with Solaris on SPARC targets 197
System profiles 197
Creating an unattended setup system profile for Solaris operating system 198
Creating a system profile from a Solaris Flash archive 198
Organizing and editing system profiles 199
Updating device mapping 200
OS configurations and fixed common parameters 200
Software modules for Solaris operating systems 202
Creating software modules 202
Editing software modules 204
Keeping command lines confidential 205
Keyword substitution 205
Customizing the software page 207
OS configuration and software bindings . . . 208
Scheduling the application of software modules 211
Task templates 211
Creating and editing deployment schemes. . . 212
Deploying Solaris 217
Deployment requirements 218
Starting a one-time deployment 218
Monitoring deployments 220

Chapter 7. Working with AIX on PowerPC targets 225
System profiles 225
Creating an unattended setup system profile for AIX operating system 225
Organizing and editing system profiles 227
Updating device mapping 227
OS configurations and fixed common parameters 228
Task templates 228
Creating and editing deployment schemes. . . 229
Deploying AIX 234
Deployment requirements 234
Starting a one-time deployment 235
Monitoring deployments 236

Chapter 8. Working with VMWare ESX Server on x86 and x86-64 targets . . . 241
System profiles 241
Creating an unattended setup system profile for VMWare 241
Organizing and editing system profiles 242
Browsing partition files 242
Changing the partition layout 243
Updating device mapping 244
OS configurations and fixed common parameters 245
Task templates 246
Customizing a screen layout 247
Creating and editing deployment schemes. . . 247
Creating media for deployment for VMWare . . . 252
Creating an OS deployment USB drive with the wizard 252

Creating an OS deployment USB drive with command lines 254
Creating OS deployment CD and DVD 255
Deploying VMWare 257
The deployment process 257
Deployment requirements 258
Starting a one-time deployment 259
Monitoring deployments 260
Bindings created during deployment 264

Chapter 9. Multiple server architecture 265
Server roles 265
OS deployment server replication 267
Replicating OS deployment servers with a schedule 270
Replicating an OS deployment server once manually 270
Replicating offline with the web interface extension 271
Replicating one time in command line 273
Server replication status and logs 274
Switching from an ODBC to a JDBC gateway 275
Removing an OS deployment server from the hierarchy 275

Chapter 10. Security 277
Security roles and access to the Web interface . . 277
Creating an HTTP authentication domain . . . 278
Creating security roles 278
Backups of server files 279
Importing and exporting RAD files 279
Importing and exporting targets lists 281
Exporting and loading configurations 282
Fault tolerance 283
Fault tolerance at the DHCP level 283
Fault tolerance at the Tivoli Provisioning Manager for OS Deployment level 284
Network security constraints 285
Avoiding new security breaches 286
Rogue PXE servers 286
Unwanted target computers 287
Security issues and the web interface 287

Chapter 11. Booting targets without using PXE 289
Creating network boot USB drive with the wizard 289
Creating a network boot CD or DVD with the wizard 290
Creating an original WinPE 3.0 network boot CD or DVD with the wizard. 291
Using a network boot CD 292
Creating a network boot USB drive with command lines 293
Creating a network boot CD or DVD with command lines 295
Booting on the network when the target is missing network drivers 296

Chapter 12. Tools. 299
Erasing hard disk content 299
Software snapshots 299
Limitation of the technology 299
Restoring software snapshots 300

Chapter 13. Migrating users 301
Capturing user settings 301
Restoring user settings 301

Chapter 14. Reference 303
Keyboard navigation on the target 303
The web interface 303
Server status 304
Server parameters 305
Server history 305
OS deployment 306
Server log files 306
Database schema 308
Database schema overview 308
Bill of Material-related tables 308
OS configuration-related tables 309
OS deployment server configuration. 310
Global parameters 310
HTTP Console Security parameters 316
Authentication domains 316
TCP tunnels 317
Creating a new hardware rule 318
Default port numbers 319
Configuring OS deployment server with a text file 320
Parameter reference 327
OS deployment object parameters 350
Target parameters and details 350
OS configuration parameters 354
Hardware compatibility list. 357
Java API 357
Getting started with Java API 358
Configuring the OS deployment server to use the Java API. 358
Examples. 360
Deployment server configuration and maintenance. 367
Deployment server tasks 372
Events. 377
Controlling API traces 378
Command-line interface 379
NetClnt 379
RbAgent 386
rembo command reference 390
dbgw command reference 391
Glossary 392
Notices 397

Chapter 1. Getting started

This section provides basic information about the main deployment features and a quick overview of the product.

Product overview

The product is a database-driven, network-based deployment solution. Using an easy-to-use interface, the product provides Windows® cloning and unattended setup, Linux® cloning, and unattended setup, Solaris cloning and unattended setup, AIX® unattended setup, and VMWare ESX unattended setup, from Windows, Linux, Solaris, and IBM® AIX servers.

Using industry standards such as Wake on LAN and vPro, PXE and OpenBOOT, ODBC and JDBC, DMI and PCI, Microsoft® system preparation tool (Sysprep), Kickstart, Autoyast, Jumpstart and NIM, the product provides ready to use installation of operating systems and selected software on tens, or even hundreds, of computers simultaneously.

The deployment source can be on the network (with either unicast or multicast downloading), on a CD or DVD, or on a disk partition.

Components

The product is made of server-side components, installed using a typical setup program, of web interface components, and of target-side applications, automatically installed by the target through the network.

The server-side components include:

- The OS deployment server, running in the background as a service. This component provides the PXE remote-boot capability and multicast file access to the remote-boot targets.
- A deployment database, accessed through the standard ODBC/JDBC interface, and through the product TCP-to-ODBC/JDBC Gateway service. This database stores the Bill of Material for every target, including information about what software modules must be installed.
- The web interface, a web-based application used to prepare and supervise deployments. Through the web interface, you can configure the OS deployment server, manage objects used during a deployment, and create recovery CDs and DVDs. You can also visualize targets into their different groups. The parameters and status of each target can be controlled (for example, when a target requests a remote-boot through PXE). The

The web interface side application, typically installed on the server, is the web interface extension, running in background as a service. Optionally, a web interface extension can be installed on targets, so that you can take control of them remotely. The web interface extension is useful, for example, to restart the computer when a deployment starts, or to perform an operating system inventory. Several features of the web interface are not available when the web interface extension is not installed on the server. The web interface extension is a piece of software designed to run as an operating system application (a Windows application or a UNIX® application).

© Copyright IBM Corp. 2006, 2010

There are two target-side tools:

Deployment engine

The deployment engine is a stand-alone mini-operating system used for all management tasks to be performed outside of the real operating system, such as operating system installation. This deployment engine never has to be installed manually on target systems, because it is downloaded automatically when target systems are instructed to boot on the OS deployment server in their DHCP configuration. After the deployment engine is started, it loads a ramdisk which can perform any kind of operations on the target, such as

- partition and format disks
- install a complete operating system such as Windows or UNIX
- wipe the disks

The deployment engine is a stand-alone management platform and does not require an operating system.

web interface extension

The *web interface extension* is a small program that can run on most operating systems (Windows, AIX, Linux , Solaris) either in the background as a service or as a command-line tool. It provides cross-operating system connectivity to the management platform for targets with an installed operating system. It can be used to collect information from targets, to remotely initiate tasks on a target from the OS deployment server, or to trigger commands on the OS deployment server from a target.

The web interface extension can access all files on the hard disk and other drives through the operating system. When running under Microsoft Windows, it can also access the system registry, and all system information published by drivers using the Windows Management Interface (WMI). In addition, for all Intel-based operating systems, the Web interface extension can install a hook on the boot process to force a PXE boot or to start the deployment engine in offline mode, to trigger pre-operating system management tasks such as migration or recovery.

The deployment engine and the web interface extension are stand-alone management platforms and do not require an operating system.

When you install the product you are installing the OS deployment server. The target side is embedded into the server, and sent to computers through a network connection such as a *network boot* or *PXE boot*, or through media such as a *CD*, a *diskette*, or a *hard disk partition*.

Note: It is also possible to boot targets over the network without using PXE.

The installation process is exemplified in Figure 1 on page 3.

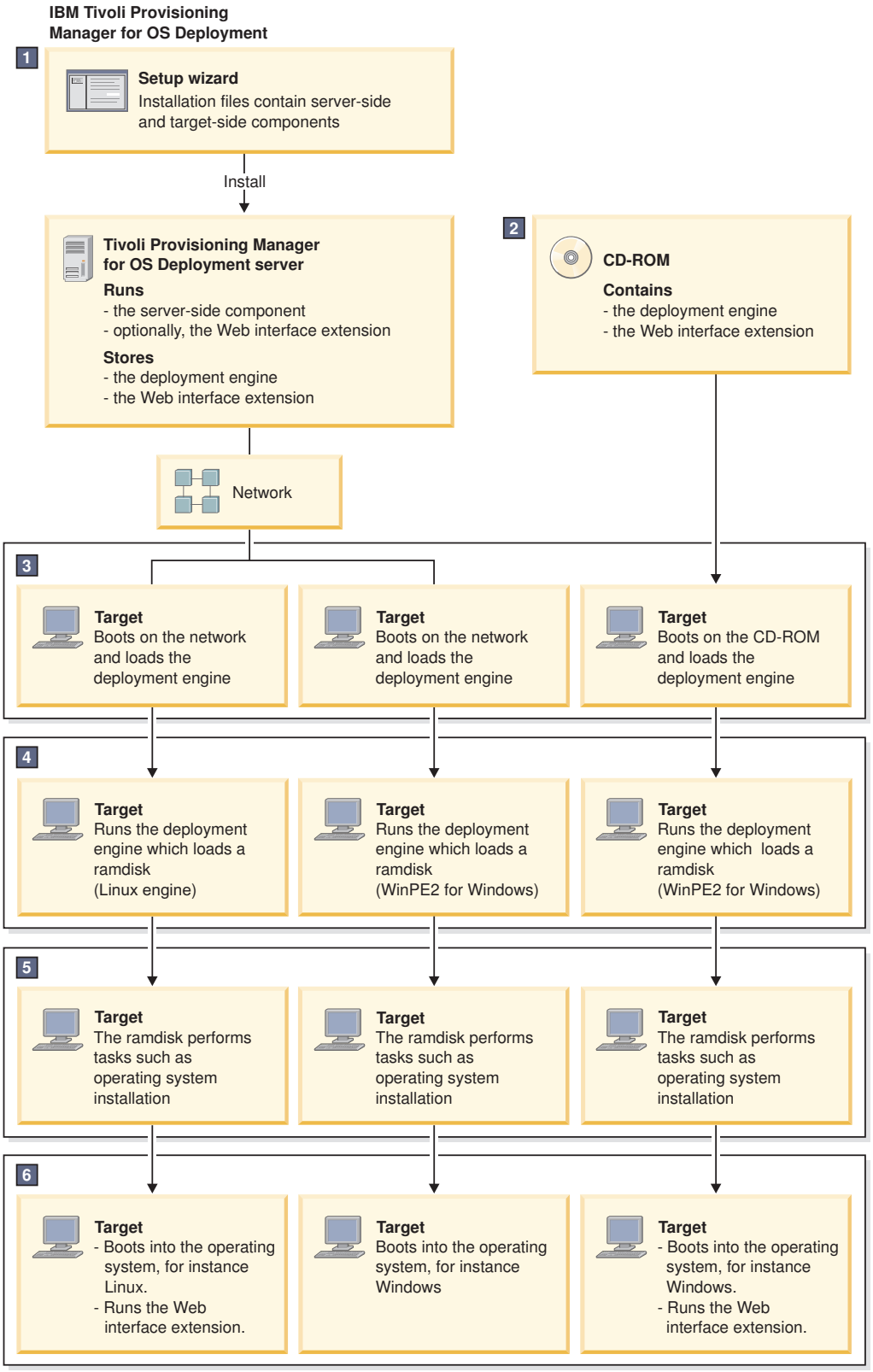


Figure 1. Tivoli Provisioning Manager for OS Deployment components

1. The installer, containing server-side and target-side components, installs the server-side and potentially the Web interface extension on the server computer.

2. A CD can then be created containing both the deployment engine and the web interface extension.
3. Computers then boot, loading the deployment engine either from the PXE server (OS deployment server), or from the CD.
4. The deployment engine loads a ramdisk, Linux deployment engine, WinPE for Windows.
5. The ramdisk can now perform all kinds of tasks, including operating system installation.
6. The targets boot into the operating system (Windows, Linux, Solaris,...) , with the web interface extension optionally running.

Setting up a system profile by unattended setup

Use unattended setup to set up a system profile without actively being involved when the process occurs.

Setting up an automated system profile by unattended setup typically involves the following steps:

- Installing the OS deployment server on a computer
- Creating a system profile by providing the installation CDs and choosing installation options
- Creating software modules for most commonly used business applications and other software, using the web interface
- Optionally, designing automatic binding rules and fixed configuration parameters for fully unattended deployment on unknown targets

Setting up a system profile by cloning

A cloning-mode system profile requires you to configure a target, prepare it for cloning, and run the cloning process directly on the computer. However, the native mode of operation of Tivoli® Provisioning Manager for OS Deployment is centered around cloning-mode system profiles, because this method of deployment is faster than unattended installation.

Setting up an automated deployment system by cloning typically involves the following steps:

- Installing the OS deployment server on a computer
- Installing a clean operating system on a target , and running the appropriate system preparation tool if needed
- Restarting the target with network-boot enabled. Instead of its own operating system, the Tivoli Provisioning Manager for OS Deployment kernel must start on the target (remote-boot through PXE).
- Creating a *system profile* and *OS configuration* with the clean operating system
- Creating software modules for most commonly used business applications and other software, using the web interface
- Optionally, designing automatic binding rules and fixed OS configurations parameters for fully unattended deployment on unknown targets

Network boot

Tivoli Provisioning Manager for OS Deployment implements a PXE network boot application. PXE is a type of DHCP extension. It is supported by the vast majority of hardware vendors, including IBM, Lenovo, HP and DELL.

Note: This topic describes the process for Intel®/AMD x86 computers, but a very similar process is applicable for recent 64-bit computers based on the UEFI standard, and SUN SPARC and IBM PowerPC® machines based on the OpenBoot standard.

A network boot application is different from any other program for a number of reasons:

- It is invoked by the BIOS during the initial bootstrap of the computer, before any operating system or disk boot manager.
- It depends on the presence of a special chip on the network card, the *PXE boot ROM*.
- It is capable of communicating with a server over the network thanks to a programming interface provided by the bootrom. PXE is the informal standard for this kind of programming interface.
- It is downloaded from a special OS deployment server, and does not depend on any local storage device. It can work on computers without hard disk, CD or diskette devices. If a local storage device is present on the computer, it is accessible to the network boot application. However, a network boot application does not fail if the storage device is corrupted or broken.
- It gets its parameters (IP address and other boot parameters) from a central DHCP server.

A typical network-boot sequence goes through the following phases:

1. *Starting up*: A user or a wake-up event starts the remote-boot computer up.
2. *Network boot*: The BIOS configuration (boot order), a hot key (for example, F12), or the wake-up event instructs the computer to boot on the network.
3. *IP address discovery*: The remote-boot target broadcasts a DHCP request for an IP address. Any DHCP server which knows the target (that is, recognizes its hardware address) or which has a pool of freely distributable dynamic addresses sends an IP address. The target takes the first answer and confirms it to the server. In addition to the IP address, the server gives some other network parameters to the target and information about the boot procedure to follow.
4. *OS deployment server discovery*: In the case of PXE remote-boot, the target computer then proceeds to the discovery of the OS deployment server. The OS deployment server is responsible for delivering a network boot program to the target. It is not necessarily the same computer as the DHCP server. The target responds to the first OS deployment server which replies and downloads a small program using a simple multicast protocol (MTFTP).
5. *Server connection*: If the network boot program is the deployment engine, it establishes a secure (encrypted) connection to the OS deployment server and receives instructions from the server to determine the name of the program to run.
6. *Pre-OS configuration*: The deployment engine then downloads from the file server everything required by the OS configuration specified by the system administrator. These file transfers are done using a secure, robust and efficient unicast or multicast protocol. Many actions can be performed at this stage: installing an operating system from scratch, repairing a corrupted operating system, or performing an inventory.
7. *OS booting*: When instructed by the OS configuration, the deployment engine removes itself from memory and allows the computer start the operating system, as if the target is booting normally from the hard disk. This ensures full compatibility with the operating system and avoids all problems of the traditional diskless remote boot.

Note: Tivoli Provisioning Manager for OS Deployment features a fault-tolerant server architecture, with the OS deployment server always available to the target. However, in case of severe network failure, targets can be configured to work in offline mode, without network access. In this scenario, the deployment engine is stored on a permanent storage medium, such a hard disk partition or a CD, and started from that medium instead of being loaded from the network.

The deployment process

The deployment process has numerous steps, including hardware discovery, hard disk partitioning, and installing operating systems and other software modules.

A typical deployment proceeds as follow:

1. Start the target remotely, possibly using Wake on LAN or vPro. The target starts up on the network. It receives IP information from the DHCP server, and a bootstrap program from the OS deployment server that directs the next steps.
2. The target performs hardware discovery through DMI and PCI system calls.
3. The target queries the ODBC/JDBC database for all relevant information regarding the operating system and the software modules to be installed, and dynamically generates a deployment plan.
4. If specified in the database, it prompts the user (or the administrator on the web interface) for additional deployment information.
5. Depending on the planned deployment steps, the bootstrap can start WinPE 3.0 for Windows or Linux deployment engine in a ramsdisk to perform some of the configuration and installation tasks.
6. If specific hardware configuration software needs to be run early (such as RAID controller configuration), it is started. After a subsequent reboot, the target resumes the process.
7. The target partitions its hard disk according to the information retrieved from the database. A small bootstrap is also written to the hard disk, to be able to take over any subsequent reboot on the hard disk.
8. The target initiates a multicast transfer for all files needed during the deployment.
9. The content of the hard disk partitions is built by merging the base operating system image and all incremental images and other software updates. The data is then written onto the disk with optimal efficiency.
10. An answer file is dynamically generated, providing all necessary information for an unattended customization of the operating system. Unattended setup command lines for software modules are prepared.
11. The target computer boots on the hard disk, and completes operating system installation, then issues unattended setup command lines for the selected software modules. The target reboots and Tivoli Provisioning Manager for OS Deployment takes control again.
12. If other software modules have been configured to be installed after one (or more) reboot, they are installed and the operating system is started again.
13. If the computer is still connected to the network, it automatically updates the deployment status in the database.
14. When everything is done, the deployment process terminates by either turning off the target, starting up into the operating system, restarting the target, or displaying a green banner page.

Universal images

With Tivoli Provisioning Manager for OS Deployment, you can create universal images for Windows operating systems.

When creating a system profile by cloning, Tivoli Provisioning Manager for OS Deployment makes a copy of the state of the hard disk of a particular hardware. The system profile thus created corresponds to a particular hardware type on which the profile can be deployed. In theory, a different image is required for each hardware type. Switching from a parallel hard disk on the cloned computer to a SCSI or a serial-ATA hard disk on the deployed computer is particularly problematic.

A universal image is a cloned image that has been prepared with all drivers for disk types and hardware abstraction layer (HAL) variants encountered in the pool of computers to be deployed. Because Tivoli Provisioning Manager for OS Deployment enables the addition, on the fly, of the appropriate driver packages (driver injection) during deployment, the image cloned from one type of hardware can be made to work appropriately with hardware of another type. Therefore, a single cloned image can be sent to many hardware types, the driver selection and installation necessary for the image to work are performed automatically.

Advantages of universal images are threefold:

Image storage space
Because Tivoli Provisioning Manager for OS Deployment can modify an image and add drivers through driver injection on the fly during an image deployment, only one image and a collection of driver packages need storage space as opposed to an image for every hardware model. This is true for the parent server and every distributed copy in the network.

Image maintenance
Instead of building a new image every time a new model of hardware or driver is released, you are only required to package the driver, to set new rules for the deployment of that driver, and to test the deployment and rules.

Image replication
Minimal images mean that less time and resources are used to move those images around the network to where they are needed, on a target for deployment or on another provisioning server, for instance.

Shared repository

Managing the data directory of a Tivoli Provisioning Manager for OS Deployment server.

In Tivoli Provisioning Manager for OS Deployment, the shared repository reduces the storage space necessary to hold all server objects. The shared repository is a content-indexed dictionary of files where every file is individually stored, indexed by a global MD5 (Message Digest 5) and a subsidiary list of MD5's for each 32KB block. In the Tivoli Provisioning Manager for OS Deployment disk image, the file content is not stored, but only the list of MD5's.

By using an MD5 algorithm to individually identify each file being stored in the image repository, it is possible to eliminate the need to store duplicates of any file. For example, one Windows XP image can take 3GB of storage, but two variations of an XP image could take less than 4GB.

Shared files are not stored individually but are embedded in common pool files that are better cached by the system. There are also some index files, for the system to quickly locate a file given its MD5 description.

Note: All these files can safely be backed up at any time, because after a content-indexed file is added, it never changes its location in the index and the consistency is therefore guaranteed at all times (this is by definition in a content-indexed repository: when a file has changed, it is considered to be a different file, because it has a different MD5 index.).

With regular use, the shared repository typically only grows as new files are added. If, for some reason, you want to remove files from the shared repository that are not used any more, run a mark and sweep procedure to first mark all the archives that you want to preserve, and then sweep all the other files out of the registry.

Note: There is normally little use to running such a process on the target, except for special applications such as frequent local backups.
Tivoli Provisioning Manager for OS Deployment provides a number of primitives for running a mark-and-sweep.

To maintain your Tivoli Provisioning Manager for OS Deployment server operational at all times, ensure you observe the following best practices:

- Tivoli Provisioning Manager for OS Deployment is not a system backup software. Use the server only to store golden images for initial deployment and not individual backup images. The size of the shared file repository used to store images is not expected to be more that 50 GB. If it is larger, then the product is being used in a way different from that which is recommended in this documentation.
- Save each deployment object created on the server, (for example, system profiles, software packages, to name a few) in a .RAD export file for backup purposes, and store each deployment object on a separate storage device.
- As for all critical applications, the administrator must perform periodic backups of the complete server data directory. The backup is essential to recover from a disk failure or from any other data corruption problem.
- On a regular basis, stop the Tivoli Provisioning Manager for OS Deployment server and perform maintenance on the shared repository, using the rembo command-line option-chkshared and then -statshared. If statshared reveals a high volume of orphans (unused) files, they can be cleaned up using the -sweepshared option. For information about the usage and syntax for the rembo command, refer to the set of command-line interface topics in the *Reference* section.

Note: The -packshared option should be used only when disk space must be recovered for other applications because it will truncate the shared repository file to the smallest possible size, possibly leading to more fragmentation of the shared repository in the future thereby decreasing performance.

Chapter 2. Working with targets

Targets are computers known to the OS deployment server. This includes the OS deployment servers themselves, the computers on which they deploy system profiles, and reference computers from which cloned system profiles are created.

An OS deployment server must know its targets to be able to work with them. Therefore, any target must be added, either automatically or manually, to an OS deployment server before it can be used.

The Target Monitor is your main interface with your targets. It allows you to view your targets and their status, to organize them into a hierarchical structure for easy retrieval, to create lists using a search function, and to view them sorted by subnet. The Target Monitor also allows you to select a default administrative group into which new targets are to be attached and assigned default settings.

Target collection types

Targets known to the OS deployment server can be sorted into administrative groups, custom lists, and subnets.

Administrative groups
determine which administrators are allowed to configure which targets.

These groups can contain a hierarchy of sub-folders.Every target belongs to exactly one administrative group.

One administrative group is the *default* group that registers unknown targets when they first contact the OS deployment server.

Note: Options defined for unknown targets might not be identical to those defined for the *default* group. An unknown target boots the first time using the options set for the unknown targets. After it is registered in the default group, it uses the options set for computers in this group for subsequent boots.

Custom target lists
are arbitrary groupings of targets built by system administrators to run tasks on several targets together. A single target can belong to several custom target lists.

These groups can contain a hierarchy of sub-folders.A custom target list can be built by adding individual targets one at a time, or through a search query. This search query is launched through the **Create a custom target list from a search query** option that appears when a custom target list folder is selected.

Subnets
implicitly group targets according to their IP address. A target can only belong to one subnet at a time. Multi-homed targets are listed in the subnet on which they last made a network-boot.

Target information

The following target information is readily visible in the Target Monitor:

- **MAC address**, the target MAC address

© Copyright IBM Corp. 2006, 2010

- **IP address**, the target IP address
- **arch**, the target platform (for example, Intel, Sun, or Mac)
- **model**, the computer model of the target
- **serial**, the serial number of the target
- **?**, the state of Tivoli Provisioning Manager for OS Deployment (illustrated with icons)
- **updated**, the last time that the state information was updated
- **status**, the last deployment status of the target.

Note: You can modify the manner in which the information columns are displayed in the Target Monitor by clicking **Arrange columns** in the contextual menu. You can personalize the size of the columns, their relative order, and which columns are displayed.

Adding targets

You must add and configure a target, before you can start a deployment for it.

The examples for preparing targets are based on the deployment of one target ; the process for deploying multiple targets is similar.

Ensure you have at least one OS configuration to deploy. Having one or two software modules ready makes the deployment more useful but is not mandatory.

The Target Monitor is used throughout this documentation to manage and deploy targets. To access it, click on the first item of the menu in the web interface.

Methods for adding targets

The very first step is to select the target on which you want to deploy the OS configuration you have created. To start the deployment, the target must be visible in the Target Monitor. There are several ways to make the target appear in the Target Monitor. In all cases it is important to configure your target to start on the network, or to press the network boot hot-key (for example, `F12`) when the targetstarts. Here are the ways that you can add the target into the Target Monitor

- Let the Target Monitor detect the target. The target is started and it boots on the network. In this case the target shows up in the targets tree on the Target Monitor page, if the OS deployment server is not running as a *closed server*.
- Create the target manually. The target must be identified by either its MAC address, its IP address, its Unique Universal Identifier (UUID), or its serial number.
- Use a target list. A target list file is a text file with comma separated values, with a `.csv` extension. Lists for targets are useful for adding large numbers of targets to the OS deployment server without having to start them up individually on the network.

Detecting targets automatically

The OS deployment server is configured to automatically answer every PXE target that requests for a network boot program. Any known PXE target is added to the target database.

1. Turn on your target and make it start on the network. At this stage, the target appears in the Target Monitor, in the target tree.

2. Select the default group (called **Default** unless you have selected another group as default) to see an icon representing the target you have just started.

If no OS configuration is bound to the target, the target shows a locked screen.

If the computer you have just booted was used to create a OS configuration or was used in a previous deployment, the locked screen might be skipped and a menu with bound OS configurations displayed instead. This happens because OS configurations are already bound to the target that you are starting.

In a network with several PXE servers

In an environment with multiple PXE servers, the easiest method for populating the target database is to:

1. Stop all PXE servers except for the OS deployment server.
2. Boot PXE targets that must be inserted in the OS deployment server database.
3. Restart the PXE servers and set OS deployment server to ignore new targets.

Adding targets to the Target Monitor manually

If you want to perform a deployment without having to start targets first, you can add targets manually into the Target Monitor or import a comma-separated text file containing a list of targets to be added.

1. Go to the **Target Monitor** page on the web interface of the OS deployment server.
2. Select either an administrative group or the **by Administrative group** folder. New targets are always inserted within an administrative group.
3. Click **Register new targets**.
4. In the window, enter at least one of the following target identifiers:
   - **MAC address**
   - **IP address**
   - **Serial number**
   - **UUID**
   - **Hostname**

   Note: The **IP address** and **Hostname** are required to deploy targets other than x86 and x86-64.

   When deploying Linux on PowerPC and Cell Blades, a default **Hostname** is provided if none was registered.
5. Click **Ok**.
6. Add another target or click **Cancel** to close the **Register target** window.

When the target is added to the database, it appears in the target tree.

Note: If you have entered a wrong identifier for a target, and you want to remove that target from the Target Monitor(and from the database), right click on the target and select **Delete** from the contextual menu.

Importing and exporting targets lists

A target list file is a text file with comma separated values, with a `.csv` extension. Importing a target list is useful for adding large numbers of targets to the OS

deployment server without having to start them up individually on the network. You can also import a PCI inventory for a single target in an `.ini` file.

Target list

Before you can import a target list, you must either export one or create a new one.

Information about each target in a target list is a collection of more than seventy items, including, :

- MAC address
- IP address
- User parameters
- Motherboard information
- Processor information

To view the complete list of items, export a target list, open it and read the beginning of the `.csv` file.

For the OS deployment server to successfully import targets in a list, you must fill at least one of the following items in:

- Serial number
- MAC address
- UUID
- IP address

The filled-in item can vary from target to target. Other items can remain empty.

Target lists above 1GB in size (about 1000 targets) cannot be imported into an OS deployment server, because of browser limitations. Therefore, you cannot use target lists for more than about 1000 targets.

Note: Do not use target lists to backup target information. To backup target information, you must backup the database used with an appropriate tool. Lists of targets are not as complete as the database. In particular, target lists do not include some crucial target information found in the database, among which

- Bindings
- Disk inventory
- PCI inventory
- Deployment history

PCI inventory

A PCI inventory is exported on a USB key or floppy disk when this media is inserted in a target, booted through a network boot media, but which does not have network drivers.

- **Importing a target list**
  1. Go to the**Target Monitor page** in the web interface.
  2. Click **Import targets**.
  3. Indicate the location of the `.csv` file.
  4. Click **Ok**.
- **Exporting a target list**
  1. Go to the**Target Monitor** page in the web interface.
  2. Click **Export targets**.

3. Click **Save**. The default file name (`hostexport.csv`) and saving location can be changed.

- **Importing a PCI inventory**
  1. Go to the**Target Monitor page** in the web interface.
  2. Click **Import targets**.
  3. Indicate the location of the `newhost.ini` file.
  4. Click **Ok**.

Configuring new targets

Targets are assigned default parameters at the time they are added to the OS deployment server. You can configure new targets by changing these default parameters.

You can configure how the OS deployment server accepts targets attempting to boot, in which group they are added, which Tivoli Provisioning Manager for OS Deployment kernel options to use, and whether to enable human interfaces.

1. To perform this Go to **Server → OS deployment → Task templates**.
2. Select **Idle Layout** and then **Idle state**.
3. Click **View idle parameters**.
4. Click **Edit** in the **Handling of unknown targets** banner. This opens the **Handling of unknown targets** dialog
5. Modify the parameters you need. You can also decide not to include targets to the OS deployment server.
6. Click **OK**.

Targets that will be added to the OS deployment server are now assigned these new default parameters.

Changing the default administrative group

You can change the administrative group to which new targets are automatically assigned. By default, new targets are assigned to the *Default* group.

To know which is the current default administrative group to which new targets are assigned, select any administrative group and read the information provided. To perform this go to **Server → OS deployment → Target Monitor**. .

To change the default administrative group:

1. Optionally, create a new administrative group.
   a. Select **by Administrative group** in the Target Monitor
   b. Click **Add a new admin** in the contextual menu to create a new administrative group.
2. To perform this Go to **Server → OS deployment → Task templates**.
3. Select **Idle Layout** and then **Idle state**.
4. Click **View idle parameters**.
5. Click **Edit** in the **Handling of unknown targets** banner.
6. In the second section of the appearing dialog, use the drop down list to select the new default administrative group.

Requirements for VMware targets

To successfully deploy system profiles on VMware, it is important that your system conforms to a number of requirements when setting up the VMware target.

Guest operating system

Always set the guest operating system to Windows 2008 or Windows Vista, even when you deploy a different Windows operating system (such as Windows 2000, Windows 2003, Windows XP, or Windows 7).

Network adapter

- Windows The Intel e1000 network adapter works correctly on all Windows editions
- Windows On Windows 64-bit, the AMD Lance network adapter is not supported. Using it results in a failed deployment with either a shutdown of the virtual machine or a blue screen.
- Linux The AMD Lance network adapter is supported for all Linux distributions, but it is very slow.
- Linux The Intel e1000 network adapter is supported on all Linux distributions, except for Red Hat Enterprise Linux 5 (REHL5).

  With REHL5, the Intel e1000 card is in a dirty state when rebooting the operating system after performing Linprep. The target can no longer connect to the network. Therefore, the deployment stops and fails. To workaround this issue, install two network cards on your VMware target:
  - The Intel e1000 as the primary boot device
  - An AMD Lance as the second boot device to use as a fallback.

  With the two cards, when Linux reboots and the Intel e1000 does not answer, the AMD Lance takes over, allowing the virtual machine boot and the deployment continue.
- SUSE For cloning and Direct Migration of SuSE Linux Enterprise Server 10, you must use the Intel e1000 network adapter.

SCSI controller

The compatibility between SCSI controllers and Windows operating systems on VMware targets is described in Table 1.

Table 1. Compatibility between Windows operating systems and SCSI controllers

Compatibilty	Windows XP	Windows 2003/Vista/2008/7	WinPE 3.0
BusLogic	Yes	No	No
LSI Logic	No	Yes	Yes

- If you intend to deploy Windows 2003/Vista/2008/7, use the LSI Logic driver.
- If you intend to deploy Windows XP, you have two options:
  - You can make a software module with the BusLogic driver, and, depending on your VMware version, bind it to or inject it offline into a WinPE deployment engine
  - You can make a software module with the LSI Logic Parallel driver, and bind it to your Windows XP system profiles
- On all other operating systems, LSI Logic is supported.

Note: LSI Logic driver for Windows Server 2003 operating system , `symmpi.inf`, version 1.28.03, has been tested successfully.

Injecting drivers on WinPE 3.0 to deploy Windows XP guests

Before installing Windows XP or Windows XP guest operating systems on a VMWare hypervisor, with BUSLogic adapter on VMWare, you must inject the VMWare SCSIAdapter BusLogic drivers into the standard WinPE 3.0. These drivers are not contained on the Vista 7 CD/DVD or installed with VMWare tools.

- Create a virtual machine and ensure that its devices are set up correctly.
- On VMware 4.1, create the WinPE 3.0 deployment engine to contain the necessary BusLogic drivers. Assign relevant matching models to this WinPE deployment engine, for example `*VMware*4.1*`.
- On VMware 4.1, disable any virus scan, to improve performance in the WinPE 3.0 update.

If you use VMware 3.5, you can user dynamic driver injection and bind your driver software modules to the WinPE deployment engine using the driver binding grid. If you use VMware 4.1, you must inject the VMware missing drivers offline in the WinPE deployment engine.

1. Extract the Microsoft drivers needed to run a Windows virtual machine on VMWare on a virtual USB key or a floppy disk.
   a. Install a Windows virtual machine on VMware.
   b. From the VMware Workstation menu, select **VM > Install VMware Tools**, The VMware Workstation connects the virtual machine CD drive to the ISO image file that contains the VMware Tools installer for your guest operating system. After the installation process, a new CD is bound to VMware and you can see all the needed drivers.
2. Create software modules for the newly extracted drivers.
3.
   - With VMware 3.5, simply bind your newly created driver software modules to the WinPE deployment engine.
   - With VMware 4.1, you must inject the driver software modules offline into an existing WinPE deployment engine.
     a. Go to **Server** → **Advanced features** → **Deployment engines**.
     b. Double-click the name of a deployment engine to view its details.
     c. Select **Inject driver** in the contextual menu.
     d. In the wizard, specify a computer running the web interface extension.
     e. Select the driver software modules to inject in the WinPE 3.0 deployment engine.

        Note: Injected drivers cannot be removed from WinPE 3.0. These drivers are started regardless of whether they are compatible with the hardware.
     f. Follow the remaining instructions in the wizard.

Your BusLogic driver is now either bound to, or contained in, your WinPE deployment engine.

You can now install Windows XP with the WinPE deployment engine on VMware, and then use your guest target like any other virtual machine.

Booting pSeries targets on the OS deployment server

pSeries® machines can be booted on the OS deployment server.

Before you can boot a pSeries target on the OS deployment server, you must

- Verify the network connectivity as follows:
  1. From the SMS menu, test the network interfaces using the **Setup Remote IPL (Initial Program Load)** menu.
  2. Select the interface to use for the deploy.
  3. Configure it and run a ping test to verify the connectivity.

     Note: Ensure that the selected interface is recognized by the operating system during the installation phase.
- Manually register the pSeries target in the OS deployment server, indicating at least the MAC address and the hostname.
- Run the **devalias** command to select the correct boot interface and add it.
- Configure the TCP/IP options.
- Start a deployment task on the target. Without a task bound to it, the target cannot boot on the OS deployment server.

How to boot a pSeries target on the OS deployment server depends on the operating system you want to install.

- AIX SUSE To install AIX and SuSE 10
  1. Boot the target using the **boot net** command.
  2. Type 1 to select **SMS Menu**.
  3. Type 5 for **Select Boot options**.
  4. Type 1 for **Select Install/Boot Device**.
  5. Type 6 for **Network**.
  6. Under **Select device**, select the network interface that you have registered in the OS deployment server. If you are not booting from the default network interface, use the alias of the interface instead of the PCI identifier.
  7. Type 2 for **Normal Mode Boot**.
  8. Type 1 (Yes) to confirm the above.

     Note: If the standard Linux operating system booting stops and you are using the serial console access, to solve the problem press any key in the `Autoyast` boot prompt. Type **linux console=hvsi0** and press enter.
- Red Hat To install RedHat
  1. Before booting ensure you are using the standard network card, otherwise perform the following steps:
     - Switch to the OpenFirmware prompt and list the boot aliases using the **devalias** command.
     - If the interface from which you are going to boot is listed in the aliases you can move forward. If the interface is not included in the devalias list then create a new alias. Run **ls** to list all the devices and find out the device address of the network card.
     - Add a new alias using **devalias** such as: `devalias net2 /pci@800000020000203/ethernet@1`
  2. Boot the target using the **boot net** command.

3. Press 8 when booting to reach the Open Firmware prompt.
4. From an Open Firmware prompt, run `boot net ks=http://serverip:serverport/linux/ks.cfg ksdevice=eth0`. `serverip` is the IP address of the OS deployment server, and `serverport` its port. Serverport is typically 8080. To boot from a different network card use the alias previously defined: `boot net2 ks=http://serverip:serverport/linux/ks.cfg ksdevice=eth2` The chosen interface is recognized as `eth2` during the operating system installation.

Booting CellBlades targets on the OS deployment server

CellBlades can be booted on the OS deployment server.

To boot on the OS deployment server the following steps must be followed:

1. Boot the target using the **boot net** command.
2. Press 8 when booting to reach the Open Firmware prompt.
3. From an Open Firmware prompt, run `boot net ks=http://serverip:serverport/linux/ks.cfg ksdevice=eth0`. `serverip` is the IP address of the OS deployment server, and `serverport` its port. Serverport is typically 8080.

If the server IP is 192.168.1.25, and the server HTTP port is 8080, type on the Open Firmware prompt the following line: `boot net ks=http://192.168.1.25:8080/linux/ks.cfg ksdevice=eth0`

Booting SPARC targets on the OS deployment server

Booting SPARC targets on the OS deployment server requires a few prerequisites and depends on whether you are doing it from OpenBoot or from a running operating system.

- DHCP option 66 must be set to the IP address of the OS deployment server.
- DHCP option 67 must be set to `rembo.fcode`.
- To network boot with Tivoli Provisioning Manager for OS Deployment, the SUN SPARC target must support WAN boot. The Open Boot version of the SPARC target must be equal or greater than 4.17.1. To verify if a SPARC target running under Solaris supports WAN boot run the following command:

  ```
  # eeprom | grep network-boot-arguments
  ```

  If the variable `network-boot-arguments` is displayed, or if the previous command returns the output `network-boot-arguments: data not available`, the OBP supports WAN boot installations. You do not must update the OBP before you perform your WAN boot installation. If the previous command does not return any output, the OBP does not support WAN boot installations. You must perform one of the following tasks.

  – Update the target OBP. See your system documentation for information about how to update the OBP.

  – After you complete the preparation tasks and are ready to install the target, perform the WAN boot installation from the Solaris Software CD in a local CD drive.

  For instructions about how to boot the client from a local CD drive, see, http://docs.sun.com/app/docs/doc/819-5776/6n7r9js6t?a=view. To continue preparing for the WAN boot installation, see http://docs.sun.com/app/docs/doc/819-5776/6n7r9js5p?a=view.

- You must register your SPARC target on the OS deployment server by indicating at least its IP address and its Hostname before it can boot on it.
- Network boot on the OS deployment server for Solaris is accepted only when a deployment task is scheduled on that target .

You can boot a SPARC target on the OS deployment server either when the target is booting, or when the Solaris operating system is running. You can also use a dynamic or a static IP address.

- From the OpenBOOT monitor (Stop-A), type `boot net:dhcp`. To make this change permanent, type `setenv boot-device net:dhcp`. Then a simple `boot` command or a cold boot are enough to boot onto the OS deployment server. If `setenv boot-device net:dhcp` does not work, use a static IP address.
- To boot with a dynamic IP address from the OpenBOOT monitor (Stop-A), type

  ```
  setenv network-boot-arguments dhcp,file=http://<OSDeploymentServerIP>:8080/sun4u
  boot net - install
  ```

  where <OSDeploymentServerIP> is the IP address of your OS deployment server.
- To boot with a static IP address from the OpenBOOT monitor (Stop-A), type

  ```
  setenv network-boot-arguments host-ip=<client-IP>,router-ip=<router-ip>,
  subnet-mask=<mask-value>,file=http://<OSDeploymentServerIP>:8080/sun4u
  boot net - install
  ```

  where:

  <client-IP>
  Is the IP address of the target.

  <router-ip>
  Is the IP address of the router.

  <mask-value>
  Is the subnet mask value.

  <OSDeploymentServerIP>
  Is the IP address of your OS deployment server.
- To force a network boot from the operating system, use

  ```
  /usr/platform/sun4u/sbin/eeprom boot-device=net:dhcp
  /usr/sbin/reboot
  ```

  Alternatively, you can force a single network boot by using the following special string, that is recognized by the bootstrap code of the OS deployment server

  ```
  /usr/platform/sun4u/sbin/eeprom boot-device="net:dhcp was: disk"
  /usr/sbin/reboot
  ```

  Note: For architectures other than `sun4u`, change the path above. Use the `uname -m` command to check the architecture.
- If you are running the web interface extension as a service on a SUN target , you can use the Target Monitor option to automatically reboot the target from the web interface. This generates the one-time change of boot device described above.

Organizing targets

Targets in the Target Monitor are organized into administrative groups, custom lists, and subnets.

An administrative group has a hierarchical, tree-like structure, and they can be used by system administrators to grant or deny access to specific web interface operators to configure particular targets groups. Custom lists are arbitrary lists built by system administrators to run tasks on several targets at the same time. A custom target list can be built by adding individual targets, or as the result of a search query. Subnets implicitly and automatically group targets according to their IP address. Multi-homed targets (targets with more than one network interface) are listed as part of the subnet on which they last made a network-boot. Subnets cannot be modified by the users.

When a new computer is added to the database, either manually or because the target was started in network boot mode, the Target Monitor automatically places this target in the *default* administrative group. To check which group is the default administrative group, select any administrative group and read the text below the target tree. The name of the default administrative group is listed.

You can move targets from one group or custom list to another:

Drag-and-drop the icon from one group or custom list to another. You might want to use the pin-board in the web interface title bar, for example if the destination folder is not visible. You will be able to temporarily leave the dragged target on the pin-board while you search for and open the folder into which you want to drop the target.Figure 2 illustrates this process.

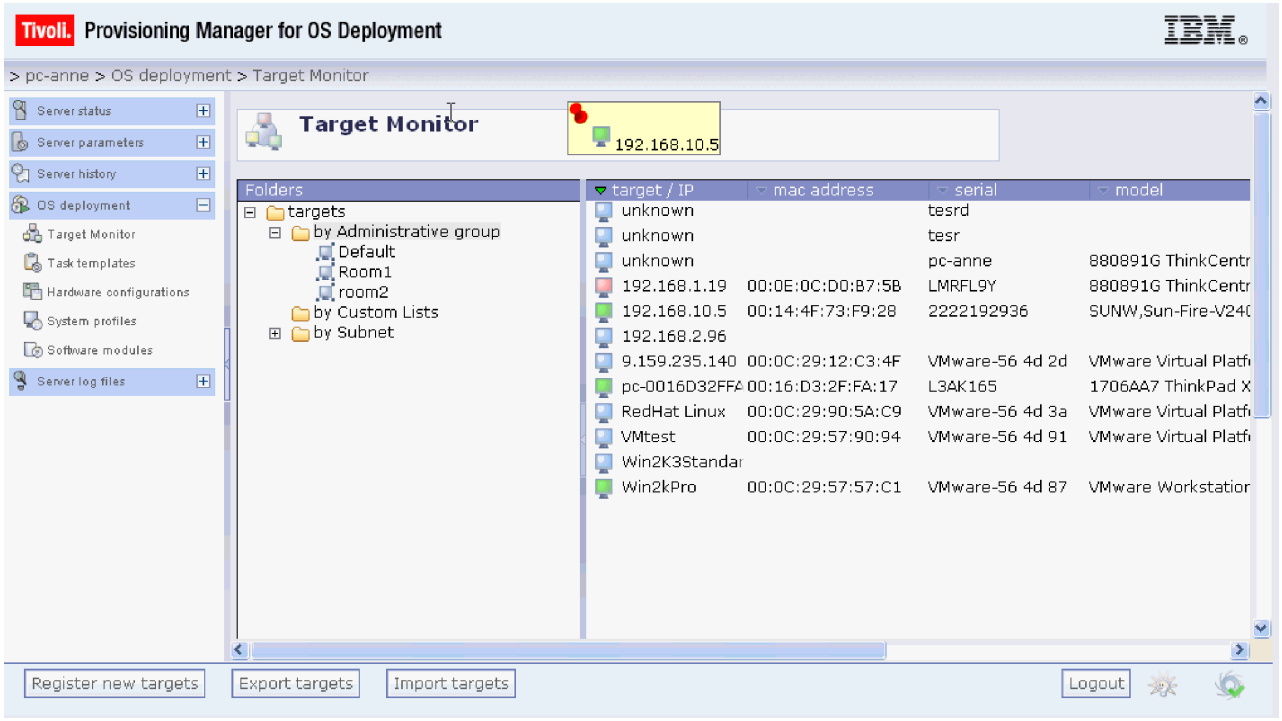


Figure 2. Pin-board of the Target Monitor

Configuring targets

Targets can be configured either individually or together, using either administrative groups, custom lists, subnets, or multiple selection.

To configure a single target:

1. Select an individual target.
2. Click **View target details** in the contextual menu
3. In turn, click **Edit** in the banner of each group of parameters you want to modify.

Configuring multiple targets

1. Select multiple targets, an administrative group, a custom list, or a subnet.
   - If you have selected multiple targets, edit links for each group of parameters appears at the bottom of the Target Monitor.
   - If you have selected an administrative group, a custom list, or a subnet, click **Edit targets in list** in the contextual menu. The **Target multi-edition** window appears.
2. In turn, click the edit link for each group of parameters.
3. Select the options that you need. Selecting an option allows you to view and select sub-options.
4. Click **Save** to close **Target multi-edition** and return to the **Target Monitor**.

Configuring targets for fully unattended OS deployments

To run fully unattended deployments, some parameters are necessary. The number and nature of these parameters vary according to the operating system which is to be deployed. Configuring targets is an alternative to providing the data in OS configurations.

You must configure your target before you start a deployment. Some values are mandatory for a fully unattended deployment and must be filled in at the target level if the information is not included in the OS configuration.

Note: If multiple targets share the same information, you can set fixed values in the OS configuration that you are deploying on these targets. Fixed values at the OS configuration level override values entered in the **Target details** page, and are used by all the targets deploying the OS configuration containing the fixed values.

1. Double-click on the target to access the details page for this target. The **Target details** page contains all of the properties specific to this target, including the target name, the serial number, and the product key to use when installing an operating system.

   Windows **For Windows deployments**

   The following fields are required by Sysprep and are asked during the deployment if they are not filled in on the properties page:
   - target name
   - Product key (The key in xxxxx-xxxxx-xxxxx-xxxxx-xxxxx format can be copied and then pasted into the first entry field at one go by pressing and holding Ctrl and then pressing V), unless you are deploying Windows Vista/2008/7 with a Volume License.
   - User full name and organization
   - An administrator password

- Workgroup or domain name

Solaris **For Solaris deployments**

- Solaris standard installation procedure includes checking for valid computer name and IP that matches DNS and DHCP. Otherwise, the deployment may fail.
- The Solaris NFS server must have name resolution properly configured to know the target name of the target. Failure to do so may lead to an interruption of the installation process.
- Four name resolution methods are available with Solaris. For each of them, a specific set of fixed properties must be set. Failing to set these properties results in a failed deployment.

  DNS For DNS, you must enter
  - At least one DNS server
  - A DNS domain
  - A DNS domain search order

  NIS and NIS+
  For NIS and NIS+, you must enter
  - A DNS domain
  - A NIS name server

  LDAP For LDAP, you must enter
  - A DNS domain
  - An LDAP name server
  - An LDAP profile
- The OS deployment server uses the root user information provided in the target specific details (or profile details) during installation. If this piece of information is not configured, the default value of root user password is ""

2. Enter the mandatory fields, and click **OK** to validate your changes.
3. If you have used the Target Monitor on this target before and the target is not displaying the *locked screen*, you might want to remove the OS configuration bindings that are forcing it into specific OS configurations.
   a. Double-click on your target.
   b. Select the **OS configurations** tab.
   c. Click **Edit**.
   d. Clear the items and click **OK** to remove the bindings.

Setting partition sizes on the targets

If you need to have different partition sizes on your targets, but you want to deploy them with the same system profile, you can set the partition size by target.

1. Edit the **User details** section of the **Target details**.
2. In **User Category 9**, type in the partition size information. Use the following syntax:resize [<existing mount point> <size in MB>].

   For example, type `resize c 5000 d 10000`.

   Note: The information provided here overrides the partition size information given in the OS configuration and on the system profile.
3. Click **Save**.

The next time you deploy this target, the partitions you have specified are resized according to the values you provided.

Chapter 3. Working with Windows operating systems on x86 and x86-64 targets

This section provides information about how to work with the product to deploy Windows operating systems.

Overview of WinPE deployment engines

WinPE deployment engines are a prerequisite for provisioning Windows operating systems.

Windows Preinstallation Environment (WinPE) is a group of files that can be loaded as a ramdisk and that allow you to perform operations on a target. Without WinPE, you cannot provision Windows operating systems.

WinPE deployment engines are stored under **Server** → **Advanced features** → **Deployment engines**.

Current version

The current version for the WinPE deployment engine is 3.0. Other versions are not compatible. WinPE 3.0 must be created from a Windows Automated Installation Kit (AIK) for Windows 7 in English.

WinPE 32-bit and WinPE 64-bit deployment engines

Two WinPE deployment engines are extracted from Windows AIK, one 32-bit version and one 64-bit version.

In version 7.1.1.3 of the product, the 32-bit WinPE deployment engine is used for all the tasks requiring a WinPE deployment engine. The 64-bit WinPE is used only to deploy Windows Vista 64-bit and Windows 2008 64-bit unattended setup system profiles. For these two operating systems, both versions of the WinPE deployment engine are used together.

Deployment engine creation

If your OS deployment server runs on a Windows operating system and if you have Windows AIK installed on the server, then the OS deployment server checks when it starts up that there is a WinPE deployment engine on the server. If not, it creates one automatically. The process takes several minutes, during which time it is not possible to log in to the web interface.

If your OS deployment server does not run on a Windows operating system, or if you want additional WinPE deployment engines, you can create them manually.

Working with several WinPE deployment engines

In most cases, you do not need to create additional WinPE deployment engines, because one per architecture is sufficient for most uses of the product.

© Copyright IBM Corp. 2006, 2010

When the WinPE deployment engine is transferred to a target, for example, during a deployment, it contains all the drivers that are bound to this deployment engine, even if only those bound for the specific target model are used. If you are binding many drivers to account for a very large range of hardware, the size of your WinPE deployment engine might become too large for some targets with a small RAM. In this case, you might want to create an additional WinPE deployment engine, match it only to the target with a small RAM, and bind to it only the drivers needed for this specific target. The size of the new WinPE deployment engine transferred to the target is therefore much smaller.

When you have several WinPE deployment engines for the same computer architecture, you must make sure that you have specified matching model patterns that allow the OS deployment server to dispatch the WinPE deployment engines to the correct targets.

The method uses the Microsoft **drvload** command to inject drivers. If this command does not work, you must inject the drivers in the standard way.

Windows Automated Installation Kit

Windows Automated Installation Kit (AIK) is needed to perform different tasks when provisioning Windows operating systems.

Windows AIK is needed to:

- Create a WinPE deployment engine
- Create an unattended setup system profile of a Windows Vista/2008/7 operating system
- Create a cloning system profile from a Windows WIM image
- Update a Windows system profile, for example, with a HotFix
- Create a Windows PE-based network boot CD/DVD

Current version

The current Windows AIK version to use with the product is Windows AIK for Windows 7 in English.

Windows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.

Best practices

Given the numerous uses of Windows AIK in the process of provisioning Windows operating systems, it is a good practice to perform all these tasks on one target that is installed with all the requirements. This target must have:

- A Windows XP/2003/Vista/2008/7 operating system
- Windows AIK for Windows 7 in English installed
- The web interface extension installed and started with local administrator privileges

If your OS deployment server is on a Windows operating system, you can use your OS deployment server as the dedicated Windows target.

Checking the version of Windows AIK

If you are unsure of the version of Windows AIK installed on a target, you can verify it.

- On Windows XP and Windows 2003:
  1. Open the **Control Panel** of your operating system.
  2. Select **Add or Remove Programs**.
  3. Select **Windows Automated Installation Kit** in the list
  4. Click **Click here for support information**.
  5. Check that the version number is 2.0.0.0, which corresponds to Windows Automated Installation Kit (AIK) for Windows 7 in English.
- On Windows Vista and Windows 2008:
  1. Open the **Control Panel** of your operating system.
  2. If you are in the **Control Panel Home** view, select **Programs**, otherwise skip this step.
  3. Select **Programs and Features**.
  4. Select **Windows Automated Installation Kit** in the list.
  5. If you cannot view the version number in the selected line, you can add a column with this information.
     a. Select **View** and then **Choose Details...**.
     b. Select **Version** and click **OK**.
  6. Check that the version number is 2.0.0.0, which corresponds to Windows Automated Installation Kit (AIK) for Windows 7 in English.
- On Windows 2008 R2 and Windows 7:
  1. Open the **Control Panel** of your operating system.
  2. If you are in the **Control Panel Home** view, select **Programs**, otherwise skip this step.
  3. Select **Programs and Features**.
  4. Select **Windows Automated Installation Kit** in the list.
  5. If you cannot view the version number at the bottom of the screen, select **Organize** → **Layout** → **Details pane** to make it visible.
  6. Check that the version number is 2.0.0.0, which corresponds to Windows Automated Installation Kit (AIK) for Windows 7 in English.

Creating a WinPE 3.0 deployment engine

To create or deploy Windows profiles, you must have created a WinPE 3.0 deployment engine.

Ensure that the computer from which you create the WinPE 3.0 deployment engine satisfies these conditions:

- Runs a Windows operating system
- Has Windows Automated Installation Kit (AIK) for Windows 7 in English installed. Windows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.
- Runs the web interface extension

The computer from which you create the WinPE 3.0 deployment engine can be:

- A local OS deployment server installed on a Windows operating system. This is the recommended option.
- Any computer with a Windows operating system.

From version 7.1.1.3 of the product, several WinPE 3.0 deployment engines can coexist on any OS deployment server.

1. Depending on what you are doing, you can create the WinPE 3.0 deployment engine from:
   - The **Deployment engine** page:
     a. Go to **Server** → **Advanced features** → **Deployment engines**.
     b. Click **New deployment engine**.
     c. Follow the instructions in the wizard.
   - The **Welcome** page:
     a. Select **Make one now** from the **For Windows scripted installation** or from the **For Windows clone installation** section.
     b. Click **Next**.
   - The **System profiles** page, if you do not already have a WinPE 3.0 deployment engine:
     – If you run an unattended setup:
       a. Go to **Server** → **OS deployment** → **System profiles**.
       b. From the contextual menu, select **Add a new profile**.
       c. Select **Unattended setup (scripted install)**.
       d. Select one of the Windows operating systems as the type of system profile to create and click **Next**.
       e. The wizard displays a warning message informing you that it did not find a WinPE 3.0 deployment engine. Click **Next** to create one.
     – If you run the capture of a cloned system profile:
       a. Go to **Server** → **OS deployment** → **System profiles**.
       b. From the contextual menu select **Add a new profile**.
       c. Select **Cloning from a reference machine**.
       d. Enter the IP address of the target that you want to clone and ensure that the referencetarget is ready to boot into the OS deployment server.
       e. Select one of the Windows operating systems as the type of system profile to create and click **Next**.
       f. The wizard displays a warning message informing you that it did not find a WinPE 3.0 deployment engine. Click **Next** to create one.
2. Specify the address of the computer on which you installed Windows AIK for Windows 7 in English and the web interface extension and click **Next**.

The resulting WinPE 3.0 deployment engines, one 32-bit WinPE 3.0 deployment engine and one 64-bit WinPE 3.0 deployment engine, are now shown under **Server** → **Advanced features** → **Deployment engines**.

You can now indicate matching target models for your WinPE 3.0 deployment engine and bind drivers to it.

After you created the WinPE 3.0 deployment engines, you can create and deploy Windows system profiles.

Note: During the deployment, do not edit the WinPE 3.0 deployment engine that you are using.

Editing the information of a WinPE deployment engine

You can edit the description and the comment attached to a WinPE deployment engine.

To edit the description and comment of a deployment engine:

1. Go to **Server** → **Advanced features** → **Deployment engines**.
2. To view the details of the deployment engine, you have two options.
   - Double-click a deployment engine.
   - Select a deployment engine, and then select **View engine details** in the contextual menu.
3. Click **Edit** above the section **Deployment engine information**.
4. Update the description and the comment to identify more easily how this WinPE deployment engine is to be used.
5. Click **OK** to save your changes and return to the **Engine details** page.

If you intend to use this deployment engine to deploy IBM servers, you might want to call your WinPE deployment engine `WinPE3 for IBM servers 32-bit`. The comment can include the server models that this WinPE deployment engine is planned to be compatible with.

If you updated the description of your WinPE deployment engine, you probably have more than one deployment engine per architecture. In this case, provide matching target models for your deployment engines.

Adding matching target models to a WinPE deployment engine

If you have several WinPE deployment engines for the same architecture, it is important that you specify with which targets a given WinPE deployment engine must be used.

If you have only one WinPE deployment engine per computer architecture, there is no reason to modify the model patterns. Only use the default * pattern, to match any target known to the OS deployment server.

To add model patterns associated with a deployment engine:

1. Go to **Server** → **Advanced features** → **Deployment engines**.
2. To view the details of the deployment engine, you have two options.
   - Double-click a deployment engine.
   - Select a deployment engine, and then select **View engine details** in the contextual menu.
3. In the **Matching models** section, click **Add a new model pattern**.
4. Enter the pattern and click **OK** to save your new pattern. The * character is used as a wildcard replacing any number of characters. The **?** character is used as a wildcard replacing exactly one character.

When deploying a target, if there are several WinPE deployment engines available, a search is performed in the list of model patterns for all WinPE deployment

engines available. The WinPE deployment engine selected has the most restrictive pattern matching the target model being deployed.

If there is no matching pattern, deployment cannot proceed.

Note: In a multiple server architecture, a WinPE deployment engine that is not fully replicated from a parent server is not yet available on the child server.

Consider that you have two WinPE deployment engines, `WinPEa` and `WinPEb`. `WinPEa` has the following patterns: `IBM Server *`, and `lenovo *`, while `WinPEb` has `lenovo m/55 *`, `lenovo T*`, and `*`.

A target with model `lenovo T61` is deployed with `WinPEb` because its model matches the `lenovo T*` pattern, because it is more restrictive than `lenovo *`.

A target with model `lenovo ThinkCenter A58` is deployed with `WinPEa` because its model matches the `lenovo *` pattern, because it is more restrictive than the generic `*` pattern.

A target with model `HP Server` is deployed with `WinPEb` because its model matches only the `*` pattern.

You can check which WinPE deployment engine is used with a given target by looking at the **Windows specific info** section in **Server → OS deployment → Target Monitor → Target details**. If you are dissatisfied with the selected WinPE deployment engine, you must adapt the target models for your WinPE deployment engines.

Binding drivers to a WinPE deployment engine

When WinPE does not contain the drivers that you need for a specific target, you must bind these drivers to the WinPE deployment engine to deploy the target.

Your WinPE deployment engine contains built-in drivers. Use them first.

If you encounter problems with the built-in drivers, if some drivers are not bound, or if some drivers are missing, bind other drivers to your WinPE deployment engine.

In this offline driver injection process, you can only bind drivers, to your WinPE deployment engine, that are driversoftware modules in your OS deployment server. You must therefore create driver software modules from the drivers that you want to bind to your WinPE deployment engine.

The product helps you select appropriate drivers for particular target models. It helps you to predict potential problems and to solve them. It does not guarantee that a specific WinPE deployment engine, with bound drivers, works with a given target.

The information used by the OS deployment server to predict the compatibility of a driver with a target model is taken from the content provided by the vendor in its driver. The OS deployment server cannot verify the accuracy of this information.

1. Check the compatibility of your WinPE deployment engine.
   a. Go to **Server → Advanced features → Deployment engine**.
   b. To view the details of the deployment engine, you have two options.

- Double-click a deployment engine.
- Select a deployment engine, and then select **View engine details** in the contextual menu.

c. Go to the section **Network and mass storage drivers**. A check is performed while the page is loading. This can take a few minutes.

If drivers are missing, or are not bound, or if several drivers are bound for the same device, the following information is provided.

Indicates a missing critical driver, or a critical driver of the wrong architecture.


Indicates that a missing non-critical driver, or a non-critical driver of the wrong architecture.

Indicates that a required driver is present on the OS deployment server, but that it is not bound.

Indicates that there are several drivers bound for the same device, or that there is a binding with a driver that is not known as compatible.

You can expand the line to get more information.

- For drivers missing on the OS deployment server, you find a suggestion of where to look for it, including, if available, a download link and the exact directory within the downloaded archive where the driver can be found.
- When drivers are present on the OS deployment server, you find suggestions of which driver to bind, in order of preference. If multiple drivers are known to possibly work for a device, the best choice is listed first. The choice is explained in the advice text, which first recommends the use of *device-specific drivers*, that is, drivers that have been specifically designed for the given hardware device. Then *compatible device drivers*, that match the device family, are recommended, even if they are not an exact rebranded variant (for example, as second choice, an Adaptec driver of the same family as an IBM ServerRaid adapter, if it is based on the same chipset). Finally, as third choice, *generic drivers*, for example, Microsoft generic AHCI driver for any AHCI controller, are recommended.

If no error is found, you do not need to modify the bindings.

2. Modify the driver bindings of the WinPE deployment engine. There are two ways to perform this.
   - Use a wizard.
     a. Click **Fix Drivers**.
     b. Follow the instructions in the wizard. After having selected a target model, you must select one of these options:

        Automatically fix issues that can be fixed for this model.
        Fixes all issues that can be automatically fixed. Such issues include, for example, a missing binding to an existing driver, multiple bindings for a device, or removing a driver tagged for another operating system.

        Manually fix issues for this model.
        Presents you with each issue in turn. Ways to solve the issue, when available, are proposed.

        Automatically bind drivers for this model.
        Erases every existing binding. New bindings are then automatically added.

Copy driver bindings for this model from a similar engine.
Copies all the bindings from a selected source engine to the current engine.

Reset all drivers bindings for this model.
Erases all the driver bindings, and does not create any new binding.

- Edit the bindings manually, using the driver binding grid.
  a. Click **Edit engine's driver bindings** on the **Engine details** page.
  A grid is loaded.
  Columns represent target models known to the OS deployment server and matching the patterns provided for the WinPE deployment engine. They can be expanded to view their network and mass storage devices, if a PCI inventory has been performed.
  The first line represents the WinPE deployment engine. Other lines represent software module folders in the OS deployment server. They can be expanded to view individual drivers. If a driver can be used only for 32-bit or 64-bit machines, a superscript *x86* or *x86-64* mark is written next to the driver name. If you do not find the drivers that you need in the list provided, create software modules for your drivers.
  b. *(Optional)* To obtain a summary of the errors and warnings, click the link provided above the grid. This helps you locate the problematic areas in the driver grid.
  c. Expand the columns of problematic target models to view the individual network and mass storage devices.
  d. Expand software module folders containing drivers to view the individual drivers.

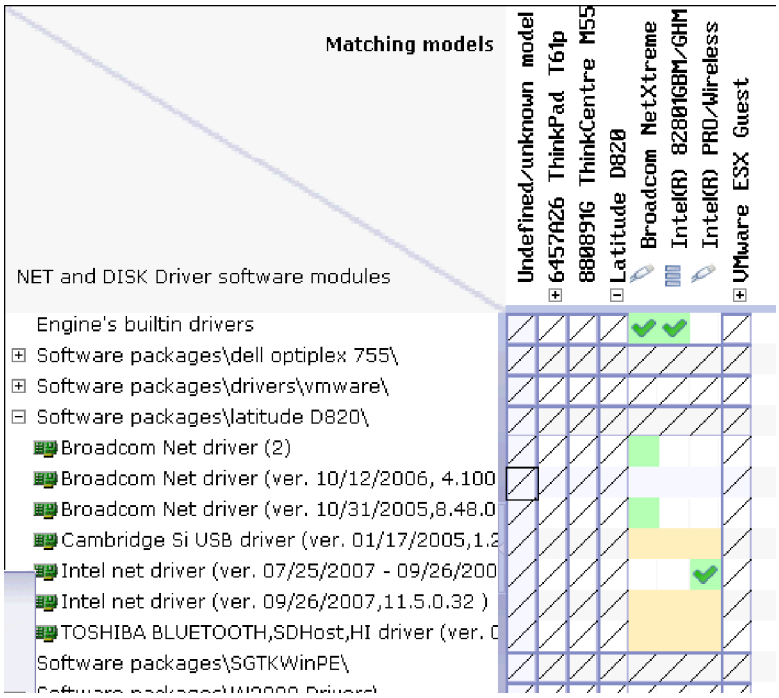


Figure 3. Driver binding grid

A cell with a green background indicates that driver information corresponds to the device. The quality of the drivers that can be selected

is illustrated by the intensity of the green background:  the best drivers are in intense green, the family drivers are in standard green, and the generic drivers are in pale green.

A cell with an orange background indicates either that the driver is not a PCI driver, or that there is no compatibility information available for the driver.

A cell with a green check mark indicates that the driver is bound to the WinPE deployment engine for use with the specific target model and device.

e. Click a green or orange background cell to add or remove bindings.

It is not possible to bind or unbind drivers from the WinPE deployment engine itself, because they are built-in drivers.

You should have one, and only one, check mark per column, indicating that you have one and only one driver for each device.

f. When you have finished modifying the bindings, click **Save**.

g. To return to the **Image details** page, click **Back**.

Potential problems with the image are recomputed, allowing you to check if your modifications have solved the detected problems.

When you have solved all the driver binding issues, you can deploy target models that match your WinPE deployment engine.

System profiles

A system profile is the partition layout and list of files to deploy an operating system, either by unattended setup or by cloning, from a reference target or from a reference image.

The main purpose of Tivoli Provisioning Manager for OS Deployment is to deploy an operating system on targets by replicating a reference system. However, unattended installation of operating systems is also possible. The latter case Tivoli Provisioning Manager for OS Deployment does not replicate a reference system, but merely provides the correct parameters to Windows or UNIX setup for a fully unattended installation.

There are a number of differences between an unattended installation and disk cloning. First, creating an unattended installation in Tivoli Provisioning Manager for OS Deployment is straightforward. All of the necessary tasks are performed on the server, using the Web interface. In contrast, a cloning-mode system profile requires you to configure a target, prepare it for cloning, and run the cloning process directly on the target. However, the native mode of operation of the product is centered around cloning-mode system profiles, because this method of deployment is faster than unattended installation. When deploying computers on a large scale, unattended installation is not possible. Novice users might start with creating unattended installation profiles because this is easier than cloning-mode profiles.

Note:

- To avoid failures in creating or deploying asystem profile, clean up the temporary directory inside the OS deployment server installation directory on a regular basis.

- To create or deploy a system profile from a physical or virtual machine at least 512 MB RAM is required.
- The exFAT filesystem is not supported.
- System profiles can have a maximum of 3 primary partitions. Therefore, you cannot clone a target with 4 primary partitions.

BitLocker compatibility

Tivoli Provisioning Manager for OS Deployment is compatible with Microsoft BitLocker Drive Encryption (BitLocker), which is available with some Windows operating systems. BitLocker is a security tool which protects data by encrypting it, rendering the content of a hard disk unreadable if stolen.

To operate, BitLocker requires at least two partitions:

- a boot partition containing the BitLocker tool and which must have a size of at least 1.5 GB
- an operating system partition which can be encrypted

Tivoli Provisioning Manager for OS Deployment can make a deployed target ready for BitLocker by creating the appropriate partition scheme during the deployment.

When you create a system profile for Windows Vista/2008/7, the Profile Wizard asks you whether you want to make your profile ready for BitLocker. In case of a positive answer, the wizard asks you the relevant questions to set up the partition scheme.

Note: When you run Microsoft System Preparation Tool (Sysprep) on a BitLocker ready target, which is necessary for cloning, Sysprep deletes some vital information about the boot and the operating system partitions. It results in a reference target which cannot boot anymore. During the cloned profile creation process, Tivoli Provisioning Manager for OS Deployment can partially repair the reference target to make it boot again. However, some manual operations with Microsoft tools remain necessary to make it BitLocker ready again.

To create a cloning profile from a BitLocker ready reference target and have this reference target operational and BitLocker ready again:

1. Make sure that the disk is not encrypted.
2. Run Sysprep on the reference target
3. In the Profile Wizard, select the option to repair the reference target to enable the target to boot again.
4. Manually modify the boot and operating system partitions with Microsoft tools to make the partition scheme BitLocker ready again.

Alternatively, if you do not want to perform manual operations to make your reference target BitLocker ready again, you can

1. Make sure that the disk is not encrypted.
2. Run Sysprep on the reference target
3. Create the cloned system profile
4. Deploy the reference target with the newly created cloned profile which is BitLocker ready

Creating system profiles

There are distinct types of system profiles. The profile wizard guides you through the creation of system profiles for each type.

Creating an unattended setup system profile for Windows operating systems

You can install operating systems using standard installation processes in unattended mode. Unattended setup simplifies the task of preparing computers for the native mode of operation of disk cloning.

- To create a Windows system profile you must have a WinPE 3.0 deployment engine on your OS deployment server. If you do not have one yet, you can create one with the profile wizard, provided you have installed Windows AIK for Windows 7 in English on the computer on which you create the WinPE 3.0 deployment engine.

Vista 2008 Windows 7

- To create an unattended Windows Vista/2008/7 setup system profile, you must use a computer running the web interface extension, where you have installed Windows AIK for Windows 7 in English, under Windows XP, Windows 2003, Windows Vista, Windows 2008, or Windows 7. You cannot run this operation on a Windows 2000 or Linux operating system. The web interface extension must be started with local administrator privileges.
- Creating an unattended Windows Vista/2008/7 installation profile with multiple CDs is not supported. You are required to use a single DVD.
- You can prepare your profile to be ready for Microsoft BitLocker Drive Encryption (BitLocker). You must have at least two partitions:
  - A partition of at least 1.5 GB is necessary to hold BitLocker and to serve as a boot partition
  - A second partition holds the operating system

  Depending on the number of partitions already created, the Profile Wizard offers to reserve one of the existing partitions for BitLocker, or to create a new one.

2003 The Windows 2003 R2 operating system is distributed on two CDs. To create a fully deployable unattended system profile of Windows 2003 R2, you must:

1. Create a system profile using the first CD only, following the steps in the wizard;
2. Create a software module with the content of the second CD (see "Creating a software module for unattended deployment of Windows 2003 R2 operating system" on page 62);
3. Bind this software module (with an automatic binding rule) to the system profile you just created.

To create a new system profile:

1. Go to **Server** → **OS deployment** → **System profiles**.
2. Click **New Profile**. A system profile wizard opens to guide you through the steps of creating a profile.
3. Select **Unattended setup** in the first pane of the profile wizard.
4. Select your operating system from the list and click **Next**.
5. Follow the instruction of the profile wizard. Vista 2008 Windows 7 If you have a volume license, select **Volume licensing, no product key required** on the product key screen of the wizard.

   Note: Tivoli Provisioning Manager for OS Deployment supports the Key Management Service (KMS) key only. If you have a Multiple Activation Key,

select **Volume licensing, no product key required** on the product key screen of the wizard. To activate your Windows license, bind the Windows Server License Manager Script (`slmgr.vbs`) as a software module to the system profile you just created.

When your first unattended installation profile is created, you can use it to deploy targets. Then you can create a cloning-mode system profile, because unattended installation profiles have a longer deployment time than cloning-mode system profiles. You can use your unattended installation profile to prepare the computer that you refer to when creating your first cloning-mode system profile.

Creating a cloning mode system profile for Windows operating system

To obtain a cloning-mode system profile from a reference target you must first prepare the reference target.

To clone a Windows operating system, your reference target must have at least 1 GB RAM.

To create a Windows system profile you must have a WinPE 3.0 deployment engine on your OS deployment server. If you do not have one yet, you can create one with the profile wizard, provided you have installed Windows AIK for Windows 7 in English on the computer on which you create the WinPE 3.0 deployment engine.

For the actual driver injection, you must use a computer running the web interface extension, where you have installed Windows AIK for Windows 7 in English, under Windows XP, Windows 2003, Windows Vista, Windows 2008, or Windows 7. You cannot run this operation on a Windows 2000 or Linux operating system. The web interface extension must be started with local administrator privileges.

1. Prepare the reference target.
2. Clone the reference target.

Preparing the reference target:

To create a cloning-mode system profile, you must first create the reference OS configuration (called the *system profile*) that you want to deploy.

You must perform this task on the reference target not on the OS deployment server.

The OS deployment server does not perform cleanup on the reference target. You are responsible for deleting useless files and services before creating a new image as follows:

- Delete the temporary Internet cache
- Delete your temporary directories and files
- Disconnect your network drives and remote printers
- Empty the recycle bin
- Delete partitions using a file system that is not supported by the product, or reformat them

Running Sysprep:

Before you can create a cloning-mode system profile for Windows operating system, you must run Microsoft System Preparation Tool (Sysprep). Where to find Sysprep and how to use it varies slightly depending on the Windows version.

Running Sysprep on Windows Vista/2008/7 operating systems:

Before cloning your Windows Vista/2008/7 image, run Sysprep to prepare your system for cloning. Tivoli Provisioning Manager for OS Deployment works with Sysprep to automate the post-cloning reconfiguration.

Sysprep cannot be used on targets that are part of a domain. The system profile image must be made on a target hat does not belong to a domain. Even if your operating system was part of the domain before you launched Sysprep, Sysprep removes it from the domain. Later, you can automatically join a domain during the deployment process.

Before running Sysprep, you must configure your target to use DHCP. Go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Then click on **Common networking info**. If your target uses a static IP address, you have a high risk IP conflicts when the target boots for the first time and it has not yet applied all Sysprep settings.

With Windows Vista/2008/7, you can run Microsoft System Preparation Tool (Sysprep) on the operating system only three times. After that, the Sysprep tool refuses to start, therefore always start from your original reference image. To work around this issue, you can also use a virtual machine.

Sysprep is available on every installed Windows Vista/2008/7 operating system. The Sysprep executable file is archived in `c:\windows\system32\sysprep\sysprep.exe`.

To start the Sysprep process, follow these instructions:

1. Log on as a user with administrator privileges.
2. Close any open applications and type the run command in the Windows Vista/2008/7 **Start Search** command prompt.
3. When the run command prompt opens, browse to the Sysprep executable file and click **OK**. A System Preparation Tool page opens.
4. From the System Cleanup action menu, select **Enter System Out-of-Box Experience (OOBE)**.
5. Select the **Generalize** check box.
6. From the **Shutdown Options menu**, select **Shutdown**.
7. Click **OK**. After a few seconds, your system shuts down automatically.

   Alternatively, you can specify these options when launching Sysprep from the command line prompt by running the command: `c:\windows\system32\sysprep\sysprep.exe /oobe /generalize /shutdown`.

   Note:

   - Sysprep can also be used in audit mode. In audit mode, when the user first boots the deployed machine, the boot process does an Out-Of-Box Experience (OOBE) stage which finalizes the OS configuration taking connected peripherals into account. This OOBE stage takes about 10 minutes. If Sysprep is used in OOBE mode, this stage is performed during deployment without significantly increasing the deployment time.

- It is possible to have a partition dedicated to Microsoft BitLocker Drive Encryption (BitLocker).
  - If the reference computer you are cloning is BitLocker ready, running Sysprep prevents it to boot anymore.The product can correct this error and allow the computer to boot again by assigning the operating system partition as boot partition. However, if you want to use BitLocker on the reference target afterward, you must manually change the boot partition back to the BitLocker partition.The product properly configures boot and root partitions on deployed computers. Thus, computers deployed with an image cloned from a BitLocker ready computer are perfectly bootable and BitLocker ready.
  - If the reference computer is not BitLocker ready, running Sysprep does not raise any difficulty. With the Profile Wizard, you can make the cloned image BitLocker ready by assigning or creating a BitLocker partition of at least 1.5 GB.

Running Sysprep on Windows XP and Windows 2003 operating systems:

Before cloning your Windows image, run Sysprep to prepare your system to be cloned. Tivoli Provisioning Manager for OS Deployment works with Sysprep to automate the post-cloning reconfiguration.

Sysprep cannot be used on targets that are part of a domain. The system profile image must be made on a target hat does not belong to a domain. Even if your operating system was part of the domain before you launched Sysprep, Sysprep removes it from the domain. Later, you can automatically join a domain during the deployment process.

Before running Sysprep, you must configure your target to use DHCP. Go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Then click on **Common networking info**. If your target uses a static IP address, you have a high risk IP conflicts when the target boots for the first time and it has not yet applied all Sysprep settings.

Depending on how Windows was installed, you might never have logged on as an administrator. If this is the case, log out and log in again as an administrator to ensure that the administrator profile is properly created. Otherwise, you might not be able to create system snapshots affecting the administrator settings.

Sysprep for Windows XP is included on the Windows XP Professional CD, and archived in the file `\Support\Tools\Deploy.cab`.

To run Sysprep:

1. Copy all the Sysprep executable files into a folder named `c:\sysprep`.
2. Close all your applications.
3. Run the command `c:\sysprep\sysprep.exe -mini -forceshutdown -reseal` from the **Start > Run** menu.

   Alternatively, you can start Sysprep with a graphical user interface by double-clicking on its icon
   a. Make sure that **Mini Setup** is checked
   b. Click **Reseal**.

   Your system shuts down automatically after a few seconds.

Running Sysprep on Windows 2000 operating system:

Before cloning your Windows 2000 image, run Microsoft system preparation tool (Sysprep) to prepare your system to be cloned. Tivoli Provisioning Manager for OS Deployment works with Sysprep to automate the post-cloning reconfiguration.

Sysprep cannot be used on targets that are part of a domain. The system profile image must be made on a target hat does not belong to a domain. Even if your operating system was part of the domain before you launched Sysprep, Sysprep removes it from the domain. Later, you can automatically join a domain during the deployment process.

Before running Sysprep, you must configure your target to use DHCP. Go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Then click on **Common networking info**. If your target uses a static IP address, you have a high risk IP conflicts when the target boots for the first time and it has not yet applied all Sysprep settings.

Sysprep for Windows 2000 is included in Windows 2000 Resource Kit, and is also available on the Microsoft Web site.

1. Copy all the Sysprep executable files into a folder named `c:\sysprep`.
2. Close all your applications.
3. Run the command `sysprep.exe` from the **Start > Run** menu. Your system shuts down automatically after a few seconds (if it does not, wait a minute or so and then turn it off).

Cloning the reference computer:

When you have prepared your reference computer, you are ready to create your system profile. You can create it from the Web interface with the profile wizard.

1. Go to **Server → OS deployment → System Profiles**.
2. Click **New profile**
3. Select **Cloning from a reference computer** and click **Next**
4. Follow the instruction of the profile wizard. If you have a volume license, select **Volume licensing, no product key required** on the product key screen of the wizard.

   Note: Tivoli Provisioning Manager for OS Deployment supports the Key Management Service (KMS) key only. If you have a Multiple Activation Key, select **Volume licensing, no product key required** on the product key screen of the wizard. To activate your Windows license, bind the Windows Server License Manager Script (`slmgr.vbs`) as a software module to the system profile you just created.

The **Volume licensing, no product key required** option does not work for all versions of Windows operating systems. If you are asked for a deployment key during deployment, go to **Server → System profiles → Profile details → OS configuration details**, click on the **Windows** tab and set **Volume Licensing** to **No**. You must then populate **Product key**. Your cloning system profile should now deploy without userinteraction.

Creating a system snapshot:

You can clone a computer with a Windows operating system without running the Sysprep tool. Such a profile is called a system snapshot. You can create it from the web interface.

1. Go to **Server → OS deployment → System Profiles**.
2. Click **New profile**.
3. Enter the IP address of the target you want to clone and click **Next**.
4. The Profile Wizard detects the operating system. Click **Next**.
5. The Profile Wizard detects that the Windows operating system has not been prepared with Sysprep. Review the warnings carefully and, if you still intend to create a system snapshot, select **I understand these limitations but I want to proceed as is** and click **Next**.
6. Follow the instructions of the wizard.

System snapshot:

You can create Windows cloning profiles without using Sysprep to prepare your reference target. Such a profile is called a system snapshot.

The purpose of creating a system snapshot is to keep a copy of a golden parent reference target before it is altered by Sysprep, thus enabling you to restore your golden parent exactly as it was before the Sysprep tool was used.

Note:

1. The product had not been designed as a backup product.
   - Do not create and restore Windows system snapshots as a backup method.
   - Do not create more than a few Windows system snapshots on any OS deployment server.
2. You cannot deploy a Windows system snapshot, you can only restore it exactly as it was created.
   - It is not possible to customize system snapshots.
   - Profile restoration does not allow the installation of software modules, including driver packages.

Creating a system profile from a reference image

You can create a system profile using a WIM image.

- The WIM image must contain only one partition. If you have two partitions in your WIM image, for example, a boot partition and a separate root partition, deployment of the cloning WIM system profile fails.
- To create a Windows system profile you must have a WinPE 3.0 deployment engine on your OS deployment server. If you do not have one yet, you can create one with the profile wizard, provided you have installed Windows AIK for Windows 7 in English on the computer on which you create the WinPE 3.0 deployment engine.
- To create a cloning profile with a Windows WIM image, you must use a computer running the web interface extension, where you have installed Windows AIK for Windows 7 in English, under Windows XP, Windows 2003, Windows Vista, Windows 2008, or Windows 7. You cannot run this operation on a Windows 2000 or Linux operating system. The web interface extension must be started with local administrator privileges.
- Creating a cloning profile from a Windows WIM image stored on multiple CDs is not supported. You are required to use a single DVD.
- You can prepare your profile to be ready for Microsoft BitLocker Drive Encryption (BitLocker). You must have at least two partitions:
  – A partition of at least 1.5 GB is necessary to hold BitLocker and to serve as a boot partition

– A second partition holds the operating system

Depending on the number of partitions already created, the Profile Wizard offers to reserve one of the existing partitions for BitLocker, or to create a new one.

To create a system profile from a reference image, you must follow these steps:

1. Go to **Server → OS deployment → System Profiles**.
2. Click **New Profile**. This opens a system profile wizard that guides you through the steps of creating a profile.
3. Select **Cloning from a reference image file** and click **Next**.
4. Select the corresponding image format and click **Next**.
5. Follow the instruction of the profile wizard. If you have a volume license, select **Volume licensing, no product key required** on the product key screen of the wizard.

   Note: Tivoli Provisioning Manager for OS Deployment supports the Key Management Service (KMS) key only. If you have a Multiple Activation Key, select **Volume licensing, no product key required** on the product key screen of the wizard. To activate your Windows license, bind the Windows Server License Manager Script (`slmgr.vbs`) as a software module to the system profile you just created.

Creating a universal system profile for Windows operating systems

When creating a software module, do not enter a hardware model because a universal system profile must be deployable on several types of hardware. If you entered a model name in the Profile Wizard, you can delete it when you edit the first set of parameters of the Profile details.

To successfully deploy your universal system profile with another type of hard disk from your reference target (for example from a parallel hard disk to a SCSI or an AHCI disk), you must inject the drivers during deployment.

There are two different scenarios

- In an unattended setup system profile, the driver MUST be injected as a created by Tivoli Provisioning Manager for OS Deployment.
- In a cloning system profile, the driver might be injected as a software module created by Tivoli Provisioning Manager for OS Deployment. If this method fails, you can use the Microsoft "Sysprep" tool.

Here are the solutions for these two different scenarios.

Deploying an unattended setup:

When a driver needs to be installed during the early stages of Windows unattended setup, you must use TEXTMODE drivers. Perform these steps:

1. Ensure that the driver files are on your server. The file txtsetup.oem must be in the driver folder. This file is provided by the hardware vendor.
2. Create a software module, type driver. Typically, Tivoli Provisioning Manager for OS Deployment recognizes that this is a TEXTMODE driver and completes the fields automatically. The installation stage must be “When the OS is installed”.
3. Bind your software module to your target profile, or bind it automatically to the hardware.

4. You can now deploy your unattended profile with the software module.

The above solution might not work when deploying a cloned system profile because Windows setup does not use the same mechanisms as Windows Sysprep for handling mass storage drivers.

Deploying a cloning system profile:

Tip: The easiest and safest solution to deploy a system profile is to start from a computer which has similar hardware to the target system, in particular regarding mass storage drivers. This will save you time and make the process deployment easier to understand and follow.

If the driver injection using the mechanism of a software module created by Tivoli Provisioning Manager for OS Deployment fails, you can use the Microsoft Sysprep tool.

You must inject the driver into the parent system profile, by performing some extra steps on the source computer before running the Sysprep tool to reseal the system profile. Prepare the drivers that you want to inject into your clone system profile in a separate folder. Perform these steps:

1. Place all the driver files on the source computer into a `C:\drivers\MyDiskController` folder.
2. Create a "Sysprep.inf" file that you place in the `c:\Sysprep` folder with the correct settings.
3. Run Sysprep.
4. Capture the cloned system profile.

When deploying the system profile, the driver injected on the target system is automatically enabled when the system starts up.

The source computer uses an EIDE controller (any type).

The target system uses an IBM ServRAID 8i controller: 0x9005(VendorID) 0x0285(DeviceID)

Copy all driver files for this controller into `arcsas.sys` and `arcsas.inf`, including all the files referenced from this file.

In your `Sysprep.inf` file, copy the following section:

```
[Sysprepmassstorage]PCI\VEN_9005&DEV_0285&SUBSYS_02f21014="%SystemDrive%\
drivers\ServeRAID8i\arcsas.inf", "\", "IBM ServeRAID 8i Controller",
"\arcsas.sys"
```

Tip:

- All the PCI numbers can be found in the web interface, in the hardware inventory tab of the target.
- To determine the appropriate driver, check the **PCI_VENxxxx&DEV_xxxx** string in the driver inf file and match it with the data reported on target hardware inventory, as reported by TPM for OS deployment. The SUBSYS_yyyyzzzz must also match the **SubVendorID (yyyy)** and **SubDeviceID (zzzz)**.
- The PCI\... key that you add to your Sysprep.inf file must be an exact copy of the one used in the driver .inf file.

Organizing and editing system profiles

After you have created a system profile, you can view it on the OS deployment server through the web interface. The profiles are listed on **> Server > OS deployment > Profiles** page, in the **System profiles** pane. Each blue jacket represents a system profile (that is, the hard-disk partition images).

If you want to organize your system profiles, you can create subfolders by following these steps:

1. Select the parent folder with a left mouse click.
2. Call the contextual menu with a right mouse click.
3. Select the **Add a new profile folder** menu item.
4. Enter the new folder name.
5. Click **OK**.

You can then move profiles (by dragging-and-dropping the profile icons) from the top folder, where they are automatically created, to the appropriate subfolder.

Editing system profiles

To display and edit the parameters associated with a given profile:

1. Double-click a system profile to open the **Profile details** page.
2. Click **Edit** on top of the parameter sections to edit the parameters.

Updating a system profile with a Language Pack or a HotFix

Vista | 2008 | Windows 7

Windows Vista/2008/7 system profiles can be updated to include a language pack or a HotFix.

To update a system profile to include either a Language Pack, or a HotFix, or both, you need an available target on which the profile will be updated.

Note:

- To update an unattended setup profile or a WIM cloning profile, you must use a computer running the web interface extension, where you have installed Windows AIK for Windows 7 in English, under Windows XP, Windows 2003, Windows Vista, Windows 2008, or Windows 7. You cannot run this operation on a Windows 2000 or Linux operating system. The web interface extension must be started with local administrator privileges.
- If you want to update a cloning system profile, the disk content of the target you will use for the update will be deleted. Make sure you use a bare-metal target or a target with no valuable content on its disks.

1. Go to **Server** → **OS deployment** → **System profiles**. Double-click on a profile to view the details.
2. Click **Update** to open the update wizard.
3. Optionally, select **Update similar profiles** to update additional system profiles at the same time. Only system profiles compatible with the current one are available for selection. Unattended setup system profiles and WIM cloning system profiles cannot be updated together with cloning system profiles.
4. Follow the wizard instructions. Depending on the type of system profile, the wizard analyses the state of the target to ensure that all prerequisites are met. If

all prerequisites are met, a new system profile is created, the old system profile taken as basis is kept. The name of the new profile is the name of the basis system profile with (updated) appended to it.

Browsing partition files

You can browse partition images stored on your server.

1. Go to **Server → OS deployment → System profiles**. Double-click on a profile to view the details.
2. In the **Original partition layout** section, click **Browse image of primary partition 1**.
3. You can expand or update the whole partition or a part of it.
   - To expand the whole or part of the partition:
     a. Right-click the folder you want and select **Expand on local disk**.
     b. Choose the computer where you want to expand and store the files contained in the selected partition.
     c. Specify the destination folder where to extract the partition files.
     d. Follow the instructions of the image wizard to expand the partition.

     Note: You must expand the partition to an empty directory. If you select a folder that is not empty the extraction fails.
   - To update the whole or part of the partition:
     a. Right-click the folder you want and select **Update from local disk**.
     b. Specify the source folder of the OS deployment server where your updated data are located.
     c. Specify the destination folder where to extract the partition files.
     d. Follow the instructions of the image wizard to update the partition.

On the **Partition image explorer** page, you can create a new directory by selecting **Add new directory** in the contextual menu. You can also modify or add files by selecting **Upload file** in the contextual menu.

Note: File upload is limited to 16 MB.

Changing the partition layout

Partition layout can be updated to resize partitions, assign mount points, change the file system.

Changing the partition layout in system profiles might render the profile unusable. It is recommended not to change the partition layout in system profiles, unless you know that the changes you want to make have no side effect.

In any case, do not transform a primary partition into a logical partition.

Note: Changing the partition layout from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose either one or the other entry point, and then perform all your changes from that entry point.

Editing the partition layout allows you to:

- Add or delete partitions.

Note: Adding or deleting partitions can lead to OS configuration problems, therefore this feature must only be used very carefully. To provide a better description to your profile, use the **Comment** field to write all necessary details.

- Resize a partition by dragging sliders, or by assigning it an absolute or relative size.
- Change the file system of a partition.
- Assign a mount point to the partition.

1. Click **Edit partition layout** on either the **Profile details** page or the **OS configuration details** page, **Disks** tab.
2.
   - To add a partition:
     a. Click **Modify partition layout**.
     b. Click into an existing partition.
     c. Click **Add a partition** in the contextual menu.
     d. Indicate the partition properties, including a mount point and click **OK**.

        Windows In a Windows profile, if you do not provide a drive letter as mount point, one is assigned randomly. Letter *W* is not available.

        Linux In a Linux profile, do not forget to assign a mount point for the new partition. To be valid, this mount point must reference an existing directory in the main image.
   - To resize partitions with the sliders, grab the slider to the right of the partition and drag it.
   - To update all other parameters, select a partition by clicking on it, and select **Edit partition** in the contextual menu.

Modified partitions are aligned on megabytes rather than on cylinders. The following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

If you want to use the same system profile with two different partition schemes, you can also duplicate a system profile by right-clicking the profile name and selecting **Duplicate profile**. The copy shares the same image files, but can have a different partition layout.

Updating device mapping

Device mapping can be updated to force logical disks to point to specific physical devices.

Note: Updating device mapping from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose one or the other entry point, and then perform all your changes from that entry point.

1. Go to **Server** → **OS deployment** → **System profiles** → **Profile details** or to **Server** → **OS deployment** → **System profiles** → **Profile details** → **OS configuration details**, **Disks** tab.
2. Click **Modify device mapping**.

3. Select to which physical device you want to map your logical disk. The column starting with **Disk 0** corresponds to an automatic detection of the first to the eighth disk, the column starting with **/dev/hda** corresponds to standard device names.

   Note: Spanning a logical disk on several physical disks is not currently available.
4. Click **OK**.
5. Repeat step 2 on page 43 to step 4 for each logical disk for which you want to update device mapping.

If the new device mapping you selected is incorrect, you receive a warning message.

OS configurations and fixed common parameters

A system profile is the partition layout and list of files to deploy, while OS configurations are operating system parameters.

At the very bottom of the **Profile details** page, there is a list of the OS configurations that correspond to your profile.

You can define several OS configurations for each system profile and duplicate them. These copies share the same image files, and the same partition layout, but can have different target parameters. You must then assign new values to some of the OS configurations parameters to make the original OS configuration and its copies distinct.

If you want to automate the assignment of parameters to targets, you can view and edit the OS configuration you are about to deploy by clicking on its name in the **Profile details** page. You are now in the **OS configuration details** page. The information is divided into panels, each displaying sets of parameters. You can modify the parameters either through the web interface or by using a parameter file.

Editing OS configuration parameters in the web interface for Windows operating systems

The web interface displays a number of OS configuration parameters divided into panes. These parameters can be edited in the web interface.

To edit parameters:

1. Click a tab to select the corresponding pane.
2. Click **Edit** on the banner of the section where you want to update parameters.
3. Modify the values.
4. Click **OK**.

Editing a Windows parameter file

You can modify OS configuration parameters by editing a file. This option allows you to modify parameters that are not displayed in the web interface. However, you must be experienced to use this option advantageously, because Tivoli Provisioning Manager for OS Deployment does not provide any syntax checking of the file. Information about the file format and syntax can be found in the documentation of the operating system itself.

1. To edit the file click **Edit custom 'unattend.xml'** on Windows Vista, Windows 2008, Windows 7 and click **Edit custom 'sysprep.inf'** on Windows XP, Windows 2003.
2. Type the parameters and their values in the syntax requested by the operating system, or copy and paste it from another editor.
3. Click **OK**.

Tivoli Provisioning Manager for OS Deployment merges the information of the edited file with the information provided on the web interface (default file). Unless otherwise specified, parameters specified in the default file override the content of the custom file.

Vista 2008 Windows 7 Content of this custom file overrides the default one created for the following parameters, which are written as tags:

- `NetworkLocation`
- `ProtectYourPC`
- `PersistAllDeviceInstalls`
- `UILanguage`
- `SystemLocale`
- `UserLocale`
- `InputLocale`

Content of the custom file is integrated within the default file for the following parameters: Value 1 is used in the default file and must not be used for these tags and attributes combinations in the custom file.

- `RunSynchronousCommand`, which takes a daughter tag `<order>`*value*`</order>`
- `LocalAccount`
- `Interface`
- `PathAndCredentials`, with the attribute `keyValue="`*value*`"`
- `DomainName`, with the attribute `keyValue="`*value*`"`
- `IPAddress`, with the attribute `keyValue="`*value*`"`

Only one tag `<component>` with attribute `processorArchitecture` is allowed. For all other tags, the values of the default file created override what the user has written in the custom file.

Troubleshooting:

If the OS configurations in the deployed operating system are not what you expected, you must examine carefully the parameter files. They are the result of the merge between the custom file and the default file created. See the log file `Windows/Panther/unattendGC/setupact.log` for problems in the file merge.

Note: Ensure you specify the full paths for the commands you use in the `unatted.xml` file.

Vista 2008 Windows 7 To troubleshoot OS configuration parameters after a successful deployment, view the two files `Windows/panther/setup.xml` and `Windows/panther/unattend.xml` which are the result of the merge between the default and custom parameter files. To troubleshoot OS configuration parameters after a failed deployment, you must look for the following files in the partition containing the operating system:

- `user_unattend.xml`, which is the file you edited

- `setup.xml`, which results from the merge
- `unattend.xml`, which results from the merge as well

XP 2003 To troubleshoot OS configuration parameters after a failed deployment, you must look for `$WIN_NT$.~BT\winnt.sif` in the partition containing the operating system. This file contains the information merged from the custom and the default files.

Binding drivers to a Windows system profile

When a system profile does not contain the drivers needed for deployment, you must bind these drivers to the system profile to be able to deploy it and obtain a working operating system.

If you encounter problem with the built-in drivers contained in your system profile, if some drivers are not bound, or if some drivers are missing, you should bind other drivers to your system profile.

You can only bind drivers to your system profile that are software modules in your OS deployment server. You must therefore create driver software modules from the drivers that you want to bind to your system profile.

Note: There are two methods to bind driver software modules to a system profile:

- the *standard binding rule method* where you can indicate profiles to bind to a software module.
- the *driver specific binding rule method* where you bind drivers per system profile and target model/device pair.

You can switch from one method to the other. In the *driver specific binding rule method*, driver bindings from the *standard binding rule method* are ignored, and vice-versa.

The method described here is the *driver specific binding rule method*.

From version 7.1.1.3 of the product onwards, it is recommended to use the *driver specific binding rule method*, which is the method by default on all new Windows system profiles.

The product helps you select appropriate drivers for particular target models. It helps you to predict potential problems and to solve them. It does not guaranty that a specific system profile, with bound drivers, works with a given target.

The information used by the OS deployment server to predict the compatibility of a driver with a target model is taken from the content provided by the vendor in its driver. The OS deployment server cannot verify the accuracy of this information.

1. Check the compatibility of your system profile.
   a. Go to **Server** → **OS deployment** → **System profiles**.
   b. To view the details of the system profile, you have two options.
      - Double-click on it.
      - Select a system profile, and then select **View profile** in the contextual menu.

c. *(Optional)* In the section **Driver handling**, click **Switch to driver specific bindings mode**. You only need to perform this step if you are in the regular software binding rule mode.

d. A check is performed while the page is loading. This may take a few minutes.

If drivers are missing, or are not bound, or if several drivers are bound for the same device, the following information is provided.

Indicates a missing critical driver, or a critical driver of the wrong architecture.


Indicates that a missing non-critical driver, or a non-critical driver of the wrong architecture.

Indicates that a required driver is present on the OS deployment server, but that it is not bound.

Indicates that there are several drivers bound for the same device, or that there is a binding with a driver that is not known as compatible.

You can expand the line to get more information.

- For drivers missing on the OS deployment server, you find a suggestion of where to look for it, including, if available, a download link and the exact directory within the downloaded archive where the driver can be found.
- When drivers are present on the OS deployment server, you find suggestions of which driver to bind, in order of preference. If multiple drivers are known to possibly work for a device, the best choice is listed first. The choice is explained in the advice text, which first recommends the use of *device-specific drivers*, that is, drivers that have been specifically designed for the given hardware device. Then *compatible device drivers*, that match the device family, are recommended, even if they are not an exact rebranded variant (for example, as second choice, an Adaptec driver of the same family as an IBM ServerRaid adapter, if it is based on the same chipset). Finally, as third choice, *generic drivers*, for example, Microsoft generic AHCI driver for any AHCI controller, are recommended.

If no error is found, you do not need to modify the bindings.

2. Modify the driver bindings of the system profile. There are two ways to perform this.
   - Use a wizard.
     a. Click **Fix Drivers**.
     b. Follow the instructions of the wizard. After having selected a target model, you have to select one of these options:

        Automatically fix issues which can be fixed for this model.
        Fixes all issues which can be automatically fixed. Such issues include a missing binding to an existing driver, or multiple bindings for a device, for example.

        Manually fix issues for this model.
        Presents you with each issue in turn. Ways to solve the issue, when available, are proposed.

        Automatically bind drivers for this model.
        Erases every existing binding. New bindings are then automatically added.

Copy driver bindings for this model from a similar profile.
Copies all the bindings from a selected source system profile to the current one.

Reset all drivers bindings for this model.
Erases all the driver bindings, and does not create any new binding.

- Edit the bindings manually.
  a. Click **Edit profile's driver bindings** on the **Profile details** page.
  A grid is loaded.
  Columns represent target models known to the OS deployment server. They can be expanded to view their devices, provided an inventory has been performed.
  The first line represents the system profile. Other lines represent software module folders in the OS deployment server. They can be expanded to view individual drivers. If a driver can be used only for 32-bit or 64-bit machines, a superscript *x86* or *x86-64* mark is written next to the driver name. If you do not find the drivers that you need in the list provided, you should first create software modules for your drivers.
  b. *(Optional)* To obtain a summary of the errors and warnings, click the link provided above the grid. This helps you locate the problematic areas in the driver grid.
  c. Expand the columns of problematic target models to view individual devices.
  d. Expand software module folders containing drivers to view the individual drivers.

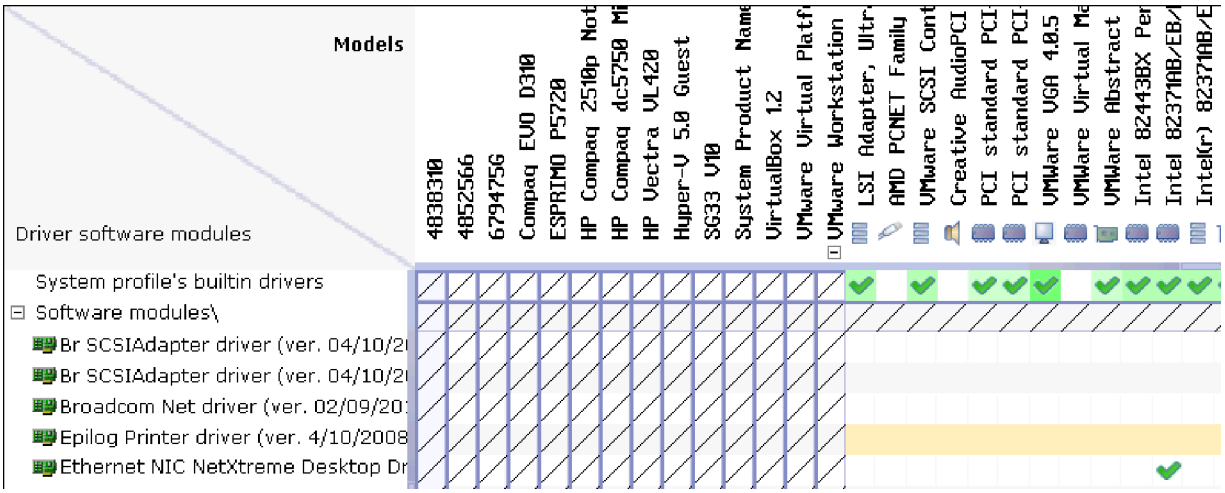


Figure 4. Driver binding grid

A cell with a green background indicates that driver information corresponds to the device. The quality of the drivers that can be selected is illustrated by the intensity of the green background: the best drivers are in intense green, the family drivers are in standard green, and the generic drivers are in pale green.

A cell with an orange background indicates either that the driver is not a PCI driver, or that there is no compatibility information available for the driver.

A cell with a green check mark indicates that the driver is bound to the system profile for use with the specific target model and device.

e. Click on a green background cell to add or remove bindings.

   It is not possible to bind or unbind drivers from the system profile itself, because they are built-in drivers.

   You should have one, and only one, check mark per column, indicating that you have one and only one driver for each device.

f. When you are done modifying the bindings, click **Save**.

g. To return to the **Profile details** page, click **Back**.

   Potential problems with the image are recomputed, allowing you to check if your modifications have solved the detected problems.

When you have solved all the driver binding issues, you can deploy targets with your system profile.

Restoring a system profile manually

If you want to check that your cloning system profile contains all necessary information, you can restore it manually either from the web interface, or from the client computer.

Note:

1. You can only restore Windows cloning system profiles.
2. System profile restoration works only on targets of the same model as the one on which the profile was created. Restoring a profile on another model of targets results in strange side effects.
3. When you restore a system profile manually, the image is restored as-is, without any automatic parametrization. Thus, restoration cannot be unattended as some parameter values are required and must be entered manually.
4. Creating a cloned profile and restoring it manually is not meant as a backup procedure and it should not be used in that way.
5. A cloned profile can be restored on one target only at a time. Restoration cannot be performed on several targets together.

To perform the manual restoration:

1. Select the wanted target in the **Target Monitor**.
2. Select **Deploy now** in the contextual menu.
3. Follow the wizard instructions.

Restoring a profile from the web interface

1. Go to the **Target Monitor** page.
2. Select a single target
3. In the contextual menu, select **Additional features**
4. Select **Restore a profile**
5. Click **Next** and follow the instructions of the wizard.

Restoring a profile from the target

1. Click the icon to restore an image.
2. Click the **Restore a system profile** icon.

   Depending on the types of images on your OS deployment server, you can also get icons for

- Restoring a software snapshot
- Restoring a virtual floppy-disk

3. Select a system profile or a software snapshot from the list provided and click **Next**.
4. Optionally select options and click **Next** to restore the profile.
   - Windows If the system profile is a Windows image, it can include the Sysprep mini-setup wizard that is typically used to perform some post-configuration on the image.

     2000 2003 XP You can disable this mini-setup wizard for Windows 2000/2003/XP if you want to start the operating system and do some modifications before reinstalling Sysprep manually. In this case, a warning message appears, telling you that some minimal post-configuration are applied anyway, to avoid the risks of potential conflicts. This option is not available on Windows 2008/Vista.
   - Some computers can have been delivered with protected partitions for emergency restore backups. At this stage, the option is given to restore protected partitions or not.
   - Additionally, if a CMOS image was included in the system profile at the time it was created, you can decide whether you want to restore it. Remember that restoring a CMOS image on a target different (or with a different BIOS version) than the original can severely damage the target.

Software modules for Windows operating systems

Software modules are images other than system profiles that can be created to address various needs.

Tivoli Provisioning Manager for OS Deployment is based on imaging technology. As administrator, you create images of components that you want to see on every target, and the automated deployment merges and restores these images on each target, automatically, when needed.

Tivoli Provisioning Manager for OS Deployment can handle most scenarios for software deployment and post-installation configuration.

Types of software modules

There are many types of software modules. Depending on the type of package and installation files, the wizard guides you through the different steps to achieve your software module with minimal effort. The types of software package supported by the wizard are listed in this section.

- Vista 2008 Windows 7 **Language pack**
- Vista 2008 Windows 7 **HotFix (MSU)**
- **A Windows application installation, using Microsoft Installer (MSI)**
- **A Windows driver to include in a deployment**
- XP 2003 **A Windows HAL to include in a clone deployment**
- **A custom action on the targets**. This includes OS configuration changes such as registry patches, commands to be run, and copying sets of files on the target.

WinPE and its uses

WinPE is widely used in all the tasks pertaining to the deployment of Windows operating systems. The product uses two different kinds of WinPE 3.0, depending on the tasks at hand.

Types of WinPE

Windows Preinstallation Environment (WinPE) is a group of files which can be loaded as a ramdisk and which allows you to perform operations on a target.

WinPE 3.0 deployment engine
This WinPE 3.0 is a prerequisite to create Windows system profiles and to deploy them.

To create a WinPE 3.0 deployment engine, you need a computer running a Windows operating system, with Windows AIK for Windows 7 in English installed and running the web interface extension.

WinPE 3.0 deployment engine creation always creates a 32-bit and a 64-bit deployment engines. The 64-bit WinPE 3.0 deployment engine is used only to deploy unattended setup of Windows 2008 64-bit GA operating system.

WinPE hardware environment
This type of WinPE is used for hardware configurations.

To create a WinPE 3.0 hardware environment, you need to start the vendor commands on a computer running a Windows operating system, with Windows AIK for Windows 7 in English installed, and the web interface extension running. You need to start the vendor commands before you start the web interface extension.

It is possible to create WinPE 1 and WinPE 2 hardware environments.

WinPE2 ramdisk
WinPE2 ramdisks are obsolete from version 7.1.1.3 of the product onwards. You may keep those that were created with an earlier version of the product, or safely delete them. You cannot create new ones.

WAIK

Windows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.

Note: Windows Automated Installation Kit for Windows Vista and Windows Server 2008 is not supported anymore. Use Windows Automated Installation Kit (AIK) for Windows 7 in English only.

You must restart your computer after having installed Windows AIK.

Good practice

If you deploy Windows operating systems, you need to create 32-bit and 64-bit WinPE 3.0 deployment engines, and potentially WinPE 3.0 hardware configurations. For each of these creations, you need a computer running a Windows operating system, with Windows Automated Installation Kit (AIK) for Windows 7 in English and the web interface extension installed. The same configuration is also needed to update Windows Vista/2008/7 system profiles.

Windows AIK for Windows 7 in English can be obtained free of charge from Microsoft, but it is rather heavy and cumbersome to install. Therefore, it is good practice to install Windows AIK and the web interface extension on a dedicated computer running a Windows operating system and to perform all operations requiring this configuration on this dedicated computer.

If your OS deployment server runs under a Windows operating system, consider making your OS deployment server the dedicated Windows computer.

Creating software modules

There are distinct types of software modules which vary according to the operating system being deployed. The software wizard guides you through the creation of software modules for each type.

Creating a Language Pack software module

Windows Language packs can be created only from a computer with a Windows operating system and running the web interface extension.

The directory containing language pack files must contain a file with a `.cab` extension.

1. Go to **Server → OS deployment → Software modules**.
2. Click **New software** to run the software wizard.
3. Select **Windows Vista/2008/7** and click **Next**.
4. Select **Language pack** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

   You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

Creating a HotFix software module

Windows HotFixes can be created only from a computer with a Windows operating system and running the web interface extension.

The directory containing the HotFix files must contain a file with a `.msu` extension.

1. Go to **Server → OS deployment → Software modules**.

2. Click **New software** to run the software wizard.
3. Select **Windows Vista/2008/7** and click **Next**.
4. Select **HotFix (MSU)** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

Creating a Microsoft Software Installer (MSI) software module

MSI software modules can be created only

- locally with a provisioning server installed on a Windows 2000/2003/2008 operating system
- from a computer with aWindows 2000/2003/2008/XP/Vista/7 operating system and running the web interface extension.

The directory containing MSI files must contain a file with a `.msi` extension. If the MSI file is located on the provisioning server, you must have placed it in a subdirectory of the `import` directory.

Note: If the folder you are looking for is not on the local computer, the provisioning server, or on another computer running the web interface extension, you might still be able to access the wanted resource using the following procedure:

Windows **Windows**

1. Create a `.lnk.yourfilename` file (where `yourfilename` is the name of your choice) that contains the path to the wanted folder (for example, `\\fileserver\export\softs\`).
2. In the wizard, enter `.lnk.yourfilename` preceded by the appropriate path.

UNIX **UNIX**

Use symlinks.

To create your software module

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select **Windows Vista/2008/7** or **Windows 2000/2003/XP** and click **Next**.

4. Select **A Windows application installation, using Microsoft Installer (MSI)** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the software module at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The path to where the installation files are restored on the target. This path is relative to the system root partition.
   - An additional command line that might be necessary to install your software module. When possible, the wizard automatically suggests the appropriate command line to run the installation unattended. However, you might need to add some additional parameters to the command.

     For example, you can specify an hour parameter to cancel an activity, if the activity does not complete before the end of the specified time. The parameter syntax format is `<=xh`, where `x` is an integer representing the number of hours after which the activity is canceled. In the following example you can specify to cancel an application installation if the installation process has not completed after one hour, by adding `<=1h` at the end of the command line:

     ```
     install.exe /sPB /rs /rps /l <=1h
     ```
   - The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

Creating a driver software module for Windows operating system

A driver package is used to provide the appropriate driver files to Sysprep or Windows unattended installation to install devices that are not activated by Windows because the driver is not present in the system profile.

The directory containing the driver files must contain a file with a `.inf` extension.

Driver packages are best used with unattended setup profiles, because standard Windows installation files do not always contain the drivers for recent hardware, and the goal of unattended setup is to have the target fully installed at the end of the process. However, driver packages can also be used with cloning-mode system

profiles, because Sysprep can use driver packages to install new devices. There is no need to run Sysprep in `PnP mode` to have new devices installed when Sysprep runs on the target.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select **Windows Vista/2008/7** or **Windows 2000/2003/XP** and click **Next**.
4. Select **A Windows driver to include in a deployment** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   When you indicate the directory in which the driver files are located, if several sub-directories contain drivers, the wizard lists all these directories. You must then select one or several directories. Selecting multiple directories allows you to create several driver packages at the same time with common binding rules.

   In case of multiple driver package creation, you can enter a folder name in which you want to store the new software modules. If the folder does not exist, the wizard creates it.

   If only one driver package is being created, the wizard presents the characteristics of the driver. This panel is skipped in the wizard in multiple driver package creation, but you can view the information in the software module details after the package has been created.

   The wizard allows you to create binding rules based on the PCI hardware ID, the baseboard ID, the computer model name, operating system architecture and targeted operating systems. Depending on your selections, the wizard provides steps with easy-to-follow instructions to create the binding rules.

   PCI hardware ID

   - If you select **Use this driver for the exact same device only**, the PCI vendor ID, device ID, and sub-device ID must match.
   - If you select **Use this driver for similar devices**, only the PCI vendor ID and the device ID must match.

   Baseboard ID

   You can either type in a substring of the baseboard name or select baseboard names extracted from the targets known to the OS deployment server.

   Computer model name

   You can either type in a substring of the computer model name or select model names extracted from the targets known to the OS deployment server.

   OS architecture

   Select **32 bit**, **64 bit**, or **Both** if you know the architecture the driver has been designed for. Select **Auto** to use the information contained in the driver to define the binding rule.

   OS targeted

   Select for which family of Windows operating systems the driver has been written for.

   The number of rules created vary depending on the selections you made, but very quickly reaches over one hundred if your rules are based on similar PCI device IDs.

Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard for single driver creation. For

multiple driver creation, the parameters are not displayed in the wizard. They can be edited in the software details page of each driver package. These parameters include:

- A description that identifies the package in the software module tree.

  Note: The pre-filled description might not be informative enough for you to know when you can use your driver. It is recommended to update the description and to include information such as operating system and architecture. You might need to use abbreviations because the description is limited to 50 characters.
- A comment with additional information about the software module.
- The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the package at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
- A file name to store your image on the OS deployment server. Packages typically have a `.pkg` extension.
- The path to where the installation files are restored on the target. This path must start with `\drivers`, because Windows unattended installation and Sysprep look in `C:\drivers` when installing new devices.

Heuristics to select drivers to work with a WinPE deployment engine:

Drivers compatible with a WinPE deployment engine are not necessarily the same as the drivers for an operating system. In any case, software modules must be created from the drivers before they can be used in the OS deployment server.

When you create driver software modules for use with a WinPE deployment engine, it is sometimes difficult to know which drivers work with WinPE. Here are a few heuristics to locate the appropriate drivers.

- If you need drivers for servers, search the server vendors' scripting toolkit for WinPE drivers.

  For example, in the IBM ServerGuide Scripting Toolkit, look for a zip file with a name similar to `ibm_utl_tsep_2.00_winpe_i386.zip`. Extract the zip file, keeping the file structure. WinPE drivers are located under a path similar to `sgdeploy\SGTKWinPE\Drivers\WinPE_x86_2010-06-10\`.
- Select drivers for the appropriate operating system architecture.
- Prefer monolithic drivers containing only a simple `.inf` file and `.sys` file (without a CoInstaller DLL). This is typically the case with drivers provided in a RIS package.
- Prefer drivers for Windows 7 and Windows Server 2008 R2 operating systems. If these drivers are not available, you can try drivers for other Windows operating systems.
- The driver must support the correct PCI device. For example, the PCI inventory for the target shows a network card with
  - VendorID: 1111
  - SubVendorID: 2222
  - DeviceID: 3333
  - SubDeviceID: 4444

Then the drivers `.INF` files should include a line ending with `PCI\VEN_1111&DEV_3333` or a line ending with `PCI\VEN_1111&DEV_3333 &SUBSYS_44442222`.

- If you need Broadcom NetXtreme II drivers, you must get the drivers in the special RIS package.

If you group your WinPE drivers within the same software module folder, it is easier to locate them when you bind drivers to your WinPE deployment engine.

Importing drivers from the IBM Web site:

You can maintain your system device drivers and firmware at the most current levels avoiding unnecessary outages by using a new agent command based on the IBM UpdateXpress System Pack Installer tool.

This command creates a batch file that launches the IBM UpdateXpress System Pack Installer tool. You can use the batch file to detect current device driver and firmware levels, remotely retrieve the device driver and firmware updates from the IBM Web site, automatically package the drivers needed, and bind them to specific hardware models.

1. Run the following command to create the batch file:

   ```
   rad-mkuxspbatch uxsp-path dest-path (BOM | model=<type1>[,<type2>,...]
   OS=<OS1>[,<OS2>...,])
   ```

   where:

   uxsp-path
   Specify the full path to the UpdateXpress setup utility.

   dest-path
   Specify the main path where all the updates are stored together with the UpdateXpress batch file.

   BOM Specify the OS deployment server database BOM table to be scanned for detecting the updates needed.

   model=
   Specify the models used for manual updates by giving a list of model types (such as `model=4190,7971`).

   OS= Specify the operating systems used for manual updates.

   OSx Specify the operating system types: `windows`, `rhel3`, `rhel4`, `rhel5`, `sles9`, `sles10`, `all`.

   A batch file `updatexpress.bat` is created in the directory `dest-path`. The generated batch file contains:

   - Commands to acquire system packs and available updates (in particular drivers) for every model and operating system combination.
   - Commands to extract every software module.

   If you specified the BOM option, the agent command automatically scans the OS deployment server database for existing models and operating systems. You can also manually specify the models and operating systems to preload software modules of systems that currently do not exist within Tivoli Provisioning Manager for OS Deployment. You can run this batch file to import the drivers needed by using the IBM UpdateXpress System Pack Installer tool located in the directory `uxsp-path`.

2. You can then run this batch file: `dest-path\updatexpress.bat`, where `updatexpress.bat` is the name of the batch file containing the sequence of commands. It acquires and extracts the drivers needed by using the IBM UpdateXpress System Pack Installer tool located in the `uxsp-path` directory.
3. Use the Tivoli Provisioning Manager for OS Deployment web interface to create software modules of the drivers acquired and extracted in the previous step:
   a. Go to **Server → OS deployment → Software modules**.
   b. Click **New software** to run the software wizard.
   c. Select the relevant operating system and click **Next**.
   d. Select **A Windows driver to include in a deployment** and click **Next**.
   e. Select the computer and the main folder in which the driver files have been extracted (such as`..\IBM_Machine_type(7971)\OS_type(windows )`). The wizard lists all the drivers contained in this folder and its sub-folders.
   f. Select the drivers you need to package according to your hardware inventory and operating system.
   g. Specify the folder name where to store all the driver packages in a software tree structure.
   h. Select **Yes, create binding rules based on:** and then **PCI hardware ID** and **Target model name**. Click **Next**.
   i. Select **Use this driver for similar devices** and click **Next**.
   j. Specify the target machine model by selecting **the model name is one of the following** and then the model in the list.
   k. For the chosen drivers select the appropriate operating system architecture (such as 32–bit) and the targeted operating system (such as `Windows Server 2003` or `Windows 2008`) and click **Next**. The driver packages are created with the specified binding rules and grouped in the folder you specified. You can also modify the binding rules by editing the software module or you can create an additional software module with other drivers and add it to the same main folder.
   l. Click **Finish**.

Examples

Here is an example of generation of the `updatexpress.bat` file. It scans the BOM table to detect current Tivoli Provisioning Manager for OS Deployment device driver and firmware levels, remotely retrieves the device driver and firmware updates from the IBM Web site, and extract them into explicit model and operating system folders.

```
rad-mkuxspbatch d:\uxsp\uxspi300.exe d:\output BOM
```

Here is an example of generation of the `updatexpress.bat` file. It remotely retrieves the device drivers of model 4190 and 7971, for Windows, RedHat 3, and SLES 10 operating systems. It connects to the IBM Web site, to retrieve, package, and extract the drivers into explicit model and operating system folders.

```
rad-mkuxspbatch d:\uxsp\uxspi300.exe d:\output model=4190,7971
OS=windows,rhel3,sles10
```

Creating driver packages for servers running Windows operating systems:

To deploy Windows operating systems on servers most efficiently, you need up-to-date drivers which are often not included with operating system installation files. These drivers can be obtained from the vendor of the server.

Before you can create your driver packages, you must obtain the appropriate driver files.

IBM drivers
For IBM drivers, download the ServerGuide. To locate the ServerGuide, search for `ServerGuide download` in a search engine. Copy the `sguide` directory.

Note: ServerGuide is different from the ServerGuide Toolkit.

HP drivers
For HP drivers, download the SmartStart. To locate the SmartStart, search for `SmartStart download`.

Dell Drivers
To locate Dell drivers, search for `Dell drivers download`.

The following task assumes that the drivers have been copied into`Files/import` on your OS deployment server.

You might have to go through the driver package creation process several times, to create different driver package directories specific for operating systems and their architecture.

1. Go to **Server** → **OS deployment** → **Software modules**. Click **New Software**.
2. Select the relevant operating system and click **Next**.
3. Select **A Windows driver to include in a deployment** and click **Next**.
4. Select **On the server itself (in the 'import' directory)** and click **Next**.
5. Select the relevant directory. For IBM drivers for a Windows 2003 operating system, this is `sguide/w2003drv/$oem$/$1/drv`.
6. Select all relevant drivers in the list provided and click **Next**. Sometimes, several versions of the same driver are available. In this case, follow these guidelines:
   - Select drivers without alternative, even if the name is misleading.
   - Select the appropriate Windows version when there are alternatives, for instance select *win2003* rather than *win2k* or *winnt* for a Windows 2003 driver.
   - Select *server* when the alternative is between *server* and *pro*.
   - Avoid selecting drivers with *powerpc* in their name.
   - Avoid selecting drivers containing hardware abstraction layer (HAL).
   - Avoid selecting drivers with another architecture.

   Note: It is better to have a few extra drivers included in the package than to miss one.
7. Give a meaningful folder name to store your drivers, for instance `IBM ServerGuide 2003 32-bit`, and click **Next**.
8. Select **Yes, create binding rules based on:** and **PCI hardware ID**. Then click **Next**.
9. Select **Use this driver for the exact same device only** and click **Next**.
10. Select the appropriate architecture and the targeted operating system and click **Next**.
11. Click **Finish**.

Now, you can check that targetshave the correct bindings.

Checking the drivers bound to a target:

1. Go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Select the **Bindings** panel.
2. Check the OS configuration bound to the target. If it does not correspond to the operating system for which you just created drivers, you must switch to a more appropriate one.
   a. Select a OS configuration with the operating system you have just created drivers for.
   b. Go back to the Target Monitor.
   c. Double-click on the target.
   d. Select the **Bindings** panel again.
3. Make sure that there is only one disk driver. If there are several, you should delete outdated drivers from your software modules, modify the binding rules, or unbind the inappropriate driver from the target if it has been bound manually. To do so
   a. Click **Edit** in the **software bindings** header.
   b. Clear any unwanted driver and click **OK**.

Creating a software module for HAL injection on a cloning system profile

2000 2003 XP

Hardware abstraction layer (HAL) can change from one computer to another depending on whether it has a single or multiple processors and on, whether it uses Advanced Programmable Interrupt Controller (APIC) and Advanced Configuration and Power Interface (ACPI). HAL also depends on the operating system. To create universal images, you might be required to have HAL versions on your system profile different from the original.

To create a HAL software module to be injected on a cloning system profile during deployment, you must:

1. Go to **Server → OS deployment → Software modules**.
2. Click **New Software**.
3. Select **Windows 2000 / 2003/ XP**.
4. Select **A Windows HAL to include in a clone deployment**.
5. Follow the wizard instructions. Different HALs are available on Windows installation CDs. The wizard offers you to create binding rules for this HAL and pre-fills some of the data to facilitate the rule creation process.

If you did not use the wizard to create binding rules, it is recommended that you bind your HAL package now to deploy it in appropriate contexts.

Creating a software module for HAL injection on an unattended setup system profile:

HAL injection on an unattended setup system profile is typically only necessary on some very specific server systems. The server vendor must then provide you with the appropriate HAL. IBM provides HALs for its servers on the ServerGuide CD

To create a HAL software module to be injected on an unattended setup system profile during deployment, you must create a HAL software module and bind it to the corresponding system profiles.

To do this:

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New Software**.
3. Select **Windows 2000 / 2003/ XP**.
4. Select **A Windows driver to include in a deployment**.
5. Follow the wizard instructions. When asked for the driver file location, provide the path to the HAL.

If you did not use the wizard to create binding rules, it is recommended that you bind your HAL package now to deploy it in appropriate contexts.

Creating a custom action software module

Software modules can also contain custom actions to be performed on the target.

They are divided into:

- An OS configuration change to perform on the target
- A set of files to copy on the target

Configuration changes are further subdivided into:

- Copy and run a single file
- Apply a Windows registry change
- Apply a Windows `.ini` file change
- Copy a single text file
- Execute a single command file
- Boot a virtual floppy disk

  Note: Virtual floppy disk software modules can only be created from a Windows operating system running the web interface extension.

In the OS configuration change wizard screen, you can select **Activate keyword substitutions**. If you use this option, you can specify which keywords must be substituted in the software module details.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select the operating system and click **Next**.
4. Select **A custom action on the target** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the software module at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.

- The path to where the installation files are restored on the target. This path is relative to the system root partition.
- An additional command line that might be necessary to install your software module. When possible, the wizard automatically suggests the appropriate command line to run the installation unattended. However, you might need to add some additional parameters to the command.

  For example, you can specify an hour parameter to cancel an activity, if the activity does not complete before the end of the specified time. The parameter syntax format is `<=xh`, where `x` is an integer representing the number of hours after which the activity is canceled. In the following example you can specify to cancel an application installation if the installation process has not completed after one hour, by adding `<=1h` at the end of the command line:

  ```
  install.exe /sPB /rs /rps /l <=1h
  ```
- The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

The complete step-by-step process of creating a software module with the content of the second CD of a Windows 2003 R2 distribution, and the compete step-by-step process of creating a ramdisk from a bootable diskette are proposed as examples.

Repeating custom actions:

Some commands must be run every time the target boots during a deployment.

This is typically the case if you want to repeatedly connect a network share. This connection is destroyed when rebooting. You can therefore create a single software module with a `netuse` command to set the network share and set this software module to run once after each reboot, starting at a specific reboot.

This option is available for

- Windows registry changes.
- Copying and executing a single file.
- Executing a single command.

1. Create your software module.
2. Double-click on the software module name in the **Software components** page to obtain the **Software details** page
3. Click **Edit** in the title of the **Package information** section.
4. Select the installation stage at which the software module must be applied first.
5. Select **Run at each software pass until end of deployment** and click **OK**.

Creating a software module for unattended deployment of Windows 2003 R2 operating system:

To prepare an unattended deployment of Windows 2003 R2, you must include some of the content of the second CD of the distribution in a software module and bind this software module to the system profile created with the first CD.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software**.

3. Select **Windows 2000 / 2003 / XP**.
4. Select **A custom action on the target**.
5. Select **A set of files to copy on the target (with an optional command to execute).**
6. Indicate on which computer the files of the second CD are located.
7. Indicate the complete path to find the files in `/CMPNENTS/R2`, for example `D:/CMPNENTS/R2`.
8. Verify the proposed description and if necessary, modify it. Optionally, enter a comment.
9. Enter the necessary parameters for this specific software module:
   - Apply the software module **After one additional reboot**.
   - Enter a meaningful package file name, with a `.pkg` extension.
   - Use `\install\R2` as destination path
   - Do not forget the command-line to be run on the target

     ```
     cmd /c \install\R2\setup2.exe /q /a /p:xxxx-xxxx-xxxx-xxxx-xxxx /cs
     ```

     where `xxxx-xxxx-xxxx-xxxx-xxxx` is the product key.
10. Wait during the package generation process and click **Finish**.

Do not forget to bind your software module to your Windows 2003 R2 unattended setup system profile.

Creating a ramdisk software module from a bootable diskette:

Creating a ramdisk software module from a bootable diskette is considered by the software module wizard to be a **Configuration change**, which itself is included in the **Custom action**.

1. On the **software modules** page, click **New software**. This opens up the software wizard.
2. Select **Windows 2000 / 2003 / XP**.
3. Select **A custom action on the target**.
4. Select a **Configuration change**
5. Select **Boot a virtual floppy disk**.
6. Specify which computer the bootable diskette must be read from. This can be either on the local computer or on another computer running the web interface extension . The option **On the server itself** must not be used.

   Note: If the diskette drive is added after the web interface extension is started (on the local or remote computer depending on your choice), it can be necessary to stop and restart the web interface extension before it can detect the diskette drive. Moreover, the diskette must not be opened by another application (such as Windows Explorer) as this can cause interference.
7. Insert the bootable diskette that you want to image and run as a ramdisk in the disk drive and click **Next**.
8. Enter a software module description and click **Next**.
9. Specify parameters for the package creation and click **Next**. The software module is created.

Editing software modules

You can edit the basic parameters of a software module, upload new files into your software module, and update drivers.

1. Go to **Server → OS deployment → Software modules**. Double-click on a software module to view the details.
2. From **Software details** page, use the links and buttons.
   - To edit the base parameters of a software module, click **Edit** at the top of the **Software module information** section.
   - To update files or add new files into the software module, click **Edit software module files**, or a link with a similar name, and select **Upload file** from the contextual menu.

     Note: File upload is limited to 16 MB.

Keeping command lines confidential

When you use command lines in your software modules, their call and their output are stored in deployment logs. In some circumstances, for example when the command line includes a password or a product key, it might be necessary to keep the information contained in the command line confidential. Three levels of confidentiality are available.

No confidentiality
: The command line is visible in the web interface and on the target during the installation, its call is logged, and its output is also logged.

The command line call is not logged
: The command line is visible in the web interface, and its output is logged, but the command line call, containing the whole command line string with all parameters, is visible in the logs neither on the web interface nor on the target.

  To apply this level of confidentiality, you must prefix the command line by one exclamation mark (!).

The command line call and output are not logged
: The command line is visible in the web interface, but its call and output are visible in the logs neither on the web interface nor on the target.

  To apply this level of confidentiality, you must prefix the command line by two exclamation marks (!!).

To keep command lines confidential:

- Enter the appropriate number of exclamation points in front of the command in the Software Wizard when first creating the software module.
- Edit the software module information
  1. Go to **Server → OS deployment → Software modules**. Double-click on a software module to view the details.
  2. Click **Edit** in the Software module information banner.
  3. Update the command line with the appropriate number of exclamation points.
  4. Click **OK**.

Keyword substitution

You can usefully use keyword which act as variables and are substituted with their values during deployments. Keywords can either refer database values or server specific values, given by the user.

Syntax

Variable substitution expressions follow the syntax given here. They start with the character { and end on the same line with }. Words between these two characters are interpreted by using one of the following schemes:

- *{$expr$}* the expression is replaced with the string resulting of the evaluation of `expr`.
- *{/expr/ab}* the expression is replaced with the string resulting of the evaluation of `expr`, but each occurrence of the character `"a"` is replaced by the character `"b"` (character-based substitution).
- *{=expr=test content=this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is equal to the text `"test content"`.
- *{!expr!test content!this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is not equal to the text `"test content"`.

Note: If a variable does not exist (for example, it contains a typing error or it is not described in `server.ini`) but it is used in a command, its value is supposed to be empty which can result in deployment errors.

Database keywords

Within an expression, database records can be referred to. Within a record, each field can be accessed using the standard C notation (`record.fieldname`). The exhaustive list of these fields can be obtained from the database records, with the following correspondences between variable and database record names:

Table 2. Records for `free-text conditions`

Variable record name	Database record name
Disk	DiskInventory
DMI	DMIInventory
Order	BOM
User	UserProfile
System	SystemProfile
PCI	PCIInventory

Below are a few examples of available fields:

- `Order.IP`: a string, the target IP address, such as `192.168.1.2`
- `Order.MAC`: a string, the target MAC address, such as `00:01:02:03:04:05`
- `Order.SN`: a string, the target Serial Number, such as `CH12345678`
- `Order.Model`: a string, the computer model name, such as `e-Vectra`
- `User.UserCateg0`: a string, without any restriction, such as`technicians`
- `DMI.Vendor`: a string, the vendor name, such as `Hewlett-Packard`
- `DMI.Product`: a string, same as `Order.Model`

- DMI.ProcModel: a string, the processor model
- Disk[0].Type: a string, the disk 0 drive type, such as ATAPI
- Disk[0].Media: a string, the disk 0 media type, such as Disk or CD
- Disk[0].DiskSize: a number, the physical size of the disk (if detected)
- PCI[0].VendorID: a string, the hexadecimal vendor ID of the device
- PCI[0].DeviceID: a string, the hexadecimal device ID of the device

For disks and PCI devices, you can use the function sizeof (sizeof(Disk) and sizeof(PCI)) to discover the number of devices present. You can then use indexes to access these devices.

As an example for keyword substitution, if BomID has OrgName Rembo SaRL, RemboServer 192.168.168.16, and IP 192.168.168.32 for value 1, the following text

```
BomID:{$Order.BomID$}
OrgName:{$User.OrgName$}/{$StrToLower(User.OrgName)$}
RemboServer:{$Order.RemboServer$}
IP:{$Order.IP$}
```

gives the following results after keywords are substituted (note the use of a Rembo-C function within the expression to be substituted):

```
BomID:1
OrgName:Rembo SaRl/rembo sarl
RemboServer:192.168.168.16
IP:192.168.168.32
```

Server specific keywords

If you want to set up server specific keywords, which are defined exclusively by the user and per server, you must edit Files/global/rad/server.ini.

Start the file with [Custom] and add a line per keyword, in the format **keyword=value**, where keyword is a word of your choice and value the value you want to give it.

To use the keyword in a command, type Server.keyword and activate keyword substitution when creating the software module.

Note: server.ini is not replicated between servers. If you use multiple servers, you must edit server.ini on each server.

Customizing the software page

You can view the software modules in a tree viewer or in a list viewer. The list viewer allows you to customize the visible information.

You must have created at least one software module, otherwise there is nothing to view.

To customize the visible information

1. Go to **Server** → **OS deployment** → **Software modules**. Then click **List view**.
2. From the list view, you can
   - Drag the column separator in the column heading to resize the column.
   - Click on the triangular arrow to the left of the column name to sort the software modules by column criteria.

- Click on the arrow on the right of the column name and select an option to filter the information. Filtering on several columns is cumulative.

3. For more options, right click anywhere to open the contextual menu and select **Arrange columns**.
   - Select the columns you want to see and clear the others.
   - Click on the minus or plus icons to decrease or increase the size of a column.
   - Select a column and use the up and down arrows to move the column relatively to the others.

   Click **OK** to save your changes. The updated version of the list view is visible in the **Software modules** page.

To return to the tree view, click **Tree view**. You can also access the details of the software modules by double-clicking on a software module name, from either view.

OS configuration and software bindings

OS configuration bindings determine which configurations are available to a target when booting the target on the network, while software bindings correspond to the list of software modules currently assigned to the target.

OS configuration and software bindings are created when:

- The Target Monitor has been used to manually modify OS configuration and software bindings for the target
- A deployment has been started with the Target Monitor. In this case, an OS configuration binding is added for the corresponding OS configuration.
- Automatic binding rules are configured in the **Details** page of OS configurations or software modules. Some of these rules have matching values for the specified criteria. These bindings cannot be modified, except by modifying the rules.

With the Target Monitor, you can browse, remove or add OS configuration and software bindings to any target present in the database. Go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details.

Binding software modules and OS configurations to targets

Bindings link software modules and OS configurations to targets to enable automatic deployment. When binding to targets, you explicitly provide the list of software modules and OS configurations to bind to your target.

To explicitly bind a software module or a OS configurations to a target, there are two methods:

- From the **Target Monitor** page
- From the **Target details** page

If you want to bind software modules or OS configurations to a group of targets, you must do it through the Target Monitor.

From the Target Monitor:

1. Select a target or a group of targets
2. Select **Bind software** or **Bind OS configurations** from the contextual menu
3. Select the items to bind from the popup window
4. Click **OK**

From the Target details page:

1. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.
2. Go to the **Bindings** panel.
3. Click **Edit** in the relevant section to add explicit bindings for OS configurations and software modules.
4. Select the items for which you want to add explicit bindings.
5. Click **OK**

You can also clear items to remove their explicit bindings. To remove a binding by rule, you must modify the rule.

Binding software modules to computer models and PCI devices

You can link software modules to target computer hardware model to enable automatic deployment. When creating a binding, you explicitly provide the list of software modules to link to a computer model and its PCI devices.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Select **View bindings** from the contextual menu. A grid is displayed representing computer model-based software bindings:
   - The rows correspond to your software modules
   - The columns correspond to the computer model. If you expand it, you can see all the corresponding PCI devices discovered in your hardware inventory.

   The background color indicates implicit bindings by existing generic rules, while checkmarks correspond to explicit one to one bindings.
3. Select a grid position corresponding to the software module and device to bind.
4. Select **Bind** from the contextual menu.

You can also remove a binding by selecting it and clicking **Unbind** from the contextual menu.

Binding software modules to a deployment scheme

Software modules can be bound to deployment schemes.

Take a company with offices in three locations: New York, Quebec City, and Mexico City. In each of these locations, the company has people in human resources, sales, logistics, and product development. For the sake of simplicity, consider further that all the employees use either one of two types of computers: a desktop, or a notebook. All desktop computers are identical (with the same network card, system board, disks, and so on) and the same applies for all notebooks.

In this scenario, the company needs two profiles, one with the image for notebooks and one with the image for desktop computers. Three configurations per profile (six in total) are necessary to integrate the different parameters of the different locations, in particular language and time zone information. Finally, schemes are set according to the employees' department, with software modules specific to the different departments bound directly to the deployment schemes.

1. Go to **Server** → **OS deployment** → **Task templates** Select the **Deployment Schemes** folder. Double-click on a deployment scheme to view its details.
2. Click **Edit** on the **Software bindings** section of the page to open the dialog to bind software modules to schemes.

3. Select which software modules you want to bind to your deployment scheme, in addition to software modules that can have been bound to targets.
4. *(Optional)* If you want to use only the software checked in the window when deploying with this scheme, select the **Discard all other software binding rules** check box.

Automatic binding rules

Automatic binding rules are used to create bindings between OS configurations and targets, or software modules and targets, without having to specifically bind a OS configuration or a software module on each target.

Rules are created in OS configurations and software modules to determine which targets are automatically bound to the OS configuration or software module.

Rules are made of criteria and values. If a target has a matching value for all criteria in the rule, the OS configuration or software module will be bound to that target. The binding will be displayed with the mention **by rule** in the OS configuration panel of the target properties for targets that match the criteria. For example, if the criteria is the model name, and the value is `Optiplex`, targets with a model name starting with `Optiplex` will be bound to the object where the rule has been defined.

Automatic binding rules are defined in Tivoli Provisioning Manager for OS deployment at the bottom of the **OS configuration details** or **Software details** page.

To create a new binding rule, click **New rule** located at the bottom of the Web interface:

1. The dialog displayed to create a new binding rule is different depending on whether you are adding a rule to an OS configuration or to a software module. When adding a binding rule to a software module, you can set values for the following criteria:
   - A deployment scheme
   - A system profile
   - A current OS configuration
   - Administrative group
   - One of the system-definable and user-definable fields of the database (only used if you have customized the database)
   - An operating system type, such as Windows 2000
   - An operating system version, such as SP2
   - An operating system language
   - An operating system architecture, such as x86-32
   - A computer model name
   - A BIOS version
   - A PCI device
   - A base board
   - MultiChassi
   - HAL Type
   - A free-text condition in `Rembo-C;` syntax

For example, to create a binding based on the operating system type between a software module and targets, you must create a new rule, click **OS type**, and select the operating system version that you want to limit this software module to.

2. When adding a binding rule to an OS configuration, you can set a condition on the deployment scheme, and on the computer model name. The next ten fields are only used if you have customized your database and want to match specific user categories.
3. Finally, you can enter a free-text condition following the `Rembo-C;` syntax. They must only be used by advanced users.

   The conditions determine the applicability of the rule and evaluate to `true` or `false`. A condition must be formed using the variables also used for keyword substitutions in software modules, combined with Java™-like logical operators, listed by order of priority in the table:

Table 3. Logical operators for free-text conditions

Operator	Meaning
<	smaller than
<=	smaller than or equal to
=>	greater than or equal to
>	greater than
==	equal to
!=	not equal to
&&	AND operator
||	OR operator

For example, a typical condition can be:

```
Disk[0].DiskSize > 10*1024*1024
```

Note: If a condition cannot be evaluated, it is considered to have the value `false`.

Scheduling the application of software modules for Windows operating systems

Tivoli Provisioning Manager for OS Deployment provides a wide flexibility in the specification of a deployment task. As several software modules can be deployed in conjunction with a system profile, you can schedule when they must be applied.

Tivoli Provisioning Manager for OS Deployment provides a wide flexibility in the specification of a deployment task. As several software modules can be deployed in conjunction with a system profile, you can schedule when they must be applied.

Typical application locations for software modules include:

- For virtual floppy-disk used as ramdisk: before disk partitioning and OS installation, to allow for the configuration of low-level hardware devices controlling the hard disk, such as RAID controllers
- For virtual floppy-disk images: in between disk partitioning and OS installation, to flash devices early
- Sysprep and unattended setup processes are automatically run during the OS installation phase, if required

- For system snapshots: right after OS installation, to deploy the software nearly at the same time as the operating system image (the most efficient). Before OS installation is forbidden, as a system snapshot needs an installed OS
- For other software: when the OS is installed or after additional reboots depending on the software module needs

Software modules are not ordered within an installation stage. If you want a software module to be installed before another between two specific reboots, create two distinct installation stages between the reboots. For example, if your first software module copies files on the target and the second one runs a command on these files, you must place the first software module in an installation stage which occurs before the one in which you run the command software module.

1. To schedule the application of software modules, go to **Server → OS deployment → Software modules**. This opens a dialog window that allows you to order the different software modules stored on your OS deployment server. The dialog shows the different steps of a deployment with disk partitioning (in green), OS installation (in purple) and reboots (in red). Software components can be installed in between all of these steps, where they are placed inside the expandable installation stages (in yellow).
2. You can add, move, and delete reboot sequences by using the buttons at the bottom of the dialog window. You can also rename software installation stages.
3. You can expand the software installation stages to view their content by clicking on the + icon. You can then move individual software modules from one stage to another by drag-and-drop. The destination stage does not need to be expanded.

   Note: Drag-and-drop is limited to the **Software Application Order** window. You cannot drag-and-drop an item from the Software Module page.

   Note: Vista 2008 Windows 7 If you have more than one HotFix (MSU) software module in stages occurring later than **When the OS is installed**, you must ensure that they each have a different destination path on the target.

When creating a recovery CD or exporting a RAD file, the software application order is automatically included.

Working with hardware configurations

It is sometimes necessary to run configuration tasks on the targets before installing the operating system, for example to update the firmware or to configure RAID volumes.

To automate this kind of operation with the product, you must perform a *hardware configuration task*, which uses a *hardware configuration object* stored on the OS deployment server. To create a hardware configuration object, you must have already created a *hardware environment*. This hardware environment contains WinPE or DOS files, updated with drivers specific to given hardware models and vendor-specific tools to perform hardware configuration tasks.

The hardware configuration tasks that you can perform with the product are

- RAID configuration
- BIOS update
- BIOS settings

- Hardware custom configuration, that is, any kind of tool that you can load into the environment and run from a command line.

You can also perform an inventory of RAID or Fiber Channel hardware.

Hardware configuration tasks are available only for targets with an x86 or an x86-64 architecture.

Example

To configure hardware with the product, for example a BIOS update with WinPE2 on an IBM target, you need to follow a number of steps.

1. Create a hardware environment with drivers and tools:
   a. Download Windows Automated Installation Kit (WAIK) from Microsoft and install it to have the WinPE2 files available.
   b. Download the latest ServerGuide scripting toolkit from IBM and extract it, for example, in directory `C:\IBM-SGTSK-WinPE2.x`.
   c. Run the SGTKWinPE.cmd command to prepare the WinPE2 environment with the needed IBM drivers. It creates the `.\sgdeploy\WinPE_ScenariosOutput\Local\RAID_Config_Only\ISO` directory, which contains both the WinPE2 binaries and the vendor-specific tools.
   d. Create a hardware environment with the hardware environment wizard.
2. Create a hardware configuration object with the hardware configuration wizard:
   a. Select **BIOS update** as the type of hardware configuration to be performed.
   b. Associate the hardware environment of step 1 and your hardware model to the new hardware configuration object you are creating.
   c. Indicate the location of the BIOS update material, that is, a set of files containing in particular `wflash.exe`.
3. Perform the actual configuration task by deploying the hardware configuration object of step 2 on your target:
   a. Select a target (or several) in the Target Monitor.
   b. Select **Deploy now** in the contextual menu.
   c. Select **Perform hardware configuration tasks** and optionally other deployment tasks in the deployment wizard.
   d. Select the hardware configuration object that you want to apply and follow the remaining instructions of the wizard.

The hardware environment now runs as a ramdisk on the target, and, using vendor-specific tools, the BIOS is updated.

Setting up your environment

To perform hardware configuration tasks, you must set up a hardware-specific environment containing the vendor-specific scripting toolkit tools and the necessary drivers to run correctly (for example, network connectivity) on the target.

The hardware environment supported are those running scripts and tools in:

- WinPE 2.x
- WinPE 1.x
- DOS

Every environment is very specific to its vendor, and must be prepared with the suitable drivers and scripting toolkit tools.

WinPE2, WinPE1, and DOS cannot perform hardware configuration tasks (for example, RAID configuration or BIOS setting) by themselves. They must contain drivers to access the hardware and tools to perform the configurations. These drivers and tools are vendor-specific and vary for each type of target model. When you create an environment with the OS deployment server, you associate either WinPE2, WinPE1, or DOS, to vendor-specific drivers and tools. You can then associate the resulting environment to a specific set of target models and a type of hardware configuration tasks to create a hardware configuration object.

Because a hardware environment is run as a ramdisk, it does not leave any trace on the target after the hardware configuration task is performed.

Hardware configuration objects and tasks

A hardware configuration object is the association, on an OS deployment server, of a vendor-dependent environment, target models, a type of hardware configuration to be performed, and possibly some other commands. A hardware configuration task is performed at deployment time by loading and running the associated hardware configuration object containing a vendor-dependent environment on the target, before installing the operating system.

Hardware configurations tasks do not impact the following operating system deployment because Tivoli Provisioning Manager for OS Deployment configures the hardware through actions run in a ramdisk before the deployment of the operating system.

The execution flow is similar, regardless of the environment to run, or the type of hardware environment task:

1. The environment is loaded in memory, as a ramdisk
2. Any additional binary or configuration files are added to the ramdisk, based on the selection made in the web interface when creating the hardware configuration object
3. The computer boots the ramdisk
4. The hardware configuration task is run
5. The computer reboots
6. Tivoli Provisioning Manager for OS Deployment resumes the deployment sequence if any was selected, but a hardware configuration object can be run also as an independent task

The following types of hardware configuration objects are available:

RAID configuration
: The hardware configuration wizard allows you to create a hardware configuration object to configure RAID adapters in a vendor-independent way. Tivoli Provisioning Manager for OS Deployment builds the vendor-specific configuration file.

BIOS update
: The hardware configuration wizard allows you to create a hardware configuration object to update the BIOS firmware on the target.

BIOS settings
: The hardware configuration wizard allows you to create a hardware

configuration object to update the BIOS or BMC (baseboard management controller) settings through an initialization file.

Hardware custom configuration
The hardware configuration wizard allows you to create a hardware configuration object to perform any kind of hardware configuration. Any tool used for preparing the environment can be packaged in a custom hardware configuration object, injected into the ramdisk and run using command lines.

Capture hardware parameters
This option is available only if you do not already have a hardware capture configuration object.

The hardware configuration wizard allows you to create a hardware configuration object to capture RAID and Fiber Channel information from a target.

RAID and Fiber Channel hardware capture

Capturing RAID and Fiber Channel information requires the use of a vendor-specific environment.

Target inventory for CPU, memory, logical disks, PCI devices, motherboard, and so on, is managed by the OS deployment engine and all information is available immediately if requested. To complete the hardware target inventory with RAID and Fibre Channel information you need the vendor-specific scripting toolkit tools. The hardware capture is done in a similar way to that of the hardware configurations, which means that you need to load the vendor-dependent environment on the target to start the specific capture tool.

The captured hardware information for Fibre Channel and RAID disks can then be seen from the web interface:

Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.

Creating a hardware environment

Having a hardware environment on your OS deployment server is a prerequisite to create a hardware configuration object, with which you can perform hardware configuration tasks on targets.

Before you can create your environment with the wizard, you must prepare the files on the OS deployment server.

Instructions are provided for preparing the files using scripting toolkits for IBM, Dell, or HP products. It is recommended that you download the latest WinPE 3.x compatible scripting tool environments and use this version. However, the instructions for, WinPE 2.x, WinPE 1.x and DOS are also provided.

IBM **IBM ServerGuide Scripting Toolkit WinPE 3.x based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site. The name of the downloaded file is similar to `ibm_utl_sgtkwin_2.30_windows_32-64.zip`.
2. Extract the toolkit into a local directory, for example, into `c:\IBM-SGSTK-WinPE3.x`

3. As described in the User's Guide in `c:\IBM-SGSTK-WinPE3.x/sgdeploy/SGTKWinPE/Docs/UserGuide.pdf`, you must then do the following:
   a. Download Windows Automated Installation Kit (AIK) for Windows 7 in EnglishWindows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.
   b. Install Windows AIK.
   c. Restart your computer.
   d. Expand files `ibm_utl_tsep_2.00_winpe_i386.zip` and `ibm_utl_tsep_2.00_winpe_x86-64.zip` located in `.\sgdeploy\updates\uxsp` into the directory in which the toolkit was extracted, for example `c:\IBM-SGSTK-WinPE3.x`
   e. Run `InstallSEPs.cmd` to install the System Enablement Pack.
   f. Run `SGTKWinPE.cmd` to create a WinPE image with the requested drivers for IBM servers. Use the option `/Image` to exclude ISO and provide `ScenarioINIs\Local\Raid_Config_Only_x86.ini` if you use a 32-bit WinPE, or `ScenarioINIs\Local\Raid_Config_Only_x64.ini` if you use a 64-bit WinPE, as properties file to include all RAID and Fibre tools and to exclude all network tools. The command finds where the Windows AIK is located by itself.

      ```
      SGTKWinPE.cmd /Image ScenarioINIs\Local\Raid_Config_Only_x86.ini
      ```

      A directory `.\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO` is created and contains the environment tools.

IBM **IBM ServerGuide Scripting Toolkit WinPE 2.x based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site. The name of the downloaded file is similar to `ibm_sw_sgtkw_2_1_windows_i386.zip`.
2. Extract the toolkit into a local directory, for example, into `c:\IBM-SGSTK-WinPE2.x`
3. As described in the User's Guide in `c:\IBM-SGSTK-WinPE2.x/sgdeploy/SGTKWinPE/Docs/UserGuide.pdf`, you must then do the following:
   a. Download the Windows Automated Installation Kit (AIK) 1.1 32-bit in English for Windows Vista SP1 and Windows Server 2008. Windows Automated Installation Kit (AIK) 1.1 is distributed by Microsoft and is available on the Microsoft Web site from the following link: Windows Automated Installation Kit (AIK).
   b. Install Windows AIK.
   c. Restart your computer.
   d. Expand files `ibm_utl_sep_1.00_winpe_i386.zip` and `ibm_utl_sep_1.00_winpe_x86-64.zip` located in `.\sgdeploy\updates\uxsp` into the directory in which the toolkit was extracted, for example `c:\IBM-SGSTK-WinPE2.x`
   e. Run `InstallSEPs.cmd` to install the System Enablement Pack.
   f. Run `SGTKWinPE.cmd` to create a WinPE image with the requested drivers for IBM servers. Use the option `/Image` to exclude ISO and provide `ScenarioINIs\Local\Raid_Config_Only_x86.ini` if you use a

32-bit WinPE2, or ScenarioINIs\Local\Raid_Config_Only_x64.ini if you use a 64-bit WinPE2, as properties file to include all RAID and Fibre tools and to exclude all network tools. The command finds where the Windows AIK is located by itself.

```
SGTKWinPE.cmd /Image ScenarioINIs\Local\Raid_Config_Only_x86.ini
```

A directory .\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO is created and contains the environment tools.

IBM **IBM ServerGuide Scripting Toolkit WinPE 1.x based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site.
2. Extract the toolkit into a local directory, for example, c:\IBM-SGSTK-WinPE1.x.
3. As described in the User's Guide in c:\IBM-SGSTK-WinPE1.x\sgdeploy\SGTKWinPE\Docs\UserGuide.pdf you must then complete the following steps:
   a. Download WinPE 2005.
   b. Run SGTKWinPE.cmd to create a WinPE image with the requested drivers for IBM servers.

IBM **IBM ServerGuide Scripting Toolkit DOS based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site
2. Extract the toolkit into a local directory, for example, c:\IBM-SGSTK-DOS.

Note: DOS tools are deprecated. They are used only to support some older hardware.

Dell **Dell DTK Scripting Toolkit WinPE 3.x based**

To set up the WinPE 3.0 environment for your Dell servers:

1. Download Windows Automated Installation Kit (AIK) for Windows 7 in English. Windows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest DTK scripting toolkit from the Dell Web site. The name of the downloaded file is similar to DTK3.2.1-WINPE-22.exe.
5. Extract the download file. For example, extract the file to the location c:\ Dell-DTK-3.2.1.
6. As described in the Dell User's Guide, in C:\Dell-DTK-3.2.1\Dell\Docs\DTKUG.pdf, you must then complete the following tasks:
   a. Open a command prompt in the directory containing the driver installation batch for WinPE3.x: WINPE3.0_driverinst.bat. For example, the directory, C:\ Dell-DTK-3.2.1\Dell\x32\Drivers\winpe3.x.
   b. Launch the file called WINPE3.0_driverinst.bat <*WINPEPATH*> <*DTKPATH*>, where <*WINPEPATH*> is the destination path to create

the directory structure for WinPE 3.0 and <*DTKPATH*> is the path to the Dell drivers in the extracted DTK toolkit. For example, the file might be called `WINPE3.0_driverinst.bat C:\Dell-DTK-3.2.1\WinPE3.x_Out C:\Dell-DTK-3.2.1\Dell\x32\drivers`. Launching this file preinstalls the Dell drivers into `winpe.wim`.

7. Copy and rename the customized `C:\Dell-DTK-3.2.1\WinPE3.x_Out\winpe.wim` to `C:\Dell-DTK-3.2.1\WinPE3.x_Out\ISO\sources\boot.wim`.

Dell **Dell DTK Scripting Toolkit WinPE 2.x based**

To set up the WinPE2 environment for your Dell servers:

1. Download the Windows Automated Installation Kit (AIK) 1.1 32-bit in English for Windows Vista SP1 and Windows Server 2008.Windows Automated Installation Kit (AIK) 1.1 is distributed by Microsoft and is available on the Microsoft Web site from the following link: Windows Automated Installation Kit (AIK).
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest DTK scripting toolkit from the Dell Web site. The name of the downloaded file is similar to `DTK2.6-WINPE-56.exe`.
5. Extract the download file. For example, extract the file to the location `c:\ Dell-DTK-2.6 5`.
6. As described in the Dell User's Guide, in C:\Dell-DTK-2.6\Dell\Toolkit\Docs\DTK25UG.pdf, you must then complete the following tasks:
   a. Open a command prompt in the directory containing the driver installation batch for `WinPE2.x: VPE_driverinst.bat`. For example, the directory, `C:\ Dell-DTK-2.6\Dell\Drivers\winpe2.x`.
   b. Launch the file called `VPE_driverinst.bat` <*WINPEPATH*> <*DTKPATH*>, where <*WINPEPATH*> is the destination path to create the directory structure for Windows PE 2.0 and <*DTKPATH*> is the path to the Dell drivers in the extracted DTK toolkit. For example, the file might be called `VPE_driverinst.bat C:\Dell-DTK-2.6\WinPE2.x_Out C:\Dell-DTK-2.6\Dell\drivers`). Launching this file preinstalls the Dell drivers into `winpe.wim`.
7. Copy and rename the customized `C:\Dell-DTK-2.6\WinPE2.x_out\winpe.wim` to `C:\Dell-DTK-2.6\WinPE2.x_Out\ISO\sources\boot.wim`.

Dell **DELL Scripting Toolkit WinPE 1.x based**

Note: Windows PE 2005 must be built from a Windows 2003 server for the Dell tools to work.

To set up the WinPE1 environment for your Dell servers:

1. Obtain a Windows PE 2005 file structure.
2. Copy it into a temporary folder, for example, `c:\winpe-dell`
3. The Windows PE 2005 directory structure should contain a directory named `I386` or `MININT`. If it contains a directory named `MININT`, rename it to `I386`.
4. Download the Deployment Toolkit from Dell.
5. Run the executable package to extract the toolkit to the disk of the OS deployment server. In the examples, it is assumed that you have extracted the toolkit into `c:\DELL-DTK`, which implies that you have a folder named `C:\DELL-DTK\Dell\Toolkit`.

6. To install the appropriate drivers for Dell servers in your WinPE image, follow the instructions of the DTK User Guide (*Running Deployment Scripts Using DTK and Windows PE*).

   In particular, you must:

   a. Install the drivers with the `driverinst.bat` script
   b. Modify `winpeoem.sif` and `winbom.ini`
   c. Add the RPC DLLs to the Windows PE directory.

      Note: Add the RPC DLLs in `i386\system32` instead of those in the Tools folder.

7. To verify that the drivers have been installed, check for the existence of the file called `c:\temp\winpedell\i386\system32\racsvc.exe`.

HP **HP SmartStart Scripting Toolkit WinPE 3.x based**

To set up the WinPE 3.0environment for your HP servers:

1. Download Windows Automated Installation Kit (AIK) for Windows 7 in EnglishWindows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest SmartStart Scripting Toolkit from the HP Web site: http://h18013.www1.hp.com/products/servers/management/toolkit/. The name of the downloaded file is similar to `SP47335.EXE`.
5. Extract the file into a directory, for example, `C:\HP-TK`.
6. As described in the *HP SmartStart Scripting Toolkit Windows Edition User Guide.pdf* in `C:\HP-TK\SWSetup\SP47335\` and the *Windows Preinstallation Environment User's Guide* (WinPE.chm) contained in Windows AIK, you must then mount the WinPE3.x base image for specific customization. For example, activate extra packages, add drivers, and so on.

   a. From the Windows AIK tools folder, run the command to create WinPE customization directory.

      ```
      C:\Program Files\Windows AIK\Tools\PETools>copype.cmd x86
      C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP
      ```

   b. Mount the base image launching Dism from the WinPE3.x_HP folder.

      ```
      Dism /Mount-Wim /WimFile:.\winpe.wim /index:1 /MountDir:.\mount
      ```

   c. Install the *neutral* WMI packages in the image.

      ```
      Dism /image:.\mount /Add-Package
      /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86
      \WinPE_FPs\winpe-wmi.cab"
      ```

      Enter the command on one line, although it does not fit on this example.

   d. Install also the language specific WMI package in the image.

      ```
      Dism /image:.\mount /Add-Package
      /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86
      \WinPE_FPs\en-us\winpe-wmi_en-us.cab"
      ```

Enter the command on one line, although it does not fit on this example.

e. Add the required drivers (.inf files) to the base image by using the `/Add-Driver` option of the `Dism` command.

```
Dism /image:<mounted image> /Add-Driver /Driver:<driverpath>
/Recurse
```

Where *<driverpath>* is the location of the .inf files found in the extracted drivers within the hpDrivers folder and `/Recurse` is an option to query all the drivers in subfolders.

```
Dism /image:.\mount /Add-Driver
/Driver:C:\HP-TK\SWSetup\SP47335\hpDrivers\Winpe30 /Recurse
```

Enter the command on one line, although it does not fit on this example.

f. Copy the `hpsstkio.sys` Toolkit I/O driver (required for the `conrep` and `rbsureset` utilities) from the HP driver directory to the Windows driver directory. For example:

```
copy C:\HP-TK\SWSetup\SP47335\hpDrivers\Winpe30\system\hpsstkio
\hpsstkio.sys C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\mount\Windows
\System32\drivers
```

Enter the command on one line, although it does not fit on this example.

g. Unmount the customized image to build the customized WinPE.wim:

```
Dism /Unmount-Wim /MountDir:.\mount /Commit
```

7. Copy and rename the customized `C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\WinPE.wim` file into `C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\ISO\sources\boot.wim`.

HP **HP SmartStart Scripting Toolkit WinPE 2.x based**

To set up the WinPE2 environment for your HP servers:

1. Download the Windows Automated Installation Kit (AIK) 1.1 32-bit in English for Windows Vista SP1 and Windows Server 2008.Windows Automated Installation Kit (AIK) 1.1 is distributed by Microsoft and is available on the Microsoft Web site from the following link: Windows Automated Installation Kit (AIK).
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest SmartStart Scripting Toolkit from the HP Web site: http://h18013.www1.hp.com/products/servers/management/toolkit/. The name of the downloaded file is similar to `SP38836.EXE`.
5. Extract the file into a directory, for example, `C:\HP-TK`.
6. As described in the *HP SmartStart Scripting Toolkit Windows Edition User Guide.pdf* in `C:\HP-TK\SWSetup\SP38836\` and the *Windows Preinstallation Environment User's Guide* (WinPE.chm) contained in Windows AIK, you must then mount the WinPE2.x base image for specific customization. For example, activate extra packages, add drivers, and so on.
   a. From the Windows AIK tools folder, run the command to create Windows PE customization directory. For example: `C:\Program Files\Windows AIK\Tools\PETools>copype.cmd x86 C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP)`

b. Mount the base image launching imagex from the WinPE2.x_HP folder. For example, `imagex /mountrw WinPE.wim 1 .\mount`.

c. Install the WMI packages in the image: `peimg /image=.\mount /install=*WMI*`

d. Add the required drivers (.inf files) to the base image by using the `peimg /inf` command.

```
peimg /inf=<driverpath> .\mount
```

Where <*driverpath*> is the location of the .inf files found in the extracted drivers within the hpDrivers folder. For example, `peimg /inf=c:\HP-TK\SWSetup\SP38836\hpDrivers\Extr-Drivers\nic\b06nd .\mount`.

e. Repeat step d. for each additional device driver.

f. Copy the `hpsstkio.sys` Toolkit I/O driver (required for the `conrep` and `rbsureset` utilities) from the HP driver directory to the Windows driver directory. For example:

```
copy C:\HP-TK\SWSetup\SP38836\hpDrivers\system\hpsstkio\hpsstkio.sys
C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP\mount\Windows\System32\drivers
```

g. When you finish customizing the image, prepare the environment image by using the `peimg /prep` command:

```
peimg /image=.\mount /prep
```

h. Unmount the customized image to build the customized WinPE.wim:

```
imagex /unmount /commit .\mount
```

7. Copy and rename the customized `C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP\WinPE.wim` file into `C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP\ISO\sources\boot.wim`.

HP **HP SmartStart Scripting Toolkit WinPE 1.x based**

The initial setup for the HP SmartStart Scripting Toolkit is very similar to the setup of the Dell Hardware Toolkit, because both Toolkits require Windows PE. Some details are therefore skipped, but you can read them in the Dell section.

1. Download the Win32 HP SmartStart Scripting Toolkit version of the toolkit on the HP site.
2. Extract it to the disk of the OS deployment server (for example, in `c:\HP-TK`).
3. Create a Windows PE 2005 folder for the HP tools:
   a. Copy a Windows PE file structure to a temporary folder (`c:\winpe_hp`)
   b. Install the HP drivers in the Windows PE directory, as explained in the User Guide for the HP Hardware Toolkit
      1) Run the executable file under hpDrivers
      2) Give the location of the i386 folder of your Windows PE folder

To create your environment, with the wizard:

1. Go to **Server** → **Advanced features** → **Hardware configurations**.
2. Click **New environment** and follow the wizard instructions. You must
   a. Ensure that the web interface extension is running on the computer where Windows AIK and the environment tools have been prepared.

b. Provide the path of the folder in which the environment tools are located, that is where you have installed the scripting toolkit. For example:

IBM `C:\IBM-SGSTK-WinPE3.x\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO`

Dell `C:\Dell-DTK-3.2.1\Dell\x32`

HP `C:\HP-TK\SWSetup\SP47335`

c. Provide the path of the folder in which the environment material is located, that is the WinPE files. For example:

IBM `C:\IBM-SGSTK-WinPE3.x\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO`

Dell `C:\Dell-DTK-3.2.1\WinPE3.x_Out\ISO`

HP `C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\ISO`

You can view the created environment by performing the following:go to **Server** → **Advanced features** → **Hardware configurations**. Alternatively, you can also view it by performing the following:go to **Server** → **OS deployment** → **Software modules**. To view it look under a specific environment folder.

Now, you can create hardware configurations using this environment.

Creating a hardware configuration object

A wizard allows you to easily create hardware configuration objects.

Before you can create a hardware configuration object, you must have created the environments needed to later perform the hardware configuration tasks.

1. Go to **Server** → **Advanced features** → **Hardware configurations**.
2. Click **New hardware config.**.
3. Select the kind of hardware configuration that you want to create.
4. Provide at least one target model and environment pair on which the hardware configuration can apply.
5. For BIOS update, BIOS settings, or Hardware custom configuration the specific files or set of files can be downloaded from the specific vendor sites.
6. Follow the wizard instructions.

To view or edit a hardware configuration, select the hardware configuration and select **View configuration details** in the contextual menu. In the **Hardware configuration details**, use the **Edit** buttons to update the different sections.

Creating a hardware capture configuration

A wizard allows you to easily create hardware capture configuration in a way similar to that for hardware configurations.

Before you can create a hardware capture configuration, you must have created the environments needed to later run the hardware capture.

- If you do not yet have a hardware capture configuration, perform the following steps:
  1. Go to **Server** → **Advanced features** → **Hardware configurations**.
  2. Click **New hardware config.**
  3. Select **Hardware discovery**.

4. Provide at least one target model and environment pair on which the hardware capture can apply.
5. Follow the instructions of the wizard.

- If you already have a hardware capture configuration, you can add target model and environment pairs, as follows:
  1. go to **Server → Advanced features → Hardware configurations**.
  2. Select **Hardware discovery**.
  3. Double-click **Hardware capture configuration**.
  4. Under **Hardware environment matching**, click **Edit**.
  5. Click **Add a new line** and select the model and environment values
  6. Repeat step 5 for each pair to be added.
  7. Click **OK**.
  8. Click **Back** to return to **Server → OS deployment → Hardware configurations**.

To view or edit the hardware capture configuration, go to **Server → Advanced features → Hardware configurations**. Select **Hardware discovery**, and double-click the hardware capture configuration. In the **Hardware configuration details** page, click **Edit** to update the different sections.

You can now capture RAID or Fiber Channel information.

Capturing hardware information using templates

When you capture hardware information with templates, this capture is done every time the template is used.

Capturing hardware information with templates requires an additional reboot to boot the specific hardware configuration environment (WinPE, DOS,...) and launch the specific scripting toolkit tools.

Note: You can not capture hardware information from a target started with a network boot media.

Capturing hardware information with templates always tries to capture both RAID and Fiber Channel. To run the capture:

1. Go to **Server → OS deployment → Task templates**.
2. Select **Idle Layout** or **Deployment Schemes**, depending on which state you want to perform the hardware capture. If you select **Deployment Schemes**, the discovery is performed at deployment time.
3. Double-click the chosen template to view its details.
4. Click **Edit** on **General settings**.
5. Under **Perform inventory on:**, select **RAID**.
6. Click **OK**.

Capturing hardware information once

When you want to capture hardware information only once for a target, or a group of targets, you do this with a specific tool.

Capturing hardware information requires an additional reboot to boot the specific hardware configuration environment (WinPE, DOS,...) and launch the specific scripting toolkit tools.

Note: You can not capture hardware information from a target started with a network boot media.

1. Go to **Server** → **OS deployment** → **Target Monitor**.
2. Select a target or a group of targets.
3. Select **Additional features** from the contextual menu.
4. Double-click the chosen template to view its details.
5. Select **Capture hardware parameters** and click **Next**.
6. Select **Raid capture**, **Fiber channel capture**, or both, and click **Next**.
7. Follow the instructions of the wizard.

When captured, the RAID and Fiber channel information can be viewed. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details. On this page look under the **Inventory** tab.

Task templates

Task templates group together elements that can be customized on a target. These elements are mostly screen layouts, which condition the appearance of the target screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment.

A deployment scheme is a specific type of task template. Together with the target display screen layout, it contains other parameters for customizing a deployment on a target.

Note: Starting with Fix Pack, version 5.1.0.2, deployment schemes are considered to be a subset of task templates. The functions of deployment schemes have not been altered. To access deployment schemes, go to the **Task template** page, and select the **deployment scheme** folder.

The task template page of the web interface contains a task template tree in the left pane with seven folders. The content of the selected folder is displayed in the right pane.

There are seven task template folders in the tree. They are described here.

Deployment Schemes
Deployment schemes contain parameters that indicate how an OS configuration must be deployed on your target. The **deployment Schemes** folder contains at least the **Default** scheme.

Idle layout
The idle layout defines what is shown on the target when there is no pending task. The **Idle Layout** folder contains at least the **Idle state** layout.

Menu Layout
The menu layout defines how deployment menus are shown to the users. Menus are used when an OS configuration and on deployment CDs. The **Menu Layout** folder contains at least the **Menu** layout.

OS Detection Layout
The operation system detection layout defines the target display when a target is busy detecting the currently installed operating system. It is used when creating a cloning profile from the web interface. The **OS Detection Layout** folder contains at least the **Detect operating system** layout.

Profile Creation Layout
The system profile creation layout defines the target display when a target is busy creating a new system profile. It is used when creating a cloning profile from the web interface. The **Profile Creation Layout** folder contains at least the **Creating cloning profile** layout.

Profile Restoration Layout
The system profile restoration layout defines the target display during the manual restoration of a system profile by the administrator.

Note: A system profile restoration is always performed as-is and must not be confused with an automated deployment resulting in a fully configured operating system installation.
The **Profile Restoration Layout** folder contains at least the **Default OS Restoration** layout.

State Capture Layout
The state capture layout defines the target display when a target is saving the operating system state for future redeployments.

State Restoration Layout
The state restoration layout defines the target display when a target is redeploying an operating system from a saved state.

When a task template is selected in the right hand pane, the bottom of the web interface contains a link to **Customize GUI**. Follow this link to modify the look of your target screen . When the selected task template is a deployment scheme, there are additional links to view and edit the current scheme.

Customizing a screen layout

You can customize the screen layout of a target.

To customize a screen layout:

1. Select the layout that you want to customize in the right pane of the **Task Templates** page of the web interface page.

   Note: An actual layout must be selected and not a layout folder (left pane)
2. At the bottom of the page, the screen layout is shown in reduced size. Click **Customize GUI** to open the screen layout editor.
3. The editor is composed of a left column, containing instructions, a *What-You-See-Is-What-You-Get* (WYSIWYG) view of the screen being edited and a bottom banner with action buttons.
4. Click on the action buttons or directly on the items that you want to modify to see their editable properties displayed in the left column. Make the wanted changes and then click **Save** to keep your new screen design. Return to the **Task Templates** page by clicking **Back**.

Creating and editing deployment schemes

By customizing your deployment schemes, you can adapt the way in which your predefined OS configurations are installed onto targets.

1. The easiest way to create a new deployment scheme is to run the deployment scheme wizard.Go to **Server** → **OS deployment** → **Task templates**. and click **New deployment scheme**.
2. Alternatively, you can modify an existing scheme by editing its parameters. To do this, select a scheme and click **View deployment parameters** and then use

Edit in the banner on top of each parameter section. Some advanced deployment scheme features are available only in this mode and not through the wizard.

3. If you prefer using a wizard to edit your scheme, click **Edit parameters using a wizard**.

   The following parameters apply for simple one-time deployments and for redeployment operations.

   Description

   The first step is to enter a name for this deployment scheme. Make it explicit enough so that you can pick it easily when starting a deployment (the web interface does not show the settings in a deployment scheme, so the choice must be made by name only). Because deployment schemes determine how the computers are installed (and not what is being deployed), use a description such as `Multicast 50 targets` or `On-site` rather than the name of a OS configuration or of a group of computers.

   When the deployment starts

   You must determine if Tivoli Provisioning Manager for OS Deployment requires user interaction during deployment (to edit individualtarget parameters) or runs completely unattended.Select:

   Always edit target-specific parameters
   to have the opportunity to change the target parameters at every deployment. The parameter edition can be made either directly on the target computer or by double-clicking the target icon in the Target Monitor.

   Edit parameters for targets that are not yet in the database
   to be prompted only during the first installation of each target. Subsequent deployments for the same targets run unattended. This is the default choice.

   Never edit parameters
   to have unattended deployments if all BOMs have been previously entered in the database. Any missing entry leads to a red banner on the target and cancels the deployment for this target.

   You must also select how Tivoli Provisioning Manager for OS Deployment behaves when the model of the computer being deployed does not match the model of the computer on which the image was created. This feature requires DMI for hardware detection.Select:

   No if you know that all your system profiles are fully hardware-independent or for deploying universal images.

   Yes, display a warning
   if you want to see all possible OS configurations for a computer, but want to avoid mistakes. This choice can require user interaction and is therefore not appropriate for a fully unattended deployment.

   Yes, abort the deployment
   if you want to prevent anyone from using an OS configuration on a computer different from the one for which it was designed.

Use 'BIOS fallback MBR' to start PXE is used when PXE activation (the process of enabling PXE when booting on the hard-disk) does not work.

The PXE boot code manages the multiple reboots needed to install a computer. To manage these reboots, the PXE boot code must intercept the boot process of the computer at every boot.

- If the computer is configured to always start on the network (LAN device first in the list of boot devices), there is nothing to do, because Tivoli Provisioning Manager for OS Deployment is loaded into memory at every boot.
- If the computer is configured to start on the hard-disk, you can change the MBR of the hard-disk and make it point to the work partition at the end of the hard-disk. Tivoli Provisioning Manager for OS Deployment is then loaded from the hard-disk when the computer starts up, instead of loading the operating system. The disadvantage of this method is that, because the computer did not use the network card to boot, PXE is not available. To enable network access, PXE is activated with a special function in the PXE card that makes it behave as though the computer had booted on the LAN. However, this is not documented in PXE, and does not work on every network card. If the network does not support this, an error is raised, and access to the OS deployment server fails (the message **Network started**, followed by an error).

  When PXE activation does not work, you can write a special MBR telling the BIOS that the hard-disk is not a valid boot device. By default, the BIOS falls back to the next device in the list, which in most computers is the network. As a result, the computer boots on the network and has full access to the network. This is the purpose of the **Use 'BIOS fallback MBR' to start PXE** check box.

Data collection

By default, Tivoli Provisioning Manager for OS Deployment automatically populates the database with an inventory of the hardware setups of all deployed computers. For Windows, a software inventory can also be populated based on the registry. If you are not interested in using those inventories, or if your computers do not comply with any of the hardware detection standards, you can disable these features. Be advised that running the hardware or software inventory on thousands of computers can produce a huge database. This inventory is performed on locked screen.

Tivoli Provisioning Manager for OS Deployment centrally reports the status of the deployment of targets to the OS deployment server and to the server database. Additionally, if you want to keep the deployment logs and the list of software modules on each deployed target, you can specify a local path where the log files are to be stored. The path that you specify is relative to the root of the operating system on the target, for example, `/` on UNIX operating systems or `c:/` on Windows operating systems. In the deployment scheme details, the label of this field is **Save deployment log to**.

When the deployment is completed

When the deployment process is finished, you can select if you want to:

- Turn off the computer automatically (if supported)

- Boot the operating system automatically (this value might not make sense with some values of the previous setting)
- Display a green banner and wait for a manual shutdown

Network usage

Note: Multicast is available only if:

- The targets have an Intel x86 or x86-64 architecture
- Multicast is selected in the deployment scheme
- The subnet supports multicast traffic
- Multicast is not disabled in the boot options of the target
- The target is not a VMWare 3.0 guest.

Depending on the number of targets that you are deploying simultaneously on your infrastructure, you must select one of the following networking modes:Select:

Unicast
to deploytargets one by one, or if you cannot use multicast. When deploying several targets simultaneously in unicast, the deployment time increases dramatically with the number of targets, as the result of network saturation.

Multicast, without explicit synchronization
to use soft-synchronized multicast protocol. Using this protocol, every target independently starts downloading images at soon as it is ready, and continues with the deployment as soon as it has downloaded all required material. When two or more targets (using the same deployment scheme) are downloading files in parallel, they automatically share the same bandwidth. The fastest target has the priority for the choice of the next shared files to be sent by the server, but the slower targets can receive them if they need them. This is a scalable solution that allows for a rolling deployment scenario.

Multicast, with the following synchronization parameters
to use a classical replicated multicast method. This mode is adequate for installing computers in batches. Enter the replication parameters (for example, the number of targets to wait for before starting the download, and the maximum timeout before starting in any case). Tivoli Provisioning Manager for OS Deployment multicast protocol can accept new download targets even after the initial replication period is over, and integrate them seamlessly into the transfer.

Note: In the first stage of an OS deployment, there are two target synchronization stages. Therefore it might seem that the maximum timeout that was set before starting the deployment is doubled.

Vista 2008 Windows 7 You can decide to use a network share on the server to download the files to the targets , rather than downloading the whole image to the hard disk of the target. Using a network share provides a shorter installation time. To use a network share:

- Select **Use network share** in the deployment scheme.

- Share the `files\global\partition` directory and provide at least read-only access to it.
- Enter the relevant server parameter in the **Network share module** section. Go to **Server** → **Server parameters** → **Configuration**. (See "Network share module" on page 314).

On-site deployment

If you are running a one-time deployment in a deployment center and do not want to use redeployment, leave the check box blank and click **Next**.

If you are running an on-site deployment, or if you plan to use redeployment, you can enable the advanced feature.

Redeployment option

Indicate if you want to keep the deployment image in a protected partition and the size of this partition. These options are valid only to configure the deployment scheme for redeployment. More information is available in deploy/tosd_redeplscheme.dita.

Note: The following parameters cannot be modified using the wizard. You must edit your deployment scheme parameters.

Request user confirmation

This parameters allows you to ask for user confirmation before running a deployment.

Unbind OS configuration at the end

This parameter, located in the **General settings** section, is set to **No** by default. Setting this parameter to **Yes** unbinds the OS configuration from the target at the end of the deployment. This OS configuration is not proposed the next time the target boots and, if no other OS configuration is bound, the target presents a locked screen to the user. Review the section on network boot scenario of the deployment process topic.

Unbind software module at the end

This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the software module at the end of the deployment. This software module is not proposed and installed the next time a deployment is performed.

Vista 2008 Windows 7 **Disable user interaction during deployment**

This parameter, located in the **General settings** section, is set to `Yes` by default. If you set this parameter to `No`, you can obtain a command prompt by pressing Shift-F10 on the target computer during a deployment to modify deployment files.

Note: This parameter is relevant for Windows Vista/2008/7 deployments only.

Send mail at end

This option can be set only if a *sendmail* tunnel has been created.

Send mail to

This parameter is available only if **Send mail at end** is not set to **No**. Use this field to enter the e-mail address to which information must be sent at the end of the deployment.

Configure Network
This parameter, located in the **Network settings** section, is set to **Before software installation** by default. The network setting of the target are set before software modules are installed, enabling the product to use the network settings during the installation of the software modules. Set the parameter to **After software installation** if you want the network settings to be applied after software modules are installed, for example if you intend to physically move the target after deployment and want it to be configured with the network settings for its final location.

Disable cancel button
This parameter, located in the **Client Display** section, allows you to prevent users from aborting a deployment by removing the cancel button. Set the parameter to **Yes** if you want to remove the cancel button from the client display

You can use the newly created deployment scheme to deploy a system profile.

To delete a deployment scheme, select the scheme, then right-click it, and click **Delete**.

Creating media for deployment for Windows operating systems

You can create deployment media such as CDs, DVDs, or USB drives to install machines without connecting them to the OS deployment server.

You can use this kind of deployment when there is no connection or connection to the OS deployment server is very slow.

Some typical situations are small branch offices with slow links and no local deployment server, isolated computers with no connection to an internal network, laptop users currently away from LAN or connected using a modem.

If the data you want to use does not fit on a single CD or DVD, use a USB drive.

Note:

- You must create the deployment media from an OS deployment server or a web interface extension installed on a computer with the same byte order (little endian or big endian) as the one on which you want to use the deployment media.
- Offline deployment media are not supported on virtual machines.

Creating an OS deployment USB drive with the wizard

Tivoli Provisioning Manager for OS Deployment can automatically generate deployment USB drives that replay the deployment process for a given system profile or for any kind of software modules available.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must be formatted as FAT32 or NTFS.

Note: SuSE Linux Enterprise Desktop cloning is not supported on USB drive deployments.

Refer to the *Troubleshooting and support* set of topics for information about problems or limitations related to deployments using a network boot USB drive, and to the product release notes or the readme file provided with the fix pack for the most up-to-date information related to problems or limitations.

The deployment USB drive is self-contained and can be used instead of a CD or DVD to provision a target entirely offline, without using the OS deployment server. These deployment USB drives can also be used to deploy computers without a PXE-compliant network adapter.

To create OS deployment USB drives:

1. Perform one of the following operations:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.
2. Click **Generate Media** or select **Create deployment media** in the contextual menu.
3. Select **Create a deployment USB key** to start the USB key wizard. Click **Next**.
4. Specify the operating system for which to build the CD or DVD. Select **Windows** to load a WinPE deployment engine, **Linux** to load a Linux deployment engine environment, or **Both** to load both.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. (*Optional*) Change settings for targets running the USB key that you are creating.

   Included objects

   When selecting objects to be included, be aware that:
   - The wizard displays all the deployment schemes, system profiles, and software modules currently stored on your OS deployment server.
   - At least one system profile and exactly one deployment scheme must be included in your image.
   - The software application order is automatically included.
7. If your USB key has already been used as a deployment media, you might choose to keep a shared repository of previous data to improve data regeneration speed. If you keep the existing shared repository, you might use more space on the USB key.
8. Plug your USB key into a machine running the web interface extensionand specify its address.
9. Choose the drive matching your USB key.
10. Click **Finish** to close the wizard.

Use the USB drive to deploy a given system profile or any kind of software module.

Creating an OS deployment USB drive with command lines

You can create an OS deployment USB drive that Tivoli Provisioning Manager for OS Deployment can use when a target cannot boot from the network.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must have boot capabilities and a FAT32 or NTFS filesystem. The drive must be already formatted; existing files on the partition are not deleted. USB keys already filled with a bootable operating system might not work.

Note: Refer to the *Troubleshooting and support* set of topics for information about problems or limitations related to deployments using a network boot USB drive, and to the product release notes or the readme file provided with the fix pack for the most up-to-date information related to problems or limitations.

The command line must be used only when the web interface is either inappropriate or unavailable.

Use this command line:

- On Windows operating systems:

```
rbagent.exe -s <OSD_server_ip_address>:<OSD_server_password>
rad-usbget <drive>
keepshared|delshared preferwpe|prefermcp nodes
```

Where:

OSD_server_ip_address
Is the IP address of the OS deployment server.

OSD_server_password
Is the password for the administrative user (typically `admin`) on your OS deployment server.

drive Is a drive letter of the Windows target where you run the `rbagent` command. The rad-usbget command adds requested files to the FAT32 or NTFS partition and makes it bootable. The drive must be already formatted. Existing files on the partition are not deleted.

keepshared
Keeps a shared repository of previous data to improve data regeneration speed. If you keep the existing shared repository, you might use more space on the USB drive.

delshared
Deletes a shared repository of previous data.

preferwpe|prefermcp
Defines if a Linux deployment engine environment or WinPE is loaded from the USB drive, when a target boots from this USB drive, without accessing the network. Only when Linux deployment engine or WinPE is running, does the target connect to the network and try to contact an OS deployment server. If you deploy only Linux, specify `prefermcp` to skip WinPE. You can specify `preferwpe` only if there is a WinPE deployment engine on the OS deployment server.

nodes Defines the deployment settings with a space-separated list of objects. Specify at least `DEPLSET:Default` for the deployment schema, and `PROFILE:SystemID` for the system profile.

You can now boot the target using the OS deployment USB drive instead of the network card. To use the PXE emulation USB key, insert the USB key into the drive and restart the target. If your machine does not boot from the USB key, check the BIOS boot list to see if your optical drive is included in the boot sequence and is listed before the hard disk. Most machines also allow you to select the temporary boot device without changing the boot sequence in BIOS.

Creating OS deployment CD and DVD

Tivoli Provisioning Manager for OS Deployment can automatically generate deployment CDs and DVDs that replay the deployment process for a given system profile or for any kind of software modules available. You can use this feature to create OS deployment CDs and DVDs that can be easily sent through the Internet or by e-mail, to refresh a computer back to its initial working state after installation.

These CDs and DVDs can also be used to deploy computers without PXE compliant network adapter. The creation of DVDs and media spanning is supported. These media can be protected using an activation code preventing unauthorized personnel from using it.

To create OS deployment CD and DVD:

1. Perform one of the following operations:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.
2. Click **Generate Media** or select **Create deployment media** in the contextual menu.
3. Select **Create a deployment CD or DVD** to start the CD and DVD wizard. Click **Next**.
4. Specify the operating system for which to build the CD or DVD. Select **Windows** to load a WinPE deployment engine, **Linux** to load a Linux deployment engine environment, or **Both** to load both.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. Follow the wizard instructions to create an ISO image.

   Included objects

   When selecting objects to be included in the ISO image, be aware that:

   - The wizard displays all the deployment schemes, system profiles, and software modules currently stored on your OS deployment server.
   - At least one system profile and exactly one deployment scheme must be included in you image.
   - The software application order is automatically included.

   Hardware options

   In the hardware options settings some boot options can be customized. By default the options are unchecked but some special cases can require changes. In particular, if the CD or DVD is to be used on a USB drive or as a secondary drive, it might be necessary to specify the option **use BIOS for CD or DVD ROM access**. When this option is selected, on some hardware it might also be necessary to select **disable enhanced disk access** (for IDE CD or DVD) or **disable USB** (for USB CD or DVD) to ensure that Tivoli Provisioning Manager for OS Deployment use of other IDE or USB devices does not interfere with the BIOS access to the CD or DVD. In addition, deploying from the second CD or DVD drive of a target only works if you can ensure that subsequent boots keeps booting on the same CD or DVD drive.

Security issues

For security issues, you might want to protect deployment from the CD with an activation code. When your computer boots on the CD, the activation code is required for the deployment to proceed.

You might also want to hide the content of the ISO image that contains sensitive information such as product keys. To do this, select **Hide the content of CD or DVD** in the CD or DVD Wizard. If you then try to access files in your ISO image, you see the content as `CDROM_content_hidden`.

Size of the ISO file

The wizard allows you to choose the size of the ISO images.

a. Enter the maximum size in the field displayed.

b. Click **Next** and the wizard starts to precompute the ISO file size.

The wizard displays the results for the number of disk images and the size required. You then have the option to:

- Download it directly from the server.
- Use the web interface extension
- Generate it on the server itself in the import directory.
- Generate it on another computer running the web interface extension

Note:

- When creating the ISO files, all objects of type *single file to copy*, *image headers*, and *WIM images* (which includes Windows Vista/2008/7 unattended setup profiles), are put on the first CD or DVD. Therefore, the first ISO file might grow larger than the requested spanning size if the total size of the files to be put on the first ISO requires it.

  For example, if you try to create an OS deployment DVD containing both Windows Vista/2008/7 unattended setup profiles, both profiles must be contained on the first ISO, but their total size is larger than 4 GB. Therefore, the ISO cannot be burned into a single layer DVD. In this case, either use a double layer DVD, or transfer the ISO without burning it.
- When deciding where to generate the ISO image, be aware that:
  - If the estimated size is bigger that 2 GB, do not use the link to download directly from the server, because of limitations of web browsers. An exception to this rule is Mozilla Firefox on Linux, which can extract files as large as 4 GB or more.
  - Because of file system limitations, do not extract files bigger than 4 GB on FAT32 partitions.

Use a CD creation tool to burn the ISO image onto disks.

Note: Vista 2008 Windows 7 Windows Vista/2008/7 unattended setup profiles contain at least one file larger than 1 GB which cannot be split. Therefore, ISO files containing Windows Vista/2008/7 unattended setup profiles must be burned on a DVD.

If you encounter problems when deploying from this CD or DVD on a virtual machine, make sure that the CD drive comes after the hard disk in the boot order.

Setting up an activation code

For security issues, you might want to protect deployment or booting from the CD with an activation code. When your computer boots on the CD, the activation code is required for the deployment or the network boot to proceed.

To prevent being asked several times for the activation code during deployment:

- The deployment scheme included on your deployment CD must have the network setting **Use 'BIOS fall back MBR' to start PXE** set to **No**.
- The boot order of your target must be set to hard disk first and you must boot on the CD manually the first time.
- To set up an activation code for the first time, when creating the deployment CD:
  1. Select **Include activation code protection** in the deployment media wizard.
  2. Enter and confirm the chosen password. You must remember this password if you want to obtain other activation codes for this CD.
  3. Set a password expiration date under **Valid until**.
- To obtain a new activation code, for example, if you must use the CD after the current activation code expiration date:
  1. Click **Generate Media** on the Profiles page to start the deployment media wizard.
  2. Select **Generate a new activation code**.
  3. Click **Next** and follow the wizard instructions to obtain your new activation code. You must remember the password given when creating the first activation code for this CD.

The wizard provides you with the generated activation code that you need when using the CD.

Deploying Windows operating systems

A deployment is the process of installing an operating system on a target, and configuring the operating system for a specific user.

The deployment process

In Tivoli Provisioning Manager for OS Deployment, a deployment is made up of several steps that are automatically run in sequence without user interaction:

1. Hardware configurations are optionally deployed, for example, to create RAID volumes.
2. Partitions are created on the hard disk, and then formatted according to information contained in the system profile.
3. All deployment objects (system profiles, partition files, and software modules) are downloaded to a temporary storage location on the hard disk.
4. Operating system files are written in the hard disk partitions, creating a bootable operating system with files and applications configured by database bindings between the *target* and *software modules*.
5. Target-specific configuration, such as the *host name* or the *product key* are gathered from the database to create a textual configuration file used by the system preparation tool.
6. The operating system is started, allowing Sysprep or LinPrep to configure the operating system according to information stored in the Tivoli Provisioning Manager for OS Deployment database.

7. Additional software is optionally installed, if it must be installed after the operating system.
8. The temporary storage location is cleaned. Installation files are removed.
9. Tivoli Provisioning Manager for OS Deployment takes control again when Sysprep or LinPrep has completed and rebooted the target, and displays a message indicating the status of the deployment.

When the deployment is complete, the operating system is installed and ready to be used by the user defined for this target in the database.

Network boot scenarios

Depending on the number of OS configurations bound to a specific target, a target behaves differently when it boots on the network:

- If no OS configuration is bound to the target (for example, when a target starts for the first time and has not been configured), a special screen is displayed that asks the administrator to configure an OS configuration binding for this target on the OS deployment server. Deployment is not possible until an OS configuration is bound to the target.
- If one or more OS configurations is bound to this target, but no deployment has been scheduled on the server, a screen is displayed with a list of all the OS configurations bound to the target. Clicking on an item in the list starts an interactive deployment for the selected OS configuration, using either the **Default** deployment scheme (if no deployment scheme has been configured for this target), or the deployment scheme used during the last deployment.
- If one or more OS configurations are bound to this target, and a deployment has been scheduled on the server for a specific OS configuration, the target immediately starts the deployment without requiring any user intervention.

Deployment requirements

To start a deployment on a target, several elements must be present in the database.

Note:

- To deploy a Windows operating system, you must have a WinPE 3.0 deployment engine stored on your OS deployment server.
- During the deployment, do not edit the WinPE deployment engine that you are using.

The following elements are required:

- A *deployment* scheme associated with the target to deploy. The deployment scheme determines how to deploy the operating system on the target. If there is no association between a deployment scheme and the target to deploy, Tivoli Provisioning Manager for OS Deployment automatically uses the **Default** deployment scheme.
- An operating system *configuration* that is used to select which operating system to install. If there is no OS configuration associated with the target to deploy, the deployment does not start.
- Optional *software modules* to install in addition to the operating system during the deployment process. If there is no software module associated with the target to deploy, the operating system image is deployed without modification.

The OS configuration and the software modules can be considered to be the *content* of the deployment, while the deployment scheme is the *how* of the deployment.

The database keeps information about associations (*bindings*) between targets and deployment schemes, between targets and OS configuration, and between targets and software modules. These bindings can be configured manually or with binding rules (for example, `deploy configuration windows XP on targets whose model name starts with Dell`).

The minimal binding required to start a deployment is an OS configuration. If no configuration is bound to a target, the deployment does not start. In practice, however, Tivoli Provisioning Manager for OS Deployment always asks for an OS configuration and deployment scheme when beginning a deployment.

Tools to start and configure deployments

Bindings between targets and deployment elements are necessary to perform a deployment. You can create and edit these bindings in the **OS configurations** panel of the **Target details** page.

The Target Monitor provides functions to prepare a deployment, start a deployment, follow the progress of a deployment, and organize targets.

Binding rules, used to create permanent implicit bindings between targets and deployment elements without having to explicitly create the binding for each target, are created using the web interface. OS configurations and software modules contain a specific section at the very bottom of the **Details** page for creating automatic binding rules.

Restrictions on user privileges

On Windows Vista/2008/7 cloning system profiles, it is not possible to give a user administrator privileges if the user name existed in the reference target without these administrator privileges. Trying to do so, either from the **Target details** page or from the **OS configuration details** page, results in a failed deployment.

Starting a one-time deployment

You start deployments in the web interface by indicating on which target or targets the deployment must occur.

Windows To deploy any Windows system profile, you must have a WinPE 3.0 deployment engine stored on your OS deployment server.

Vista 2008 Windows 7 Here are the requirements to deploy Windows Vista/2008/7.

- To deploy an unattended setup profile for Windows Vista/2008/7 32-bit, the minimal size of the hard drive of your target is about 10 GB.
- To deploy an unattended setup profile for Windows Vista/2008/7 64-bit, the minimal size of the hard drive of your target is about 20 GB.
- If you have a Volume Licence edition, the product key field in the system profile details must be empty for the deployment to succeed.

To start a deployment:

1. Select a single target or multiple targets on the Target Monitor page. To select multiple targets or deployment, select an administrative group, a custom list, a subnet, or click on individual target names while holding down the Ctrl key.
2. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
3. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
4. If you do not choose this option, select the type or types of deployment you want to perform. You can install additional software only if you deploy an operating system.
   a. If you have selected **Perform hardware configuration tasks**, indicate which hardware configurations you want to deploy.
5. Select **Simple deployment** and click **Next**
6. Follow the deployment wizard instructions to select a deployment scheme, an OS configuration and optionally software modules, and to set up deployment options.

   Three options are available in the deployment wizard to deploy targets without physically interacting with the computers.

   - **Try to wake up targets currently powered off using WOL** asks the Target Monitor to send IBM Wake on LAN packets to wake up targets. Waking up targets only works on carefully designed modern computers. A target can only be woken up if its network adapter and its system board support Wake on LAN packets, and if the network adapter has been shut down properly. If the network adapter is not in the appropriate power state, Wake on LAN packets will not wake the computer up. This is not specific to Tivoli Provisioning Manager for OS Deployment, but is rather a general limitation of the Wake on LAN technology.
   - **Try to wake up targets using management interface** tries to turn on the target using command lines to the management interface. This works only for targets with an interface management known to the product and properly identified.
   - **Try to reboot targets running theweb interface extension** asks the Target Monitor to contact the targets if they are running under Windows and send a reboot request. If you are not running Windows, you cannot reboot targets remotely. They must be rebooted manually. If you are running Windows, you need the web interface extension running with the correct privileges to run a remote boot.

   If you have not selected one of these options or if they do not work, and if the target you are trying to deploy is not powered on, turn it on now and make it start on the network.
7. If your system profile uses the **driver specific bindings mode**, a check is performed to warn you of potential driver issues. If your system profile uses the **regular software binding rules**, this check cannot be performed and a warning message is displayed. If you want to switch from one binding mode to another, you must do it on the system profile itself, on the **Profile details** page.

For a Windows cloning deployment, the target goes through the following stages:

1. **Prepare one partition** Tivoli Provisioning Manager for OS Deployment creates partitions on the hard-disk according to the information stored in the system profile associated with the OS configuration being deployed.
2. **Install Operating System files** Tivoli Provisioning Manager for OS Deployment downloads deployment files on the hard-disk and installs the operating system.

3. **Generate Windows Sysprep configuration file** Tivoli Provisioning Manager for OS Deployment creates the files needed by Sysprep.
4. **Windows Sysprep Mini-Setup** Tivoli Provisioning Manager for OS Deployment runs Sysprep Mini-Setup.
5. **Install additional software** Tivoli Provisioning Manager for OS Deployment installs the various software modules according to their application order, handling multiple reboots if required.
6. **Complete Windows post-OS configuration** Tivoli Provisioning Manager for OS Deployment finalizes operating system settings not configured by Sysprep.
7. **Cleanup deployment data** Tivoli Provisioning Manager for OS Deployment deletes deployment files.

When the deployment is complete, the server either displays a green banner on the target, boots in the operating system, or powers the target off, depending on how the deployment scheme is configured.

After deployment, the following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Partitions are aligned on megabytes rather than on cylinders. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

If you want to test the operating system deployed, you must first ensure that the target is not configured to start on the network, or you will get a menu allowing you to run the deployment again.

Windows Vista/2008/7 offline servicing

Offline servicing allows the OS deployment server to patch Windows Vista/2008/7 image with HotFixes and language packs before the deployed operating system needs to be connected to the network, thus preventing the risk of contracting viruses before the operating system is fully functional and is patched with security updates.

Offline servicing also enables you to use language packs with versions of Windows Vista other than Enterprise or Ultimate.

Offline servicing is automatically performed on aWindows Vista/2008/7 deployment when

- A HotFix (.msu) or a language pack is bound to the OS configuration.
- The `unattend.xml` parameter file contains a `<servicing>` tag.

To perform offline servicing, you must have a WinPE 3.0 deployment engine on you server.

Deploying a hardware configuration

A wizard allows you to effortlessly deploy hardware configurations.

To start a hardware configuration deployment you must first have at least a hardware configuration environment and a hardware configuration.

Note: You can not deploy a hardware configuration from a target started with a network boot media.

1. Select a single target or multiple targets on the Target Monitor page. To do this go to **Server → OS deployment → Target Monitor**. To select multiple targets or deployment, select an **administrative group** , a **custom list**, a **subnet**, or click on **individual target** names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the second screen of the deployment wizard, you must select at least **Perform hardware configuration** tasks and if you want to proceed with the Operating System/Software deployments you can also select another option.
4. Select one or several hardware configuration(s) you want to apply on target. RAID Configurations, BIOS Settings, BIOS Updates or Hardware custom configurations are classified in a matching folders.
5. Follow the deployment Wizard as it is described in the chapter Deploying depending on the options you chose above.

Every configuration you selected will automatically use the appropriate environment and only be applied if the model is matching the target.

Redeploying

This function is a special deployment scheme that gives you the ability to rapidly restore an image to a computer from a hidden partition on the computers hard-disk.

During the original image deployment to the computer, Tivoli Provisioning Manager for OS Deployment creates a hidden partition on the hard-disk of the target computer. When it has finished deploying the master image on the computer, it stores a reference image into the hidden partition. It is possible to store one or more reference images into a hidden partition on the computer.

Note: Before running a deployment task on a machine with a redeployment partition, ensure you remove the hard disk partition content.

Each time the system is booted, either off the hard-disk or using network boot, Tivoli Provisioning Manager for OS Deployment intercepts the boot process of the computer and presents a customizable menu of possible actions. Those actions are:

- Boot the system off the current image on the hard-disk.
- Do a quick cleanup of the currently deployed image against the reference image and restore the image from the hidden partition.
- Do a format and full restore of the reference image from the hidden partition. Using this function, it is possible to effectively have a fresh image deployment every day for the optimum performance of a system.
- Choose and deploy another configuration available on the hidden partition. This option takes as long as the format and restore option.

The purpose of redeployment

A computer generally works the best and the fastest on the day that it is installed. At that time, the system is completely clean, free of any undesirable processor-consuming gadgets, and all programs are configured for their optimal use by the system administrator. The purpose of redeployment is to ensure that the system is reset to this optimal state at every boot (or at some fixed interval).

There are three categories of systems that experience the most visible need for the redeployment technology:

Public computers
such as schools, universities, and Internet cafes, where users cannot be relied on to preserve the computer integrity, because the computer is not their own

Critical systems
such as banks, insurance companies, and industrial plants, where the company cannot afford to risk computers being reconfigured or infected by malicious software

Embedded systems
such as ticket machines, airport information systems and ATMs, that must be quickly rebuilt to their original OS configuration, without using a specific infrastructure

Because redeployment often occurs at the user's desk, it is necessary to find a solution that is quick, easy to use, does not require any significant infrastructure, and does not affect the work process of other users. This rules out standard deployment tools, because they impose a significant load on the network and affect other users' ability to perform their tasks.

Note: The redeployment feature is not intended to be used on virtual machines. On virtual machines, you should leverage the snapshot feature of your hypervisor rather than use the redeployment feature.

The redeployment process

Redeployment involves several steps, including creating a reference image of the target, and saving it as a redeployment partition.

Redeployment steps

Tivoli Provisioning Manager for OS Deployment addresses the challenge of redeployment with the following steps:

- At the end of a deployment, Tivoli Provisioning Manager for OS Deployment creates a reference image of the target, and saves it into a protected redeployment partition (invisible to the user and to the operating system itself). This increases deployment time by roughly 10% compared to a simple deployment,, as most of the files are already present as file archives on the disk at that time.
- Every time a target starts, Tivoli Provisioning Manager for OS Deployment hooks the boot process before the operating system starts (using PXE or a special Master Boot Record).
- If configured to do so, Tivoli Provisioning Manager for OS Deployment authenticates the user of the target against the server database to restrict the use or the maintenance of the target to authorized persons only.
- If configured to do so, Tivoli Provisioning Manager for OS Deployment offers the choice of several OS configurations available on the target (multiboot), and of several levels of "cleaning".
- Using the reference image saved during deployment, Tivoli Provisioning Manager for OS Deployment resynchronizes the hard-disk content to its reference state. This typically takes only a few seconds, but can take up to a few minutes if everything on the hard disk has been deleted.

Offline redeployment limitations

Offline redeployment behaves slightly differently from online redeployment as the OS deployment server cannot be contacted for information. These limitations are removed after the target contacts the OS deployment server again. For example, interrupted tasks are not automatically resumed and changes to the partition scheme cannot be recovered.

Moreover, authentication with offline redeployment does not work. A message warns the user.

Note: If you plan to use redeployment with multiple OS configurations offline, make sure that all the preloaded OS configurations have exactly the same partition layout (number and size), because Tivoli Provisioning Manager for OS Deployment cannot create new partitions offline or to resize existing partitions offline. Failure to do so prevents you from redeploying offline some of the preloaded OS configurations.

Redeployment with multiple operating systems

You can preload up to three operating systems on a target, with a menu allowing the user to select which operating system to start.

Scenario

You want to provision the computers of a classroom with three different operating systems (for example, Windows XP by cloning, Windows Vista, by unattended setup, and SuSE 10 by cloning). When entering the classroom, the student must choose between the three operating systems. For security reasons, you want to make sure that the operating system which is started is always in a clean state. You also want the selected operating system to install and start quickly.

Principles

To achieve this, you must install each operating system in its own partition, save the OS configurations in a protected partition. Before you start an operating system, you do a rapid verification of the operating system partition with the information in the protected partition.

Requirements

For you multiple operating systems to cohabit in a single target and to be able to start them individually, you must follow these guidelines strictly:

- The hard disk of the targets must be large enough to contain the three operating systems and the protected partition.
- You must create a separate system profile for each operating system.
- All the profiles must have the same number of partitions, in the same format.
- Each operating system must be in a distinct partition, and all other partitions must be empty during the system profile creation.
- Each operating system must be in a primary partition, and there is a maximum of three primary partitions.
- In the system profiles, partition numbers cannot be modified.
- An offline refresh does not update the partition table.

Procedure

1. Create your Windows XP cloning system profile.
   a. Start a target with the Windows XP CD.
   b. Create and format one large partition.
   c. Install Windows XP on the partition.
   d. Customize your installation.
   e. Determine the best size for the partition.
   f. Clear the administrator password.
   g. Run Sysprep.
   h. From the web interface, clone your target to create a new system profile.
2. Create your Windows Vista unattended system profile.
   a. From the web interface, create a new unattended system profile with the profile wizard, following the instructions.
      1) Create a small partition 1.
      2) Create a large partition 2 for Windows Vista.
   b. Customize your OS configuration.
      1) Set the **administrator name** in the configuration.
      2) Optionally, bind software modules.
      3) Determine the best size for partition 2.
3. Create your SuSE 10 cloning system profile.
   a. Start a target with the SuSE 10 CD.
   b. Delete partitions 1 and 2. Recreate and format two small partitions.
   c. Create one large primary partition (EXT2) for / (partition 3).
   d. Create a swap partition of 1 GB (logical partition).
   e. Install SuSE 10 in partition 3.
   f. Customize your installation.
   g. Determine the best size for the partition.
   h. From the web interface, clone your target to create a new system profile.
4. Update the OS configurations.
   a. Edit the partition scheme for each OS configuration so that partitions have the same size on each OS configuration.
   b. Use the best size found for each operating system.
   c. Set the options**Must be deployed** and **Must be redeployed** so that only the partition containing the operating system is actually deployed or redeployed for each system profile.
5. Test each system profile. Each operating system installs in the correct partition, without impacting other partitions.
6. Create a specific deployment scheme for this redeployment.
   a. Export the three system profiles into a RAD file.
   b. With the deployment scheme wizard, create a new deployment scheme enabling redeployment.
   c. For **Protected redeployment partition size**, give 200% of the size of the RAD file you have just created.
7. Preload the system profiles on your targets.
   a. Select the targets in the web interface.
   b. Select **Deploy now** in the contextual menu.

c. Select **Redeployment preload** in the deployment wizard.
d. Select the deployment scheme you have just created.
e. Select the three OS configurations that you have prepared.
f. Optionally, select additional software modules.
g. Click **Customize GUI** if you want to customize the boot menu appearing in the target.

When your targets boot, they now display a menu with the three possible operating systems in which they can start.

Configuring a deployment scheme for redeployment

Redeployment is a feature that affects *how* the target is being preinstalled, not *what* is in the deployed OS configuration. Redeployment is enabled by customizing a deployment scheme.

Because redeployment is basically the replay of a standard deployment operation, you must first configure a regular deployment process, and try it on a test computer. When you have performed these two stages, follow the instructions provided to turn your one-time deployment OS configuration into a redeployment OS configuration.

To customize a deployment scheme for redeployment, you can

- Create a new deployment scheme with the deployment Scheme Wizard
- Modify an existing deployment scheme with the deployment Wizard
- Edit the parameters of an existing deployment scheme manually

The following steps are based on the first and second options, which are very similar.

1. Follow the first alternative to create a completely new scheme, and the second alternative to modify an existing scheme with the wizard:
   - Go to the **Task templates** page and click **New deployment scheme**. This launches the deployment Scheme Wizard, which guides you through the customization of deployment parameters.
   - Go to the **Task templates** page. Select a deployment scheme, and click **Edit parameters using a wizard**.
2. Follow the instructions of the wizard in the same way as for a regular deployment, until you reach the panel called **On-site deployment features**.
3. Select **Enable support for quick redeployment of the same OS configuration** and click **Next**.
4. On the next panel, **Redeployment option**, select **Yes, keep** IBM Tivoli Provisioning Manager for OS Deployment **images in a protected partition**. Optionally modify the space that you want to allocate to this special partition, and click **Next**.

   Note:

   a. The protected partition size must be at least as large as the total size of all system and software images to be deployed on the computer, because it retains all these images. If you are unsure of the space required, start with approximately 800 MB for a Windows 2000 configuration, 1500 MB for a Windows XP configuration, or 1500 MB for a Linux configuration. If you want a more precise number, check the image sizes reported in a deployment log, and round up the total to accommodate the miscellaneous structures used for redeployment.

b. The space that you allocate to the redeployment partition is subtracted from the hard-disk total capacity detected by Windows or Linux. The user cannot detect, access, or delete this protected area from the operating system disk manager. It is not simply a hidden partition, but a hardware-protected area, as defined in ATA-5 specification. If necessary, you can recover this space by running another deployment operation.

5. Click **Finish** to complete the customization process and obtain a deployment scheme ready for redeployment.

Edit the parameters manually:

1. Go to **Server** → **OS deployment** → **Task templates**.
2. Select a deployment scheme
3. Click **View deployment parameters**
4. Click **Edit** in the section header in which you want to modify parameters.

Preloading for redeployment

Before you can redeploy a target, you must preload one or several OS configurations.

For a successful redeployment, targets must not **Boot on hard-disk if idle**. Make sure this target parameter is not selected for the targets you want to redeploy.

After you have created an appropriate redeployment scheme, you can begin the preload of the OS configurations of your choice on the target. This operation must be initiated using the Target Monitor page of the web interface.

1. Select the targets to deploy and select **Deploy now** from the contextual menu to start the deployment wizard.
2. Select **Redeployment preload** and click **Next**.
3. Follow the instructions of the deployment wizard.

   Note:

   a. When you select a deployment scheme, only those configured for redeployment are displayed. If you do not have any scheme ready for redeployment, a warning message appears.

   b. Preloading more than one OS configuration is supported, but increases the preload time.

The preload automatically starts when the targets boot, just like with regular one-time deployments. The process goes through the same steps, with one exception. When Sysprep or `LinPrep` has completed and after all software modules have been installed, an image of the fully configured target is stored on the redeployment partition. If you have selected multiple OS configurations, the process repeats for all OS configurations in turn, until all redeployment images are ready.

Customizing the redeployment menu

You can customize the menu entries that you see in the user interface when starting a target in redeployment mode. Each OS configuration can define one or more menu entries, and the complete menu is the union of all entries defined by all available OS configurations.

After having selected **Redeployment preload** in the deployment wizard and selected the deployment objects:

1. Click **Customize GUI** in the deployment wizard. This opens the menu customization interface which is divided into three parts:
   - A left column with instructions on how to modify the menus and editable fields
   - A bottom banner with action buttons
   - A view of the target screen as it will appear
2. Click **New menu item**.
3. Modify the captions and actions.
4. You can select one of the following actions:
   - **Format and restore**
   - **Quick restore**
   - **Boot on OS**
5. If you want to protect a specific menu item from unauthorized users, you can set up a global password or user authentication for that user by selecting an appropriate value under **Authentication**. To make full use of this feature, you must first have defined authentication domains in the **Server parameters**. Three authentication formalisms are available

   Authenticate locally on RAD *group*
   uses the local user database to authenticate a user. The optional *group* parameter can be used to restrict the verification to a specific group of users. This type of domain is supported by both Windows NT® and UNIX versions of the OS deployment server.

   Authenticate on NT server *server:group*
   forwards authentication requests to the NT server specified by the mandatory parameter *server*. The optional parameter *group* can be used to restrict the verification to a specific group of users. This type of domain is supported by the Windows implementation of the OS deployment server only.

   Authenticate on Radius server *ipaddr:secret*
   forwards authentication requests to the Radius-compliant device specified by the parameter *ipaddr*. The value of the parameter *secret* is used as the secret for the Radius communication, and must match the secret stored in the configuration of the Radius device for the protocol to work.

   Note: Authentication with redeployment does not work if the target is offline (the target has no network connection and boots from the hard disk). A message warns the user. If you plan to redeploy offline, use a global password rather than user authentication.
6. Click **Save** and then **Close** to exit this window.

Formatting hard disk and restoring files:

With this option, your partitions are always reformatted and all the files restored before you boot into the operating system.

After your targets are preinstalled for redeployment, they always boot into the user interface, independently of the selected boot order in the BIOS (disk or network). If user authentication has been configured, targets connect to the OS deployment server using the PXE network adapter even if they start from the hard-disk.

On the target, select the OS configuration to be restored.

After an OS configuration has been selected, Tivoli Provisioning Manager for OS Deployment completely format the disk and then restore all files. The default behavior is to:

1. Format the disk partitions as specified in the system profile.
2. Restore all the files from the hidden partition.
3. Boot on the selected operating system.

Using quick redeployment:

This option is the typical way to use redeployment. A fast verification of partitions and files is run and, fixes are performed if needed before the target boots into the operating system.

After your targets are preinstalled for redeployment, they always boot into the user interface, independently of the selected boot order in the BIOS (disk or network). If user authentication has been configured, targets connect to the OS deployment server using the PXE network adapter even if they start from the hard-disk.

On the target, select the OS configuration to be restored.

After an OS configuration has been selected, Tivoli Provisioning Manager for OS Deployment automatically restores it as quickly as possible. The default behavior (which typically takes only a few seconds to run) is to:

1. Verify that the disk partitions match the wanted system profile, and fix them if needed.
2. Verify that all partitions have the appropriate file content, and fix them if needed.
3. Boot on the selected operating system.

Booting on the installed operating system:

This option allows you to boot on the currently installed operating system, without any verification. It is fast, but it does not prevent operating system corruption.

After your targets are preinstalled for redeployment, they always boot into the user interface, independently of the selected boot order in the BIOS (disk or network).

If user authentication has been configured, targets connect to the OS deployment server using the PXE network adapter even if they start from the hard-disk.

On the target, select the option that allows you to boot on the operating system.

The target boots directly in the installed operating system, without any disk partition or file verification.

Monitoring deployments

There are several ways available to monitor the deployment progress.

Monitoring deployment progress with theTarget Monitor

You can use the Target Monitor to monitor deployments remotely. Information is located on the **Target Monitor** page and on several tabs of the **Target details** page.

On the **Target Monitor** page, the target color changes during the deployment. When PXE is activated, targets are monitored on a regular basis. The color of the icon is updated as soon as the status changes. By pointing to the target icon, you can get a description of the target status.

Note: A successfully deployed computer can continue to have a yellow icon (indicating that the deployment is still in progress). This reflects a PXE activation problem. The computer, having booted on the hard disk, is not using the network to inform the OS deployment server of its status. To remedy this, select the **Use 'BIOS fallback MBR' to start PXE** check box in the deployment scheme wizard. This forces the computer to boot through the network first.

If the deployment scheme used is configured to collect inventory information about target hardware (which is the default), you can see information about target hardware in the **Inventory** panel of the **Target details** page for that target (double-click on the target to go to the details page).

At the end of the deployment, the target icon shows either a green screen (success), or a red screen (failure). The deployment logs stored on the OS deployment server provide information about the deployment process. They are particularly useful in case of deployment failure to track its cause. To access the logs, double-click the wanted target. This opens the **Target details** page. Select the **Logs** tab to display a list of logs. To view a specific log, click its description. To download it, click **download** immediately after the log description.

Note: Logs are text files with UTF-8 formatting. If you are using a Windows operating system, you can view log files adequately by opening them in Microsoft WordPad.

There is only one log file for each deployment. This log file contains information about the different stages of the deployment process, including reboots and information provided by the operating system being deployed.

If any log information needs to be propagated to the OS deployment server outside of any task, an *idle* log file is created to store this information. The idle log file is created on demand and does not therefore exist for all the targets.

Another place of interest for information about a current deployment or another current task is the **Task history** tab, where each task of the target is listed. For each current task, the following information is provided:

- Description
- Status
- Scheduled date
- Start date
- Progress rate
- End date
- Download link to the log file
- Download link to the task file
- Download link to the `bom` file

Note: You must scroll to the right of the **Task history** tab to see all the fields.

The log file contains the target log. The task file contains all the parameters of the task. The `bom` file contains target-specific parameters for the given task.

The log file, the task file, and the `bom` file are needed by the development team to fix defects. Make sure that you download these three files if you suspect the presence of a defect in the software.

To cancel or destroy a task, select the task and select **Cancel target task** or **Destroy target task** in the contextual menu. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on running tasks, because the task destruction can cause strange behavior.

To view tasks scheduled for a later time, go to the **Tasks** page.

Monitoring deployment progress with the Tasks page for Windows operating systems

The **Tasks** page is also a useful source of information to monitor a deployment (and other tasks). You can also cancel tasks from there.

Go to **Server** → **Server history** → **Tasks**. .

The description field of each deployment in the **Tasks** page is headed by the keyword *Deploy* for easy retrieval. The information provided includes

Description
Is headed by specific keywords, indicating the type of task. *Deploy* is the keyword for deployment tasks.

Execution
Is the scheduled date and time for the execution of the task.

State Uses icons to represent if the task is pending, in progress, completed, and so on. If in doubt to the meaning of a state icon, browse over it to get a state name.

Progress
Indicates the rate of completion of the task as a percentage.

Expire Indicates when the task information is going to be removed from the page.

Tasks are expandable by clicking their **+** sign. An expanded task displays information about its targets. The target information fields are:

- IP address
- Hostname
- Start date and time of the task
- State
- Progress rate
- Status date

If, for any reason, you want to cancel a running or scheduled task, you can easily do so by following these steps:

1. Expand the task
2. Select the target for which you want to cancel the task
3. Select **Cancel task** from the contextual menu
4. It is also possible to *destroy* tasks. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on a running task, because its destruction can cause strange behavior. To permanently delete tasks:

a. Select one or several tasks. To select multiple tasks, use the Shift key for a range of tasks and the Ctrl key for individual tasks.
b. Select **Destroy task** from the contextual menu

Receiving an e-mail notification

To receive an e-mail notification at the end of a deployment, you must configure a TCP tunnel called *sendmail*.

To receive an e-mail notification at the end of a deployment, you must have configured a *sendmail* TCP tunnel.

Note: The OS deployment server supports only sendmail servers without authentication.

There are two options to configure a deployment to receive an e-mail notification:

- You can edit the deployment scheme used for deployment to include the notification information.
  1. Go to **Server** → **OS deployment** → **Task Templates**.
  2. Select **Deployment schemes** and double-click a specific deployment scheme name to edit it.
  3. Click **Edit** in the **General settings** section.
  4. Under **Send mail at end:**, select the type of notification that you want. You can choose among:
     - **No**
     - **Whatever the notification is**
     - **If deployment is successfully completed**
     - **If deployment failed**
  5. If you selected a notification, you must now enter a valid e-mail address to which the notification is sent, under **Send mail to:**.
- You can modify the settings of the deployment scheme in the deployment wizard. Step deploy/tosd_sendmail.dita#Receivinganemailnotification/first and possibly step deploy/tosd_sendmail.dita#Receivinganemailnotification/second are available.

Depending on your selection, you will receive an e-mail notification at the end of the deployment.

The notification e-mail looks like this:

The target with MAC *xxxx* / IP **xxx** has completed an activity *activity description*.

You can now deploy targets with the edited deployment scheme and receive e-mail notification at the end of the task.

Creating a sendmail TCP tunnel:

A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Note: OS deployment server supports only sendmail servers without authentication.

1. Go to **Server** → **Server parameters** → **Predefined channels**.
2. Click **New tunnel**.

3. In the TCP tunnel information screen enter,
   a. The name of the tunnel. In this case the name is `sendmail`.
   b. The host name or IP address of the target.
   c. The TCP port of the target.
4. Click **Save**.

You can now configure deployment schemes to send an e-mail notification at the end of a deployment.

Bindings created during deployment

The Target Monitor creates a binding between the OS configuration chosen for the deployment and the targets being deployed. This binding is added into the database and can be later removed using the Target Monitor.

Because at least one configuration binding now exists, targets that have been deployed no longer show the locked screen. They show a boot menu with a list of the OS configurations that are bound to the target. This allows the target user to manually restart the deployment of an already deployed OS configuration by clicking on the corresponding line in the menu.

You can remove, add, or modify OS configurations and software bindings using the Target Monitor.

Chapter 4. Working with Linux operating systems on x86 and x86-64 targets

This section provides information on how to work with the product to deploy Linux operating systems.

System profiles

A system profile is the partition layout and list of files to deploy an operating system, either by unattended setup or by cloning, from a reference target or from a reference image.

The main purpose of Tivoli Provisioning Manager for OS Deployment is to deploy an operating system on targets by replicating a reference system. However, unattended installation of operating systems is also possible. The latter case Tivoli Provisioning Manager for OS Deployment does not replicate a reference system, but merely provides the correct parameters to Windows or UNIX setup for a fully unattended installation.

There are a number of differences between an unattended installation and disk cloning. First, creating an unattended installation in Tivoli Provisioning Manager for OS Deployment is straightforward. All of the necessary tasks are performed on the server, using the Web interface. In contrast, a cloning-mode system profile requires you to configure a target, prepare it for cloning, and run the cloning process directly on the target. However, the native mode of operation of the product is centered around cloning-mode system profiles, because this method of deployment is faster than unattended installation. When deploying computers on a large scale, unattended installation is not possible. Novice users might start with creating unattended installation profiles because this is easier than cloning-mode profiles.

Note:

- To avoid failures in creating or deploying asystem profile, clean up the temporary directory inside the OS deployment server installation directory on a regular basis.
- To create or deploy a system profile from a physical or virtual machine at least 512 MB RAM is required.
- The exFAT filesystem is not supported.
- System profiles can have a maximum of 3 primary partitions. Therefore, you cannot clone a target with 4 primary partitions.

Creating system profiles

There are distinct types of system profiles. The profile wizard guides you through the creation of system profiles for each type.

Creating an unattended setup system profile for Linux operating systems

You can install operating systems using standard installation processes in unattended mode. Unattended setup simplifies the task of preparing computers for the native mode of operation of disk cloning.

© Copyright IBM Corp. 2006, 2010

During deployment of a Linux unattended setup profile, /swap is used as temporary cache partition. It should be at least 2 GB to hold all the necessary files.

To create a new system profile:

1. Go to **Server → OS deployment → System profiles**.
2. Click **New Profile**. A system profile wizard opens to guide you through the steps of creating a profile.
3. Select **Unattended setup** in the first pane of the profile wizard.
4. Select your operating system from the list and click **Next**.
5. Follow the instruction of the profile wizard.

When your first unattended installation profile is created, you can use it to deploy targets. Then you can create a cloning-mode system profile, because unattended installation profiles have a longer deployment time than cloning-mode system profiles. You can use your unattended installation profile to prepare the computer that you refer to when creating your first cloning-mode system profile.

Creating a cloning-mode system profile for Linux operating systems

To obtain a cloning-mode system profile from a reference target you must first prepare the reference target.

Note: As of version 7.1 of the product, LVM2 is supported for cloning. During the cloning process, the physical volume under LVM2 is moved to an extended partition.

The product supports only one volume group per disk and a volume group cannot span over two disks.

1. Prepare the reference target.
2. Clone the reference target.

Preparing the reference target:

To create a cloning-mode system profile, you must first create the reference OS configuration (called the *system profile*) that you want to deploy.

You must perform this task on the reference target not on the OS deployment server.

The OS deployment server does not perform cleanup on the reference target. You are responsible for deleting useless files and services before creating a new image as follows:

- Delete the temporary Internet cache
- Delete your temporary directories and files
- Disconnect your network drives and remote printers
- Empty the recycle bin
- Delete partitions using a file system that is not supported by the product, or reformat them

Linux-specific advice:

When preparing a Linux system profile, there are two main issues: the space for the temporary cache partition and the bootloader.

You must ensure that the partitioning scheme provides enough space for the temporary cache partition during deployment. For Linux cloning, `/boot` is used as temporary cache partition. It must be large enough to hold image file headers and software modules. The recommended size is 256 to 512 MB, unless you have very large software modules and must augment this size. If you do not want to change the `/boot` partition of your reference computer, you can edit the size of the `/boot` partition directly in the cloned system profile.

The Linux bootloader is also important. Tivoli Provisioning Manager for OS Deployment supports only Grand Unified Bootloader (GNU GRUB). You can install GRUB on the bootsector of the Linux `/boot` partition or on the root partition. If you plan to use redeployment, it is mandatory to install GRUB in the boot sector of the Linux `/boot` partition. To start your system properly with GRUB, ensure that you have a standard MBR on the disk, with the boot partition flagged as bootable.

You do not must run a system preparation tool for deploying Linux using Tivoli Provisioning Manager for OS Deployment. Tivoli Provisioning Manager for OS Deployment automatically installs and runs its own system preparation tool, LinPrep.

The Xen virtualization package part of RHEL5 is not supported. You must remove the Xen package from your reference computer before you clone it.

NTFS and exFAT partitions are not supported in Linux system profiles. Use FAT 32 partitions instead.

Cloning the reference computer:

When you have prepared your reference computer, you are ready to create your system profile. You can create it from the Web interface with the profile wizard.

1. Go to **Server** → **OS deployment** → **System Profiles**.
2. Click **New profile**
3. Select **Cloning from a reference computer** and click **Next**
4. Follow the instruction of the profile wizard.

Creating a universal system profile for Linux operating systems

When creating a software module, do not enter a hardware model because a universal system profile must be deployable on several types of hardware. If you entered a model name in the Profile Wizard, you can delete it when you edit the first set of parameters of the Profile details.

To deploy your universal system profile with a type of hard disk different from that of your reference target (for example, going from a parallel hard disk to an SCSI or an AHCI disk), the system handles hardware changes by rebuilding the initial ramdisk (or initrd) during deployment. The rebuilding of initrd is available for 32-bit cloned images only.

Organizing and editing system profiles

After you have created a system profile, you can view it on the OS deployment server through the web interface. The profiles are listed on **> Server > OS deployment > Profiles** page, in the **System profiles** pane. Each blue jacket represents a system profile (that is, the hard-disk partition images).

If you want to organize your system profiles, you can create subfolders by following these steps:

1. Select the parent folder with a left mouse click.
2. Call the contextual menu with a right mouse click.
3. Select the **Add a new profile folder** menu item.
4. Enter the new folder name.
5. Click **OK**.

You can then move profiles (by dragging-and-dropping the profile icons) from the top folder, where they are automatically created, to the appropriate subfolder.

Editing system profiles

To display and edit the parameters associated with a given profile:

1. Double-click a system profile to open the **Profile details** page.
2. Click **Edit** on top of the parameter sections to edit the parameters.

Browsing partition files

You can browse partition images stored on your server.

1. Go to **Server** → **OS deployment** → **System profiles**. Double-click on a profile to view the details.
2. In the **Original partition layout** section, click **Browse image of primary partition 1**.
3. You can expand or update the whole partition or a part of it.
   - To expand the whole or part of the partition:
     a. Right-click the folder you want and select **Expand on local disk**.
     b. Choose the computer where you want to expand and store the files contained in the selected partition.
     c. Specify the destination folder where to extract the partition files.
     d. Follow the instructions of the image wizard to expand the partition.

     Note: You must expand the partition to an empty directory. If you select a folder that is not empty the extraction fails.
   - To update the whole or part of the partition:
     a. Right-click the folder you want and select **Update from local disk**.
     b. Specify the source folder of the OS deployment server where your updated data are located.
     c. Specify the destination folder where to extract the partition files.
     d. Follow the instructions of the image wizard to update the partition.

On the **Partition image explorer** page, you can create a new directory by selecting **Add new directory** in the contextual menu. You can also modify or add files by selecting **Upload file** in the contextual menu.

Note: File upload is limited to 16 MB.

Changing the partition layout

Partition layout can be updated to resize partitions, assign mount points, change the file system.

Changing the partition layout in system profiles might render the profile unusable. It is recommended not to change the partition layout in system profiles, unless you know that the changes you want to make have no side effect.

In any case, do not:

- Transform a primary partition into a logical partition.

Note: Changing the partition layout from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose either one or the other entry point, and then perform all your changes from that entry point.

Editing the partition layout allows you to:

- Add or delete partitions.

  Note: Adding or deleting partitions can lead to OS configuration problems, therefore this feature must only be used very carefully. To provide a better description to your profile, use the **Comment** field to write all necessary details.
- Resize a partition by dragging sliders, or by assigning it an absolute or relative size.
- Change the file system of a partition.
- Assign a mount point to the partition.

1. Click **Edit partition layout** on either the **Profile details** page or the **OS configuration details** page, **Disks** tab.
2.
   - To add a partition:
     a. Click **Modify partition layout**.
     b. Click into an existing partition.
     c. Click **Add a partition** in the contextual menu.
     d. Indicate the partition properties, including a mount point and click **OK**.

        Windows In a Windows profile, if you do not provide a drive letter as mount point, one is assigned randomly. Letter *W* is not available.

        Linux In a Linux profile, do not forget to assign a mount point for the new partition. To be valid, this mount point must reference an existing directory in the main image.
   - To resize partitions with the sliders, grab the slider to the right of the partition and drag it.
   - To update all other parameters, select a partition by clicking on it, and select **Edit partition** in the contextual menu.

Modified partitions are aligned on megabytes rather than on cylinders. The following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

If you want to use the same system profile with two different partition schemes, you can also duplicate a system profile by right-clicking the profile name and selecting **Duplicate profile**. The copy shares the same image files, but can have a different partition layout.

Updating device mapping

Device mapping can be updated to force logical disks to point to specific physical devices.

Note: Updating device mapping from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose one or the other entry point, and then perform all your changes from that entry point.

1. Go to **Server** → **OS deployment** → **System profiles** → **Profile details** or to **Server** → **OS deployment** → **System profiles** → **Profile details** → **OS configuration details**, **Disks** tab.
2. Click **Modify device mapping**.
3. Select to which physical device you want to map your logical disk. The column starting with **Disk 0** corresponds to an automatic detection of the first to the eighth disk, the column starting with **/dev/hda** corresponds to standard device names.

   Note: Spanning a logical disk on several physical disks is not currently available.
4. Click **OK**.
5. Repeat step 2 to step 4 for each logical disk for which you want to update device mapping.

If the new device mapping you selected is incorrect, you receive a warning message.

OS configurations and fixed common parameters

A system profile is the partition layout and list of files to deploy, while OS configurations are operating system parameters.

At the very bottom of the **Profile details** page, there is a list of the OS configurations that correspond to your profile.

You can define several OS configurations for each system profile and duplicate them. These copies share the same image files, and the same partition layout, but can have different target parameters. You must then assign new values to some of the OS configurations parameters to make the original OS configuration and its copies distinct.

If you want to automate the assignment of parameters to targets, you can view and edit the OS configuration you are about to deploy by clicking on its name in the **Profile details** page. You are now in the **OS configuration details** page. The information is divided into panels, each displaying sets of parameters. You can modify the parameters either through the web interface or by using a parameter file.

Editing OS configuration parameters in the web interface for Linux

The web interface displays a number of OS configuration parameters divided into panes. These parameters can be edited in the web interface.

To edit parameters:

1. Click a tab to select the corresponding pane.

2. Click **Edit** on the banner of the section where you want to update parameters.
3. Modify the values.
4. Click **OK**.

Editing a Linux parameter file

Note: Since version 7.1.1 of the product, information about partitions in the custom configuration file is not normally taken into account.

SUSE For partitioning information in the custom configuration file to be taken into account, and to replace any information in the default file, these conditions must be fulfilled:

- The version of the product must be 7.1.1.3 or higher
- The deployment must be performed by HTTP
- The system profile must be of type *unattended setup*
- The operating system being deployed must be SuSE

Information in **Common networking info** is overwritten by the information in the custom configuration file. However, information in the **Advanced network settings** are not because they are applied in a post-configuration stage.

1.
   - Red Hat Click **Edit custom 'ks.cfg'** to edit the file.
   - SUSE Click **Edit custom 'autoinst.xml'** to edit the file.
2. Type the parameters and their values in the syntax requested by the operating system, or copy and paste it from another editor.
3. Click **OK**.

Tivoli Provisioning Manager for OS Deployment merges the information of the edited file with the information provided on the web interface (default file). The resulting configuration is the union of the values in the custom and default files, with the following restrictions:

- The result of conflicting values between the custom and default files is undefined.
- Partition information in the custom file is taken into account only for SuSE unattended setup by HTTP, in which case only the information in the custom file is taken into account.
- Advanced network settings are always applied, because they are performed at a later stage.

SUSE Here is an example of a `autoinst.xml` file which adds a new user during setup.

```
<profile xmlns="http://www.suse.com/1.0/yast2ns"
         xmlns:config="http://www.suse.com/1.0/configns">
 <users config:type="list">
   <user>
     <username>jdoe</username>
     <user_password>tOpsEcreT</user_password>
     <encrypted config:type="boolean">false</encrypted>
     <forename>John</forename>
     <surname>Doe</surname>
   </user>
 </users>
</profile>
```

Do not omit the `xmlns` and `xmlns:config` attributes of the `profile` tag.

Troubleshooting:

If the OS configurations in the deployed operating system are not what you expected, you must examine the parameter files carefully. They are the result of the merge between the custom file and the default file created.

Red Hat To troubleshoot OS configuration parameters after a failed deployment, there are two options:

- Without rebooting the target:
  1. Type `Alt+F2` on the target. This opens a shell.
  2. In the opened shell, view the file `/tmp/anaconda.log`.
- You must look for `ks.cfg` at the root of the partition labeled `rembo`. The file contains the information merged from the custom and the default files.

SUSE To troubleshoot OS configuration parameters after a failed deployment, there are two options:

- Without rebooting the target:
  1. Type `Alt+F2` on the target. This opens a shell.
  2. In the opened shell, view the file`/var/log/YaST2/y2log` .
- You must look for `autoinst.xml` at the root of the partition labelled `rembo`. The file contains the information merged from the custom and the default files.

Software modules for Linux operating systems

Software modules are images other than system profiles that can be created to address various needs.

Tivoli Provisioning Manager for OS Deployment is based on imaging technology. As administrator, you create images of components that you want to see on every target, and the automated deployment merges and restores these images on each target, automatically, when needed.

Tivoli Provisioning Manager for OS Deployment can handle most scenarios for software deployment and post-installation configuration.

Types of software modules

There are many types of software modules. Depending on the type of package and installation files, the wizard guides you through the different steps to achieve your software module with minimal effort. The types of software package supported by the wizard are listed in this section.

- **A Linux application installation, using RPM**
- **A custom action on the target computer**. This includes OS configuration changes such as commands to be run, and copying sets of files on the target.

Creating software modules

There are distinct types of software modules which vary according to the operating system being deployed. The software wizard guides you through the creation of software modules for each type.

Creating software modules with RPM for Linux operating systems

Using RPM is current for Linux software installation.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select **Linux** and click **Next**.
4. Select **A Linux application installation, using RPM** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the software module at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The path to where the installation files are restored on the target. This path is relative to the system root partition.
   - An additional command line that might be necessary to install your software module. When possible, the wizard automatically suggests the appropriate command line to run the installation unattended. However, you might need to add some additional parameters to the command.

     For example, you can specify an hour parameter to cancel an activity, if the activity does not complete before the end of the specified time. The parameter syntax format is `<=xh`, where `x` is an integer representing the number of hours after which the activity is canceled. In the following example you can specify to cancel an application installation if the installation process has not completed after one hour, by adding `<=1h` at the end of the command line:

     ```
     install.exe /sPB /rs /rps /l <=1h
     ```

   - The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

Creating a custom action software module for Linux operating systems

Software modules can also contain custom actions to be performed on the target.

They are divided into:

- An OS configuration change to perform on the target
- A set of files to copy on the target

Configuration changes are further subdivided. Depending on the operating system, you can:

- Copy a single text file
- Run a single command file, this can be a batch file or a vb script file.
- Boot a virtual floppy disk

In the OS configuration change wizard screen, you can select **Activate keyword substitutions**. If you use this option, you can specify which keywords must be substituted in the software module details.

1. Go to **Server → OS deployment → Software modules**.
2. Click **New software** to run the software wizard.
3. Select the operating system and click **Next**.
4. Select **A custom action on the target** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the software module at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The path to where the installation files are restored on the target. This path is relative to the system root partition.
   - An additional command line that might be necessary to install your software module. When possible, the wizard automatically suggests the appropriate command line to run the installation unattended. However, you might need to add some additional parameters to the command.

     For example, you can specify an hour parameter to cancel an activity, if the activity does not complete before the end of the specified time. The parameter syntax format is `<=xh`, where `x` is an integer representing the number of hours after which the activity is canceled. In the following example you can specify to cancel an application installation if the installation process has not completed after one hour, by adding `<=1h` at the end of the command line:

     ```
     install.exe /sPB /rs /rps /l <=1h
     ```

   - The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

Repeating custom actions:

Some commands must be run every time the target boots during a deployment.

This is typically the case if you want to repeatedly connect a network share. This connection is destroyed when rebooting. You can therefore create a single software module with a `netuse` command to set the network share and set this software module to run once after each reboot, starting at a specific reboot.

This option is available for executing a single command.

1. Create your software module.
2. Double-click on the software module name in the **Software components** page to obtain the **Software details** page
3. Click **Edit** in the title of the **Package information** section.
4. Select the installation stage at which the software module must be applied first.
5. Select **Run at each software pass until end of deployment** and click **OK**.

Importing drivers from the IBM Web site

You can maintain your system device drivers and firmware at the most current levels avoiding unnecessary outages by using a new agent command based on the IBM UpdateXpress System Pack Installer tool.

This command creates a batch file that launches the IBM UpdateXpress System Pack Installer tool. You can use the batch file to detect current device driver and firmware levels, remotely retrieve the device driver and firmware updates from the IBM Web site, automatically package the drivers needed, and bind them to specific hardware models.

1. Run the following command to create the batch file:

   ```
   rad-mkuxspbatch uxsp-path dest-path (BOM | model=<type1>[,<type2>,...]
   OS=<OS1>[,<OS2>...,])
   ```

   where:

   uxsp-path
   Specify the full path to the UpdateXpress setup utility.

   dest-path
   Specify the main path where all the updates are stored together with the UpdateXpress batch file.

   BOM Specify the OS deployment server database BOM table to be scanned for detecting the updates needed.

   model=
   Specify the models used for manual updates by giving a list of model types (such as `model=4190,7971`).

   OS= Specify the operating systems used for manual updates.

   OSx Specify the operating system types: `windows`, `rhel3`, `rhel4`, `rhel5`, `sles9`, `sles10`, `all`.

   A batch file `updatexpress.bat` is created in the directory `dest-path`. The generated batch file contains:

   - Commands to acquire system packs and available updates (in particular drivers) for every model and operating system combination.
   - Commands to extract every software module.

   If you specified the BOM option, the agent command automatically scans the OS deployment server database for existing models and operating systems. You can also manually specify the models and operating systems to preload software modules of systems that currently do not exist within Tivoli

Provisioning Manager for OS Deployment. You can run this batch file to import the drivers needed by using the IBM UpdateXpress System Pack Installer tool located in the directory `uxsp-path`.

2. You can then run this batch file: `dest-path\updatexpress.bat`, where `updatexpress.bat` is the name of the batch file containing the sequence of commands. It acquires and extracts the drivers needed by using the IBM UpdateXpress System Pack Installer tool located in the `uxsp-path` directory.
3. Use the Tivoli Provisioning Manager for OS Deployment web interface to create software modules of the drivers acquired and extracted in the previous step:
   a. Go to **Server → OS deployment → Software modules**.
   b. Click **New software** to run the software wizard.
   c. Select the relevant operating system and click **Next**.
   d. Select **A Windows driver to include in a deployment** and click **Next**.
   e. Select the computer and the main folder in which the driver files have been extracted (such as`..\IBM_Machine_type(7971)\OS_type(windows )`). The wizard lists all the drivers contained in this folder and its sub-folders.
   f. Select the drivers you need to package according to your hardware inventory and operating system.
   g. Specify the folder name where to store all the driver packages in a software tree structure.
   h. Select **Yes, create binding rules based on:** and then **PCI hardware ID** and **Target model name**. Click **Next**.
   i. Select **Use this driver for similar devices** and click **Next**.
   j. Specify the target machine model by selecting **the model name is one of the following** and then the model in the list.
   k. For the chosen drivers select the appropriate operating system architecture (such as 32–bit) and the targeted operating system (such as `Windows Server 2003` or `Windows 2008`) and click **Next**. The driver packages are created with the specified binding rules and grouped in the folder you specified. You can also modify the binding rules by editing the software module or you can create an additional software module with other drivers and add it to the same main folder.
   l. Click **Finish**.

Examples

Here is an example of generation of the `updatexpress.bat` file. It scans the BOM table to detect current Tivoli Provisioning Manager for OS Deployment device driver and firmware levels, remotely retrieves the device driver and firmware updates from the IBM Web site, and extract them into explicit model and operating system folders.

```
rad-mkuxspbatch d:\uxsp\uxspi300.exe d:\output BOM
```

Here is an example of generation of the `updatexpress.bat` file. It remotely retrieves the device drivers of model 4190 and 7971, for Windows, RedHat 3, and SLES 10 operating systems. It connects to the IBM Web site, to retrieve, package, and extract the drivers into explicit model and operating system folders.

```
rad-mkuxspbatch d:\uxsp\uxspi300.exe d:\output model=4190,7971
OS=windows,rhel3,sles10
```

Editing software modules

You can edit the basic parameters of a software module, upload new files into your software module, and update drivers.

1. Go to **Server** → **OS deployment** → **Software modules**. Double-click on a software module to view the details.
2. From **Software details** page, use the links and buttons. Links vary according to the type of software module. Not all the links listed are necessary available.
   - To edit the base parameters of a software module, click **Edit** at the top of the **Software module information** section.
   - To update files or add new files into the software module, click **Edit software module files**, or a link with a similar name, and select **Upload file** from the contextual menu.

     Note: File upload is limited to 16 MB.

Keeping command lines confidential

When you use command lines in your software modules, their call and their output are stored in deployment logs. In some circumstances, for example when the command line includes a password or a product key, it might be necessary to keep the information contained in the command line confidential. Three levels of confidentiality are available.

No confidentiality
The command line is visible in the web interface and on the target during the installation, its call is logged, and its output is also logged.

The command line call is not logged
The command line is visible in the web interface, and its output is logged, but the command line call, containing the whole command line string with all parameters, is visible in the logs neither on the web interface nor on the target.

To apply this level of confidentiality, you must prefix the command line by one exclamation mark (!).

The command line call and output are not logged
The command line is visible in the web interface, but its call and output are visible in the logs neither on the web interface nor on the target.

To apply this level of confidentiality, you must prefix the command line by two exclamation marks (!!).

To keep command lines confidential:

- Enter the appropriate number of exclamation points in front of the command in the Software Wizard when first creating the software module.
- Edit the software module information
  1. Go to **Server** → **OS deployment** → **Software modules**. Double-click on a software module to view the details.
  2. Click **Edit** in the Software module information banner.
  3. Update the command line with the appropriate number of exclamation points.
  4. Click **OK**.

Keyword substitution

You can usefully use keyword which act as variables and are substituted with their values during deployments. Keywords can either refer database values or server specific values, given by the user.

Syntax

Variable substitution expressions follow the syntax given here. They start with the character { and end on the same line with }. Words between these two characters are interpreted by using one of the following schemes:

- *{$expr$}* the expression is replaced with the string resulting of the evaluation of `expr`.
- *{/expr/ab}* the expression is replaced with the string resulting of the evaluation of `expr`, but each occurrence of the character `"a"` is replaced by the character `"b"` (character-based substitution).
- *{=expr=test content=this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is equal to the text `"test content"`.
- *{!expr!test content!this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is not equal to the text `"test content"`.

Note: If a variable does not exist (for example, it contains a typing error or it is not described in `server.ini`) but it is used in a command, its value is supposed to be empty which can result in deployment errors.

Database keywords

Within an expression, database records can be referred to. Within a record, each field can be accessed using the standard C notation (`record.fieldname`). The exhaustive list of these fields can be obtained from the database records, with the following correspondences between variable and database record names:

Table 4. Records for `free-text conditions`

Variable record name	Database record name
Disk	DiskInventory
DMI	DMIInventory
Order	BOM
User	UserProfile
System	SystemProfile
PCI	PCIInventory

Below are a few examples of available fields:

- `Order.IP`: a string, the target IP address, such as `192.168.1.2`
- `Order.MAC`: a string, the target MAC address, such as `00:01:02:03:04:05`
- `Order.SN`: a string, the target Serial Number, such as `CH12345678`
- `Order.Model`: a string, the computer model name, such as `e-Vectra`
- `User.UserCateg0`: a string, without any restriction, such as`technicians`
- `DMI.Vendor`: a string, the vendor name, such as `Hewlett-Packard`
- `DMI.Product`: a string, same as `Order.Model`

- DMI.ProcModel: a string, the processor model
- Disk[0].Type: a string, the disk 0 drive type, such as ATAPI
- Disk[0].Media: a string, the disk 0 media type, such as Disk or CD
- Disk[0].DiskSize: a number, the physical size of the disk (if detected)
- PCI[0].VendorID: a string, the hexadecimal vendor ID of the device
- PCI[0].DeviceID: a string, the hexadecimal device ID of the device

For disks and PCI devices, you can use the function sizeof (sizeof(Disk) and sizeof(PCI)) to discover the number of devices present. You can then use indexes to access these devices.

As an example for keyword substitution, if BomID has OrgName Rembo SaRL, RemboServer 192.168.168.16, and IP 192.168.168.32 for value 1, the following text

```
BomID:{$Order.BomID$}
OrgName:{$User.OrgName$}/{$StrToLower(User.OrgName)$}
RemboServer:{$Order.RemboServer$}
IP:{$Order.IP$}
```

gives the following results after keywords are substituted (note the use of a Rembo-C function within the expression to be substituted):

```
BomID:1
OrgName:Rembo SaRl/rembo sarl
RemboServer:192.168.168.16
IP:192.168.168.32
```

Server specific keywords

If you want to set up server specific keywords, which are defined exclusively by the user and per server, you must edit Files/global/rad/server.ini.

Start the file with [Custom] and add a line per keyword, in the format **keyword=value**, where keyword is a word of your choice and value the value you want to give it.

To use the keyword in a command, type Server.keyword and activate keyword substitution when creating the software module.

Note: server.ini is not replicated between servers. If you use multiple servers, you must edit server.ini on each server.

Customizing the software page

You can view the software modules in a tree viewer or in a list viewer. The list viewer allows you to customize the visible information.

You must have created at least one software module, otherwise there is nothing to view.

To customize the visible information

1. Go to **Server** → **OS deployment** → **Software modules**. Then click **List view**.
2. From the list view, you can
   - Drag the column separator in the column heading to resize the column.
   - Click on the triangular arrow to the left of the column name to sort the software modules by column criteria.

- Click on the arrow on the right of the column name and select an option to filter the information. Filtering on several columns is cumulative.

3. For more options, right click anywhere to open the contextual menu and select **Arrange columns**.
   - Select the columns you want to see and clear the others.
   - Click on the minus or plus icons to decrease or increase the size of a column.
   - Select a column and use the up and down arrows to move the column relatively to the others.

   Click **OK** to save your changes. The updated version of the list view is visible in the **Software modules** page.

To return to the tree view, click **Tree view**. You can also access the details of the software modules by double-clicking on a software module name, from either view.

OS configuration and software bindings

OS configuration bindings determine which configurations are available to a target when booting the target on the network, while software bindings correspond to the list of software modules currently assigned to the target.

OS configuration and software bindings are created when:

- The Target Monitor has been used to manually modify OS configuration and software bindings for the target
- A deployment has been started with the Target Monitor. In this case, an OS configuration binding is added for the corresponding OS configuration.
- Automatic binding rules are configured in the **Details** page of OS configurations or software modules. Some of these rules have matching values for the specified criteria. These bindings cannot be modified, except by modifying the rules.

With the Target Monitor, you can browse, remove or add OS configuration and software bindings to any target present in the database. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.

Binding software modules and OS configurations to targets

Bindings link software modules and OS configurations to targets to enable automatic deployment. When binding to targets, you explicitly provide the list of software modules and OS configurations to bind to your target.

To explicitly bind a software module or a OS configurations to a target, there are two methods:

- From the **Target Monitor** page
- From the **Target details** page

If you want to bind software modules or OS configurations to a group of targets, you must do it through the Target Monitor.

From the Target Monitor:

1. Select a target or a group of targets
2. Select **Bind software** or **Bind OS configurations** from the contextual menu
3. Select the items to bind from the popup window
4. Click **OK**

From the Target details page:

1. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.
2. Go to the **Bindings** panel.
3. Click **Edit** in the relevant section to add explicit bindings for OS configurations and software modules.
4. Select the items for which you want to add explicit bindings.
5. Click **OK**

You can also clear items to remove their explicit bindings. To remove a binding by rule, you must modify the rule.

Binding software modules to computer models and PCI devices

You can link software modules to target computer hardware model to enable automatic deployment. When creating a binding, you explicitly provide the list of software modules to link to a computer model and its PCI devices.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Select **View bindings** from the contextual menu. A grid is displayed representing computer model-based software bindings:
   - The rows correspond to your software modules
   - The columns correspond to the computer model. If you expand it, you can see all the corresponding PCI devices discovered in your hardware inventory.

   The background color indicates implicit bindings by existing generic rules, while checkmarks correspond to explicit one to one bindings.
3. Select a grid position corresponding to the software module and device to bind.
4. Select **Bind** from the contextual menu.

You can also remove a binding by selecting it and clicking **Unbind** from the contextual menu.

Binding software modules to a deployment scheme

Software modules can be bound to deployment schemes.

Take a company with offices in three locations: New York, Quebec City, and Mexico City. In each of these locations, the company has people in human resources, sales, logistics, and product development. For the sake of simplicity, consider further that all the employees use either one of two types of computers: a desktop, or a notebook. All desktop computers are identical (with the same network card, system board, disks, and so on) and the same applies for all notebooks.

In this scenario, the company needs two profiles, one with the image for notebooks and one with the image for desktop computers. Three configurations per profile (six in total) are necessary to integrate the different parameters of the different locations, in particular language and time zone information. Finally, schemes are set according to the employees' department, with software modules specific to the different departments bound directly to the deployment schemes.

1. Go to **Server** → **OS deployment** → **Task templates** Select the **Deployment Schemes** folder. Double-click on a deployment scheme to view its details.
2. Click **Edit** on the **Software bindings** section of the page to open the dialog to bind software modules to schemes.

3. Select which software modules you want to bind to your deployment scheme, in addition to software modules that can have been bound to targets.
4. *(Optional)* If you want to use only the software checked in the window when deploying with this scheme, select the **Discard all other software binding rules** check box.

Automatic binding rules

Automatic binding rules are used to create bindings between OS configurations and targets, or software modules and targets, without having to specifically bind a OS configuration or a software module on each target.

Rules are created in OS configurations and software modules to determine which targets are automatically bound to the OS configuration or software module.

Rules are made of criteria and values. If a target has a matching value for all criteria in the rule, the OS configuration or software module will be bound to that target. The binding will be displayed with the mention **by rule** in the OS configuration panel of the target properties for targets that match the criteria. For example, if the criteria is the model name, and the value is `Optiplex`, targets with a model name starting with `Optiplex` will be bound to the object where the rule has been defined.

Automatic binding rules are defined in Tivoli Provisioning Manager for OS deployment at the bottom of the **OS configuration details** or **Software details** page.

To create a new binding rule, click **New rule** located at the bottom of the Web interface:

1. The dialog displayed to create a new binding rule is different depending on whether you are adding a rule to an OS configuration or to a software module. When adding a binding rule to a software module, you can set values for the following criteria:
   - A deployment scheme
   - A system profile
   - A current OS configuration
   - Administrative group
   - One of the system-definable and user-definable fields of the database (only used if you have customized the database)
   - An operating system type, such as Windows 2000
   - An operating system version, such as SP2
   - An operating system language
   - An operating system architecture, such as x86-32
   - A computer model name
   - A BIOS version
   - A PCI device
   - A base board
   - MultiChassi
   - HAL Type
   - A free-text condition in `Rembo-C;` syntax

For example, to create a binding based on the operating system type between a software module and targets, you must create a new rule, click **OS type**, and select the operating system version that you want to limit this software module to.

2. When adding a binding rule to an OS configuration, you can set a condition on the deployment scheme, and on the computer model name. The next ten fields are only used if you have customized your database and want to match specific user categories.
3. Finally, you can enter a free-text condition following the `Rembo-C;` syntax. They must only be used by advanced users.

   The conditions determine the applicability of the rule and evaluate to `true` or `false`. A condition must be formed using the variables also used for keyword substitutions in software modules, combined with Java-like logical operators, listed by order of priority in the table:

Table 5. Logical operators for free-text conditions

Operator	Meaning
<	smaller than
<=	smaller than or equal to
=>	greater than or equal to
>	greater than
==	equal to
!=	not equal to
&&	AND operator
||	OR operator

For example, a typical condition can be:

```
Disk[0].DiskSize > 10*1024*1024
```

Note: If a condition cannot be evaluated, it is considered to have the value `false`.

Scheduling the application of software modules for Linux operating systems

Tivoli Provisioning Manager for OS Deployment provides a wide flexibility in the specification of a deployment task. As several software modules can be deployed in conjunction with a system profile, you can schedule when they must be applied.

Tivoli Provisioning Manager for OS Deployment provides a wide flexibility in the specification of a deployment task. As several software modules can be deployed in conjunction with a system profile, you can schedule when they must be applied.

Typical application locations for software modules include:

- For virtual floppy-disk used as ramdisk: before disk partitioning and OS installation, to allow for the configuration of low-level hardware devices controlling the hard disk, such as RAID controllers
- For virtual floppy-disk images: in between disk partitioning and OS installation, to flash devices early
- Sysprep and unattended setup processes are automatically run during the OS installation phase, if required

- For system snapshots: right after OS installation, to deploy the software nearly at the same time as the operating system image (the most efficient). Before OS installation is forbidden, as a system snapshot needs an installed OS
- For other software: when the OS is installed or after additional reboots depending on the software module needs

Software modules are not ordered within an installation stage. If you want a software module to be installed before another between two specific reboots, create two distinct installation stages between the reboots. For example, if your first software module copies files on the target and the second one runs a command on these files, you must place the first software module in an installation stage which occurs before the one in which you run the command software module.

1. To schedule the application of software modules, go to **Server → OS deployment → Software modules**. This opens a dialog window that allows you to order the different software modules stored on your OS deployment server. The dialog shows the different steps of a deployment with disk partitioning (in green), OS installation (in purple) and reboots (in red). Software components can be installed in between all of these steps, where they are placed inside the expandable installation stages (in yellow).
2. You can add, move, and delete reboot sequences by using the buttons at the bottom of the dialog window. You can also rename software installation stages.
3. You can expand the software installation stages to view their content by clicking on the **+** icon. You can then move individual software modules from one stage to another by drag-and-drop. The destination stage does not need to be expanded.

   Note: Drag-and-drop is limited to the **Software Application Order** window. You cannot drag-and-drop an item from the Software Module page.

When creating a recovery CD or exporting a RAD file, the software application order is automatically included.

Working with hardware configurations

It is sometimes necessary to run configuration tasks on the targets before installing the operating system, for example to update the firmware or to configure RAID volumes.

To automate this kind of operation with the product, you must perform a *hardware configuration task*, which uses a *hardware configuration object* stored on the OS deployment server. To create a hardware configuration object, you must have already created a *hardware environment*. This hardware environment contains WinPE or DOS files, updated with drivers specific to given hardware models and vendor-specific tools to perform hardware configuration tasks.

The hardware configuration tasks that you can perform with the product are

- RAID configuration
- BIOS update
- BIOS settings
- Hardware custom configuration, that is, any kind of tool that you can load into the environment and run from a command line.

You can also perform an inventory of RAID or Fiber Channel hardware.

Hardware configuration tasks are available only for targets with an x86 or an x86-64 architecture.

Example

To configure hardware with the product, for example a BIOS update with WinPE2 on an IBM target, you need to follow a number of steps.

1. Create a hardware environment with drivers and tools:
   a. Download Windows Automated Installation Kit (WAIK) from Microsoft and install it to have the WinPE2 files available.
   b. Download the latest ServerGuide scripting toolkit from IBM and extract it, for example, in directory `C:\IBM-SGTSK-WinPE2.x`.
   c. Run the SGTKWinPE.cmd command to prepare the WinPE2 environment with the needed IBM drivers. It creates the `.\sgdeploy\WinPE_ScenariosOutput\Local\RAID_Config_Only\ISO` directory, which contains both the WinPE2 binaries and the vendor-specific tools.
   d. Create a hardware environment with the hardware environment wizard.
2. Create a hardware configuration object with the hardware configuration wizard:
   a. Select **BIOS update** as the type of hardware configuration to be performed.
   b. Associate the hardware environment of step 1 and your hardware model to the new hardware configuration object you are creating.
   c. Indicate the location of the BIOS update material, that is, a set of files containing in particular `wflash.exe`.
3. Perform the actual configuration task by deploying the hardware configuration object of step 2 on your target:
   a. Select a target (or several) in the Target Monitor.
   b. Select **Deploy now** in the contextual menu.
   c. Select **Perform hardware configuration tasks** and optionally other deployment tasks in the deployment wizard.
   d. Select the hardware configuration object that you want to apply and follow the remaining instructions of the wizard.

The hardware environment now runs as a ramdisk on the target, and, using vendor-specific tools, the BIOS is updated.

Setting up your environment

To perform hardware configuration tasks, you must set up a hardware-specific environment containing the vendor-specific scripting toolkit tools and the necessary drivers to run correctly (for example, network connectivity) on the target.

The hardware environment supported are those running scripts and tools in:

- WinPE 2.x
- WinPE 1.x
- DOS

Every environment is very specific to its vendor, and must be prepared with the suitable drivers and scripting toolkit tools.

WinPE2, WinPE1, and DOS cannot perform hardware configuration tasks (for example, RAID configuration or BIOS setting) by themselves. They must contain

drivers to access the hardware and tools to perform the configurations. These drivers and tools are vendor-specific and vary for each type of target model. When you create an environment with the OS deployment server, you associate either WinPE2, WinPE1, or DOS, to vendor-specific drivers and tools. You can then associate the resulting environment to a specific set of target models and a type of hardware configuration tasks to create a hardware configuration object.

Because a hardware environment is run as a ramdisk, it does not leave any trace on the target after the hardware configuration task is performed.

Hardware configuration objects and tasks

A hardware configuration object is the association, on an OS deployment server, of a vendor-dependent environment, target models, a type of hardware configuration to be performed, and possibly some other commands. A hardware configuration task is performed at deployment time by loading and running the associated hardware configuration object containing a vendor-dependent environment on the target, before installing the operating system.

Hardware configurations tasks do not impact the following operating system deployment because Tivoli Provisioning Manager for OS Deployment configures the hardware through actions run in a ramdisk before the deployment of the operating system.

The execution flow is similar, regardless of the environment to run, or the type of hardware environment task:

1. The environment is loaded in memory, as a ramdisk
2. Any additional binary or configuration files are added to the ramdisk, based on the selection made in the web interface when creating the hardware configuration object
3. The computer boots the ramdisk
4. The hardware configuration task is run
5. The computer reboots
6. Tivoli Provisioning Manager for OS Deployment resumes the deployment sequence if any was selected, but a hardware configuration object can be run also as an independent task

The following types of hardware configuration objects are available:

RAID configuration
The hardware configuration wizard allows you to create a hardware configuration object to configure RAID adapters in a vendor-independent way. Tivoli Provisioning Manager for OS Deployment builds the vendor-specific configuration file.

BIOS update
The hardware configuration wizard allows you to create a hardware configuration object to update the BIOS firmware on the target.

BIOS settings
The hardware configuration wizard allows you to create a hardware configuration object to update the BIOS or BMC (baseboard management controller) settings through an initialization file.

Hardware custom configuration
The hardware configuration wizard allows you to create a hardware configuration object to perform any kind of hardware configuration. Any

tool used for preparing the environment can be packaged in a custom hardware configuration object, injected into the ramdisk and run using command lines.

Capture hardware parameters
This option is available only if you do not already have a hardware capture configuration object.

The hardware configuration wizard allows you to create a hardware configuration object to capture RAID and Fiber Channel information from a target.

RAID and Fiber Channel hardware capture

Capturing RAID and Fiber Channel information requires the use of a vendor-specific environment.

Target inventory for CPU, memory, logical disks, PCI devices, motherboard, and so on, is managed by the OS deployment engine and all information is available immediately if requested. To complete the hardware target inventory with RAID and Fibre Channel information you need the vendor-specific scripting toolkit tools. The hardware capture is done in a similar way to that of the hardware configurations, which means that you need to load the vendor-dependent environment on the target to start the specific capture tool.

The captured hardware information for Fibre Channel and RAID disks can then be seen from the web interface:

Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.

Creating a hardware environment

Having a hardware environment on your OS deployment server is a prerequisite to create a hardware configuration object, with which you can perform hardware configuration tasks on targets.

Before you can create your environment with the wizard, you must prepare the files on the OS deployment server.

Instructions are provided for preparing the files using scripting toolkits for IBM, Dell, or HP products. It is recommended that you download the latest WinPE 3.x compatible scripting tool environments and use this version. However, the instructions for, WinPE 2.x, WinPE 1.x and DOS are also provided.

IBM **IBM ServerGuide Scripting Toolkit WinPE 3.x based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site. The name of the downloaded file is similar to `ibm_utl_sgtkwin_2.30_windows_32-64.zip`.
2. Extract the toolkit into a local directory, for example, into `c:\IBM-SGSTK-WinPE3.x`
3. As described in the User's Guide in `c:\IBM-SGSTK-WinPE3.x/sgdeploy/SGTKWinPE/Docs/UserGuide.pdf`, you must then do the following:
   a. Download Windows Automated Installation Kit (AIK) for Windows 7 in EnglishWindows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication:

http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.

b. Install Windows AIK.

c. Restart your computer.

d. Expand files `ibm_utl_tsep_2.00_winpe_i386.zip` and `ibm_utl_tsep_2.00_winpe_x86-64.zip` located in `.\sgdeploy\updates\uxsp` into the directory in which the toolkit was extracted, for example `c:\IBM-SGSTK-WinPE3.x`

e. Run `InstallSEPs.cmd` to install the System Enablement Pack.

f. Run `SGTKWinPE.cmd` to create a WinPE image with the requested drivers for IBM servers. Use the option `/Image` to exclude ISO and provide `ScenarioINIs\Local\Raid_Config_Only_x86.ini` if you use a 32-bit WinPE, or `ScenarioINIs\Local\Raid_Config_Only_x64.ini` if you use a 64-bit WinPE, as properties file to include all RAID and Fibre tools and to exclude all network tools. The command finds where the Windows AIK is located by itself.

```
SGTKWinPE.cmd /Image ScenarioINIs\Local\Raid_Config_Only_x86.ini
```

A directory `.\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO` is created and contains the environment tools.

IBM **IBM ServerGuide Scripting Toolkit WinPE 2.x based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site. The name of the downloaded file is similar to `ibm_sw_sgtkw_2_1_windows_i386.zip`.
2. Extract the toolkit into a local directory, for example, into `c:\IBM-SGSTK-WinPE2.x`
3. As described in the User's Guide in `c:\IBM-SGSTK-WinPE2.x/sgdeploy/SGTKWinPE/Docs/UserGuide.pdf`, you must then do the following:

   a. Download the Windows Automated Installation Kit (AIK) 1.1 32-bit in English for Windows Vista SP1 and Windows Server 2008. Windows Automated Installation Kit (AIK) 1.1 is distributed by Microsoft and is available on the Microsoft Web site from the following link: Windows Automated Installation Kit (AIK).

   b. Install Windows AIK.

   c. Restart your computer.

   d. Expand files `ibm_utl_sep_1.00_winpe_i386.zip` and `ibm_utl_sep_1.00_winpe_x86-64.zip` located in `.\sgdeploy\updates\uxsp` into the directory in which the toolkit was extracted, for example `c:\IBM-SGSTK-WinPE2.x`

   e. Run `InstallSEPs.cmd` to install the System Enablement Pack.

   f. Run `SGTKWinPE.cmd` to create a WinPE image with the requested drivers for IBM servers. Use the option `/Image` to exclude ISO and provide `ScenarioINIs\Local\Raid_Config_Only_x86.ini` if you use a 32-bit WinPE2, or `ScenarioINIs\Local\Raid_Config_Only_x64.ini` if you use a 64-bit WinPE2, as properties file to include all RAID and Fibre tools and to exclude all network tools. The command finds where the Windows AIK is located by itself.

```
SGTKWinPE.cmd /Image ScenarioINIs\Local\Raid_Config_Only_x86.ini
```

A directory `.\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO` is created and contains the environment tools.

IBM **IBM ServerGuide Scripting Toolkit WinPE 1.x based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site.
2. Extract the toolkit into a local directory, for example, `c:\IBM-SGSTK-WinPE1.x`.
3. As described in the User's Guide in `c:\IBM-SGSTK-WinPE1.x\sgdeploy\SGTKWinPE\Docs\UserGuide.pdf` you must then complete the following steps:
   a. Download WinPE 2005.
   b. Run `SGTKWinPE.cmd` to create a WinPE image with the requested drivers for IBM servers.

IBM **IBM ServerGuide Scripting Toolkit DOS based**

1. Download the latest ServerGuide scripting toolkit from the IBM Web site
2. Extract the toolkit into a local directory, for example, `c:\IBM-SGSTK-DOS`.

Note: DOS tools are deprecated. They are used only to support some older hardware.

Dell **Dell DTK Scripting Toolkit WinPE 3.x based**

To set up the WinPE 3.0 environment for your Dell servers:

1. Download Windows Automated Installation Kit (AIK) for Windows 7 in English. Windows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest DTK scripting toolkit from the Dell Web site. The name of the downloaded file is similar to `DTK3.2.1-WINPE-22.exe`.
5. Extract the download file. For example, extract the file to the location `c:\ Dell-DTK-3.2.1`.
6. As described in the Dell User's Guide, in `C:\Dell-DTK-3.2.1\Dell\Docs\DTKUG.pdf`, you must then complete the following tasks:
   a. Open a command prompt in the directory containing the driver installation batch for `WinPE3.x: WINPE3.0_driverinst.bat`. For example, the directory, `C:\ Dell-DTK-3.2.1\Dell\x32\Drivers\winpe3.x`.
   b. Launch the file called `WINPE3.0_driverinst.bat` *<WINPEPATH> <DTKPATH>*, where *<WINPEPATH>* is the destination path to create the directory structure for WinPE 3.0 and *<DTKPATH>* is the path to the Dell drivers in the extracted DTK toolkit. For example, the file might be called `WINPE3.0_driverinst.bat C:\Dell-DTK-3.2.1\WinPE3.x_Out C:\Dell-DTK-3.2.1\Dell\x32\drivers`. Launching this file preinstalls the Dell drivers into `winpe.wim`.

7. Copy and rename the customized C:\Dell-DTK-3.2.1\WinPE3.x_Out\winpe.wim to C:\Dell-DTK-3.2.1\WinPE3.x_Out\ISO\sources\boot.wim.

Dell **Dell DTK Scripting Toolkit WinPE 2.x based**

To set up the WinPE2 environment for your Dell servers:

1. Download the Windows Automated Installation Kit (AIK) 1.1 32-bit in English for Windows Vista SP1 and Windows Server 2008.Windows Automated Installation Kit (AIK) 1.1 is distributed by Microsoft and is available on the Microsoft Web site from the following link: Windows Automated Installation Kit (AIK).
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest DTK scripting toolkit from the Dell Web site. The name of the downloaded file is similar to DTK2.6-WINPE-56.exe.
5. Extract the download file. For example, extract the file to the location c:\ Dell-DTK-2.6 5.
6. As described in the Dell User's Guide, in C:\Dell-DTK-2.6\Dell\Toolkit\Docs\DTK25UG.pdf, you must then complete the following tasks:
   a. Open a command prompt in the directory containing the driver installation batch for WinPE2.x: VPE_driverinst.bat. For example, the directory, C:\ Dell-DTK-2.6\Dell\Drivers\winpe2.x.
   b. Launch the file called VPE_driverinst.bat *<WINPEPATH>* *<DTKPATH>*, where *<WINPEPATH>* is the destination path to create the directory structure for Windows PE 2.0 and *<DTKPATH>* is the path to the Dell drivers in the extracted DTK toolkit. For example, the file might be called VPE_driverinst.bat C:\Dell-DTK-2.6\WinPE2.x_Out C:\Dell-DTK-2.6\Dell\drivers). Launching this file preinstalls the Dell drivers into winpe.wim.
7. Copy and rename the customized C:\Dell-DTK-2.6\WinPE2.x_out\winpe.wim to C:\Dell-DTK-2.6\WinPE2.x_Out\ISO\sources\boot.wim.

Dell **DELL Scripting Toolkit WinPE 1.x based**

Note: Windows PE 2005 must be built from a Windows 2003 server for the Dell tools to work.

To set up the WinPE1 environment for your Dell servers:

1. Obtain a Windows PE 2005 file structure.
2. Copy it into a temporary folder, for example, c:\winpe-dell
3. The Windows PE 2005 directory structure should contain a directory named I386 or MININT. If it contains a directory named MININT, rename it to I386.
4. Download the Deployment Toolkit from Dell.
5. Run the executable package to extract the toolkit to the disk of the OS deployment server. In the examples, it is assumed that you have extracted the toolkit into c:\DELL-DTK, which implies that you have a folder named C:\DELL-DTK\Dell\Toolkit.
6. To install the appropriate drivers for Dell servers in your WinPE image, follow the instructions of the DTK User Guide (*Running Deployment Scripts Using DTK and Windows PE*).

   In particular, you must:

   a. Install the drivers with the driverinst.bat script

b. Modify `winpeoem.sif` and `winbom.ini`

c. Add the RPC DLLs to the Windows PE directory.

Note: Add the RPC DLLs in `i386\system32` instead of those in the Tools folder.

7. To verify that the drivers have been installed, check for the existence of the file called `c:\temp\winpedell\i386\system32\racsvc.exe`.

HP **HP SmartStart Scripting Toolkit WinPE 3.x based**

To set up the WinPE 3.0environment for your HP servers:

1. Download Windows Automated Installation Kit (AIK) for Windows 7 in EnglishWindows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest SmartStart Scripting Toolkit from the HP Web site: http://h18013.www1.hp.com/products/servers/management/toolkit/. The name of the downloaded file is similar to `SP47335.EXE`.
5. Extract the file into a directory, for example, `C:\HP-TK`.
6. As described in the *HP SmartStart Scripting Toolkit Windows Edition User Guide.pdf* in `C:\HP-TK\SWSetup\SP47335\` and the *Windows Preinstallation Environment User's Guide* (WinPE.chm) contained in Windows AIK, you must then mount the WinPE3.x base image for specific customization. For example, activate extra packages, add drivers, and so on.

   a. From the Windows AIK tools folder, run the command to create WinPE customization directory.

   ```
   C:\Program Files\Windows AIK\Tools\PETools>copype.cmd x86
   C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP
   ```

   b. Mount the base image launching Dism from the WinPE3.x_HP folder.

   ```
   Dism /Mount-Wim /WimFile:.\winpe.wim /index:1 /MountDir:.\mount
   ```

   c. Install the *neutral* WMI packages in the image.

   ```
   Dism /image:.\mount /Add-Package
   /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86
   \WinPE_FPs\winpe-wmi.cab"
   ```

   Enter the command on one line, although it does not fit on this example.

   d. Install also the language specific WMI package in the image.

   ```
   Dism /image:.\mount /Add-Package
   /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86
   \WinPE_FPs\en-us\winpe-wmi_en-us.cab"
   ```

   Enter the command on one line, although it does not fit on this example.

   e. Add the required drivers (.inf files) to the base image by using the `/Add-Driver` option of the `Dism` command.

   ```
   Dism /image:<mounted image> /Add-Driver /Driver:<driverpath>
   /Recurse
   ```

Where <*driverpath*> is the location of the .inf files found in the extracted drivers within the hpDrivers folder and /Recurse is an option to query all the drivers in subfolders.

```
Dism /image:.\mount /Add-Driver
/Driver:C:\HP-TK\SWSetup\SP47335\hpDrivers\Winpe30 /Recurse
```

Enter the command on one line, although it does not fit on this example.

f. Copy the hpsstkio.sys Toolkit I/O driver (required for the conrep and rbsureset utilities) from the HP driver directory to the Windows driver directory. For example:

```
copy C:\HP-TK\SWSetup\SP47335\hpDrivers\Winpe30\system\hpsstkio
\hpsstkio.sys C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\mount\Windows
\System32\drivers
```

Enter the command on one line, although it does not fit on this example.

g. Unmount the customized image to build the customized WinPE.wim:

```
Dism /Unmount-Wim /MountDir:.\mount /Commit
```

7. Copy and rename the customized C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\WinPE.wim file into C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\ISO\sources\boot.wim.

HP **HP SmartStart Scripting Toolkit WinPE 2.x based**

To set up the WinPE2 environment for your HP servers:

1. Download the Windows Automated Installation Kit (AIK) 1.1 32-bit in English for Windows Vista SP1 and Windows Server 2008.Windows Automated Installation Kit (AIK) 1.1 is distributed by Microsoft and is available on the Microsoft Web site from the following link: Windows Automated Installation Kit (AIK).
2. Install Windows AIK.
3. Restart your computer.
4. Download the latest SmartStart Scripting Toolkit from the HP Web site: http://h18013.www1.hp.com/products/servers/management/toolkit/. The name of the downloaded file is similar to SP38836.EXE.
5. Extract the file into a directory, for example, C:\HP-TK.
6. As described in the *HP SmartStart Scripting Toolkit Windows Edition User Guide.pdf* in C:\HP-TK\SWSetup\SP38836\ and the *Windows Preinstallation Environment User's Guide* (WinPE.chm) contained in Windows AIK, you must then mount the WinPE2.x base image for specific customization. For example, activate extra packages, add drivers, and so on.
   a. From the Windows AIK tools folder, run the command to create Windows PE customization directory. For example: C:\Program Files\Windows AIK\Tools\PETools>copype.cmd x86 C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP)
   b. Mount the base image launching imagex from the WinPE2.x_HP folder. For example, imagex /mountrw WinPE.wim 1 .\mount.
   c. Install the WMI packages in the image: peimg /image=.\mount /install=*WMI*
   d. Add the required drivers (.inf files) to the base image by using the peimg /inf command.

```
peimg /inf=<driverpath> .\mount
```

Where <*driverpath*> is the location of the .inf files found in the extracted drivers within the hpDrivers folder. For example, `peimg /inf=c:\HP-TK\SWSetup\SP38836\hpDrivers\Extr-Drivers\nic\b06nd .\mount`.

e. Repeat step d. for each additional device driver.

f. Copy the `hpsstkio.sys` Toolkit I/O driver (required for the `conrep` and `rbsureset` utilities) from the HP driver directory to the Windows driver directory. For example:

```
copy C:\HP-TK\SWSetup\SP38836\hpDrivers\system\hpsstkio\hpsstkio.sys
C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP\mount\Windows\System32\drivers
```

g. When you finish customizing the image, prepare the environment image by using the `peimg /prep` command:

```
peimg /image=.\mount /prep
```

h. Unmount the customized image to build the customized WinPE.wim:

```
imagex /unmount /commit .\mount
```

7. Copy and rename the customized `C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP\WinPE.wim` file into `C:\HP-TK\SWSetup\SP38836\WinPE2.x_HP\ISO\sources\boot.wim`.

HP **HP SmartStart Scripting Toolkit WinPE 1.x based**

The initial setup for the HP SmartStart Scripting Toolkit is very similar to the setup of the Dell Hardware Toolkit, because both Toolkits require Windows PE. Some details are therefore skipped, but you can read them in the Dell section.

1. Download the Win32 HP SmartStart Scripting Toolkit version of the toolkit on the HP site.
2. Extract it to the disk of the OS deployment server (for example, in `c:\HP-TK`).
3. Create a Windows PE 2005 folder for the HP tools:
   a. Copy a Windows PE file structure to a temporary folder (`c:\winpe_hp`)
   b. Install the HP drivers in the Windows PE directory, as explained in the User Guide for the HP Hardware Toolkit
      1) Run the executable file under hpDrivers
      2) Give the location of the i386 folder of your Windows PE folder

To create your environment, with the wizard:

1. Go to **Server** → **Advanced features** → **Hardware configurations**.
2. Click **New environment** and follow the wizard instructions. You must
   a. Ensure that the web interface extension is running on the computer where Windows AIK and the environment tools have been prepared.
   b. Provide the path of the folder in which the environment tools are located, that is where you have installed the scripting toolkit. For example:

      IBM `C:\IBM-SGSTK-WinPE3.x\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO`

      Dell `C:\Dell-DTK-3.2.1\Dell\x32`

      HP `C:\HP-TK\SWSetup\SP47335`

   c. Provide the path of the folder in which the environment material is located, that is the WinPE files. For example:

IBM C:\IBM-SGSTK-WinPE3.x\sgdeploy\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\ISO

Dell C:\Dell-DTK-3.2.1\WinPE3.x_Out\ISO

HP C:\HP-TK\SWSetup\SP47335\WinPE3.x_HP\ISO

You can view the created environment by performing the following:go to **Server** → **Advanced features** → **Hardware configurations**. Alternatively, you can also view it by performing the following:go to **Server** → **OS deployment** → **Software modules**. To view it look under a specific environment folder.

Now, you can create hardware configurations using this environment.

Creating a hardware configuration object

A wizard allows you to easily create hardware configuration objects.

Before you can create a hardware configuration object, you must have created the environments needed to later perform the hardware configuration tasks.

1. Go to **Server** → **Advanced features** → **Hardware configurations**.
2. Click **New hardware config.**.
3. Select the kind of hardware configuration that you want to create.
4. Provide at least one target model and environment pair on which the hardware configuration can apply.
5. For BIOS update, BIOS settings, or Hardware custom configuration the specific files or set of files can be downloaded from the specific vendor sites.
6. Follow the wizard instructions.

To view or edit a hardware configuration, select the hardware configuration and select **View configuration details** in the contextual menu. In the **Hardware configuration details**, use the **Edit** buttons to update the different sections.

Creating a hardware capture configuration

A wizard allows you to easily create hardware capture configuration in a way similar to that for hardware configurations.

Before you can create a hardware capture configuration, you must have created the environments needed to later run the hardware capture.

- If you do not yet have a hardware capture configuration, perform the following steps:
  1. Go to **Server** → **Advanced features** → **Hardware configurations**.
  2. Click **New hardware config.**
  3. Select **Hardware discovery**.
  4. Provide at least one target model and environment pair on which the hardware capture can apply.
  5. Follow the instructions of the wizard.
- If you already have a hardware capture configuration, you can add target model and environment pairs, as follows:
  1. go to **Server** → **Advanced features** → **Hardware configurations**.
  2. Select **Hardware discovery**.
  3. Double-click **Hardware capture configuration**.

4. Under **Hardware environment matching**, click **Edit**.
5. Click **Add a new line** and select the model and environment values
6. Repeat step 5 for each pair to be added.
7. Click **OK**.
8. Click **Back** to return to **Server** → **OS deployment** → **Hardware configurations**.

To view or edit the hardware capture configuration, go to **Server** → **Advanced features** → **Hardware configurations**. Select **Hardware discovery**, and double-click the hardware capture configuration. In the **Hardware configuration details** page, click **Edit** to update the different sections.

You can now capture RAID or Fiber Channel information.

Capturing hardware information using templates

When you capture hardware information with templates, this capture is done every time the template is used.

Capturing hardware information with templates requires an additional reboot to boot the specific hardware configuration environment (WinPE, DOS,...) and launch the specific scripting toolkit tools.

Note: You can not capture hardware information from a target started with a network boot media.

Capturing hardware information with templates always tries to capture both RAID and Fiber Channel. To run the capture:

1. Go to **Server** → **OS deployment** → **Task templates**.
2. Select **Idle Layout** or **Deployment Schemes**, depending on which state you want to perform the hardware capture. If you select **Deployment Schemes**, the discovery is performed at deployment time.
3. Double-click the chosen template to view its details.
4. Click **Edit** on **General settings**.
5. Under **Perform inventory on:**, select **RAID**.
6. Click **OK**.

Capturing hardware information once

When you want to capture hardware information only once for a target, or a group of targets, you do this with a specific tool.

Capturing hardware information requires an additional reboot to boot the specific hardware configuration environment (WinPE, DOS,...) and launch the specific scripting toolkit tools.

Note: You can not capture hardware information from a target started with a network boot media.

1. Go to **Server** → **OS deployment** → **Target Monitor**.
2. Select a target or a group of targets.
3. Select **Additional features** from the contextual menu.
4. Double-click the chosen template to view its details.
5. Select **Capture hardware parameters** and click **Next**.
6. Select **Raid capture**, **Fiber channel capture**, or both, and click **Next**.

7. Follow the instructions of the wizard.

When captured, the RAID and Fiber channel information can be viewed. Go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. On this page look under the **Inventory** tab.

Task templates

Task templates group together elements that can be customized on a target. These elements are mostly screen layouts, which condition the appearance of the target screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment.

A deployment scheme is a specific type of task template. Together with the target display screen layout, it contains other parameters for customizing a deployment on a target.

Note: Starting with Fix Pack, version 5.1.0.2, deployment schemes are considered to be a subset of task templates. The functions of deployment schemes have not been altered. To access deployment schemes, go to the **Task template** page, and select the **deployment scheme** folder.

The task template page of the web interface contains a task template tree in the left pane with seven folders. The content of the selected folder is displayed in the right pane.

There are seven task template folders in the tree. They are described here.

Deployment Schemes
Deployment schemes contain parameters that indicate how an OS configuration must be deployed on your target. The **deployment Schemes** folder contains at least the **Default** scheme.

Idle layout
The idle layout defines what is shown on the target when there is no pending task. The **Idle Layout** folder contains at least the **Idle state** layout.

Menu Layout
The menu layout defines how deployment menus are shown to the users. Menus are used when an OS configuration and on deployment CDs. The **Menu Layout** folder contains at least the **Menu** layout.

OS Detection Layout
The operation system detection layout defines the target display when a target is busy detecting the currently installed operating system. It is used when creating a cloning profile from the web interface. The **OS Detection Layout** folder contains at least the **Detect operating system** layout.

Profile Creation Layout
The system profile creation layout defines the target display when a target is busy creating a new system profile. It is used when creating a cloning profile from the web interface. The **Profile Creation Layout** folder contains at least the **Creating cloning profile** layout.

Profile Restoration Layout
The system profile restoration layout defines the target display during the manual restoration of a system profile by the administrator.

Note: A system profile restoration is always performed as-is and must not be confused with an automated deployment resulting in a fully configured operating system installation.
The **Profile Restoration Layout** folder contains at least the **Default OS Restoration** layout.

State Capture Layout
The state capture layout defines the target display when a target is saving the operating system state for future redeployments.

State Restoration Layout
The state restoration layout defines the target display when a target is redeploying an operating system from a saved state.

When a task template is selected in the right hand pane, the bottom of the web interface contains a link to **Customize GUI**. Follow this link to modify the look of your target screen . When the selected task template is a deployment scheme, there are additional links to view and edit the current scheme.

Customizing a screen layout

You can customize the screen layout of a target.

To customize a screen layout:

1. Select the layout that you want to customize in the right pane of the **Task Templates** page of the web interface page.

   Note: An actual layout must be selected and not a layout folder (left pane)
2. At the bottom of the page, the screen layout is shown in reduced size. Click **Customize GUI** to open the screen layout editor.
3. The editor is composed of a left column, containing instructions, a *What-You-See-Is-What-You-Get* (WYSIWYG) view of the screen being edited and a bottom banner with action buttons.
4. Click on the action buttons or directly on the items that you want to modify to see their editable properties displayed in the left column. Make the wanted changes and then click **Save** to keep your new screen design. Return to the **Task Templates** page by clicking **Back**.

Creating and editing deployment schemes

By customizing your deployment schemes, you can adapt the way in which your predefined OS configurations are installed onto targets.

1. The easiest way to create a new deployment scheme is to run the deployment scheme wizard.Go to **Server** → **OS deployment** → **Task templates**. and click **New deployment scheme**.
2. Alternatively, you can modify an existing scheme by editing its parameters. To do this, select a scheme and click **View deployment parameters** and then use **Edit** in the banner on top of each parameter section. Some advanced deployment scheme features are available only in this mode and not through the wizard.
3. If you prefer using a wizard to edit your scheme, click **Edit parameters using a wizard**.

   The following parameters apply for simple one-time deployments and for redeployment operations.

   Description

The first step is to enter a name for this deployment scheme. Make it explicit enough so that you can pick it easily when starting a deployment (the web interface does not show the settings in a deployment scheme, so the choice must be made by name only). Because deployment schemes determine how the computers are installed (and not what is being deployed), use a description such as `Multicast 50 targets` or `On-site` rather than the name of a OS configuration or of a group of computers.

When the deployment starts

You must determine if Tivoli Provisioning Manager for OS Deployment requires user interaction during deployment (to edit individualtarget parameters) or runs completely unattended.Select:

Always edit target-specific parameters
to have the opportunity to change the target parameters at every deployment. The parameter edition can be made either directly on the target computer or by double-clicking the target icon in the Target Monitor.

Edit parameters for targets that are not yet in the database
to be prompted only during the first installation of each target. Subsequent deployments for the same targets run unattended. This is the default choice.

Never edit parameters
to have unattended deployments if all BOMs have been previously entered in the database. Any missing entry leads to a red banner on the target and cancels the deployment for this target.

You must also select how Tivoli Provisioning Manager for OS Deployment behaves when the model of the computer being deployed does not match the model of the computer on which the image was created. This feature requires DMI for hardware detection.Select:

No if you know that all your system profiles are fully hardware-independent or for deploying universal images.

Yes, display a warning
if you want to see all possible OS configurations for a computer, but want to avoid mistakes. This choice can require user interaction and is therefore not appropriate for a fully unattended deployment.

Yes, abort the deployment
if you want to prevent anyone from using an OS configuration on a computer different from the one for which it was designed.

Use 'BIOS fallback MBR' to start PXE is used when PXE activation (the process of enabling PXE when booting on the hard-disk) does not work.

The PXE boot code manages the multiple reboots needed to install a computer. To manage these reboots, the PXE boot code must intercept the boot process of the computer at every boot.

- If the computer is configured to always start on the network (LAN device first in the list of boot devices), there is nothing to do, because Tivoli Provisioning Manager for OS Deployment is loaded into memory at every boot.
- If the computer is configured to start on the hard-disk, you can change the MBR of the hard-disk and make it point to the work partition at the end of the hard-disk. Tivoli Provisioning Manager for OS Deployment is then loaded from the hard-disk when the computer starts up, instead of loading the operating system. The disadvantage of this method is that, because the computer did not use the network card to boot, PXE is not available. To enable network access, PXE is activated with a special function in the PXE card that makes it behave as though the computer had booted on the LAN. However, this is not documented in PXE, and does not work on every network card. If the network does not support this, an error is raised, and access to the OS deployment server fails (the message **Network started**, followed by an error).

  When PXE activation does not work, you can write a special MBR telling the BIOS that the hard-disk is not a valid boot device. By default, the BIOS falls back to the next device in the list, which in most computers is the network. As a result, the computer boots on the network and has full access to the network. This is the purpose of the **Use 'BIOS fallback MBR' to start PXE** check box.

Data collection

By default, Tivoli Provisioning Manager for OS Deployment automatically populates the database with an inventory of the hardware setups of all deployed computers. For Windows, a software inventory can also be populated based on the registry. If you are not interested in using those inventories, or if your computers do not comply with any of the hardware detection standards, you can disable these features. Be advised that running the hardware or software inventory on thousands of computers can produce a huge database. This inventory is performed on locked screen.

Tivoli Provisioning Manager for OS Deployment centrally reports the status of the deployment of targets to the OS deployment server and to the server database. Additionally, if you want to keep the deployment logs and the list of software modules on each deployed target, you can specify a local path where the log files are to be stored. The path that you specify is relative to the root of the operating system on the target, for example, `/` on UNIX operating systems or `c:/` on Windows operating systems. In the deployment scheme details, the label of this field is **Save deployment log to**.

When the deployment is completed

When the deployment process is finished, you can select if you want to:

- Turn off the computer automatically (if supported)
- Boot the operating system automatically (this value might not make sense with some values of the previous setting)
- Display a green banner and wait for a manual shutdown

Network usage

Note: Multicast is available only if:

- The targets have an Intel x86 or x86-64 architecture
- Multicast is selected in the deployment scheme
- The subnet supports multicast traffic
- Multicast is not disabled in the boot options of the target
- The target is not a VMWare 3.0 guest.

Depending on the number of targets that you are deploying simultaneously on your infrastructure, you must select one of the following networking modes:Select:

Unicast
to deploytargets one by one, or if you cannot use multicast. When deploying several targets simultaneously in unicast, the deployment time increases dramatically with the number of targets, as the result of network saturation.

Multicast, without explicit synchronization
to use soft-synchronized multicast protocol. Using this protocol, every target independently starts downloading images at soon as it is ready, and continues with the deployment as soon as it has downloaded all required material. When two or more targets (using the same deployment scheme) are downloading files in parallel, they automatically share the same bandwidth. The fastest target has the priority for the choice of the next shared files to be sent by the server, but the slower targets can receive them if they need them. This is a scalable solution that allows for a rolling deployment scenario.

Multicast, with the following synchronization parameters
to use a classical replicated multicast method. This mode is adequate for installing computers in batches. Enter the replication parameters (for example, the number of targets to wait for before starting the download, and the maximum timeout before starting in any case). Tivoli Provisioning Manager for OS Deployment multicast protocol can accept new download targets even after the initial replication period is over, and integrate them seamlessly into the transfer.

Note: In the first stage of an OS deployment, there are two target synchronization stages. Therefore it might seem that the maximum timeout that was set before starting the deployment is doubled.

Vista 2008 Windows 7 You can decide to use a network share on the server to download the files to the targets , rather than downloading the whole image to the hard disk of the target. Using a network share provides a shorter installation time. To use a network share:

- Select **Use network share** in the deployment scheme.
- Share the `files\global\partition` directory and provide at least read-only access to it.
- Enter the relevant server parameter in the **Network share module** section. Go to **Server** → **Server parameters** → **Configuration**. (See "Network share module" on page 314).

On-site deployment

If you are running a one-time deployment in a deployment center and do not want to use redeployment, leave the check box blank and click **Next**.

If you are running an on-site deployment, or if you plan to use redeployment, you can enable the advanced feature.

Redeployment option
Indicate if you want to keep the deployment image in a protected partition and the size of this partition. These options are valid only to configure the deployment scheme for redeployment. More information is available in deploy/tosd_redeplscheme.dita.

Note: The following parameters cannot be modified using the wizard. You must edit your deployment scheme parameters.

Request user confirmation
This parameters allows you to ask for user confirmation before running a deployment.

Unbind OS configuration at the end
This parameter, located in the **General settings** section, is set to **No** by default. Setting this parameter to **Yes** unbinds the OS configuration from the target at the end of the deployment. This OS configuration is not proposed the next time the target boots and, if no other OS configuration is bound, the target presents a locked screen to the user. Review the section on network boot scenario of the deployment process topic.

Unbind software module at the end
This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the software module at the end of the deployment. This software module is not proposed and installed the next time a deployment is performed.

Vista 2008 Windows 7 **Disable user interaction during deployment**
This parameter, located in the **General settings** section, is set to `Yes` by default. If you set this parameter to `No`, you can obtain a command prompt by pressing Shift-F10 on the target computer during a deployment to modify deployment files.

Note: This parameter is relevant for Windows Vista/2008/7 deployments only.

Send mail at end
This option can be set only if a *sendmail* tunnel has been created.

Send mail to
This parameter is available only if **Send mail at end** is not set to **No**. Use this field to enter the e-mail address to which information must be sent at the end of the deployment.

Configure Network
This parameter, located in the **Network settings** section, is set to **Before software installation** by default. The network setting of the target are set before software modules are installed, enabling the product to use the network settings during the installation of the software modules. Set the parameter to **After software installation** if you want the network settings to be applied after software modules are installed, for

example if you intend to physically move the target after deployment and want it to be configured with the network settings for its final location.

Disable cancel button
This parameter, located in the **Client Display** section, allows you to prevent users from aborting a deployment by removing the cancel button. Set the parameter to **Yes** if you want to remove the cancel button from the client display

You can use the newly created deployment scheme to deploy a system profile.

To delete a deployment scheme, select the scheme, then right-click it, and click **Delete**.

Creating media for deployment for Linux operating systems

You can create deployment media such as CDs, DVDs, or USB drives to install machines without connecting them to the OS deployment server.

You can use this kind of deployment when there is no connection or connection to the OS deployment server is very slow.

Some typical situations are small branch offices with slow links and no local deployment server, isolated computers with no connection to an internal network, laptop users currently away from LAN or connected using a modem.

If the data you want to use does not fit on a single CD or DVD, use a USB drive.

Note:

- You must create the deployment media from an OS deployment server or a web interface extension installed on a computer with the same byte order (little endian or big endian) as the one on which you want to use the deployment media.
- Offline deployment media are not supported on virtual machines.

Deploying Linux from a deployment media

- When you use a deployment media to deploy a Linux operating system, the target keyboard layout cannot be changed.
- When you use a deployment media containing both Windows and Linux system profiles and you want to deploy a Linux system profile, you are asked twice to select your system profile.

Creating an OS deployment USB drive with the wizard

Tivoli Provisioning Manager for OS Deployment can automatically generate deployment USB drives that replay the deployment process for a given system profile or for any kind of software modules available.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must be formatted as FAT32 or NTFS.

Note: SuSE Linux Enterprise Desktop cloning is not supported on USB drive deployments.

Refer to the *Troubleshooting and support* set of topics for information about problems or limitations related to deployments using a network boot USB drive, and to the product release notes or the readme file provided with the fix pack for the most up-to-date information related to problems or limitations.

The deployment USB drive is self-contained and can be used instead of a CD or DVD to provision a target entirely offline, without using the OS deployment server. These deployment USB drives can also be used to deploy computers without a PXE-compliant network adapter.

To create OS deployment USB drives:

1. Perform one of the following operations:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.
2. Click **Generate Media** or select **Create deployment media** in the contextual menu.
3. Select **Create a deployment USB key** to start the USB key wizard. Click **Next**.
4. Specify the operating system for which to build the CD or DVD. Select **Windows** to load a WinPE deployment engine, **Linux** to load a Linux deployment engine environment, or **Both** to load both.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. (*Optional*) Change settings for targets running the USB key that you are creating.

   Included objects

   When selecting objects to be included, be aware that:
   - The wizard displays all the deployment schemes, system profiles, and software modules currently stored on your OS deployment server.
   - At least one system profile must be included in your image.
   - One, and only one, deployment scheme must be included in your image. In this deployment scheme, do not select the **Download files with a network share when applicable** option in the **Network settings** section, because HTTP deployment is not available offline.
   - The software application order is automatically included.
7. If your USB key has already been used as a deployment media, you might choose to keep a shared repository of previous data to improve data regeneration speed. If you keep the existing shared repository, you might use more space on the USB key.
8. Plug your USB key into a machine running the web interface extensionand specify its address.
9. Choose the drive matching your USB key.
10. Click **Finish** to close the wizard.

Use the USB drive to deploy a given system profile or any kind of software module.

Creating an OS deployment USB drive with command lines

You can create an OS deployment USB drive that Tivoli Provisioning Manager for OS Deployment can use when a target cannot boot from the network.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must have boot capabilities and a FAT32 or NTFS filesystem. The drive must be already formatted; existing files on the partition are not deleted. USB keys already filled with a bootable operating system might not work.

Note: Refer to the *Troubleshooting and support* set of topics for information about problems or limitations related to deployments using a network boot USB drive, and to the product release notes or the readme file provided with the fix pack for the most up-to-date information related to problems or limitations.

The command line must be used only when the web interface is either inappropriate or unavailable.

Use this command line:

- On Windows operating systems:

```
rbagent.exe -s <OSD_server_ip_address>:<OSD_server_password>
rad-usbget <drive>
keepshared|delshared preferwpe|prefermcp nodes
```

Where:

OSD_server_ip_address
Is the IP address of the OS deployment server.

OSD_server_password
Is the password for the administrative user (typically `admin`) on your OS deployment server.

drive Is a drive letter of the Windows target where you run the `rbagent` command. The rad-usbget command adds requested files to the FAT32 or NTFS partition and makes it bootable. The drive must be already formatted. Existing files on the partition are not deleted.

keepshared
Keeps a shared repository of previous data to improve data regeneration speed. If you keep the existing shared repository, you might use more space on the USB drive.

delshared
Deletes a shared repository of previous data.

preferwpe|prefermcp
Defines if a Linux deployment engine or WinPE is loaded from the USB drive, when a target boots from this USB drive, without accessing the network. Only when Linux deployment engine or WinPE is running, does the target connect to the network and try to contact an OS deployment server. If you deploy only Linux, specify `prefermcp` to skip WinPE. You can specify `preferwpe` only if there is a WinPE deployment engine on the OS deployment server.

nodes Defines the deployment settings with a space-separated list of objects. Specify at least `DEPLSET:Default` for the deployment schema, and `PROFILE:SystemID` for the system profile.

You can now boot the target using the OS deployment USB drive instead of the network card. To use the PXE emulation USB key, insert the USB key into the drive

and restart the target. If your machine does not boot from the USB key, check the BIOS boot list to see if your optical drive is included in the boot sequence and is listed before the hard disk. Most machines also allow you to select the temporary boot device without changing the boot sequence in BIOS.

Creating OS deployment CD and DVD

Tivoli Provisioning Manager for OS Deployment can automatically generate deployment CDs and DVDs that replay the deployment process for a given system profile or for any kind of software modules available. You can use this feature to create OS deployment CDs and DVDs that can be easily sent through the Internet or by e-mail, to refresh a computer back to its initial working state after installation.

These CDs and DVDs can also be used to deploy computers without PXE compliant network adapter. The creation of DVDs and media spanning is supported. These media can be protected using an activation code preventing unauthorized personnel from using it.

To create OS deployment CD and DVD:

1. Perform one of the following operations:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.
2. Click **Generate Media** or select **Create deployment media** in the contextual menu.
3. Select **Create a deployment CD or DVD** to start the CD and DVD wizard. Click **Next**.
4. Specify the operating system for which to build the CD or DVD. Select **Windows** to load a WinPE deployment engine, **Linux** to load a Linux deployment engine environment, or **Both** to load both.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. Follow the wizard instructions to create an ISO image.

   Included objects

   When selecting objects to be included in the ISO image, be aware that:
   - The wizard displays all the deployment schemes, system profiles, and software modules currently stored on your OS deployment server.
   - At least one system profile must be included in your image.
   - One, and only one, deployment scheme must be included in your image. In this deployment scheme, do not select the **Download files with a network share when applicable** option in the **Network settings** section, because HTTP deployment is not available offline.
   - The software application order is automatically included.

   Hardware options

   In the hardware options settings some boot options can be customized. By default the options are unchecked but some special cases can require changes. In particular, if the CD or DVD is to be used on a USB drive or as a secondary drive, it might be necessary to specify the option **use**

BIOS for CD or DVD ROM access. When this option is selected, on some hardware it might also be necessary to select **disable enhanced disk access** (for IDE CD or DVD) or **disable USB** (for USB CD or DVD) to ensure that Tivoli Provisioning Manager for OS Deployment use of other IDE or USB devices does not interfere with the BIOS access to the CD or DVD. In addition, deploying from the second CD or DVD drive of a target only works if you can ensure that subsequent boots keeps booting on the same CD or DVD drive.

Security issues

For security issues, you might want to protect deployment from the CD with an activation code. When your computer boots on the CD, the activation code is required for the deployment to proceed.

You might also want to hide the content of the ISO image that contains sensitive information such as product keys. To do this, select **Hide the content of CD or DVD** in the CD or DVD Wizard. If you then try to access files in your ISO image, you see the content as `CDROM_content_hidden`.

Size of the ISO file

The wizard allows you to choose the size of the ISO images.

a. Enter the maximum size in the field displayed.

b. Click **Next** and the wizard starts to precompute the ISO file size.

The wizard displays the results for the number of disk images and the size required. You then have the option to:

- Download it directly from the server.
- Use the web interface extension
- Generate it on the server itself in the import directory.
- Generate it on another computer running the web interface extension

Note:

- When creating the ISO files, all objects of type *single file to copy*, *image headers*, and *WIM images* (which includes Windows Vista/2008/7 unattended setup profiles), are put on the first CD or DVD. Therefore, the first ISO file might grow larger than the requested spanning size if the total size of the files to be put on the first ISO requires it.

  For example, if you try to create an OS deployment DVD containing both Windows Vista/2008/7 unattended setup profiles, both profiles must be contained on the first ISO, but their total size is larger than 4 GB. Therefore, the ISO cannot be burned into a single layer DVD. In this case, either use a double layer DVD, or transfer the ISO without burning it.
- When deciding where to generate the ISO image, be aware that:
  - If the estimated size is bigger that 2 GB, do not use the link to download directly from the server, because of limitations of web browsers. An exception to this rule is Mozilla Firefox on Linux, which can extract files as large as 4 GB or more.
  - Because of file system limitations, do not extract files bigger than 4 GB on FAT32 partitions.

Use a CD creation tool to burn the ISO image onto disks.

Note: Vista 2008 Windows 7 Windows Vista/2008/7 unattended setup profiles contain at least one file larger than 1 GB which cannot be split. Therefore, ISO files containing Windows Vista/2008/7 unattended setup profiles must be burned on a DVD.

If you encounter problems when deploying from this CD or DVD on a virtual machine, make sure that the CD drive comes after the hard disk in the boot order.

Setting up an activation code

For security issues, you might want to protect deployment or booting from the CD with an activation code. When your computer boots on the CD, the activation code is required for the deployment or the network boot to proceed.

To prevent being asked several times for the activation code during deployment:

- The deployment scheme included on your deployment CD must have the network setting **Use 'BIOS fall back MBR' to start PXE** set to **No**.
- The boot order of your target must be set to hard disk first and you must boot on the CD manually the first time.
- To set up an activation code for the first time, when creating the deployment CD:
  1. Select **Include activation code protection** in the deployment media wizard.
  2. Enter and confirm the chosen password. You must remember this password if you want to obtain other activation codes for this CD.
  3. Set a password expiration date under **Valid until**.
- To obtain a new activation code, for example, if you must use the CD after the current activation code expiration date:
  1. Click **Generate Media** on the Profiles page to start the deployment media wizard.
  2. Select **Generate a new activation code**.
  3. Click **Next** and follow the wizard instructions to obtain your new activation code. You must remember the password given when creating the first activation code for this CD.

The wizard provides you with the generated activation code that you need when using the CD.

Deploying Linux

A deployment is the process of installing an operating system on a target, and configuring the operating system for a specific user.

The deployment process

In Tivoli Provisioning Manager for OS Deployment, a deployment is made up of several steps that are automatically run in sequence without user interaction:

1. Hardware configurations are optionally deployed, for example, to create RAID volumes.
2. Partitions are created on the hard disk, and then formatted according to information contained in the system profile.
3. All deployment objects (system profiles, partition files, and software modules) are downloaded to a temporary storage location on the hard disk.

4. Operating system files are written in the hard disk partitions, creating a bootable operating system with files and applications configured by database bindings between the *target* and *software modules*.
5. Target-specific configuration, such as the *host name* or the *product key* are gathered from the database to create a textual configuration file used by the system preparation tool.
6. The operating system is started, allowing LinPrep to configure the operating system according to information stored in the Tivoli Provisioning Manager for OS Deployment database.
7. Additional software is optionally installed, if it must be installed after the operating system.
8. The temporary storage location is cleaned. Installation files are removed.
9. Tivoli Provisioning Manager for OS Deployment takes control again when LinPrep has completed and rebooted the target, and displays a message indicating the status of the deployment.

When the deployment is complete, the operating system is installed and ready to be used by the user defined for this target in the database.

HTTP deployment

From version 7.1.1.3 of the product, the deployment of Linux system profiles in unicast can be performed through HTTP.

To activate or deactivate HTTP downloads in unicast, change the value of **Download files with a network share or Linux HTTP when applicable** in the **Network settings** section of your deployment scheme. By default, the value is set to **Yes**.

The deployment of Linux system profiles in multicast remains unchanged.

Network boot scenarios

Depending on the number of OS configurations bound to a specific target, a target behaves differently when it boots on the network:

- If no OS configuration is bound to the target (for example, when a target starts for the first time and has not been configured), a special screen is displayed that asks the administrator to configure an OS configuration binding for this target on the OS deployment server. Deployment is not possible until an OS configuration is bound to the target.
- If one or more OS configurations is bound to this target, but no deployment has been scheduled on the server, a screen is displayed with a list of all the OS configurations bound to the target. Clicking on an item in the list starts an interactive deployment for the selected OS configuration, using either the **Default** deployment scheme (if no deployment scheme has been configured for this target), or the deployment scheme used during the last deployment.
- If one or more OS configurations are bound to this target, and a deployment has been scheduled on the server for a specific OS configuration, the target immediately starts the deployment without requiring any user intervention.

Deployment requirements

To start a deployment on a target, several elements must be present in the database.

The following elements are required:

- A *deployment* scheme associated with the target to deploy. The deployment scheme determines how to deploy the operating system on the target. If there is no association between a deployment scheme and the target to deploy, Tivoli Provisioning Manager for OS Deployment automatically uses the **Default** deployment scheme.
- An operating system *configuration* that is used to select which operating system to install. If there is no OS configuration associated with the target to deploy, the deployment does not start.
- Optional *software modules* to install in addition to the operating system during the deployment process. If there is no software module associated with the target to deploy, the operating system image is deployed without modification.

The OS configuration and the software modules can be considered to be the *content* of the deployment, while the deployment scheme is the *how* of the deployment.

The database keeps information about associations (*bindings*) between targets and deployment schemes, between targets and OS configuration, and between targets and software modules. These bindings can be configured manually or with binding rules (for example, `deploy configuration windows XP on targets whose model name starts with Dell`).

The minimal binding required to start a deployment is an OS configuration. If no configuration is bound to a target, the deployment does not start. In practice, however, Tivoli Provisioning Manager for OS Deployment always asks for an OS configuration and deployment scheme when beginning a deployment.

Tools to start and configure deployments

Bindings between targets and deployment elements are necessary to perform a deployment. You can create and edit these bindings in the **OS configurations** panel of the **Target details** page.

The Target Monitor provides functions to prepare a deployment, start a deployment, follow the progress of a deployment, and organize targets.

Binding rules, used to create permanent implicit bindings between targets and deployment elements without having to explicitly create the binding for each target, are created using the web interface. OS configurations and software modules contain a specific section at the very bottom of the **Details** page for creating automatic binding rules.

Starting a one-time deployment

You start deployments in the web interface by indicating on which target or targets the deployment must occur.

When your target has different network card interfaces, before deploying a system profile, ensure you define the target network configuration by performing the following steps:

1. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details. Click **Switch to Advanced IP settings mode** in the **Common networking info** section.
2. Click **Edit**.
3. Set the connection name in the **Connection name** field.

To start a deployment:

1. Select a single target or multiple targets on the Target Monitor page. To select multiple targets or deployment, select an administrative group, a custom list, a subnet, or click on individual target names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
4. If you do not choose this option, select the type or types of deployment you want to perform. You can install additional software only if you deploy an operating system.
   a. If you have selected **Perform hardware configuration tasks**, indicate which hardware configurations you want to deploy.
5. Select **Simple deployment** and click **Next**
6. Follow the deployment wizard instructions to select a deployment scheme, an OS configuration and optionally software modules, and to set up deployment options.

   Three options are available in the deployment wizard to deploy targets without physically interacting with the computers.

   - **Try to wake up targets currently powered off using WOL** asks the Target Monitor to send IBM Wake on LAN packets to wake up targets. Waking up targets only works on carefully designed modern computers. A target can only be woken up if its network adapter and its system board support Wake on LAN packets, and if the network adapter has been shut down properly. If the network adapter is not in the appropriate power state, Wake on LAN packets will not wake the computer up. This is not specific to Tivoli Provisioning Manager for OS Deployment, but is rather a general limitation of the Wake on LAN technology.
   - **Try to wake up targets using management interface** tries to turn on the target using command lines to the management interface. This works only for targets with an interface management known to the product and properly identified.
   - **Try to reboot targets running theweb interface extension** asks the Target Monitor to contact the targets if they are running under Windows and send a reboot request. If you are not running Windows, you cannot reboot targets remotely. They must be rebooted manually. If you are running Windows, you need the web interface extension running with the correct privileges to run a remote boot.

   If you have not selected one of these options or if they do not work, and if the target you are trying to deploy is not powered on, turn it on now and make it start on the network.

When the deployment is complete, the server either displays a green banner on the target, boots in the operating system, or powers the target off, depending on how the deployment scheme is configured.

After deployment, the following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Partitions are aligned on megabytes rather than on cylinders. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

Deploying a hardware configuration

A wizard allows you to effortlessly deploy hardware configurations.

To start a hardware configuration deployment you must first have at least a hardware configuration environment and a hardware configuration.

Note: You can not deploy a hardware configuration from a target started with a network boot media.

1. Select a single target or multiple targets on the Target Monitor page. To do this go to **Server → OS deployment → Target Monitor**. To select multiple targets or deployment, select an **administrative group** , a **custom list**, a **subnet**, or click on **individual target** names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the second screen of the deployment wizard, you must select at least **Perform hardware configuration** tasks and if you want to proceed with the Operating System/Software deployments you can also select another option.
4. Select one or several hardware configuration(s) you want to apply on target. RAID Configurations, BIOS Settings, BIOS Updates or Hardware custom configurations are classified in a matching folders.
5. Follow the deployment Wizard as it is described in the chapter Deploying depending on the options you chose above.

Every configuration you selected will automatically use the appropriate environment and only be applied if the model is matching the target.

Redeploying

This function is a special deployment scheme that gives you the ability to rapidly restore an image to a computer from a hidden partition on the computers hard-disk.

During the original image deployment to the computer, Tivoli Provisioning Manager for OS Deployment creates a hidden partition on the hard-disk of the target computer. When it has finished deploying the master image on the computer, it stores a reference image into the hidden partition. It is possible to store one or more reference images into a hidden partition on the computer.

Note: Before running a deployment task on a machine with a redeployment partition, ensure you remove the hard disk partition content.

Each time the system is booted, either off the hard-disk or using network boot, Tivoli Provisioning Manager for OS Deployment intercepts the boot process of the computer and presents a customizable menu of possible actions. Those actions are:

- Boot the system off the current image on the hard-disk.
- Do a quick cleanup of the currently deployed image against the reference image and restore the image from the hidden partition.
- Do a format and full restore of the reference image from the hidden partition. Using this function, it is possible to effectively have a fresh image deployment every day for the optimum performance of a system.
- Choose and deploy another configuration available on the hidden partition. This option takes as long as the format and restore option.

The purpose of redeployment

A computer generally works the best and the fastest on the day that it is installed. At that time, the system is completely clean, free of any undesirable processor-consuming gadgets, and all programs are configured for their optimal use by the system administrator. The purpose of redeployment is to ensure that the system is reset to this optimal state at every boot (or at some fixed interval).

There are three categories of systems that experience the most visible need for the redeployment technology:

Public computers
: such as schools, universities, and Internet cafes, where users cannot be relied on to preserve the computer integrity, because the computer is not their own

Critical systems
: such as banks, insurance companies, and industrial plants, where the company cannot afford to risk computers being reconfigured or infected by malicious software

Embedded systems
: such as ticket machines, airport information systems and ATMs, that must be quickly rebuilt to their original OS configuration, without using a specific infrastructure

Because redeployment often occurs at the user's desk, it is necessary to find a solution that is quick, easy to use, does not require any significant infrastructure, and does not affect the work process of other users. This rules out standard deployment tools, because they impose a significant load on the network and affect other users' ability to perform their tasks.

Note: The redeployment feature is not intended to be used on virtual machines. On virtual machines, you should leverage the snapshot feature of your hypervisor rather than use the redeployment feature.

The redeployment process

Redeployment involves several steps, including creating a reference image of the target, and saving it as a redeployment partition.

Redeployment steps

Tivoli Provisioning Manager for OS Deployment addresses the challenge of redeployment with the following steps:

- At the end of a deployment, Tivoli Provisioning Manager for OS Deployment creates a reference image of the target, and saves it into a protected redeployment partition (invisible to the user and to the operating system itself). This increases deployment time by roughly 10% compared to a simple deployment,, as most of the files are already present as file archives on the disk at that time.
- Every time a target starts, Tivoli Provisioning Manager for OS Deployment hooks the boot process before the operating system starts (using PXE or a special Master Boot Record).
- If configured to do so, Tivoli Provisioning Manager for OS Deployment authenticates the user of the target against the server database to restrict the use or the maintenance of the target to authorized persons only.

- If configured to do so, Tivoli Provisioning Manager for OS Deployment offers the choice of several OS configurations available on the target (multiboot), and of several levels of "cleaning".
- Using the reference image saved during deployment, Tivoli Provisioning Manager for OS Deployment resynchronizes the hard-disk content to its reference state. This typically takes only a few seconds, but can take up to a few minutes if everything on the hard disk has been deleted.

Offline redeployment limitations

Offline redeployment behaves slightly differently from online redeployment as the OS deployment server cannot be contacted for information. These limitations are removed after the target contacts the OS deployment server again. For example, interrupted tasks are not automatically resumed and changes to the partition scheme cannot be recovered.

Moreover, authentication with offline redeployment does not work. A message warns the user.

Note: If you plan to use redeployment with multiple OS configurations offline, make sure that all the preloaded OS configurations have exactly the same partition layout (number and size), because Tivoli Provisioning Manager for OS Deployment cannot create new partitions offline or to resize existing partitions offline. Failure to do so prevents you from redeploying offline some of the preloaded OS configurations.

Redeployment with multiple operating systems

You can preload up to three operating systems on a target, with a menu allowing the user to select which operating system to start.

Scenario

You want to provision the computers of a classroom with three different operating systems (for example, Windows XP by cloning, Windows Vista, by unattended setup, and SuSE 10 by cloning). When entering the classroom, the student must choose between the three operating systems. For security reasons, you want to make sure that the operating system which is started is always in a clean state. You also want the selected operating system to install and start quickly.

Principles

To achieve this, you must install each operating system in its own partition, save the OS configurations in a protected partition. Before you start an operating system, you do a rapid verification of the operating system partition with the information in the protected partition.

Requirements

For you multiple operating systems to cohabit in a single target and to be able to start them individually, you must follow these guidelines strictly:

- The hard disk of the targets must be large enough to contain the three operating systems and the protected partition.
- You must create a separate system profile for each operating system.
- All the profiles must have the same number of partitions, in the same format.

- Each operating system must be in a distinct partition, and all other partitions must be empty during the system profile creation.
- Each operating system must be in a primary partition, and there is a maximum of three primary partitions.
- In the system profiles, partition numbers cannot be modified.
- An offline refresh does not update the partition table.

Procedure

1. Create your Windows XP cloning system profile.
   a. Start a target with the Windows XP CD.
   b. Create and format one large partition.
   c. Install Windows XP on the partition.
   d. Customize your installation.
   e. Determine the best size for the partition.
   f. Clear the administrator password.
   g. Run Sysprep.
   h. From the web interface, clone your target to create a new system profile.
2. Create your Windows Vista unattended system profile.
   a. From the web interface, create a new unattended system profile with the profile wizard, following the instructions.
      1) Create a small partition 1.
      2) Create a large partition 2 for Windows Vista.
   b. Customize your OS configuration.
      1) Set the **administrator name** in the configuration.
      2) Optionally, bind software modules.
      3) Determine the best size for partition 2.
3. Create your SuSE 10 cloning system profile.
   a. Start a target with the SuSE 10 CD.
   b. Delete partitions 1 and 2. Recreate and format two small partitions.
   c. Create one large primary partition (EXT2) for / (partition 3).
   d. Create a swap partition of 1 GB (logical partition).
   e. Install SuSE 10 in partition 3.
   f. Customize your installation.
   g. Determine the best size for the partition.
   h. From the web interface, clone your target to create a new system profile.
4. Update the OS configurations.
   a. Edit the partition scheme for each OS configuration so that partitions have the same size on each OS configuration.
   b. Use the best size found for each operating system.
   c. Set the options**Must be deployed** and **Must be redeployed** so that only the partition containing the operating system is actually deployed or redeployed for each system profile.
5. Test each system profile. Each operating system installs in the correct partition, without impacting other partitions.
6. Create a specific deployment scheme for this redeployment.
   a. Export the three system profiles into a RAD file.

b. With the deployment scheme wizard, create a new deployment scheme enabling redeployment.
c. For **Protected redeployment partition size**, give 200% of the size of the RAD file you have just created.

7. Preload the system profiles on your targets.
   a. Select the targets in the web interface.
   b. Select **Deploy now** in the contextual menu.
   c. Select **Redeployment preload** in the deployment wizard.
   d. Select the deployment scheme you have just created.
   e. Select the three OS configurations that you have prepared.
   f. Optionally, select additional software modules.
   g. Click **Customize GUI** if you want to customize the boot menu appearing in the target.

When your targets boot, they now display a menu with the three possible operating systems in which they can start.

Configuring a deployment scheme for redeployment

Redeployment is a feature that affects *how* the target is being preinstalled, not *what* is in the deployed OS configuration. Redeployment is enabled by customizing a deployment scheme.

Because redeployment is basically the replay of a standard deployment operation, you must first configure a regular deployment process, and try it on a test computer. When you have performed these two stages, follow the instructions provided to turn your one-time deployment OS configuration into a redeployment OS configuration.

To customize a deployment scheme for redeployment, you can

- Create a new deployment scheme with the deployment Scheme Wizard
- Modify an existing deployment scheme with the deployment Wizard
- Edit the parameters of an existing deployment scheme manually

The following steps are based on the first and second options, which are very similar.

1. Follow the first alternative to create a completely new scheme, and the second alternative to modify an existing scheme with the wizard:
   - Go to the **Task templates** page and click **New deployment scheme**. This launches the deployment Scheme Wizard, which guides you through the customization of deployment parameters.
   - Go to the **Task templates** page. Select a deployment scheme, and click **Edit parameters using a wizard**.
2. Follow the instructions of the wizard in the same way as for a regular deployment, until you reach the panel called **On-site deployment features**.
3. Select **Enable support for quick redeployment of the same OS configuration** and click **Next**.
4. On the next panel, **Redeployment option**, select **Yes, keep** IBM Tivoli Provisioning Manager for OS Deployment **images in a protected partition**. Optionally modify the space that you want to allocate to this special partition, and click **Next**.

   Note:

a. The protected partition size must be at least as large as the total size of all system and software images to be deployed on the computer, because it retains all these images. If you are unsure of the space required, start with approximately 800 MB for a Windows 2000 configuration, 1500 MB for a Windows XP configuration, or 1500 MB for a Linux configuration. If you want a more precise number, check the image sizes reported in a deployment log, and round up the total to accommodate the miscellaneous structures used for redeployment.

b. The space that you allocate to the redeployment partition is subtracted from the hard-disk total capacity detected by Windows or Linux. The user cannot detect, access, or delete this protected area from the operating system disk manager. It is not simply a hidden partition, but a hardware-protected area, as defined in ATA-5 specification. If necessary, you can recover this space by running another deployment operation.

5. Click **Finish** to complete the customization process and obtain a deployment scheme ready for redeployment.

Edit the parameters manually:

1. Go to **Server** → **OS deployment** → **Task templates**.
2. Select a deployment scheme
3. Click **View deployment parameters**
4. Click **Edit** in the section header in which you want to modify parameters.

Preloading for redeployment

Before you can redeploy a target, you must preload one or several OS configurations.

For a successful redeployment, targets must not **Boot on hard-disk if idle**. Make sure this target parameter is not selected for the targets you want to redeploy.

LVM partitions are not supported for redeployment. Make sure the OS configuration you want to redeploy does not contain an LVM partition.

After you have created an appropriate redeployment scheme, you can begin the preload of the OS configurations of your choice on the target. This operation must be initiated using the Target Monitor page of the web interface.

1. Select the targets to deploy and select **Deploy now** from the contextual menu to start the deployment wizard.
2. Select **Redeployment preload** and click **Next**.
3. Follow the instructions of the deployment wizard.

   Note:

   a. When you select a deployment scheme, only those configured for redeployment are displayed. If you do not have any scheme ready for redeployment, a warning message appears.

   b. Preloading more than one OS configuration is supported, but increases the preload time.

The preload automatically starts when the targets boot, just like with regular one-time deployments. The process goes through the same steps, with one exception. When Sysprep or `LinPrep` has completed and after all software modules have been installed, an image of the fully configured target is stored on the

redeployment partition. If you have selected multiple OS configurations, the process repeats for all OS configurations in turn, until all redeployment images are ready.

Customizing the redeployment menu

You can customize the menu entries that you see in the user interface when starting a target in redeployment mode. Each OS configuration can define one or more menu entries, and the complete menu is the union of all entries defined by all available OS configurations.

After having selected **Redeployment preload** in the deployment wizard and selected the deployment objects:

1. Click **Customize GUI** in the deployment wizard. This opens the menu customization interface which is divided into three parts:
   - A left column with instructions on how to modify the menus and editable fields
   - A bottom banner with action buttons
   - A view of the target screen as it will appear
2. Click **New menu item**.
3. Modify the captions and actions.
4. You can select one of the following actions:
   - **Format and restore**
   - **Quick restore**
   - **Boot on OS**
5. If you want to protect a specific menu item from unauthorized users, you can set up a global password or user authentication for that user by selecting an appropriate value under **Authentication**. To make full use of this feature, you must first have defined authentication domains in the **Server parameters**. Three authentication formalisms are available

   Authenticate locally on RAD *group*
   uses the local user database to authenticate a user. The optional *group* parameter can be used to restrict the verification to a specific group of users. This type of domain is supported by both Windows NT and UNIX versions of the OS deployment server.

   Authenticate on NT server *server:group*
   forwards authentication requests to the NT server specified by the mandatory parameter *server*. The optional parameter *group* can be used to restrict the verification to a specific group of users. This type of domain is supported by the Windows implementation of the OS deployment server only.

   Authenticate on Radius server *ipaddr:secret*
   forwards authentication requests to the Radius-compliant device specified by the parameter *ipaddr*. The value of the parameter *secret* is used as the secret for the Radius communication, and must match the secret stored in the configuration of the Radius device for the protocol to work.

   Note: Authentication with redeployment does not work if the target is offline (the target has no network connection and boots from the hard disk). A message warns the user. If you plan to redeploy offline, use a global password rather than user authentication.
6. Click **Save** and then **Close** to exit this window.

Formatting hard disk and restoring files:

With this option, your partitions are always reformatted and all the files restored before you boot into the operating system.

After your targets are preinstalled for redeployment, they always boot into the user interface, independently of the selected boot order in the BIOS (disk or network). If user authentication has been configured, targets connect to the OS deployment server using the PXE network adapter even if they start from the hard-disk.

Note: Before redeploying Linux profiles, assign a label to the each profile partition or define a corresponding device name in the `/etc/fstab` file.

On the target, select the OS configuration to be restored.

After an OS configuration has been selected, Tivoli Provisioning Manager for OS Deployment completely format the disk and then restore all files. The default behavior is to:

1. Format the disk partitions as specified in the system profile.
2. Restore all the files from the hidden partition.
3. Boot on the selected operating system.

Using quick redeployment:

This option is the typical way to use redeployment. A fast verification of partitions and files is run and, fixes are performed if needed before the target boots into the operating system.

After your targets are preinstalled for redeployment, they always boot into the user interface, independently of the selected boot order in the BIOS (disk or network). If user authentication has been configured, targets connect to the OS deployment server using the PXE network adapter even if they start from the hard-disk.

On the target, select the OS configuration to be restored.

After an OS configuration has been selected, Tivoli Provisioning Manager for OS Deployment automatically restores it as quickly as possible. The default behavior (which typically takes only a few seconds to run) is to:

1. Verify that the disk partitions match the wanted system profile, and fix them if needed.
2. Verify that all partitions have the appropriate file content, and fix them if needed.
3. Boot on the selected operating system.

Booting on the installed operating system:

This option allows you to boot on the currently installed operating system, without any verification. It is fast, but it does not prevent operating system corruption.

After your targets are preinstalled for redeployment, they always boot into the user interface, independently of the selected boot order in the BIOS (disk or network).

If user authentication has been configured, targets connect to the OS deployment server using the PXE network adapter even if they start from the hard-disk.

On the target, select the option that allows you to boot on the operating system.

The target boots directly in the installed operating system, without any disk partition or file verification.

Monitoring deployments

There are several ways available to monitor the deployment progress.

Monitoring deployment progress with theTarget Monitor

You can use the Target Monitor to monitor deployments remotely. Information is located on the **Target Monitor** page and on several tabs of the **Target details** page.

On the **Target Monitor** page, the target color changes during the deployment. When PXE is activated, targets are monitored on a regular basis. The color of the icon is updated as soon as the status changes. By pointing to the target icon, you can get a description of the target status.

Note: A successfully deployed computer can continue to have a yellow icon (indicating that the deployment is still in progress). This reflects a PXE activation problem. The computer, having booted on the hard disk, is not using the network to inform the OS deployment server of its status. To remedy this, select the **Use 'BIOS fallback MBR' to start PXE** check box in the deployment scheme wizard. This forces the computer to boot through the network first.

If the deployment scheme used is configured to collect inventory information about target hardware (which is the default), you can see information about target hardware in the **Inventory** panel of the **Target details** page for that target (double-click on the target to go to the details page).

At the end of the deployment, the target icon shows either a green screen (success), or a red screen (failure). The deployment logs stored on the OS deployment server provide information about the deployment process. They are particularly useful in case of deployment failure to track its cause. To access the logs, double-click the wanted target. This opens the **Target details** page. Select the **Logs** tab to display a list of logs. To view a specific log, click its description. To download it, click **download** immediately after the log description.

Note: Logs are text files with UTF-8 formatting. If you are using a Windows operating system, you can view log files adequately by opening them in Microsoft WordPad.

There is only one log file for each deployment. This log file contains information about the different stages of the deployment process, including reboots and information provided by the operating system being deployed.

If any log information needs to be propagated to the OS deployment server outside of any task, an *idle* log file is created to store this information. The idle log file is created on demand and does not therefore exist for all the targets.

Another place of interest for information about a current deployment or another current task is the **Task history** tab, where each task of the target is listed. For each current task, the following information is provided:

- Description
- Status
- Scheduled date

- Start date
- Progress rate
- End date
- Download link to the log file
- Download link to the task file
- Download link to the `bom` file

Note: You must scroll to the right of the **Task history** tab to see all the fields.

The log file contains the target log. The task file contains all the parameters of the task. The `bom` file contains target-specific parameters for the given task.

The log file, the task file, and the `bom` file are needed by the development team to fix defects. Make sure that you download these three files if you suspect the presence of a defect in the software.

To cancel or destroy a task, select the task and select **Cancel target task** or **Destroy target task** in the contextual menu. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on running tasks, because the task destruction can cause strange behavior.

To view tasks scheduled for a later time, go to the **Tasks** page.

Monitoring deployment progress with the Tasks page for Linux

The **Tasks** page is also a useful source of information to monitor a deployment (and other tasks). You can also cancel tasks from there.

Go to **Server** → **Server history** → **Tasks**. .

The description field of each deployment in the **Tasks** page is headed by the keyword *Deploy* for easy retrieval. The information provided includes

Description
Is headed by specific keywords, indicating the type of task. *Deploy* is the keyword for deployment tasks.

Execution
Is the scheduled date and time for the execution of the task.

State Uses icons to represent if the task is pending, in progress, completed, and so on. If in doubt to the meaning of a state icon, browse over it to get a state name.

Progress
Indicates the rate of completion of the task as a percentage.

Expire Indicates when the task information is going to be removed from the page.

Tasks are expandable by clicking their **+** sign. An expanded task displays information about its targets. The target information fields are:

- IP address
- Hostname
- Start date and time of the task
- State
- Progress rate
- Status date

If, for any reason, you want to cancel a running or scheduled task, you can easily do so by following these steps:

1. Expand the task
2. Select the target for which you want to cancel the task
3. Select **Cancel task** from the contextual menu
4. It is also possible to *destroy* tasks. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on a running task, because its destruction can cause strange behavior. To permanently delete tasks:
   a. Select one or several tasks. To select multiple tasks, use the Shift key for a range of tasks and the Ctrl key for individual tasks.
   b. Select **Destroy task** from the contextual menu

Receiving an e-mail notification

To receive an e-mail notification at the end of a deployment, you must configure a TCP tunnel called *sendmail*.

To receive an e-mail notification at the end of a deployment, you must have configured a *sendmail* TCP tunnel.

Note: The OS deployment server supports only sendmail servers without authentication.

There are two options to configure a deployment to receive an e-mail notification:

- You can edit the deployment scheme used for deployment to include the notification information.
  1. Go to **Server** → **OS deployment** → **Task Templates**.
  2. Select **Deployment schemes** and double-click a specific deployment scheme name to edit it.
  3. Click **Edit** in the **General settings** section.
  4. Under **Send mail at end:**, select the type of notification that you want. You can choose among:
     – **No**
     – **Whatever the notification is**
     – **If deployment is successfully completed**
     – **If deployment failed**
  5. If you selected a notification, you must now enter a valid e-mail address to which the notification is sent, under **Send mail to:**.
- You can modify the settings of the deployment scheme in the deployment wizard. Step deploy/tosd_sendmail.dita#Receivinganemailnotification/first and possibly step deploy/tosd_sendmail.dita#Receivinganemailnotification/second are available.

Depending on your selection, you will receive an e-mail notification at the end of the deployment.

The notification e-mail looks like this:

The target with MAC *xxxx* / IP **xxx** has completed an activity *activity description*.

You can now deploy targets with the edited deployment scheme and receive e-mail notification at the end of the task.

Creating a sendmail TCP tunnel:

A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Note: OS deployment server supports only sendmail servers without authentication.

1. Go to **Server → Server parameters → Predefined channels**.
2. Click **New tunnel**.
3. In the TCP tunnel information screen enter,
   a. The name of the tunnel. In this case the name is `sendmail`.
   b. The host name or IP address of the target.
   c. The TCP port of the target.
4. Click **Save**.

You can now configure deployment schemes to send an e-mail notification at the end of a deployment.

Bindings created during deployment

The Target Monitor creates a binding between the OS configuration chosen for the deployment and the targets being deployed. This binding is added into the database and can be later removed using the Target Monitor.

Because at least one configuration binding now exists, targets that have been deployed no longer show the locked screen. They show a boot menu with a list of the OS configurations that are bound to the target. This allows the target user to manually restart the deployment of an already deployed OS configuration by clicking on the corresponding line in the menu.

You can remove, add, or modify OS configurations and software bindings using the Target Monitor.

Chapter 5. Working with Linux on PowerPC and Cell targets

To work with Linux system profiles on PowerPC and Cell targets, you must take into account some specificities of these targets.

DHCP specificities

There are specific considerations for setting Dynamic Host Configuration Protocol (DHCP) options. Make sure you set them appropriately.

Note: Microsoft DHCP server does not work well with somePowerPC firmware. Use IBM recommended DHCP servers.

Registering new targets

You must add targets manually into the Target Monitor or import a comma-separated text file containing a list of targets to be added.

PReP boot and /boot partitions

The PReP boot partition is mandatory to deploy Linux on PowerPC. It must be the first partition of the disk. If your profile contains a /boot partition, this partition must be the second partition. Set both partitions to a fixed size in MB and not in percentage of the total disk size.

System profiles

A system profile is the partition layout and list of files to deploy an operating system, either by unattended setup or by cloning, from a reference target or from a reference image.

The main purpose of Tivoli Provisioning Manager for OS Deployment is to deploy an operating system on targets by replicating a reference system. However, unattended installation of operating systems is also possible. The latter case Tivoli Provisioning Manager for OS Deployment does not replicate a reference system, but merely provides the correct parameters to Windows or UNIX setup for a fully unattended installation.

There are a number of differences between an unattended installation and disk cloning. First, creating an unattended installation in Tivoli Provisioning Manager for OS Deployment is straightforward. All of the necessary tasks are performed on the server, using the Web interface. In contrast, a cloning-mode system profile requires you to configure a target, prepare it for cloning, and run the cloning process directly on the target. However, the native mode of operation of the product is centered around cloning-mode system profiles, because this method of deployment is faster than unattended installation. When deploying computers on a large scale, unattended installation is not possible. Novice users might start with creating unattended installation profiles because this is easier than cloning-mode profiles.

Note:

© Copyright IBM Corp. 2006, 2010

- To avoid failures in creating or deploying asystem profile, clean up the temporary directory inside the OS deployment server installation directory on a regular basis.
- To create or deploy a system profile from a physical or virtual machine at least 512 MB RAM is required.
- The exFAT filesystem is not supported.
- System profiles can have a maximum of 3 primary partitions. Therefore, you cannot clone a target with 4 primary partitions.

Creating an unattended setup system profile for Linux on PowerPC

SUSE

SuSE Linux Enterprise Server (SLES) 10 unattended system profiles for PowerPC must be created on a Linux target, running the web interface extension. It can either be the OS deployment server itself, or a remote target which IP address must be entered in the profile wizard.

You can install operating systems using standard installation processes in unattended mode.

To create a new system profile:

1. Go to **Server** → **OS deployment** → **System profiles**.
2. Click **New Profile**. A system profile wizard opens to guide you through the steps of creating a profile.
3. Select **Unattended setup** in the first pane of the profile wizard.
4. Select your operating system from the list and click **Next**.
5. Follow the instruction of the profile wizard.

When your first unattended installation profile is created, you can use it to deploy computers. The profile wizard for a Linux unattended installation helps you to create a partition layout for this profile. Mandatory partitions are:

- **PReP Boot**
- **Boot**
- **Swap**
- **Root**

The **PReP Boot** partition has a size of 256 MB, the **Boot** partition of 100 MB. **Swap** and **Root** partition sizes are editable. The suggested settings in the profile wizard should be kept if there is any doubt in the allocation of disk space.

Organizing and editing system profiles

After you have created a system profile, you can view it on the OS deployment server through the web interface. The profiles are listed on **> Server > OS deployment > Profiles** page, in the **System profiles** pane. Each blue jacket represents a system profile (that is, the hard-disk partition images).

If you want to organize your system profiles, you can create subfolders by following these steps:

1. Select the parent folder with a left mouse click.
2. Call the contextual menu with a right mouse click.

3. Select the **Add a new profile folder** menu item.
4. Enter the new folder name.
5. Click **OK**.

You can then move profiles (by dragging-and-dropping the profile icons) from the top folder, where they are automatically created, to the appropriate subfolder.

Editing system profiles

To display and edit the parameters associated with a given profile:

1. Double-click a system profile to open the **Profile details** page.
2. Click **Edit** on top of the parameter sections to edit the parameters.

Changing the partition layout

Partition layout can be updated to resize partitions, assign mount points, change the file system.

Changing the partition layout in system profiles might render the profile unusable. It is recommended not to change the partition layout in system profiles, unless you know that the changes you want to make have no side effect.

In any case, do not:

- Add or remove a partition before the *root* partition.
- Transform a primary partition into a logical partition.
- Move a primary partition before *boot* or *root* to LVM.

Note: Changing the partition layout from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose either one or the other entry point, and then perform all your changes from that entry point.

Editing the partition layout allows you to:

- Add or delete partitions.

  Note: Adding or deleting partitions can lead to OS configuration problems, therefore this feature must only be used very carefully. To provide a better description to your profile, use the **Comment** field to write all necessary details.
- Resize a partition by dragging sliders, or by assigning it an absolute or relative size.
- Change the file system of a partition.
- Assign a mount point to the partition.

1. Click **Edit partition layout** on either the **Profile details** page or the **OS configuration details** page, **Disks** tab.
2.
   - To add a partition:
     a. Click **Modify partition layout**.
     b. Click into an existing partition.
     c. Click **Add a partition** in the contextual menu.
     d. Indicate the partition properties, including a mount point and click **OK**.

        Windows In a Windows profile, if you do not provide a drive letter as mount point, one is assigned randomly. Letter *W* is not available.

Linux In a Linux profile, do not forget to assign a mount point for the new partition. To be valid, this mount point must reference an existing directory in the main image.

- To resize partitions with the sliders, grab the slider to the right of the partition and drag it.
- To update all other parameters, select a partition by clicking on it, and select **Edit partition** in the contextual menu.

Modified partitions are aligned on megabytes rather than on cylinders. The following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

If you want to use the same system profile with two different partition schemes, you can also duplicate a system profile by right-clicking the profile name and selecting **Duplicate profile**. The copy shares the same image files, but can have a different partition layout.

Updating device mapping

Device mapping can be updated to force logical disks to point to specific physical devices.

Note: Updating device mapping from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose one or the other entry point, and then perform all your changes from that entry point.

1. Go to **Server** → **OS deployment** → **System profiles** → **Profile details** or to **Server** → **OS deployment** → **System profiles** → **Profile details** → **OS configuration details**, **Disks** tab.
2. Click **Modify device mapping**.
3. Select to which physical device you want to map your logical disk. The column starting with **Disk 0** corresponds to an automatic detection of the first to the eighth disk, the column starting with **/dev/hda** corresponds to standard device names.

   Note: Spanning a logical disk on several physical disks is not currently available.
4. Click **OK**.
5. Repeat step 2 to step 4 for each logical disk for which you want to update device mapping.

If the new device mapping you selected is incorrect, you receive a warning message.

OS configurations and fixed common parameters

A system profile is the partition layout and list of files to deploy, while OS configurations are operating system parameters.

At the very bottom of the **Profile details** page, there is a list of the OS configurations that correspond to your profile.

You can define several OS configurations for each system profile and duplicate them. These copies share the same image files, and the same partition layout, but can have different target parameters. You must then assign new values to some of the OS configurations parameters to make the original OS configuration and its copies distinct.

If you want to automate the assignment of parameters to targets, you can view and edit the OS configuration you are about to deploy by clicking on its name in the **Profile details** page. You are now in the **OS configuration details** page. The information is divided into panels, each displaying sets of parameters. You can modify the parameters either through the web interface or by using a parameter file.

Editing OS configuration parameters in the web interface for Linux on PowerPC

The web interface displays a number of OS configuration parameters divided into panes. These parameters can be edited in the web interface.

To edit parameters:

1. Click a tab to select the corresponding pane.
2. Click **Edit** on the banner of the section where you want to update parameters.
3. Modify the values.
4. Click **OK**.

Editing a Linux parameter file for Linux on PowerPC

Note: Since version 7.1.1 of the product, information about partitions in the custom configuration file is not normally taken into account.

SUSE For partitioning information in the custom configuration file to be taken into account, and to replace any information in the default file, these conditions must be fulfilled:

- The version of the product must be 7.1.1.3 or higher
- The deployment must be performed by HTTP
- The system profile must be of type *unattended setup*
- The operating system being deployed must be SuSE

Information in **Common networking info** is overwritten by the information in the custom configuration file. However, information in the **Advanced network settings** are not because they are applied in a post-configuration stage.

1.
   - Red Hat Click **Edit custom 'ks.cfg'** to edit the file.
   - SUSE Click **Edit custom 'autoinst.xml'** to edit the file.
2. Type the parameters and their values in the syntax requested by the operating system, or copy and paste it from another editor.
3. Click **OK**.

Tivoli Provisioning Manager for OS Deployment merges the information of the edited file with the information provided on the web interface (default file). The resulting configuration is the union of the values in the custom and default files, with the following restrictions:

- The result of conflicting values between the custom and default files is undefined.
- Partition information in the custom file is taken into account only for SuSE unattended setup by HTTP, in which case only the information in the custom file is taken into account.
- Advanced network settings are always applied, because they are performed at a later stage.

SUSE Here is an example of a `autoinst.xml` file which adds a new user during setup.

```
<profile xmlns="http://www.suse.com/1.0/yast2ns"
         xmlns:config="http://www.suse.com/1.0/configns">
 <users config:type="list">
   <user>
     <username>jdoe</username>
     <user_password>tOpsEcreT</user_password>
     <encrypted config:type="boolean">false</encrypted>
     <forename>John</forename>
     <surname>Doe</surname>
   </user>
 </users>
</profile>
```

Do not omit the `xmlns` and `xmlns:config` attributes of the `profile` tag.

Troubleshooting:

If the OS configurations in the deployed operating system are not what you expected, you must examine the parameter files carefully. They are the result of the merge between the custom file and the default file created.

Red Hat To troubleshoot OS configuration parameters after a failed deployment, there are two options:

- Without rebooting the target:
  1. Type `Alt+F2` on the target. This opens a shell.
  2. In the opened shell, view the file `/tmp/anaconda.log`.
- You must look for `ks.cfg` at the root of the partition labeled `rembo`. The file contains the information merged from the custom and the default files.

SUSE To troubleshoot OS configuration parameters after a failed deployment, there are two options:

- Without rebooting the target:
  1. Type `Alt+F2` on the target. This opens a shell.
  2. In the opened shell, view the file`/var/log/YaST2/y2log` .
- You must look for `autoinst.xml` at the root of the partition labelled `rembo`. The file contains the information merged from the custom and the default files.

Software modules for Linux operating systems on PowerPC

Software modules are images other than system profiles that can be created to address various needs.

Tivoli Provisioning Manager for OS Deployment is based on imaging technology. As administrator, you create images of components that you want to see on every target, and the automated deployment merges and restores these images on each target, automatically, when needed.

Tivoli Provisioning Manager for OS Deployment can handle most scenarios for software deployment and post-installation configuration.

Types of software modules

There are many types of software modules. Depending on the type of package and installation files, the wizard guides you through the different steps to achieve your software module with minimal effort. The types of software package supported by the wizard are listed in this section.

- **A Linux application installation, using RPM**
- **A custom action on the target computer**. This includes OS configuration changes such as commands to be run, and copying sets of files on the target.

Creating software modules

There are distinct types of software modules which vary according to the operating system being deployed. The software wizard guides you through the creation of software modules for each type.

Creating software modules with RPM for Linux operating systems

Using RPM is current for Linux software installation.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select **Linux** and click **Next**.
4. Select **A Linux application installation, using RPM** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the software module at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The path to where the installation files are restored on the target. This path is relative to the system root partition.
   - An additional command line that might be necessary to install your software module. When possible, the wizard automatically suggests the appropriate command line to run the installation unattended. However, you might need to add some additional parameters to the command.

     For example, you can specify an hour parameter to cancel an activity, if the activity does not complete before the end of the specified time. The

parameter syntax format is <=xh, where x is an integer representing the number of hours after which the activity is canceled. In the following example you can specify to cancel an application installation if the installation process has not completed after one hour, by adding <=1h at the end of the command line:

```
install.exe /sPB /rs /rps /l <=1h
```

- The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

Creating a custom action software module for Linux operating systems

Software modules can also contain custom actions to be performed on the target.

They are divided into:

- An OS configuration change to perform on the target
- A set of files to copy on the target

Configuration changes are further subdivided. Depending on the operating system, you can:

- Copy a single text file
- Run a single command file, this can be a batch file or a vb script file.
- Boot a virtual floppy disk

In the OS configuration change wizard screen, you can select **Activate keyword substitutions**. If you use this option, you can specify which keywords must be substituted in the software module details.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select the operating system and click **Next**.
4. Select **A custom action on the target** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when your software module must be installed: when the OS is installed, or after one or more additional reboot. Most of the time, you must install the software module at the same time as the operating system. However, you can decide to install them in a specified order to avoid software-specific conflicts.
   - A file name to store your image on the OS deployment server. Software modules typically have a .pkg extension.

- The path to where the installation files are restored on the target. This path is relative to the system root partition.
- An additional command line that might be necessary to install your software module. When possible, the wizard automatically suggests the appropriate command line to run the installation unattended. However, you might need to add some additional parameters to the command.

  For example, you can specify an hour parameter to cancel an activity, if the activity does not complete before the end of the specified time. The parameter syntax format is `<=xh`, where `x` is an integer representing the number of hours after which the activity is canceled. In the following example you can specify to cancel an application installation if the installation process has not completed after one hour, by adding `<=1h` at the end of the command line:

  ```
  install.exe /sPB /rs /rps /l <=1h
  ```
- The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

Repeating custom actions:

Some commands must be run every time the target boots during a deployment.

This is typically the case if you want to repeatedly connect a network share. This connection is destroyed when rebooting. You can therefore create a single software module with a `netuse` command to set the network share and set this software module to run once after each reboot, starting at a specific reboot.

This option is available for executing a single command.

1. Create your software module.
2. Double-click on the software module name in the **Software components** page to obtain the **Software details** page
3. Click **Edit** in the title of the **Package information** section.
4. Select the installation stage at which the software module must be applied first.
5. Select **Run at each software pass until end of deployment** and click **OK**.

Editing software modules

You can edit the basic parameters of a software module, upload new files into your software module, and update drivers.

1. Go to **Server** → **OS deployment** → **Software modules**. Double-click on a software module to view the details.
2. From **Software details** page, use the links and buttons. Links vary according to the type of software module. Not all the links listed are necessary available.
   - To edit the base parameters of a software module, click **Edit** at the top of the **Software module information** section.
   - To update files or add new files into the software module, click **Edit software module files**, or a link with a similar name, and select **Upload file** from the contextual menu.

     Note: File upload is limited to 16 MB.

Keeping command lines confidential

When you use command lines in your software modules, their call and their output are stored in deployment logs. In some circumstances, for example when the command line includes a password or a product key, it might be necessary to keep the information contained in the command line confidential. Three levels of confidentiality are available.

No confidentiality
The command line is visible in the web interface and on the target during the installation, its call is logged, and its output is also logged.

The command line call is not logged
The command line is visible in the web interface, and its output is logged, but the command line call, containing the whole command line string with all parameters, is visible in the logs neither on the web interface nor on the target.

To apply this level of confidentiality, you must prefix the command line by one exclamation mark (`!`).

The command line call and output are not logged
The command line is visible in the web interface, but its call and output are visible in the logs neither on the web interface nor on the target.

To apply this level of confidentiality, you must prefix the command line by two exclamation marks (`!!`).

To keep command lines confidential:

- Enter the appropriate number of exclamation points in front of the command in the Software Wizard when first creating the software module.
- Edit the software module information
  1. Go to **Server** → **OS deployment** → **Software modules**. Double-click on a software module to view the details.
  2. Click **Edit** in the Software module information banner.
  3. Update the command line with the appropriate number of exclamation points.
  4. Click **OK**.

Keyword substitution

You can usefully use keyword which act as variables and are substituted with their values during deployments. Keywords can either refer database values or server specific values, given by the user.

Syntax

Variable substitution expressions follow the syntax given here. They start with the character { and end on the same line with }. Words between these two characters are interpreted by using one of the following schemes:

- *{$expr$}* the expression is replaced with the string resulting of the evaluation of `expr`.
- *{/expr/ab}* the expression is replaced with the string resulting of the evaluation of `expr`, but each occurrence of the character `"a"` is replaced by the character `"b"` (character-based substitution).

- *{=expr=test content=this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is equal to the text `"test content"`.
- *{!expr!test content!this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is not equal to the text `"test content"`.

Note: If a variable does not exist (for example, it contains a typing error or it is not described in `server.ini`) but it is used in a command, its value is supposed to be empty which can result in deployment errors.

Database keywords

Within an expression, database records can be referred to. Within a record, each field can be accessed using the standard C notation (`record.fieldname`). The exhaustive list of these fields can be obtained from the database records, with the following correspondences between variable and database record names:

Table 6. Records for `free-text conditions`

Variable record name	Database record name
Disk	DiskInventory
DMI	DMIInventory
Order	BOM
User	UserProfile
System	SystemProfile
PCI	PCIInventory

Below are a few examples of available fields:

- `Order.IP`: a string, the target IP address, such as `192.168.1.2`
- `Order.MAC`: a string, the target MAC address, such as `00:01:02:03:04:05`
- `Order.SN`: a string, the target Serial Number, such as `CH12345678`
- `Order.Model`: a string, the computer model name, such as `e-Vectra`
- `User.UserCateg0`: a string, without any restriction, such as`technicians`
- `DMI.Vendor`: a string, the vendor name, such as `Hewlett-Packard`
- `DMI.Product`: a string, same as `Order.Model`
- `DMI.ProcModel`: a string, the processor model
- `Disk[0].Type`: a string, the disk 0 drive type, such as `ATAPI`
- `Disk[0].Media`: a string, the disk 0 media type, such as `Disk` or `CD`
- `Disk[0].DiskSize`: a number, the physical size of the disk (if detected)
- `PCI[0].VendorID`: a string, the hexadecimal vendor ID of the device
- `PCI[0].DeviceID`: a string, the hexadecimal device ID of the device

For disks and PCI devices, you can use the function `sizeof` (`sizeof(Disk)` and `sizeof(PCI)`) to discover the number of devices present. You can then use indexes to access these devices.

As an example for keyword substitution, if `BomID` has `OrgName Rembo SaRL`, `RemboServer 192.168.168.16`, and `IP 192.168.168.32` for value `1`, the following text

```
BomID:{$Order.BomID$}
OrgName:{$User.OrgName$}/{$StrToLower(User.OrgName)$}
RemboServer:{$Order.RemboServer$}
IP:{$Order.IP$}
```

gives the following results after keywords are substituted (note the use of a Rembo-C function within the expression to be substituted):

```
BomID:1
OrgName:Rembo SaRl/rembo sarl
RemboServer:192.168.168.16
IP:192.168.168.32
```

Server specific keywords

If you want to set up server specific keywords, which are defined exclusively by the user and per server, you must edit `Files/global/rad/server.ini`.

Start the file with `[Custom]` and add a line per keyword, in the format **keyword=value**, where `keyword` is a word of your choice and value the `value` you want to give it.

To use the keyword in a command, type `Server.keyword` and activate keyword substitution when creating the software module.

Note: `server.ini` is not replicated between servers. If you use multiple servers, you must edit `server.ini` on each server.

Customizing the software page

You can view the software modules in a tree viewer or in a list viewer. The list viewer allows you to customize the visible information.

You must have created at least one software module, otherwise there is nothing to view.

To customize the visible information

1. Go to **Server** → **OS deployment** → **Software modules**. Then click **List view**.
2. From the list view, you can
   - Drag the column separator in the column heading to resize the column.
   - Click on the triangular arrow to the left of the column name to sort the software modules by column criteria.
   - Click on the arrow on the right of the column name and select an option to filter the information. Filtering on several columns is cumulative.
3. For more options, right click anywhere to open the contextual menu and select **Arrange columns**.
   - Select the columns you want to see and clear the others.
   - Click on the minus or plus icons to decrease or increase the size of a column.
   - Select a column and use the up and down arrows to move the column relatively to the others.

   Click **OK** to save your changes. The updated version of the list view is visible in the **Software modules** page.

To return to the tree view, click **Tree view**. You can also access the details of the software modules by double-clicking on a software module name, from either view.

OS configuration and software bindings

OS configuration bindings determine which configurations are available to a target when booting the target on the network, while software bindings correspond to the list of software modules currently assigned to the target.

OS configuration and software bindings are created when:

- The Target Monitor has been used to manually modify OS configuration and software bindings for the target
- A deployment has been started with the Target Monitor. In this case, an OS configuration binding is added for the corresponding OS configuration.
- Automatic binding rules are configured in the **Details** page of OS configurations or software modules. Some of these rules have matching values for the specified criteria. These bindings cannot be modified, except by modifying the rules.

With the Target Monitor, you can browse, remove or add OS configuration and software bindings to any target present in the database. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.

Binding software modules and OS configurations to targets

Bindings link software modules and OS configurations to targets to enable automatic deployment. When binding to targets, you explicitly provide the list of software modules and OS configurations to bind to your target.

To explicitly bind a software module or a OS configurations to a target, there are two methods:

- From the **Target Monitor** page
- From the **Target details** page

If you want to bind software modules or OS configurations to a group of targets, you must do it through the Target Monitor.

From the Target Monitor:

1. Select a target or a group of targets
2. Select **Bind software** or **Bind OS configurations** from the contextual menu
3. Select the items to bind from the popup window
4. Click **OK**

From the Target details page:

1. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.
2. Go to the **Bindings** panel.
3. Click **Edit** in the relevant section to add explicit bindings for OS configurations and software modules.
4. Select the items for which you want to add explicit bindings.
5. Click **OK**

You can also clear items to remove their explicit bindings. To remove a binding by rule, you must modify the rule.

Binding software modules to computer models and PCI devices

You can link software modules to target computer hardware model to enable automatic deployment. When creating a binding, you explicitly provide the list of software modules to link to a computer model and its PCI devices.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Select **View bindings** from the contextual menu. A grid is displayed representing computer model-based software bindings:
   - The rows correspond to your software modules
   - The columns correspond to the computer model. If you expand it, you can see all the corresponding PCI devices discovered in your hardware inventory.

   The background color indicates implicit bindings by existing generic rules, while checkmarks correspond to explicit one to one bindings.
3. Select a grid position corresponding to the software module and device to bind.
4. Select **Bind** from the contextual menu.

You can also remove a binding by selecting it and clicking **Unbind** from the contextual menu.

Binding software modules to a deployment scheme

Software modules can be bound to deployment schemes.

Take a company with offices in three locations: New York, Quebec City, and Mexico City. In each of these locations, the company has people in human resources, sales, logistics, and product development. For the sake of simplicity, consider further that all the employees use either one of two types of computers: a desktop, or a notebook. All desktop computers are identical (with the same network card, system board, disks, and so on) and the same applies for all notebooks.

In this scenario, the company needs two profiles, one with the image for notebooks and one with the image for desktop computers. Three configurations per profile (six in total) are necessary to integrate the different parameters of the different locations, in particular language and time zone information. Finally, schemes are set according to the employees' department, with software modules specific to the different departments bound directly to the deployment schemes.

1. Go to **Server** → **OS deployment** → **Task templates** Select the **Deployment Schemes** folder. Double-click on a deployment scheme to view its details.
2. Click **Edit** on the **Software bindings** section of the page to open the dialog to bind software modules to schemes.
3. Select which software modules you want to bind to your deployment scheme, in addition to software modules that can have been bound to targets.
4. *(Optional)* If you want to use only the software checked in the window when deploying with this scheme, select the **Discard all other software binding rules** check box.

Automatic binding rules

Automatic binding rules are used to create bindings between OS configurations and targets, or software modules and targets, without having to specifically bind a OS configuration or a software module on each target.

Rules are created in OS configurations and software modules to determine which targets are automatically bound to the OS configuration or software module.

Rules are made of criteria and values. If a target has a matching value for all criteria in the rule, the OS configuration or software module will be bound to that target. The binding will be displayed with the mention **by rule** in the OS configuration panel of the target properties for targets that match the criteria. For

example, if the criteria is the model name, and the value is `Optiplex`, targets with a model name starting with `Optiplex` will be bound to the object where the rule has been defined.

Automatic binding rules are defined in Tivoli Provisioning Manager for OS deployment at the bottom of the **OS configuration details** or **Software details** page.

To create a new binding rule, click **New rule** located at the bottom of the Web interface:

1. The dialog displayed to create a new binding rule is different depending on whether you are adding a rule to an OS configuration or to a software module. When adding a binding rule to a software module, you can set values for the following criteria:
   - A deployment scheme
   - A system profile
   - A current OS configuration
   - Administrative group
   - One of the system-definable and user-definable fields of the database (only used if you have customized the database)
   - An operating system type, such as Windows 2000
   - An operating system version, such as SP2
   - An operating system language
   - An operating system architecture, such as x86-32
   - A computer model name
   - A BIOS version
   - A PCI device
   - A base board
   - MultiChassi
   - HAL Type
   - A free-text condition in `Rembo-C;` syntax

   For example, to create a binding based on the operating system type between a software module and targets, you must create a new rule, click **OS type**, and select the operating system version that you want to limit this software module to.
2. When adding a binding rule to an OS configuration, you can set a condition on the deployment scheme, and on the computer model name. The next ten fields are only used if you have customized your database and want to match specific user categories.
3. Finally, you can enter a free-text condition following the `Rembo-C;` syntax. They must only be used by advanced users.

   The conditions determine the applicability of the rule and evaluate to `true` or `false`. A condition must be formed using the variables also used for keyword substitutions in software modules, combined with Java-like logical operators, listed by order of priority in the table:

Table 7. Logical operators for free-text conditions

Operator	Meaning
<	smaller than
<=	smaller than or equal to

Table 7. Logical operators for free-text conditions (continued)

Operator	Meaning
=>	greater than or equal to
>	greater than
==	equal to
!=	not equal to
&&	AND operator
||	OR operator

For example, a typical condition can be:

```
Disk[0].DiskSize > 10*1024*1024
```

Note: If a condition cannot be evaluated, it is considered to have the value `false`.

Scheduling the application of software modules

It is not possible to schedule the application of software modules for this operating system and hardware.

All software modules are applied **When the OS is installed**, regardless of the set stage in the **Software application order** window. Reboots are not handled either.

Task templates

Task templates group together elements that can be customized on a target. These elements are mostly screen layouts, which condition the appearance of the target screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment.

A deployment scheme is a specific type of task template. Together with the target display screen layout, it contains other parameters for customizing a deployment on a target.

Note: Starting with Fix Pack, version 5.1.0.2, deployment schemes are considered to be a subset of task templates. The functions of deployment schemes have not been altered. To access deployment schemes, go to the **Task template** page, and select the **deployment scheme** folder.

The task template page of the web interface contains a task template tree in the left pane with seven folders. The content of the selected folder is displayed in the right pane.

There are seven task template folders in the tree. They are described here.

Deployment Schemes
: Deployment schemes contain parameters that indicate how an OS configuration must be deployed on your target. The **deployment Schemes** folder contains at least the **Default** scheme.

Idle layout
: The idle layout defines what is shown on the target when there is no pending task. The **Idle Layout** folder contains at least the **Idle state** layout.

Menu Layout
The menu layout defines how deployment menus are shown to the users. Menus are used when an OS configuration and on deployment CDs. The **Menu Layout** folder contains at least the **Menu** layout.

OS Detection Layout
The operation system detection layout defines the target display when a target is busy detecting the currently installed operating system. It is used when creating a cloning profile from the web interface. The **OS Detection Layout** folder contains at least the **Detect operating system** layout.

Profile Creation Layout
The system profile creation layout defines the target display when a target is busy creating a new system profile. It is used when creating a cloning profile from the web interface. The **Profile Creation Layout** folder contains at least the **Creating cloning profile** layout.

Profile Restoration Layout
The system profile restoration layout defines the target display during the manual restoration of a system profile by the administrator.

Note: A system profile restoration is always performed as-is and must not be confused with an automated deployment resulting in a fully configured operating system installation.
The **Profile Restoration Layout** folder contains at least the **Default OS Restoration** layout.

State Capture Layout
The state capture layout defines the target display when a target is saving the operating system state for future redeployments.

State Restoration Layout
The state restoration layout defines the target display when a target is redeploying an operating system from a saved state.

When a task template is selected in the right hand pane, the bottom of the web interface contains a link to **Customize GUI**. Follow this link to modify the look of your target screen . When the selected task template is a deployment scheme, there are additional links to view and edit the current scheme.

Creating and editing deployment schemes

By customizing your deployment schemes, you can adapt the way in which your predefined OS configurations are installed onto targets.

Deployment schemes are not linked to architecture of the target or the operating system. Therefore, the deployment scheme wizard always offers to set all modifiable parameters. When deploying, parameters incompatible with either the architecture of the target or the operating system being deployed are not taken into account.

1. The easiest way to create a new deployment scheme is to run the deployment scheme wizard, by clicking **New deployment scheme** from the **Task templates** page.
2. Alternatively, you can modify an existing scheme by editing its parameters. To do this, select a scheme and click **View deployment parameters** and then use **Edit** in the banner on top of each parameter section.
3. If you prefer using a wizard to edit your scheme, click **Edit parameters using a wizard**.

The following parameters apply for simple one-time deployments and for redeployment operations.

Description

The first step is to enter a name for this deployment scheme. Make it explicit enough so that you can pick it easily when starting a deployment (the web interface does not show the settings in a deployment scheme, therefore the choice must be made by name only). Because deployment schemes determine how the computers are installed (and not what is being deployed), use a description such as `Multicast 50 targets` or `On-site` rather than the name of a OS configuration or of a group of computers.

When the deployment starts

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

You must determine if Tivoli Provisioning Manager for OS Deployment requires user interaction during deployment (to edit individualtarget parameters) or runs completely unattended.Select:

Always edit target-specific parameters
to have the opportunity to change the target parameters at every deployment. The parameter edition can be made either directly on the target computer or by double-clicking the target icon in the Target Monitor.

Edit parameters for targets that are not yet in the database
to be prompted only during the first installation of each target. Subsequent deployments for the same targets run unattended. This is the default choice.

Never edit parameters
to have unattended deployments if all BOMs have been previously entered in the database. Any missing entry leads to a red banner on the target and cancels the deployment for this target.

You must also select how Tivoli Provisioning Manager for OS Deployment behaves when the model of the computer being deployed does not match the model of the computer on which the image was created. This feature requires DMI for hardware detection.Select:

No if you know that all your system profiles are fully hardware-independent or for deploying universal images.

Yes, display a warning
if you want to see all possible OS configurations for a computer, but want to avoid mistakes. This choice can require user interaction and is therefore not appropriate for a fully unattended deployment.

Yes, abort the deployment
if you want to prevent anyone from using an OS configuration on a computer different from the one for which it was designed.

Use 'BIOS fallback MBR' to start PXE is used when PXE activation (the process of enabling PXE when booting on the hard-disk) does not work.

The PXE boot code manages the multiple reboots needed to install a computer. To manage these reboots, the PXE boot code must intercept the boot process of the computer at every boot.

- If the computer is configured to always start on the network (LAN device first in the list of boot devices), there is nothing to do, becauseTivoli Provisioning Manager for OS Deployment is loaded into memory at every boot.
- If the computer is configured to start on the hard-disk, you can change the MBR of the hard-disk and make it point to the work partition at the end of the hard-disk. Tivoli Provisioning Manager for OS Deployment is then loaded from the hard-disk when the computer starts up, instead of loading the operating system. The disadvantage of this method is that, because the computer did not use the network card to boot, PXE is not available. To enable network access, PXE is activated with a special function in the PXE card that makes it behave as though the computer had booted on the LAN. However, this is not documented in PXE, and does not work on every network card. If the network does not support this, an error is raised, and access to the OS deployment server fails (the message **Network started**, followed by an error).

  When PXE activation does not work, you can write a special MBR telling the BIOS that the hard-disk is not a valid boot device. By default, the BIOS falls back to the next device in the list, which in most computers is the network. As a result, the computer boots on the network and has full access to the network. This is the purpose of the **Use 'BIOS fallback MBR' to start PXE** check box.

Data collection

Note: These parameters have no effect on PowerPC and Cell targets.

By default, Tivoli Provisioning Manager for OS Deployment automatically populates the database with an inventory of the hardware setups of all deployed computers. For Windows, a software inventory can also be populated based on the registry. If you are not interested in using those inventories, or if your computers do not comply with any of the hardware detection standards, you can disable these features. Be advised that running the hardware or software inventory on thousands of computers can produce a huge database. This inventory is performed on locked screen.

Tivoli Provisioning Manager for OS Deployment centrally reports the status of the deployment of target computers to the OS deployment server and to the server database. Additionally, if you want to keep the deployment logs and the list of software modules on each deployed computer, you can specify a local path where the log files are to be stored. The path that you specify is relative to the root of the operating system on the target, for example, /. In the deployment scheme details, the label of this field is **Save deployment log to**.

When the deployment is completed

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

When the deployment process is finished, you can select if you want to:

- Turn off the computer automatically (if supported)

- Boot the operating system automatically (this value might not make sense with some values of the previous setting)
- Display a green banner and wait for a manual shutdown

Network usage

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

Note: Multicast is available only if

- The targets have an Intel x86 or x86-64 architecture
- Multicast is selected in the deployment scheme
- The subnet supports multicast traffic
- Multicast is not disabled in the boot options of the target
- The target is not a VMWare 3.0 guest.

Depending on the number of computers that you are deploying simultaneously on your infrastructure, you must select one of the following networking modes:Select:

Unicast
to deploytargets one by one, or if you cannot use multicast. When deploying several targets simultaneously in unicast, the deployment time increases dramatically with the number of targets, as the result of network saturation.

Multicast, without explicit synchronization
to use soft-synchronized multicast protocol. Using this protocol, every target independently starts downloading images at soon as it is ready, and continues with the deployment as soon as it has downloaded all required material. When two or more targets (using the same deployment scheme) are downloading files in parallel, they automatically share the same bandwidth. The fastest target has the priority for the choice of the next shared files to be sent by the server, but the slower targets can receive them if they need them. This is a scalable solution that allows for a rolling deployment scenario.

Multicast, with the following synchronization parameters
to use a classical replicated multicast method. This mode is adequate for installing computers in batches. Enter the replication parameters (for example, the number of targets to wait for before starting the download, and the maximum timeout before starting in any case). Tivoli Provisioning Manager for OS Deployment multicast protocol can accept new download targets even after the initial replication period is over, and integrate them seamlessly into the transfer.

Note: In the first stage of an OS deployment, there are two target synchronization stages. Therefore it might seem that the maximum timeout that was set before starting the deployment is doubled.

You can use a network share or Linux HTTP protocol on the server to download the files to the target computers, rather than downloading the whole image to the hard disk of the target. Using a network share

or Linux HTTP protocol provides a shorter operating system installation time. To use a network share or Linux HTTP protocol:

- Select **Download files with a network share or Linux HTTP when applicable** in the deployment scheme.
- Share the `files\global\partition` directory and provide at least read-only access to it.
- Enter the relevant server parameter. Go to **Server** → **Server parameters** → **Configuration**. (See "Network share module" on page 314). .

On-site deployment

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

If you are running a one-time deployment in a deployment center and do not want to use redeployment, leave the check box blank and click **Next**.

If you are running an on-site deployment, or if you plan to use redeployment, you can enable the advanced feature.

Redeployment parameters

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
Indicate if you want to keep the deployment image in a protected partition and the size of this partition.

Note: The following parameters cannot be modified using the wizard. You must edit your deployment scheme parameters.

Request user confirmation
This parameters allows you to ask for user confirmation before running a deployment.

Unbind OS configuration at the end

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the OS configuration from the target at the end of the deployment. This OS configuration is not proposed the next time the target boots and, if no other OS configuration is bound, the target presents a locked screen to the user.

Unbind software module at the end

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the software module at the end of the deployment. This software module is not proposed and installed the next time a deployment is performed.

Vista 2008 Windows 7 **Disable user interaction during deployment**

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

This parameter, located in the **General settings** section, is set to `Yes` by default. If you set this parameter to `No`, you can obtain a command prompt by pressing Shift-F10 on the target computer during a deployment to modify deployment files.

Note: This parameter is currently relevant forWindows Vista/2008/7deployments only.

Send mail at end
This option can be set only if a *sendmail* tunnel has been created.

Send mail to
This parameter is available only if **Send mail at end** is not set to **No**. Use this field to enter the e-mail address to which information must be sent at the end of the deployment.

Configure Network

Note: This parameter has no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **Network settings** section, is set to **Before software installation** by default. The network setting of the target are set before software modules are installed, enabling the product to use the network settings during the installation of the software modules. Set the parameter to **After software installation** if you want the network settings to be applied after software modules are installed, for example if you intend to physically move the target after deployment and want it to be configured with the network settings for its final location.

Disable cancel button
This parameter, located in the **Client Display** section, allows you to prevent users from aborting a deployment by removing the cancel button. Set the parameter to **Yes** if you want to remove the cancel button from the client display

You can use the newly created deployment scheme to deploy a system profile.

To delete a deployment scheme, select the scheme, then right-click it, and click **Delete**.

Deploying Linux on PowerPC

A deployment is the process of installing an operating system on a target, and configuring the operating system for a specific user.

Deployment requirements

To start a deployment on a target, several elements must be present in the database.

The following elements are required:

- A *deployment* scheme associated with the target to deploy. The deployment scheme determines how to deploy the operating system on the target. If there is no association between a deployment scheme and the target to deploy, Tivoli Provisioning Manager for OS Deployment automatically uses the **Default** deployment scheme.

- An operating system *configuration* that is used to select which operating system to install. If there is no OS configuration associated with the target to deploy, the deployment does not start.
- Optional *software modules* to install in addition to the operating system during the deployment process. If there is no software module associated with the target to deploy, the operating system image is deployed without modification.

The OS configuration and the software modules can be considered to be the *content* of the deployment, while the deployment scheme is the *how* of the deployment.

The database keeps information about associations (*bindings*) between targets and deployment schemes, between targets and OS configuration, and between targets and software modules. These bindings can be configured manually or with binding rules (for example, `deploy configuration windows XP on targets whose model name starts with Dell`).

The minimal binding required to start a deployment is an OS configuration. If no configuration is bound to a target, the deployment does not start. In practice, however, Tivoli Provisioning Manager for OS Deployment always asks for an OS configuration and deployment scheme when beginning a deployment.

Tools to start and configure deployments

Bindings between targets and deployment elements are necessary to perform a deployment. You can create and edit these bindings in the **OS configurations** panel of the **Target details** page.

The Target Monitor provides functions to prepare a deployment, start a deployment, follow the progress of a deployment, and organize targets.

Binding rules, used to create permanent implicit bindings between targets and deployment elements without having to explicitly create the binding for each target, are created using the web interface. OS configurations and software modules contain a specific section at the very bottom of the **Details** page for creating automatic binding rules.

Starting a one-time deployment

You start deployments in the web interface by indicating on which target or targets the deployment must occur.

To start a deployment:

1. Select a single target or multiple targets on the Target Monitor page. To select multiple targets or deployment, select an administrative group, a custom list, a subnet, or click on individual target names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
4. Select **Simple deployment** and click **Next**
5. Follow the deployment wizard instructions to select a deployment scheme, an OS configuration and optionally software modules, and to set up deployment options.

   Note: The Wake on LAN options are available only on Intel targets.

When the deployment is complete, the server either displays a green banner on the target, boots in the operating system, or powers the target off, depending on how the deployment scheme is configured.

After deployment, the following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Partitions are aligned on megabytes rather than on cylinders. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

Monitoring deployments

There are several ways available to monitor the deployment progress.

Monitoring deployment progress with theTarget Monitor

You can use the Target Monitor to monitor deployments remotely. Information is located on the **Target Monitor** page and on several tabs of the **Target details** page.

On the **Target Monitor** page, the target color changes during the deployment. When PXE is activated, targets are monitored on a regular basis. The color of the icon is updated as soon as the status changes. By pointing to the target icon, you can get a description of the target status.

Note: A successfully deployed computer can continue to have a yellow icon (indicating that the deployment is still in progress). This reflects a PXE activation problem. The computer, having booted on the hard disk, is not using the network to inform the OS deployment server of its status. To remedy this, select the **Use 'BIOS fallback MBR' to start PXE** check box in the deployment scheme wizard. This forces the computer to boot through the network first.

If the deployment scheme used is configured to collect inventory information about target hardware (which is the default), you can see information about target hardware in the **Inventory** panel of the **Target details** page for that target (double-click on the target to go to the details page).

At the end of the deployment, the target icon shows either a green screen (success), or a red screen (failure). The deployment logs stored on the OS deployment server provide information about the deployment process. They are particularly useful in case of deployment failure to track its cause. To access the logs, double-click the wanted target. This opens the **Target details** page. Select the **Logs** tab to display a list of logs. To view a specific log, click its description. To download it, click **download** immediately after the log description.

Note: Logs are text files with UTF-8 formatting. If you are using a Windows operating system, you can view log files adequately by opening them in Microsoft WordPad.

There is only one log file for each deployment. This log file contains information about the different stages of the deployment process, including reboots and information provided by the operating system being deployed.

If any log information needs to be propagated to the OS deployment server outside of any task, an *idle* log file is created to store this information. The idle log file is created on demand and does not therefore exist for all the targets.

Another place of interest for information about a current deployment or another current task is the **Task history** tab, where each task of the target is listed. For each current task, the following information is provided:

- Description
- Status
- Scheduled date
- Start date
- Progress rate
- End date
- Download link to the log file
- Download link to the task file
- Download link to the `bom` file

Note: You must scroll to the right of the **Task history** tab to see all the fields.

The log file contains the target log. The task file contains all the parameters of the task. The `bom` file contains target-specific parameters for the given task.

The log file, the task file, and the `bom` file are needed by the development team to fix defects. Make sure that you download these three files if you suspect the presence of a defect in the software.

To cancel or destroy a task, select the task and select **Cancel target task** or **Destroy target task** in the contextual menu. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on running tasks, because the task destruction can cause strange behavior.

To view tasks scheduled for a later time, go to the **Tasks** page.

Monitoring deployment progress with the Tasks page for Linux on PowerPC

The **Tasks** page is also a useful source of information to monitor a deployment (and other tasks). You can also cancel tasks from there.

Go to **Server** → **Server history** → **Tasks**. .

The description field of each deployment in the **Tasks** page is headed by the keyword *Deploy* for easy retrieval. The information provided includes

Description
: Is headed by specific keywords, indicating the type of task. *Deploy* is the keyword for deployment tasks.

Execution
: Is the scheduled date and time for the execution of the task.

State
: Uses icons to represent if the task is pending, in progress, completed, and so on. If in doubt to the meaning of a state icon, browse over it to get a state name.

Progress
: Indicates the rate of completion of the task as a percentage.

Expire
: Indicates when the task information is going to be removed from the page.

Tasks are expandable by clicking their + sign. An expanded task displays information about its targets. The target information fields are:

- IP address
- Hostname
- Start date and time of the task
- State
- Progress rate
- Status date

If, for any reason, you want to cancel a running or scheduled task, you can easily do so by following these steps:

1. Expand the task
2. Select the target for which you want to cancel the task
3. Select **Cancel task** from the contextual menu
4. It is also possible to *destroy* tasks. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on a running task, because its destruction can cause strange behavior. To permanently delete tasks:
   a. Select one or several tasks. To select multiple tasks, use the Shift key for a range of tasks and the Ctrl key for individual tasks.
   b. Select **Destroy task** from the contextual menu

Receiving an e-mail notification

To receive an e-mail notification at the end of a deployment, you must configure a TCP tunnel called *sendmail*.

To receive an e-mail notification at the end of a deployment, you must have configured a *sendmail* TCP tunnel.

Note: The OS deployment server supports only sendmail servers without authentication.

There are two options to configure a deployment to receive an e-mail notification:

- You can edit the deployment scheme used for deployment to include the notification information.
  1. Go to **Server → OS deployment → Task Templates**.
  2. Select **Deployment schemes** and double-click a specific deployment scheme name to edit it.
  3. Click **Edit** in the **General settings** section.
  4. Under **Send mail at end:**, select the type of notification that you want. You can choose among:
     – **No**
     – **Whatever the notification is**
     – **If deployment is successfully completed**
     – **If deployment failed**
  5. If you selected a notification, you must now enter a valid e-mail address to which the notification is sent, under **Send mail to:**.

- You can modify the settings of the deployment scheme in the deployment wizard. Step deploy/tosd_sendmail.dita#Receivinganemailnotification/first and possibly step deploy/tosd_sendmail.dita#Receivinganemailnotification/second are available.

Depending on your selection, you will receive an e-mail notification at the end of the deployment.

The notification e-mail looks like this:

The target with MAC *xxxx* / IP **xxx** has completed an activity *activity description*.

You can now deploy targets with the edited deployment scheme and receive e-mail notification at the end of the task.

Creating a sendmail TCP tunnel:

A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Note: OS deployment server supports only sendmail servers without authentication.

1. Go to **Server** → **Server parameters** → **Predefined channels**.
2. Click **New tunnel**.
3. In the TCP tunnel information screen enter,
   a. The name of the tunnel. In this case the name is `sendmail`.
   b. The host name or IP address of the target.
   c. The TCP port of the target.
4. Click **Save**.

You can now configure deployment schemes to send an e-mail notification at the end of a deployment.

Chapter 6. Working with Solaris on SPARC targets

Deploying Solaris operating systems has a number of specificities and prerequisites.

Note: Tivoli Provisioning Manager for OS Deployment does not support the operating system deployment on Fujitsu SPARC targets.

To deploy Solaris, you must have installed a Solaris install server which is also running the web interface extension.

1. Set up and configure a Solaris install server.
   a. Set up a Solaris install server
   b. Configure it for operating system content.
   c. Configure it for Flash Archive content.
   d. Install the web interface extension on the Solaris install server.
2. Register new targets. You must add SPARC targets manually into the Target Monitor or import a comma-separated text file containing a list of targets to be added.
3. Setup the specific SPARC DHCP options for these targets.

You can then create your Solaris system profiles and software modules.

You can then also boot SPARC targets on the OS deployment server.

System profiles

A system profile is the partition layout and list of files to deploy an operating system, either by unattended setup or by cloning, from a reference target or from a reference image.

The main purpose of Tivoli Provisioning Manager for OS Deployment is to deploy an operating system on targets by replicating a reference system. However, unattended installation of operating systems is also possible. The latter case Tivoli Provisioning Manager for OS Deployment does not replicate a reference system, but merely provides the correct parameters to Windows or UNIX setup for a fully unattended installation.

There are a number of differences between an unattended installation and disk cloning. First, creating an unattended installation in Tivoli Provisioning Manager for OS Deployment is straightforward. All of the necessary tasks are performed on the server, using the Web interface. In contrast, a cloning-mode system profile requires you to configure a target, prepare it for cloning, and run the cloning process directly on the target. However, the native mode of operation of the product is centered around cloning-mode system profiles, because this method of deployment is faster than unattended installation. When deploying computers on a large scale, unattended installation is not possible. Novice users might start with creating unattended installation profiles because this is easier than cloning-mode profiles.

Note:

© Copyright IBM Corp. 2006, 2010

- To avoid failures in creating or deploying asystem profile, clean up the temporary directory inside the OS deployment server installation directory on a regular basis.
- To create or deploy a system profile from a physical or virtual machine at least 512 MB RAM is required.
- The exFAT filesystem is not supported.
- System profiles can have a maximum of 3 primary partitions. Therefore, you cannot clone a target with 4 primary partitions.

Creating an unattended setup system profile for Solaris operating system

You can install operating systems using standard installation processes in unattended mode.

- You must have set up a Solaris install server, as described in the Installation Guide, Chapter 6, sectioninstall/tosd_solariscontent.dita.
- If you want to create a system profile from Solaris 10 Update 6 or higher, do not forget to modify the wanboot directory.
  - Create a directory named `interim_dir` by running:

    ```
    mkdir /export/install/sol10-miniroot/interim_dir
    ```

  - Copy the `platform` subdirectory from `Solaris_10/Tools/Boot` into the `/sol10-miniroot/interim_dir` directory as follows:

    ```
    (cd /export/install/Solaris_10/Tools/Boot ; tar cf - platform) |
    (cd /export/install/sol10-miniroot/interim_dir ; tar xvf - )
    ```

- You must create your system profile from a Solaris target and the NFS server must also reside on a Solaris target.
- Make sure the web interface extension is running.

1. Go to **Server** → **OS deployment** → **System profiles**.
2. Click **New Profile**. A system profile wizard opens to guide you through the steps of creating a profile.
3. Select **Unattended setup** in the first pane of the profile wizard.
4. Select **A Solaris system profile**.
5. Follow the instruction of the profile wizard.

When your first unattended installation profile is created, you can use it to deploy targets.

Now that you have created your Solaris unattended profile, you can optionally move your Solaris install server to a UNIX target with an NFS server compatible with Solaris targets. In this case, you must edit your profile configuration to update the value of **NFS install source** to the new NFS server.

Note: When using a Linux NFS server, the NFS share should force to NFS 3 since NFS 4 from Solaris is not compatible with NFS 4 from Linux.

Creating a system profile from a Solaris Flash archive

You can create a cloning system profile from a Solaris Flash archive (a file with a `.flar` extension).

To be able to create your system profile, you need not only the Solaris Flash archive on your NFS server, but also the complete installation files for a Solaris operating system. The Profile Wizard asks you first for the directory in which the

operating system installation files are located. It checks whether the `.cdtoc` hidden file is present before asking you for the exact location of the Flash Archive you want to use for your system profile creation.

To create a system profile from a reference image, you must follow these steps:

1. Go to **Server** → **OS deployment** → **System Profiles**.
2. Click **New Profile**. This opens a system profile wizard that guides you through the steps of creating a profile.
3. Select **Cloning from a reference image file** and click **Next**.
4. Select the corresponding image format and click **Next**.
5. Follow the instruction of the profile wizard.

Creating Flash archives

Although Tivoli Provisioning Manager for OS Deployment is not involved in the creation of Flash archives, the process is described for convenience.

For more information, see the SUN Solaris documentation.

Creating flash archives in Solaris is a relatively simple process.

1. Mount the flash archive directory on the install server.
   a. Create a local mount point, a directory that you can reference locally.
      ```
      mkdir /export/flash
      ```
   b. Mount the remote flash archive directory
      ```
      mount certdev-sun2:/export/flars /export/flash
      ```
2. Run the flash archive creation command
   ```
   flarcreate -n flarname.flar -x /export/flash -c /export/flash/flarname.flar
   ```
3. Restart the computer to make sure that all unnecessary file handles are closed.
4. Check that the new flash archive is created and sent to the Flash directory of the Solaris install Server.

Note: There can be installation specific issues with Flash archives. In particular, some symbolic links may prevent flash archives to be restored properly. As a workaround, remove the symbolic links and copy the actual files in the appropriate directory.

Organizing and editing system profiles

After you have created a system profile, you can view it on the OS deployment server through the web interface. The profiles are listed on **> Server > OS deployment > Profiles** page, in the **System profiles** pane. Each blue jacket represents a system profile (that is, the hard-disk partition images).

If you want to organize your system profiles, you can create subfolders by following these steps:

1. Select the parent folder with a left mouse click.
2. Call the contextual menu with a right mouse click.
3. Select the **Add a new profile folder** menu item.
4. Enter the new folder name.
5. Click **OK**.

You can then move profiles (by dragging-and-dropping the profile icons) from the top folder, where they are automatically created, to the appropriate subfolder.

Editing system profiles

To display and edit the parameters associated with a given profile:

1. Double-click a system profile to open the **Profile details** page.
2. Click **Edit** on top of the parameter sections to edit the parameters.

Updating device mapping

Device mapping can be updated to force logical disks to point to specific physical devices.

Note: Updating device mapping from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose one or the other entry point, and then perform all your changes from that entry point.

1. Go to **Server** → **OS deployment** → **System profiles** → **Profile details** or to **Server** → **OS deployment** → **System profiles** → **Profile details** → **OS configuration details**, **Disks** tab.
2. Click **Modify device mapping**.
3. Select to which physical device you want to map your logical disk. The column starting with **Disk 0** corresponds to an automatic detection of the first to the eighth disk, the column starting with **/dev/hda** corresponds to standard device names.

   Note: Spanning a logical disk on several physical disks is not currently available.
4. Click **OK**.
5. Repeat step 2 to step 4 for each logical disk for which you want to update device mapping.

If the new device mapping you selected is incorrect, you receive a warning message.

OS configurations and fixed common parameters

A system profile is the partition layout and list of files to deploy, while OS configurations are operating system parameters.

At the very bottom of the **Profile details** page, there is a list of the OS configurations that correspond to your profile.

You can define several OS configurations for each system profile and duplicate them. These copies share the same image files, and the same partition layout, but can have different target parameters. You must then assign new values to some of the OS configurations parameters to make the original OS configuration and its copies distinct.

If you want to automate the assignment of parameters to targets, you can view and edit the OS configuration you are about to deploy by clicking on its name in the **Profile details** page. You are now in the **OS configuration details** page. The

information is divided into panels, each displaying sets of parameters. You can modify the parameters either through the web interface or by using a parameter file.

Editing OS configuration parameters in the web interface for Solaris

The web interface displays a number of OS configuration parameters divided into panes. These parameters can be edited in the web interface.

To edit parameters:

1. Click a tab to select the corresponding pane.
2. Click **Edit** on the banner of the section where you want to update parameters.
3. Modify the values.
4. Click **OK**.

Editing a Solaris parameter file

You can modify OS configuration parameters by editing a file. This option allows you to modify parameters that are not displayed in the web interface. However, you must be experienced to use this option advantageously, because Tivoli Provisioning Manager for OS Deployment does not provide any syntax checking of the file. Information about the file format and syntax can be found in the documentation of the operating system itself.

1. Click **Edit custom 'solaris.profile'** to edit the file.
2. Type the parameters and their values in the syntax requested by the operating system, or copy and paste it from another editor.
3. Click **OK**.

Tivoli Provisioning Manager for OS Deployment merges the information of the edited file with the information provided on the web interface (default file). Unless otherwise specified, parameters specified in the default file override the content of the custom file.

Here is an example of a disk layout described in a `solaris.profile` file:

```
partitioning    explicit
filesys         rootdisk.s0     free    /
filesys         rootdisk.s1     2048    swap
cluster         SUNWCpm         delete
cluster         SUNWCpmx        delete
cluster         SUNWCdial       delete
cluster         SUNWCdialx      delete
cluster         SUNWCadm
cluster         SUNWCcpc
```

By default the deployment provides its own pre-installation and post-installation scripts for generating profiles dynamically and installing software modules specified in the database.

If you want to add your own code in the pre-installation and post-installation scripts, you can do so by adding sections in the custom profile configuration file `solaris.profile`.

```
SI_BEGIN:
echo 'This is the pre-installation script'
...
SI_PROFILE:
echo 'This is the profile configuration'
```

```
partitioning    explicit
filesys         rootdisk.s0     free    /
filesys         rootdisk.s1     2048    swap
cluster         SUNWCpm         delete
cluster         SUNWCpmx        delete
cluster         SUNWCdial       delete
cluster         SUNWCdialx      delete
cluster         SUNWCadm
cluster         SUNWCcpc
SI_FINISH:
echo 'This is the post-installation script'
...
```

Note: The cluster command is not supported in `solaris.profile` files attached to cloning system profiles.

Software modules for Solaris operating systems

Software modules are images other than system profiles that can be created to address various needs.

Tivoli Provisioning Manager for OS Deployment is based on imaging technology. As administrator, you create images of components that you want to see on every target, and the automated deployment merges and restores these images on each target, automatically, when needed.

Tivoli Provisioning Manager for OS Deployment can handle most scenarios for software deployment and post-installation configuration.

Types of software modules

There are many types of software modules. Depending on the type of package and installation files, the wizard guides you through the different steps to achieve your software module with minimal effort. The types of software package supported by the wizard are listed in this section.

- **A Solaris package installation, using pkgadd**
- **A custom action on the target computer**. This includes OS configuration changes such as commands to be run, and copying sets of files on the target.

Creating software modules

There are distinct types of software modules which vary according to the operating system being deployed. The software wizard guides you through the creation of software modules for each type.

Creating Solaris software modules with pkgadd

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select **Solaris** and click **Next**.
4. Select **A Solaris package installation, using pkgadd** and click **Next**.
5. Follow the instructions of the wizard to create your software module

   Note: Make sure the folder containing Solaris package also includes the corresponding `pkginfo`. The software module cannot be created if `pkginfo` is not found in the folder.

Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

- A description that identifies the software module in the software module tree.
- A comment with additional information about the software module.
- The stage of the deployment when yoursoftware module must be installed. Solaris software modules must always be installed with the operating system.
- A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
- The path to where the installation files are restored on the target. This path is relative to the system root partition.
- An additional command line that might be necessary to install your software module. When possible, the wizard suggests automatically the appropriate command line to run the installation unattended. However, you might must add some additional parameters to the command.

  All software packages executed during installation run in a specific environment where the newly installed system is mounted under directory `/a`. To write a file in the root directory, you must use the `/a` path. The `/a` prefix is automatically added to the destination path when copying packages, so this only applies to command lines referring to specific paths on the newly installed system
- The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

You can organize your software modules by creating software module subfolders following the same procedure as for system profiles.

Creating a custom action software module for Solaris operating systems

Software modules can also contain custom actions to be performed on the target.

They are divided into:

- An OS configuration change to perform on the target
- A set of files to copy on the target

Configuration changes are further subdivided. Depending on the operating system, you can:

- Copy a single text file
- Run a single command file.

In the OS configuration change wizard screen, you can select **Activate keyword substitutions**. If you use this option, you can specify which keywords must be substituted in the software module details.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Click **New software** to run the software wizard.
3. Select the operating system and click **Next**.

4. Select **A custom action on the target** and click **Next**.
5. Follow the instructions of the wizard to create your software module.

   Parameters of the software module are pre-filled for you but they can be modified in the appropriate step of the software wizard. These parameters include:

   - A description that identifies the software module in the software module tree.
   - A comment with additional information about the software module.
   - The stage of the deployment when yoursoftware module must be installed. Solaris software modules must always be installed with the operating system.
   - A file name to store your image on the OS deployment server. Software modules typically have a `.pkg` extension.
   - The path to where the installation files are restored on the target. This path is relative to the system root partition.
   - An additional command line that might be necessary to install your software module. When possible, the wizard suggests automatically the appropriate command line to run the installation unattended. However, you might must add some additional parameters to the command.

     All software packages executed during installation run in a specific environment where the newly installed system is mounted under directory `/a`. To write a file in the root directory, you must use the `/a` path. The `/a` prefix is automatically added to the destination path when copying packages, so this only applies to command lines referring to specific paths on the newly installed system
   - The operating system with which the software module is compatible. The deployment wizard offers to deploy only software modules compatible with the operating system being deployed. Moreover, this parameter prevents the deployment of a bound software module if the it is not compatible with the operating system. Additionally, you can also sort and filter software modules by this parameter in list view.

Repeating custom actions:

Some commands must be run every time the target boots during a deployment.

This is typically the case if you want to repeatedly connect a network share. This connection is destroyed when rebooting. You can therefore create a single software module with a `netuse` command to set the network share and set this software module to run once after each reboot, starting at a specific reboot.

This option is available for executing a single command.

1. Create your software module.
2. Double-click on the software module name in the **Software components** page to obtain the **Software details** page
3. Click **Edit** in the title of the **Package information** section.
4. Select the installation stage at which the software module must be applied first.
5. Select **Run at each software pass until end of deployment** and click **OK**.

Editing software modules

You can edit the basic parameters of a software module, upload new files into your software module, and update drivers.

1. Go to **Server** → **OS deployment** → **Software modules**. Double-click on a software module to view the details.
2. From **Software details** page, use the links and buttons. Links vary according to the type of software module. Not all the links listed are necessary available.
   - To edit the base parameters of a software module, click **Edit** at the top of the **Software module information** section.
   - To update files or add new files into the software module, click **Edit software module files**, or a link with a similar name, and select **Upload file** from the contextual menu.

     Note: File upload is limited to 16 MB.

Keeping command lines confidential

When you use command lines in your software modules, their call and their output are stored in deployment logs. In some circumstances, for example when the command line includes a password or a product key, it might be necessary to keep the information contained in the command line confidential. Three levels of confidentiality are available.

No confidentiality
: The command line is visible in the web interface and on the target during the installation, its call is logged, and its output is also logged.

The command line call is not logged
: The command line is visible in the web interface, and its output is logged, but the command line call, containing the whole command line string with all parameters, is visible in the logs neither on the web interface nor on the target.

  To apply this level of confidentiality, you must prefix the command line by one exclamation mark (!).

The command line call and output are not logged
: The command line is visible in the web interface, but its call and output are visible in the logs neither on the web interface nor on the target.

  To apply this level of confidentiality, you must prefix the command line by two exclamation marks (!!).

To keep command lines confidential:

- Enter the appropriate number of exclamation points in front of the command in the Software Wizard when first creating the software module.
- Edit the software module information
  1. Go to **Server** → **OS deployment** → **Software modules**. Double-click on a software module to view the details.
  2. Click **Edit** in the Software module information banner.
  3. Update the command line with the appropriate number of exclamation points.
  4. Click **OK**.

Keyword substitution

You can usefully use keyword which act as variables and are substituted with their values during deployments. Keywords can either refer database values or server specific values, given by the user.

Syntax

Variable substitution expressions follow the syntax given here. They start with the character { and end on the same line with }. Words between these two characters are interpreted by using one of the following schemes:

- *{$expr$}* the expression is replaced with the string resulting of the evaluation of `expr`.
- *{/expr/ab}* the expression is replaced with the string resulting of the evaluation of `expr`, but each occurrence of the character `"a"` is replaced by the character `"b"` (character-based substitution).
- *{=expr=test content=this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is equal to the text `"test content"`.
- *{!expr!test content!this is a test}* the text `"this is a test"` is included in the destination file only if the string resulting of the evaluation of `expr` is not equal to the text `"test content"`.

Note: If a variable does not exist (for example, it contains a typing error or it is not described in `server.ini`) but it is used in a command, its value is supposed to be empty which can result in deployment errors.

Database keywords

Within an expression, database records can be referred to. Within a record, each field can be accessed using the standard C notation (`record.fieldname`). The exhaustive list of these fields can be obtained from the database records, with the following correspondences between variable and database record names:

Table 8. Records for `free-text conditions`

Variable record name	Database record name
Disk	DiskInventory
DMI	DMIInventory
Order	BOM
User	UserProfile
System	SystemProfile
PCI	PCIInventory

Below are a few examples of available fields:

- `Order.IP`: a string, the target IP address, such as `192.168.1.2`
- `Order.MAC`: a string, the target MAC address, such as `00:01:02:03:04:05`
- `Order.SN`: a string, the target Serial Number, such as `CH12345678`
- `Order.Model`: a string, the computer model name, such as `e-Vectra`
- `User.UserCateg0`: a string, without any restriction, such as`technicians`
- `DMI.Vendor`: a string, the vendor name, such as `Hewlett-Packard`
- `DMI.Product`: a string, same as `Order.Model`
- `DMI.ProcModel`: a string, the processor model
- `Disk[0].Type`: a string, the disk 0 drive type, such as `ATAPI`
- `Disk[0].Media`: a string, the disk 0 media type, such as `Disk` or `CD`
- `Disk[0].DiskSize`: a number, the physical size of the disk (if detected)

- PCI[0].VendorID: a string, the hexadecimal vendor ID of the device
- PCI[0].DeviceID: a string, the hexadecimal device ID of the device

For disks and PCI devices, you can use the function sizeof (sizeof(Disk) and sizeof(PCI)) to discover the number of devices present. You can then use indexes to access these devices.

As an example for keyword substitution, if BomID has OrgName Rembo SaRL, RemboServer 192.168.168.16, and IP 192.168.168.32 for value 1, the following text

```
BomID:{$Order.BomID$}
OrgName:{$User.OrgName$}/{$StrToLower(User.OrgName)$}
RemboServer:{$Order.RemboServer$}
IP:{$Order.IP$}
```

gives the following results after keywords are substituted (note the use of a Rembo-C function within the expression to be substituted):

```
BomID:1
OrgName:Rembo SaRl/rembo sarl
RemboServer:192.168.168.16
IP:192.168.168.32
```

Server specific keywords

If you want to set up server specific keywords, which are defined exclusively by the user and per server, you must edit Files/global/rad/server.ini.

Start the file with [Custom] and add a line per keyword, in the format **keyword=value**, where keyword is a word of your choice and value the value you want to give it.

To use the keyword in a command, type Server.keyword and activate keyword substitution when creating the software module.

Note: server.ini is not replicated between servers. If you use multiple servers, you must edit server.ini on each server.

Customizing the software page

You can view the software modules in a tree viewer or in a list viewer. The list viewer allows you to customize the visible information.

You must have created at least one software module, otherwise there is nothing to view.

To customize the visible information

1. Go to **Server** → **OS deployment** → **Software modules**. Then click **List view**.
2. From the list view, you can
   - Drag the column separator in the column heading to resize the column.
   - Click on the triangular arrow to the left of the column name to sort the software modules by column criteria.
   - Click on the arrow on the right of the column name and select an option to filter the information. Filtering on several columns is cumulative.
3. For more options, right click anywhere to open the contextual menu and select **Arrange columns**.

- Select the columns you want to see and clear the others.
- Click on the minus or plus icons to decrease or increase the size of a column.
- Select a column and use the up and down arrows to move the column relatively to the others.

Click **OK** to save your changes. The updated version of the list view is visible in the **Software modules** page.

To return to the tree view, click **Tree view**. You can also access the details of the software modules by double-clicking on a software module name, from either view.

OS configuration and software bindings

OS configuration bindings determine which configurations are available to a target when booting the target on the network, while software bindings correspond to the list of software modules currently assigned to the target.

OS configuration and software bindings are created when:

- The Target Monitor has been used to manually modify OS configuration and software bindings for the target
- A deployment has been started with the Target Monitor. In this case, an OS configuration binding is added for the corresponding OS configuration.
- Automatic binding rules are configured in the **Details** page of OS configurations or software modules. Some of these rules have matching values for the specified criteria. These bindings cannot be modified, except by modifying the rules.

With the Target Monitor, you can browse, remove or add OS configuration and software bindings to any target present in the database. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.

Binding software modules and OS configurations to targets

Bindings link software modules and OS configurations to targets to enable automatic deployment. When binding to targets, you explicitly provide the list of software modules and OS configurations to bind to your target.

To explicitly bind a software module or a OS configurations to a target, there are two methods:

- From the **Target Monitor** page
- From the **Target details** page

If you want to bind software modules or OS configurations to a group of targets, you must do it through the Target Monitor.

From the Target Monitor:

1. Select a target or a group of targets
2. Select **Bind software** or **Bind OS configurations** from the contextual menu
3. Select the items to bind from the popup window
4. Click **OK**

From the Target details page:

1. Go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.
2. Go to the **Bindings** panel.

3. Click **Edit** in the relevant section to add explicit bindings for OS configurations and software modules.
4. Select the items for which you want to add explicit bindings.
5. Click **OK**

You can also clear items to remove their explicit bindings. To remove a binding by rule, you must modify the rule.

Binding software modules to computer models and PCI devices

You can link software modules to target computer hardware model to enable automatic deployment. When creating a binding, you explicitly provide the list of software modules to link to a computer model and its PCI devices.

1. Go to **Server** → **OS deployment** → **Software modules**.
2. Select **View bindings** from the contextual menu. A grid is displayed representing computer model-based software bindings:
   - The rows correspond to your software modules
   - The columns correspond to the computer model. If you expand it, you can see all the corresponding PCI devices discovered in your hardware inventory.

   The background color indicates implicit bindings by existing generic rules, while checkmarks correspond to explicit one to one bindings.
3. Select a grid position corresponding to the software module and device to bind.
4. Select **Bind** from the contextual menu.

You can also remove a binding by selecting it and clicking **Unbind** from the contextual menu.

Binding software modules to a deployment scheme

Software modules can be bound to deployment schemes.

Take a company with offices in three locations: New York, Quebec City, and Mexico City. In each of these locations, the company has people in human resources, sales, logistics, and product development. For the sake of simplicity, consider further that all the employees use either one of two types of computers: a desktop, or a notebook. All desktop computers are identical (with the same network card, system board, disks, and so on) and the same applies for all notebooks.

In this scenario, the company needs two profiles, one with the image for notebooks and one with the image for desktop computers. Three configurations per profile (six in total) are necessary to integrate the different parameters of the different locations, in particular language and time zone information. Finally, schemes are set according to the employees' department, with software modules specific to the different departments bound directly to the deployment schemes.

1. Go to **Server** → **OS deployment** → **Task templates** Select the **Deployment Schemes** folder. Double-click on a deployment scheme to view its details.
2. Click **Edit** on the **Software bindings** section of the page to open the dialog to bind software modules to schemes.
3. Select which software modules you want to bind to your deployment scheme, in addition to software modules that can have been bound to targets.
4. *(Optional)* If you want to use only the software checked in the window when deploying with this scheme, select the **Discard all other software binding rules** check box.

Automatic binding rules

Automatic binding rules are used to create bindings between OS configurations and targets, or software modules and targets, without having to specifically bind a OS configuration or a software module on each target.

Rules are created in OS configurations and software modules to determine which targets are automatically bound to the OS configuration or software module.

Rules are made of criteria and values. If a target has a matching value for all criteria in the rule, the OS configuration or software module will be bound to that target. The binding will be displayed with the mention **by rule** in the OS configuration panel of the target properties for targets that match the criteria. For example, if the criteria is the model name, and the value is `Optiplex`, targets with a model name starting with `Optiplex` will be bound to the object where the rule has been defined.

Automatic binding rules are defined in Tivoli Provisioning Manager for OS deployment at the bottom of the **OS configuration details** or **Software details** page.

To create a new binding rule, click **New rule** located at the bottom of the Web interface:

1. The dialog displayed to create a new binding rule is different depending on whether you are adding a rule to an OS configuration or to a software module. When adding a binding rule to a software module, you can set values for the following criteria:
   - A deployment scheme
   - A system profile
   - A current OS configuration
   - Administrative group
   - One of the system-definable and user-definable fields of the database (only used if you have customized the database)
   - An operating system type, such as Windows 2000
   - An operating system version, such as SP2
   - An operating system language
   - An operating system architecture, such as x86-32
   - A computer model name
   - A BIOS version
   - A PCI device
   - A base board
   - MultiChassi
   - HAL Type
   - A free-text condition in `Rembo-C;` syntax

   For example, to create a binding based on the operating system type between a software module and targets, you must create a new rule, click **OS type**, and select the operating system version that you want to limit this software module to.
2. When adding a binding rule to an OS configuration, you can set a condition on the deployment scheme, and on the computer model name. The next ten fields are only used if you have customized your database and want to match specific user categories.

3. Finally, you can enter a free-text condition following the `Rembo-C;` syntax. They must only be used by advanced users.

   The conditions determine the applicability of the rule and evaluate to `true` or `false`. A condition must be formed using the variables also used for keyword substitutions in software modules, combined with Java-like logical operators, listed by order of priority in the table:

Table 9. Logical operators for free-text conditions

Operator	Meaning
<	smaller than
<=	smaller than or equal to
=>	greater than or equal to
>	greater than
==	equal to
!=	not equal to
&&	AND operator
||	OR operator

   For example, a typical condition can be:

```
Disk[0].DiskSize > 10*1024*1024
```

   Note: If a condition cannot be evaluated, it is considered to have the value `false`.

Scheduling the application of software modules

It is not possible to schedule the application of software modules for this operating system and hardware.

All software modules are applied **When the OS is installed**, regardless of the set stage in the **Software application order** window. Reboots are not handled either.

Task templates

Task templates group together elements that can be customized on a target. These elements are mostly screen layouts, which condition the appearance of the target screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment.

A deployment scheme is a specific type of task template. Together with the target display screen layout, it contains other parameters for customizing a deployment on a target.

Note: Starting with Fix Pack, version 5.1.0.2, deployment schemes are considered to be a subset of task templates. The functions of deployment schemes have not been altered. To access deployment schemes, go to the **Task template** page, and select the **deployment scheme** folder.

The task template page of the web interface contains a task template tree in the left pane with seven folders. The content of the selected folder is displayed in the right pane.

There are seven task template folders in the tree. They are described here.

Deployment Schemes
Deployment schemes contain parameters that indicate how an OS configuration must be deployed on your target. The **deployment Schemes** folder contains at least the **Default** scheme.

Idle layout
The idle layout defines what is shown on the target when there is no pending task. The **Idle Layout** folder contains at least the **Idle state** layout.

Menu Layout
The menu layout defines how deployment menus are shown to the users. Menus are used when an OS configuration and on deployment CDs. The **Menu Layout** folder contains at least the **Menu** layout.

OS Detection Layout
The operation system detection layout defines the target display when a target is busy detecting the currently installed operating system. It is used when creating a cloning profile from the web interface. The **OS Detection Layout** folder contains at least the **Detect operating system** layout.

Profile Creation Layout
The system profile creation layout defines the target display when a target is busy creating a new system profile. It is used when creating a cloning profile from the web interface. The **Profile Creation Layout** folder contains at least the **Creating cloning profile** layout.

Profile Restoration Layout
The system profile restoration layout defines the target display during the manual restoration of a system profile by the administrator.

Note: A system profile restoration is always performed as-is and must not be confused with an automated deployment resulting in a fully configured operating system installation.
The **Profile Restoration Layout** folder contains at least the **Default OS Restoration** layout.

State Capture Layout
The state capture layout defines the target display when a target is saving the operating system state for future redeployments.

State Restoration Layout
The state restoration layout defines the target display when a target is redeploying an operating system from a saved state.

When a task template is selected in the right hand pane, the bottom of the web interface contains a link to **Customize GUI**. Follow this link to modify the look of your target screen . When the selected task template is a deployment scheme, there are additional links to view and edit the current scheme.

Creating and editing deployment schemes

By customizing your deployment schemes, you can adapt the way in which your predefined OS configurations are installed onto targets.

Deployment schemes are not linked to architecture of the target or the operating system. Therefore, the deployment scheme wizard always offers to set all modifiable parameters. When deploying, parameters incompatible with either the architecture of the target or the operating system being deployed are not taken into account.

1. The easiest way to create a new deployment scheme is to run the deployment scheme wizard, by clicking **New deployment scheme** from the **Task templates** page.
2. Alternatively, you can modify an existing scheme by editing its parameters. To do this, select a scheme and click **View deployment parameters** and then use **Edit** in the banner on top of each parameter section.
3. If you prefer using a wizard to edit your scheme, click **Edit parameters using a wizard**.

   The following parameters apply for simple one-time deployments and for redeployment operations.

   Description

   The first step is to enter a name for this deployment scheme. Make it explicit enough so that you can pick it easily when starting a deployment (the web interface does not show the settings in a deployment scheme, therefore the choice must be made by name only). Because deployment schemes determine how the computers are installed (and not what is being deployed), use a description such as `Multicast 50 targets` or `On-site` rather than the name of a OS configuration or of a group of computers.

   When the deployment starts

   Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

   You must determine if Tivoli Provisioning Manager for OS Deployment requires user interaction during deployment (to edit individualtarget parameters) or runs completely unattended.Select:

   Always edit target-specific parameters
   to have the opportunity to change the target parameters at every deployment. The parameter edition can be made either directly on the target computer or by double-clicking the target icon in the Target Monitor.

   Edit parameters for targets that are not yet in the database
   to be prompted only during the first installation of each target. Subsequent deployments for the same targets run unattended. This is the default choice.

   Never edit parameters
   to have unattended deployments if all BOMs have been previously entered in the database. Any missing entry leads to a red banner on the target and cancels the deployment for this target.

   You must also select how Tivoli Provisioning Manager for OS Deployment behaves when the model of the computer being deployed does not match the model of the computer on which the image was created. This feature requires DMI for hardware detection.Select:

   No if you know that all your system profiles are fully hardware-independent or for deploying universal images.

   Yes, display a warning
   if you want to see all possible OS configurations for a

computer, but want to avoid mistakes. This choice can require user interaction and is therefore not appropriate for a fully unattended deployment.

Yes, abort the deployment
if you want to prevent anyone from using an OS configuration on a computer different from the one for which it was designed.

Use 'BIOS fallback MBR' to start PXE is used when PXE activation (the process of enabling PXE when booting on the hard-disk) does not work.

The PXE boot code manages the multiple reboots needed to install a computer. To manage these reboots, the PXE boot code must intercept the boot process of the computer at every boot.

- If the computer is configured to always start on the network (LAN device first in the list of boot devices), there is nothing to do, becauseTivoli Provisioning Manager for OS Deployment is loaded into memory at every boot.
- If the computer is configured to start on the hard-disk, you can change the MBR of the hard-disk and make it point to the work partition at the end of the hard-disk. Tivoli Provisioning Manager for OS Deployment is then loaded from the hard-disk when the computer starts up, instead of loading the operating system. The disadvantage of this method is that, because the computer did not use the network card to boot, PXE is not available. To enable network access, PXE is activated with a special function in the PXE card that makes it behave as though the computer had booted on the LAN. However, this is not documented in PXE, and does not work on every network card. If the network does not support this, an error is raised, and access to the OS deployment server fails (the message **Network started**, followed by an error).

  When PXE activation does not work, you can write a special MBR telling the BIOS that the hard-disk is not a valid boot device. By default, the BIOS falls back to the next device in the list, which in most computers is the network. As a result, the computer boots on the network and has full access to the network. This is the purpose of the **Use 'BIOS fallback MBR' to start PXE** check box.

Data collection

Note: These parameters have no effect on PowerPC and Cell targets.

By default, Tivoli Provisioning Manager for OS Deployment automatically populates the database with an inventory of the hardware setups of all deployed computers. For Windows, a software inventory can also be populated based on the registry. If you are not interested in using those inventories, or if your computers do not comply with any of the hardware detection standards, you can disable these features. Be advised that running the hardware or software inventory on thousands of computers can produce a huge database. This inventory is performed on locked screen.

Tivoli Provisioning Manager for OS Deployment centrally reports the status of the deployment of target computers to the OS deployment server and to the server database. Additionally, if you want to keep the deployment logs and the list of software modules on each deployed

computer, you can specify a local path where the log files are to be stored. The path that you specify is relative to the root of the operating system on the target, for example, /. In the deployment scheme details, the label of this field is **Save deployment log to**.

When the deployment is completed

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

When the deployment process is finished, you can select if you want to:

- Turn off the computer automatically (if supported)
- Boot the operating system automatically (this value might not make sense with some values of the previous setting)
- Display a green banner and wait for a manual shutdown

Network usage

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

Note: Multicast is available only if

- The targets have an Intel x86 or x86-64 architecture
- Multicast is selected in the deployment scheme
- The subnet supports multicast traffic
- Multicast is not disabled in the boot options of the target
- The target is not a VMWare 3.0 guest.

Depending on the number of computers that you are deploying simultaneously on your infrastructure, you must select one of the following networking modes:Select:

Unicast
to deploytargets one by one, or if you cannot use multicast. When deploying several targets simultaneously in unicast, the deployment time increases dramatically with the number of targets, as the result of network saturation.

Multicast, without explicit synchronization
to use soft-synchronized multicast protocol. Using this protocol, every target independently starts downloading images at soon as it is ready, and continues with the deployment as soon as it has downloaded all required material. When two or more targets (using the same deployment scheme) are downloading files in parallel, they automatically share the same bandwidth. The fastest target has the priority for the choice of the next shared files to be sent by the server, but the slower targets can receive them if they need them. This is a scalable solution that allows for a rolling deployment scenario.

Multicast, with the following synchronization parameters
to use a classical replicated multicast method. This mode is adequate for installing computers in batches. Enter the replication parameters (for example, the number of targets to wait for before starting the download, and the maximum timeout before starting in any case). Tivoli Provisioning Manager for OS Deployment multicast protocol can accept new

download targets even after the initial replication period is over, and integrate them seamlessly into the transfer.

Note: In the first stage of an OS deployment, there are two target synchronization stages. Therefore it might seem that the maximum timeout that was set before starting the deployment is doubled.

You can use a network share or Linux HTTP protocol on the server to download the files to the target computers, rather than downloading the whole image to the hard disk of the target. Using a network share or Linux HTTP protocol provides a shorter operating system installation time. To use a network share or Linux HTTP protocol:

- Select **Download files with a network share or Linux HTTP when applicable** in the deployment scheme.
- Share the `files\global\partition` directory and provide at least read-only access to it.
- Enter the relevant server parameter. Go to **Server** → **Server parameters** → **Configuration**. (See "Network share module" on page 314). .

On-site deployment

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

If you are running a one-time deployment in a deployment center and do not want to use redeployment, leave the check box blank and click **Next**.

If you are running an on-site deployment, or if you plan to use redeployment, you can enable the advanced feature.

Redeployment parameters

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
Indicate if you want to keep the deployment image in a protected partition and the size of this partition.

Note: The following parameters cannot be modified using the wizard. You must edit your deployment scheme parameters.

Request user confirmation
This parameters allows you to ask for user confirmation before running a deployment.

Unbind OS configuration at the end

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the OS configuration from the target at the end of the deployment. This OS configuration is not proposed the next time the target boots and, if no other OS configuration is bound, the target presents a locked screen to the user.

Unbind software module at the end

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the software module at the end of the deployment. This software module is not proposed and installed the next time a deployment is performed.

Vista | 2008 | Windows 7 **Disable user interaction during deployment**

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `Yes` by default. If you set this parameter to `No`, you can obtain a command prompt by pressing Shift-F10 on the target computer during a deployment to modify deployment files.

Note: This parameter is currently relevant forWindows Vista/2008/7deployments only.

Send mail at end
This option can be set only if a *sendmail* tunnel has been created.

Send mail to
This parameter is available only if **Send mail at end** is not set to **No**. Use this field to enter the e-mail address to which information must be sent at the end of the deployment.

Configure Network

Note: This parameter has no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **Network settings** section, is set to **Before software installation** by default. The network setting of the target are set before software modules are installed, enabling the product to use the network settings during the installation of the software modules. Set the parameter to **After software installation** if you want the network settings to be applied after software modules are installed, for example if you intend to physically move the target after deployment and want it to be configured with the network settings for its final location.

Disable cancel button
This parameter, located in the **Client Display** section, allows you to prevent users from aborting a deployment by removing the cancel button. Set the parameter to **Yes** if you want to remove the cancel button from the client display

You can use the newly created deployment scheme to deploy a system profile.

To delete a deployment scheme, select the scheme, then right-click it, and click **Delete**.

Deploying Solaris

A deployment is the process of installing an operating system on a target, and configuring the operating system for a specific user.

Deployment requirements

To start a deployment on a target, several elements must be present in the database.

The following elements are required:

- A *deployment* scheme associated with the target to deploy. The deployment scheme determines how to deploy the operating system on the target. If there is no association between a deployment scheme and the target to deploy, Tivoli Provisioning Manager for OS Deployment automatically uses the **Default** deployment scheme.
- An operating system *configuration* that is used to select which operating system to install. If there is no OS configuration associated with the target to deploy, the deployment does not start.
- Optional *software modules* to install in addition to the operating system during the deployment process. If there is no software module associated with the target to deploy, the operating system image is deployed without modification.

The OS configuration and the software modules can be considered to be the *content* of the deployment, while the deployment scheme is the *how* of the deployment.

The database keeps information about associations (*bindings*) between targets and deployment schemes, between targets and OS configuration, and between targets and software modules. These bindings can be configured manually or with binding rules (for example, `deploy configuration windows XP on targets whose model name starts with Dell`).

The minimal binding required to start a deployment is an OS configuration. If no configuration is bound to a target, the deployment does not start. In practice, however, Tivoli Provisioning Manager for OS Deployment always asks for an OS configuration and deployment scheme when beginning a deployment.

Tools to start and configure deployments

Bindings between targets and deployment elements are necessary to perform a deployment. You can create and edit these bindings in the **OS configurations** panel of the **Target details** page.

The Target Monitor provides functions to prepare a deployment, start a deployment, follow the progress of a deployment, and organize targets.

Binding rules, used to create permanent implicit bindings between targets and deployment elements without having to explicitly create the binding for each target, are created using the web interface. OS configurations and software modules contain a specific section at the very bottom of the **Details** page for creating automatic binding rules.

Starting a one-time deployment

You start deployments in the web interface by indicating on which target or targets the deployment must occur.

When deploying a Solaris system profile, you must have set the following target properties:

IP address
To edit this field, go to **Server** → **OS deployment** → **Target Monitor**.

Double-click on a target to view its details. Click **Switch to Advanced IP settings mode** in the **Common networking info** section.

Network mask
To edit this field, go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Click **Switch to Advanced IP settings mode** in the **Common networking info** section.

Default Gateway
To edit this field, go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Click **Switch to Advanced IP settings mode** in the **Common networking info** section.

Name resolution method
To edit this field, go to **Server → OS deployment → Target Monitor**. Double-click on a target to view its details. Click **Edit** in the **UNIX-specific info** section. If you use DNS, then you must also set

- DNS server
- DNS domain
- DNS domain search order

in the **Common networking info** section.

Moreover, the target must already be registered in the DNS server. The target name (unqualified) entered in the OS deployment server database must match the DNS record.

If these requirements are not met, Jumpstart switches to interactive mode.

If you want to use DHCP, you must have set a DHCP reservation on your DHCP server to ensure a coherent name and IP address for your target during deployment.

To deploy Solaris system profiles, the OS deployment server must have access to the NFS share where the content of the installation CDs have been copied, because the content of the system profile is not stored on the OS deployment server. For the deployment, the NFS share can be on a Solaris target or they can have been moved to a UNIX target compatible with the Solaris targets

To start a deployment:

1. Select a single target or multiple targets on the Target Monitor page. To select multiple targets or deployment, select an administrative group, a custom list, a subnet, or click on individual target names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
4. Select **Simple deployment** and click **Next**
5. Follow the deployment wizard instructions to select a deployment scheme, an OS configuration and optionally software modules, and to set up deployment options.

   Note: The Wake on LAN options are available only on Intel targets.

When the deployment is complete, the target boots into the operating system.

After deploying a Solaris unattended setup profile, you might see the following message:

```
sunablade0 console login: line 24: WARNING: loghost could not be resolved
```

This is standard Jumpstart behavior, allowing you to redirect loghost to a separate computer provided by DNS.

After deployment, the following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Partitions are aligned on megabytes rather than on cylinders. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

Monitoring deployments

There are several ways available to monitor the deployment progress.

Monitoring deployment progress with theTarget Monitor

You can use the Target Monitor to monitor deployments remotely. Information is located on the **Target Monitor** page and on several tabs of the **Target details** page.

On the **Target Monitor** page, the target color changes during the deployment. When PXE is activated, targets are monitored on a regular basis. The color of the icon is updated as soon as the status changes. By pointing to the target icon, you can get a description of the target status.

Note: A successfully deployed computer can continue to have a yellow icon (indicating that the deployment is still in progress). This reflects a PXE activation problem. The computer, having booted on the hard disk, is not using the network to inform the OS deployment server of its status. To remedy this, select the **Use 'BIOS fallback MBR' to start PXE** check box in the deployment scheme wizard. This forces the computer to boot through the network first.

If the deployment scheme used is configured to collect inventory information about target hardware (which is the default), you can see information about target hardware in the **Inventory** panel of the **Target details** page for that target (double-click on the target to go to the details page).

At the end of the deployment, the target icon shows either a green screen (success), or a red screen (failure). The deployment logs stored on the OS deployment server provide information about the deployment process. They are particularly useful in case of deployment failure to track its cause. To access the logs, double-click the wanted target. This opens the **Target details** page. Select the **Logs** tab to display a list of logs. To view a specific log, click its description. To download it, click **download** immediately after the log description.

Note: Logs are text files with UTF-8 formatting. If you are using a Windows operating system, you can view log files adequately by opening them in Microsoft WordPad.

There is only one log file for each deployment. This log file contains information about the different stages of the deployment process, including reboots and information provided by the operating system being deployed.

If any log information needs to be propagated to the OS deployment server outside of any task, an *idle* log file is created to store this information. The idle log file is created on demand and does not therefore exist for all the targets.

Another place of interest for information about a current deployment or another current task is the **Task history** tab, where each task of the target is listed. For each current task, the following information is provided:

- Description
- Status
- Scheduled date
- Start date
- Progress rate
- End date
- Download link to the log file
- Download link to the task file
- Download link to the `bom` file

Note: You must scroll to the right of the **Task history** tab to see all the fields.

The log file contains the target log. The task file contains all the parameters of the task. The `bom` file contains target-specific parameters for the given task.

The log file, the task file, and the `bom` file are needed by the development team to fix defects. Make sure that you download these three files if you suspect the presence of a defect in the software.

To cancel or destroy a task, select the task and select **Cancel target task** or **Destroy target task** in the contextual menu. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on running tasks, because the task destruction can cause strange behavior.

To view tasks scheduled for a later time, go to the **Tasks** page.

Monitoring deployment progress with the Tasks page for Solaris

The **Tasks** page is also a useful source of information to monitor a deployment (and other tasks). You can also cancel tasks from there.

Go to **Server** → **Server history** → **Tasks**. .

The description field of each deployment in the **Tasks** page is headed by the keyword *Deploy* for easy retrieval. The information provided includes

Description
Is headed by specific keywords, indicating the type of task. *Deploy* is the keyword for deployment tasks.

Execution
Is the scheduled date and time for the execution of the task.

State Uses icons to represent if the task is pending, in progress, completed, and so on. If in doubt to the meaning of a state icon, browse over it to get a state name.

Progress
Indicates the rate of completion of the task as a percentage.

Expire Indicates when the task information is going to be removed from the page.

Tasks are expandable by clicking their + sign. An expanded task displays information about its targets. The target information fields are:

- IP address
- Hostname
- Start date and time of the task
- State
- Progress rate
- Status date

If, for any reason, you want to cancel a running or scheduled task, you can easily do so by following these steps:

1. Expand the task
2. Select the target for which you want to cancel the task
3. Select **Cancel task** from the contextual menu
4. It is also possible to *destroy* tasks. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on a running task, because its destruction can cause strange behavior. To permanently delete tasks:
   a. Select one or several tasks. To select multiple tasks, use the Shift key for a range of tasks and the Ctrl key for individual tasks.
   b. Select **Destroy task** from the contextual menu

Receiving an e-mail notification

To receive an e-mail notification at the end of a deployment, you must configure a TCP tunnel called *sendmail*.

To receive an e-mail notification at the end of a deployment, you must have configured a *sendmail* TCP tunnel.

Note: The OS deployment server supports only sendmail servers without authentication.

There are two options to configure a deployment to receive an e-mail notification:

- You can edit the deployment scheme used for deployment to include the notification information.
  1. Go to **Server** → **OS deployment** → **Task Templates**.
  2. Select **Deployment schemes** and double-click a specific deployment scheme name to edit it.
  3. Click **Edit** in the **General settings** section.
  4. Under **Send mail at end:**, select the type of notification that you want. You can choose among:
     – **No**
     – **Whatever the notification is**
     – **If deployment is successfully completed**
     – **If deployment failed**
  5. If you selected a notification, you must now enter a valid e-mail address to which the notification is sent, under **Send mail to:**.

- You can modify the settings of the deployment scheme in the deployment wizard. Step deploy/tosd_sendmail.dita#Receivinganemailnotification/first and possibly step deploy/tosd_sendmail.dita#Receivinganemailnotification/second are available.

Depending on your selection, you will receive an e-mail notification at the end of the deployment.

The notification e-mail looks like this:

The target with MAC *xxxx* / IP **xxx** has completed an activity *activity description*.

You can now deploy targets with the edited deployment scheme and receive e-mail notification at the end of the task.

Creating a sendmail TCP tunnel:

A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Note: OS deployment server supports only sendmail servers without authentication.

1. Go to **Server** → **Server parameters** → **Predefined channels**.
2. Click **New tunnel**.
3. In the TCP tunnel information screen enter,
   a. The name of the tunnel. In this case the name is `sendmail`.
   b. The host name or IP address of the target.
   c. The TCP port of the target.
4. Click **Save**.

You can now configure deployment schemes to send an e-mail notification at the end of a deployment.

Chapter 7. Working with AIX on PowerPC targets

To work with AIX system profiles on PowerPC targets, you must take into account some specificities of these targets.

Types of system profiles

Only unattended system profiles are supported on AIX. Cloning system profiles are not supported.

DHCP specificities

There are specific considerations for setting Dynamic Host Configuration Protocol (DHCP) options. Make sure you set them appropriately.

Note: Microsoft DHCP server does not work well with somePowerPC firmware. Use IBM recommended DHCP servers.

Registering new targets

You must add targets manually into the Target Monitor or import a comma-separated text file containing a list of targets to be added.

System profiles

A system profile is the partition layout and list of files to deploy an operating system.

Note:

- To avoid failures in creating or deploying asystem profile, clean up the temporary directory inside the OS deployment server installation directory on a regular basis.
- To create or deploy a system profile from a physical or virtual machine at least 512 MB RAM is required.
- The exFAT filesystem is not supported.
- System profiles can have a maximum of 3 primary partitions. Therefore, you cannot clone a target with 4 primary partitions.

Creating an unattended setup system profile for AIX operating system

An unattended setup system profile allows you to install operating systems using standard installation processes in unattended mode.

To create an AIX unattended system profile, you must work on an AIX operating system of the same version as the profile you want to create, and the web interface extension must be running.

Note: Using AIX NFS is not an easy process: if you have no DNS and no entry inside `/etc/hosts`, another computer can never mount the exported path. This may prevent a target to access the installation source during deployment. After profile creation, it is recommended to move the installation file to another NFS server.

© Copyright IBM Corp. 2006, 2010

1. Copy the AIX installation CD or DVD on the hard disk.
2. Export the path of the folder in which you have copied the installation files by NFS. This folder must have write permissions.
   a. Verify that NFS is already running by typing the command `lssrc -g nfs`. The output should indicate that the `nfsd` and the `rpc.mountd` daemons are active. If they are not, you must start the NFS daemons.
   b. At a command line, enter `smit mknfsexp`.
   c. Specify appropriate values in the fields
      - **Pathname of directory to export**
      - **Mode to export directory**
      - **Export directory now, system restart or both**
   d. Specify any other optional characteristics you want, or accept the default values by leaving the remaining fields as they are.
   e. When you have finished making your changes, SMIT updates the `/etc/exports` file. If the `/etc/exports` file does not exist, it is created.
   f. Repeat steps a through e for each directory you want to export.
3. Open the web interface, go to the menu and select **Profiles**.
4. Click **New Profile**. A system profile wizard opens to guide you through the steps of creating a profile.
5. Select **Unattended setup** in the first pane of the profile wizard.
6. Select your operating system from the list and click **Next**.
7. Follow the instructions of the wizard.

   When an AIX 6.11 profile is created, an `lpp` directory is created in the directory where the image was created. The `RPMS` and `installp` directories are moved to `lpp`, and the `usr/swlag` directory is moved to `lpp/usr`. The same principle applies also for AIX 5.3 system profiles. However, the directory names are not the same.

- When your first unattended installation profile is created, you can use it to deploy targets.
- If you want to create a new system profile, you must copy the AIX CD or DVD again, as indicated in step 1.

  Note: Copy the files in the same directory as previously only when you create a new profile, because otherwise deployment will not find the necessary files and fail.
- Although all the files on the installation media were necessary for the profile creation, only some are needed at deployment time. If you want to delete superfluous files:

  for AIX 5L™ 5.3
  make sure *NFS_install* points to `install/ppc`. You can then delete the content of other installation directories.

  for AIX 6.11
  make sure *NFS_install* points to `lpp`. You must keep the `lpp` directory and all its sub directories, but you can delete the content of other installation directories.
- Because AIX NFS server is not easy to use, it is recommended to move the AIX installation content on a Linux or Solaris NFS server. You must also edit the OS configuration to update **NFS installation source** to the new value.

Organizing and editing system profiles

After you have created a system profile, you can view it on the OS deployment server through the web interface. The profiles are listed on **> Server > OS deployment > Profiles** page, in the **System profiles** pane. Each blue jacket represents a system profile (that is, the hard-disk partition images).

If you want to organize your system profiles, you can create subfolders by following these steps:

1. Select the parent folder with a left mouse click.
2. Call the contextual menu with a right mouse click.
3. Select the **Add a new profile folder** menu item.
4. Enter the new folder name.
5. Click **OK**.

You can then move profiles (by dragging-and-dropping the profile icons) from the top folder, where they are automatically created, to the appropriate subfolder.

Editing system profiles

To display and edit the parameters associated with a given profile:

1. Double-click a system profile to open the **Profile details** page.
2. Click **Edit** on top of the parameter sections to edit the parameters.

Updating device mapping

Device mapping can be updated to force logical disks to point to specific physical devices.

Note: Updating device mapping from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose one or the other entry point, and then perform all your changes from that entry point.

1. Go to **Server** → **OS deployment** → **System profiles** → **Profile details** or to **Server** → **OS deployment** → **System profiles** → **Profile details** → **OS configuration details**, **Disks** tab.
2. Click **Modify device mapping**.
3. Select to which physical device you want to map your logical disk. The column starting with **Disk 0** corresponds to an automatic detection of the first to the eighth disk, the column starting with **/dev/hda** corresponds to standard device names.

   Note: Spanning a logical disk on several physical disks is not currently available.
4. Click **OK**.
5. Repeat step 2 to step 4 for each logical disk for which you want to update device mapping.

If the new device mapping you selected is incorrect, you receive a warning message.

OS configurations and fixed common parameters

A system profile is the partition layout and list of files to deploy, while OS configurations are operating system parameters.

At the very bottom of the **Profile details** page, there is a list of the OS configurations that correspond to your profile.

You can define several OS configurations for each system profile and duplicate them. These copies share the same image files, and the same partition layout, but can have different target parameters. You must then assign new values to some of the OS configurations parameters to make the original OS configuration and its copies distinct.

If you want to automate the assignment of parameters to targets, you can view and edit the OS configuration you are about to deploy by clicking on its name in the **Profile details** page. You are now in the **OS configuration details** page. The information is divided into panels, each displaying sets of parameters. You can modify the parameters either through the web interface or by using a parameter file.

Editing OS configuration parameters in the web interface for AIX

The web interface displays a number of OS configuration parameters divided into panes. These parameters can be edited in the web interface.

To edit parameters:

1. Click a tab to select the corresponding pane.
2. Click **Edit** on the banner of the section where you want to update parameters.
3. Modify the values.
4. Click **OK**.

Task templates

Task templates group together elements that can be customized on a target. These elements are mostly screen layouts, which condition the appearance of the target screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment.

A deployment scheme is a specific type of task template. Together with the target display screen layout, it contains other parameters for customizing a deployment on a target.

Note: Starting with Fix Pack, version 5.1.0.2, deployment schemes are considered to be a subset of task templates. The functions of deployment schemes have not been altered. To access deployment schemes, go to the **Task template** page, and select the **deployment scheme** folder.

The task template page of the web interface contains a task template tree in the left pane with seven folders. The content of the selected folder is displayed in the right pane.

There are seven task template folders in the tree. They are described here.

Deployment Schemes
Deployment schemes contain parameters that indicate how an OS

configuration must be deployed on your target. The **deployment Schemes** folder contains at least the **Default** scheme.

Idle layout
The idle layout defines what is shown on the target when there is no pending task. The **Idle Layout** folder contains at least the **Idle state** layout.

Menu Layout
The menu layout defines how deployment menus are shown to the users. Menus are used when an OS configuration and on deployment CDs. The **Menu Layout** folder contains at least the **Menu** layout.

OS Detection Layout
The operation system detection layout defines the target display when a target is busy detecting the currently installed operating system. It is used when creating a cloning profile from the web interface. The **OS Detection Layout** folder contains at least the **Detect operating system** layout.

Profile Creation Layout
The system profile creation layout defines the target display when a target is busy creating a new system profile. It is used when creating a cloning profile from the web interface. The **Profile Creation Layout** folder contains at least the **Creating cloning profile** layout.

Profile Restoration Layout
The system profile restoration layout defines the target display during the manual restoration of a system profile by the administrator.

Note: A system profile restoration is always performed as-is and must not be confused with an automated deployment resulting in a fully configured operating system installation.
The **Profile Restoration Layout** folder contains at least the **Default OS Restoration** layout.

State Capture Layout
The state capture layout defines the target display when a target is saving the operating system state for future redeployments.

State Restoration Layout
The state restoration layout defines the target display when a target is redeploying an operating system from a saved state.

When a task template is selected in the right hand pane, the bottom of the web interface contains a link to **Customize GUI**. Follow this link to modify the look of your target screen . When the selected task template is a deployment scheme, there are additional links to view and edit the current scheme.

Creating and editing deployment schemes

By customizing your deployment schemes, you can adapt the way in which your predefined OS configurations are installed onto targets.

Deployment schemes are not linked to architecture of the target or the operating system. Therefore, the deployment scheme wizard always offers to set all modifiable parameters. When deploying, parameters incompatible with either the architecture of the target or the operating system being deployed are not taken into account.

1. The easiest way to create a new deployment scheme is to run the deployment scheme wizard, by clicking **New deployment scheme** from the **Task templates** page.

2. Alternatively, you can modify an existing scheme by editing its parameters. To do this, select a scheme and click **View deployment parameters** and then use **Edit** in the banner on top of each parameter section.
3. If you prefer using a wizard to edit your scheme, click **Edit parameters using a wizard**.

   The following parameters apply for simple one-time deployments and for redeployment operations.

   Description

   The first step is to enter a name for this deployment scheme. Make it explicit enough so that you can pick it easily when starting a deployment (the web interface does not show the settings in a deployment scheme, therefore the choice must be made by name only). Because deployment schemes determine how the computers are installed (and not what is being deployed), use a description such as `Multicast 50 targets` or `On-site` rather than the name of a OS configuration or of a group of computers.

   When the deployment starts

   Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

   You must determine if Tivoli Provisioning Manager for OS Deployment requires user interaction during deployment (to edit individualtarget parameters) or runs completely unattended.Select:

   Always edit target-specific parameters
   to have the opportunity to change the target parameters at every deployment. The parameter edition can be made either directly on the target computer or by double-clicking the target icon in the Target Monitor.

   Edit parameters for targets that are not yet in the database
   to be prompted only during the first installation of each target. Subsequent deployments for the same targets run unattended. This is the default choice.

   Never edit parameters
   to have unattended deployments if all BOMs have been previously entered in the database. Any missing entry leads to a red banner on the target and cancels the deployment for this target.

   You must also select how Tivoli Provisioning Manager for OS Deployment behaves when the model of the computer being deployed does not match the model of the computer on which the image was created. This feature requires DMI for hardware detection.Select:

   No if you know that all your system profiles are fully hardware-independent or for deploying universal images.

   Yes, display a warning
   if you want to see all possible OS configurations for a computer, but want to avoid mistakes. This choice can require user interaction and is therefore not appropriate for a fully unattended deployment.

Yes, abort the deployment
if you want to prevent anyone from using an OS configuration on a computer different from the one for which it was designed.

Use 'BIOS fallback MBR' to start PXE is used when PXE activation (the process of enabling PXE when booting on the hard-disk) does not work.

The PXE boot code manages the multiple reboots needed to install a computer. To manage these reboots, the PXE boot code must intercept the boot process of the computer at every boot.

- If the computer is configured to always start on the network (LAN device first in the list of boot devices), there is nothing to do, becauseTivoli Provisioning Manager for OS Deployment is loaded into memory at every boot.
- If the computer is configured to start on the hard-disk, you can change the MBR of the hard-disk and make it point to the work partition at the end of the hard-disk. Tivoli Provisioning Manager for OS Deployment is then loaded from the hard-disk when the computer starts up, instead of loading the operating system. The disadvantage of this method is that, because the computer did not use the network card to boot, PXE is not available. To enable network access, PXE is activated with a special function in the PXE card that makes it behave as though the computer had booted on the LAN. However, this is not documented in PXE, and does not work on every network card. If the network does not support this, an error is raised, and access to the OS deployment server fails (the message **Network started**, followed by an error).

  When PXE activation does not work, you can write a special MBR telling the BIOS that the hard-disk is not a valid boot device. By default, the BIOS falls back to the next device in the list, which in most computers is the network. As a result, the computer boots on the network and has full access to the network. This is the purpose of the **Use 'BIOS fallback MBR' to start PXE** check box.

Data collection

Note: These parameters have no effect on PowerPC and Cell targets.

By default, Tivoli Provisioning Manager for OS Deployment automatically populates the database with an inventory of the hardware setups of all deployed computers. For Windows, a software inventory can also be populated based on the registry. If you are not interested in using those inventories, or if your computers do not comply with any of the hardware detection standards, you can disable these features. Be advised that running the hardware or software inventory on thousands of computers can produce a huge database. This inventory is performed on locked screen.

Tivoli Provisioning Manager for OS Deployment centrally reports the status of the deployment of target computers to the OS deployment server and to the server database. Additionally, if you want to keep the deployment logs and the list of software modules on each deployed computer, you can specify a local path where the log files are to be stored. The path that you specify is relative to the root of the operating system on the target, for example, /. In the deployment scheme details, the label of this field is **Save deployment log to**.

When the deployment is completed

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

When the deployment process is finished, you can select if you want to:

- Turn off the computer automatically (if supported)
- Boot the operating system automatically (this value might not make sense with some values of the previous setting)
- Display a green banner and wait for a manual shutdown

Network usage

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

Note: Multicast is available only if

- The targets have an Intel x86 or x86-64 architecture
- Multicast is selected in the deployment scheme
- The subnet supports multicast traffic
- Multicast is not disabled in the boot options of the target
- The target is not a VMWare 3.0 guest.

Depending on the number of computers that you are deploying simultaneously on your infrastructure, you must select one of the following networking modes:Select:

Unicast
to deploytargets one by one, or if you cannot use multicast. When deploying several targets simultaneously in unicast, the deployment time increases dramatically with the number of targets, as the result of network saturation.

Multicast, without explicit synchronization
to use soft-synchronized multicast protocol. Using this protocol, every target independently starts downloading images at soon as it is ready, and continues with the deployment as soon as it has downloaded all required material. When two or more targets (using the same deployment scheme) are downloading files in parallel, they automatically share the same bandwidth. The fastest target has the priority for the choice of the next shared files to be sent by the server, but the slower targets can receive them if they need them. This is a scalable solution that allows for a rolling deployment scenario.

Multicast, with the following synchronization parameters
to use a classical replicated multicast method. This mode is adequate for installing computers in batches. Enter the replication parameters (for example, the number of targets to wait for before starting the download, and the maximum timeout before starting in any case). Tivoli Provisioning Manager for OS Deployment multicast protocol can accept new download targets even after the initial replication period is over, and integrate them seamlessly into the transfer.

Note: In the first stage of an OS deployment, there are two target synchronization stages. Therefore it might seem that the maximum timeout that was set before starting the deployment is doubled.

You can use a network share or Linux HTTP protocol on the server to download the files to the target computers, rather than downloading the whole image to the hard disk of the target. Using a network share or Linux HTTP protocol provides a shorter operating system installation time. To use a network share or Linux HTTP protocol:

- Select **Download files with a network share or Linux HTTP when applicable** in the deployment scheme.
- Share the `files\global\partition` directory and provide at least read-only access to it.
- Enter the relevant server parameter. Go to **Server → Server parameters → Configuration**. (See "Network share module" on page 314). .

On-site deployment

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.

If you are running a one-time deployment in a deployment center and do not want to use redeployment, leave the check box blank and click **Next**.

If you are running an on-site deployment, or if you plan to use redeployment, you can enable the advanced feature.

Redeployment parameters

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
Indicate if you want to keep the deployment image in a protected partition and the size of this partition.

Note: The following parameters cannot be modified using the wizard. You must edit your deployment scheme parameters.

Request user confirmation
This parameters allows you to ask for user confirmation before running a deployment.

Unbind OS configuration at the end

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the OS configuration from the target at the end of the deployment. This OS configuration is not proposed the next time the target boots and, if no other OS configuration is bound, the target presents a locked screen to the user.

Unbind software module at the end

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to `No` by

default. Setting this parameter to Yes unbinds the software module at the end of the deployment. This software module is not proposed and installed the next time a deployment is performed.

Vista 2008 Windows 7 **Disable user interaction during deployment**

Note: These parameters have no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **General settings** section, is set to Yes by default. If you set this parameter to No, you can obtain a command prompt by pressing Shift-F10 on the target computer during a deployment to modify deployment files.

Note: This parameter is currently relevant forWindows Vista/2008/7deployments only.

Send mail at end
This option can be set only if a *sendmail* tunnel has been created.

Send mail to
This parameter is available only if **Send mail at end** is not set to **No**. Use this field to enter the e-mail address to which information must be sent at the end of the deployment.

Configure Network

Note: This parameter has no effect on PowerPC, Cell, and SPARC targets.
This parameter, located in the **Network settings** section, is set to **Before software installation** by default. The network setting of the target are set before software modules are installed, enabling the product to use the network settings during the installation of the software modules. Set the parameter to **After software installation** if you want the network settings to be applied after software modules are installed, for example if you intend to physically move the target after deployment and want it to be configured with the network settings for its final location.

Disable cancel button
This parameter, located in the **Client Display** section, allows you to prevent users from aborting a deployment by removing the cancel button. Set the parameter to **Yes** if you want to remove the cancel button from the client display

You can use the newly created deployment scheme to deploy a system profile.

To delete a deployment scheme, select the scheme, then right-click it, and click **Delete**.

Deploying AIX

A deployment is the process of installing an operating system on a target, and configuring the operating system for a specific user.

Deployment requirements

To start a deployment on a target, several elements must be present in the database.

The following elements are required:

- A *deployment* scheme associated with the target to deploy. The deployment scheme determines how to deploy the operating system on the target. If there is no association between a deployment scheme and the target to deploy, Tivoli Provisioning Manager for OS Deployment automatically uses the **Default** deployment scheme.
- An operating system *configuration* that is used to select which operating system to install. If there is no OS configuration associated with the target to deploy, the deployment does not start.
- Optional *software modules* to install in addition to the operating system during the deployment process. If there is no software module associated with the target to deploy, the operating system image is deployed without modification.

The OS configuration and the software modules can be considered to be the *content* of the deployment, while the deployment scheme is the *how* of the deployment.

The database keeps information about associations (*bindings*) between targets and deployment schemes, between targets and OS configuration, and between targets and software modules. These bindings can be configured manually or with binding rules (for example, `deploy configuration windows XP on targets whose model name starts with Dell`).

The minimal binding required to start a deployment is an OS configuration. If no configuration is bound to a target, the deployment does not start. In practice, however, Tivoli Provisioning Manager for OS Deployment always asks for an OS configuration and deployment scheme when beginning a deployment.

Tools to start and configure deployments

Bindings between targets and deployment elements are necessary to perform a deployment. You can create and edit these bindings in the **OS configurations** panel of the **Target details** page.

The Target Monitor provides functions to prepare a deployment, start a deployment, follow the progress of a deployment, and organize targets.

Binding rules, used to create permanent implicit bindings between targets and deployment elements without having to explicitly create the binding for each target, are created using the web interface. OS configurations and software modules contain a specific section at the very bottom of the **Details** page for creating automatic binding rules.

Starting a one-time deployment

You start deployments in the web interface by indicating on which target or targets the deployment must occur.

To start a deployment:

1. Select a single target or multiple targets on the Target Monitor page. To select multiple targets or deployment, select an administrative group, a custom list, a subnet, or click on individual target names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
4. Select **Simple deployment** and click **Next**

5. Follow the deployment wizard instructions to select a deployment scheme, an OS configuration and optionally software modules, and to set up deployment options.

   Note: The Wake on LAN options are available only on Intel targets.

When the deployment is complete, the server either displays a green banner on the target, boots in the operating system, or powers the target off, depending on how the deployment scheme is configured.

After deployment, the following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Partitions are aligned on megabytes rather than on cylinders. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

Monitoring deployments

There are several ways available to monitor the deployment progress.

Monitoring deployment progress with theTarget Monitor

You can use the Target Monitor to monitor deployments remotely. Information is located on the **Target Monitor** page and on several tabs of the **Target details** page.

On the **Target Monitor** page, the target color changes during the deployment. When PXE is activated, targets are monitored on a regular basis. The color of the icon is updated as soon as the status changes. By pointing to the target icon, you can get a description of the target status.

Note: A successfully deployed computer can continue to have a yellow icon (indicating that the deployment is still in progress). This reflects a PXE activation problem. The computer, having booted on the hard disk, is not using the network to inform the OS deployment server of its status. To remedy this, select the **Use 'BIOS fallback MBR' to start PXE** check box in the deployment scheme wizard. This forces the computer to boot through the network first.

If the deployment scheme used is configured to collect inventory information about target hardware (which is the default), you can see information about target hardware in the **Inventory** panel of the **Target details** page for that target (double-click on the target to go to the details page).

At the end of the deployment, the target icon shows either a green screen (success), or a red screen (failure). The deployment logs stored on the OS deployment server provide information about the deployment process. They are particularly useful in case of deployment failure to track its cause. To access the logs, double-click the wanted target. This opens the **Target details** page. Select the **Logs** tab to display a list of logs. To view a specific log, click its description. To download it, click **download** immediately after the log description.

Note: Logs are text files with UTF-8 formatting. If you are using a Windows operating system, you can view log files adequately by opening them in Microsoft WordPad.

There is only one log file for each deployment. This log file contains information about the different stages of the deployment process, including reboots and information provided by the operating system being deployed.

If any log information needs to be propagated to the OS deployment server outside of any task, an *idle* log file is created to store this information. The idle log file is created on demand and does not therefore exist for all the targets.

Another place of interest for information about a current deployment or another current task is the **Task history** tab, where each task of the target is listed. For each current task, the following information is provided:

- Description
- Status
- Scheduled date
- Start date
- Progress rate
- End date
- Download link to the log file
- Download link to the task file
- Download link to the `bom` file

Note: You must scroll to the right of the **Task history** tab to see all the fields.

The log file contains the target log. The task file contains all the parameters of the task. The `bom` file contains target-specific parameters for the given task.

The log file, the task file, and the `bom` file are needed by the development team to fix defects. Make sure that you download these three files if you suspect the presence of a defect in the software.

To cancel or destroy a task, select the task and select **Cancel target task** or **Destroy target task** in the contextual menu. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on running tasks, because the task destruction can cause strange behavior.

To view tasks scheduled for a later time, go to the **Tasks** page.

Monitoring deployment progress with the Tasks page for AIX

The **Tasks** page is also a useful source of information to monitor a deployment (and other tasks). You can also cancel tasks from there.

Go to **Server** → **Server history** → **Tasks**. .

The description field of each deployment in the **Tasks** page is headed by the keyword *Deploy* for easy retrieval. The information provided includes

Description
: Is headed by specific keywords, indicating the type of task. *Deploy* is the keyword for deployment tasks.

Execution
: Is the scheduled date and time for the execution of the task.

State
: Uses icons to represent if the task is pending, in progress, completed, and so on. If in doubt to the meaning of a state icon, browse over it to get a state name.

Progress
: Indicates the rate of completion of the task as a percentage.

Expire Indicates when the task information is going to be removed from the page.

Tasks are expandable by clicking their + sign. An expanded task displays information about its targets. The target information fields are:

- IP address
- Hostname
- Start date and time of the task
- State
- Progress rate
- Status date

If, for any reason, you want to cancel a running or scheduled task, you can easily do so by following these steps:

1. Expand the task
2. Select the target for which you want to cancel the task
3. Select **Cancel task** from the contextual menu
4. It is also possible to *destroy* tasks. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on a running task, because its destruction can cause strange behavior. To permanently delete tasks:
   a. Select one or several tasks. To select multiple tasks, use the Shift key for a range of tasks and the Ctrl key for individual tasks.
   b. Select **Destroy task** from the contextual menu

Receiving an e-mail notification

To receive an e-mail notification at the end of a deployment, you must configure a TCP tunnel called *sendmail*.

To receive an e-mail notification at the end of a deployment, you must have configured a *sendmail* TCP tunnel.

Note: The OS deployment server supports only sendmail servers without authentication.

There are two options to configure a deployment to receive an e-mail notification:

- You can edit the deployment scheme used for deployment to include the notification information.
  1. Go to **Server** → **OS deployment** → **Task Templates**.
  2. Select **Deployment schemes** and double-click a specific deployment scheme name to edit it.
  3. Click **Edit** in the **General settings** section.
  4. Under **Send mail at end:**, select the type of notification that you want. You can choose among:
     – **No**
     – **Whatever the notification is**
     – **If deployment is successfully completed**
     – **If deployment failed**
  5. If you selected a notification, you must now enter a valid e-mail address to which the notification is sent, under **Send mail to:**.

- You can modify the settings of the deployment scheme in the deployment wizard. Step deploy/tosd_sendmail.dita#Receivinganemailnotification/first and possibly step deploy/tosd_sendmail.dita#Receivinganemailnotification/second are available.

Depending on your selection, you will receive an e-mail notification at the end of the deployment.

The notification e-mail looks like this:

The target with MAC *xxxx* / IP **xxx** has completed an activity *activity description*.

You can now deploy targets with the edited deployment scheme and receive e-mail notification at the end of the task.

Creating a sendmail TCP tunnel:

A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Note: OS deployment server supports only sendmail servers without authentication.

1. Go to **Server** → **Server parameters** → **Predefined channels**.
2. Click **New tunnel**.
3. In the TCP tunnel information screen enter,
   a. The name of the tunnel. In this case the name is `sendmail`.
   b. The host name or IP address of the target.
   c. The TCP port of the target.
4. Click **Save**.

You can now configure deployment schemes to send an e-mail notification at the end of a deployment.

Chapter 8. Working with VMWare ESX Server on x86 and x86-64 targets

This section provides information on how to work with the product to deploy VMWare ESX Server.

System profiles

A system profile is the partition layout and list of files to deploy an operating system, either by unattended setup or by cloning, from a reference target or from a reference image.

The main purpose of Tivoli Provisioning Manager for OS Deployment is to deploy an operating system on targets by replicating a reference system. However, unattended installation of operating systems is also possible. The latter case Tivoli Provisioning Manager for OS Deployment does not replicate a reference system, but merely provides the correct parameters to Windows or UNIX setup for a fully unattended installation.

There are a number of differences between an unattended installation and disk cloning. First, creating an unattended installation in Tivoli Provisioning Manager for OS Deployment is straightforward. All of the necessary tasks are performed on the server, using the Web interface. In contrast, a cloning-mode system profile requires you to configure a target, prepare it for cloning, and run the cloning process directly on the target. However, the native mode of operation of the product is centered around cloning-mode system profiles, because this method of deployment is faster than unattended installation. When deploying computers on a large scale, unattended installation is not possible. Novice users might start with creating unattended installation profiles because this is easier than cloning-mode profiles.

Note:

- To avoid failures in creating or deploying asystem profile, clean up the temporary directory inside the OS deployment server installation directory on a regular basis.
- To create or deploy a system profile from a physical or virtual machine at least 512 MB RAM is required.
- The exFAT filesystem is not supported.
- System profiles can have a maximum of 3 primary partitions. Therefore, you cannot clone a target with 4 primary partitions.

Creating an unattended setup system profile for VMWare

VMWare unattended system profiles must be created on a Linux target, running the web interface extension. It can either be the OS deployment server itself, or a remote target whose IP address must be entered in the profile wizard.

To create an unattended setup system profile for VMWare ESX 3.5, you must download the binary file named *ESX Server 3.5 Update 2 CD image* (596 MB). Creating the profile from *ESX Server 3i U2 Installable* (238 MB) results in a failed deployment.

© Copyright IBM Corp. 2006, 2010

To create a new system profile:

1. Go to **Server → OS deployment → System profiles**.
2. Click **New Profile**. A system profile wizard opens to guide you through the steps of creating a profile.
3. Select **Unattended setup** in the first pane of the profile wizard.
4. Select your operating system from the list and click **Next**.
5. Follow the instruction of the profile wizard.

On VMware ESX 4.0, the service console runs in a virtual machine. When you view the partitions of your system profile, this virtual machine is shown as a logical disk named *esx console*, which is not mapped to any physical device. You can therefore deploy the profile on a target with only one disk. Do not edit the *esx console* logical disk.

When your first unattended installation profile is created, you can use it to deploy targets.

Organizing and editing system profiles

After you have created a system profile, you can view it on the OS deployment server through the web interface. The profiles are listed on **> Server > OS deployment > Profiles** page, in the **System profiles** pane. Each blue jacket represents a system profile (that is, the hard-disk partition images).

If you want to organize your system profiles, you can create subfolders by following these steps:

1. Select the parent folder with a left mouse click.
2. Call the contextual menu with a right mouse click.
3. Select the **Add a new profile folder** menu item.
4. Enter the new folder name.
5. Click **OK**.

You can then move profiles (by dragging-and-dropping the profile icons) from the top folder, where they are automatically created, to the appropriate subfolder.

Editing system profiles

To display and edit the parameters associated with a given profile:

1. Double-click a system profile to open the **Profile details** page.
2. Click **Edit** on top of the parameter sections to edit the parameters.

Browsing partition files

You can browse partition images stored on your server.

1. Go to **Server → OS deployment → System profiles**. Double-click on a profile to view the details.
2. In the **Original partition layout** section, click **Browse image of primary partition 1**.
3. You can expand or update the whole partition or a part of it.
   - To expand the whole or part of the partition:
     a. Right-click the folder you want and select **Expand on local disk**.

b. Choose the computer where you want to expand and store the files contained in the selected partition.
c. Specify the destination folder where to extract the partition files.
d. Follow the instructions of the image wizard to expand the partition.

Note: You must expand the partition to an empty directory. If you select a folder that is not empty the extraction fails.

- To update the whole or part of the partition:
  a. Right-click the folder you want and select **Update from local disk**.
  b. Specify the source folder of the OS deployment server where your updated data are located.
  c. Specify the destination folder where to extract the partition files.
  d. Follow the instructions of the image wizard to update the partition.

On the **Partition image explorer** page, you can create a new directory by selecting **Add new directory** in the contextual menu. You can also modify or add files by selecting **Upload file** in the contextual menu.

Note: File upload is limited to 16 MB.

Changing the partition layout

Partition layout can be updated to resize partitions, assign mount points, change the file system.

Changing the partition layout in system profiles might render the profile unusable. It is recommended not to change the partition layout in system profiles, unless you know that the changes you want to make have no side effect.

In any case, do not:

- Add or remove a partition before the *root* partition.
- Transform a primary partition into a logical partition.
- Move a primary partition before *boot* or *root* to LVM.

Note: Changing the partition layout from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose either one or the other entry point, and then perform all your changes from that entry point.

Editing the partition layout allows you to:

- Add or delete partitions.

  Note: Adding or deleting partitions can lead to OS configuration problems, therefore this feature must only be used very carefully. To provide a better description to your profile, use the **Comment** field to write all necessary details.
- Resize a partition by dragging sliders, or by assigning it an absolute or relative size.
- Change the file system of a partition.
- Assign a mount point to the partition.

1. Click **Edit partition layout** on either the **Profile details** page or the **OS configuration details** page, **Disks** tab.
2.

- To add a partition:
  a. Click **Modify partition layout**.
  b. Click into an existing partition.
  c. Click **Add a partition** in the contextual menu.
  d. Indicate the partition properties, including a mount point and click **OK**.

     Windows In a Windows profile, if you do not provide a drive letter as mount point, one is assigned randomly. Letter *W* is not available.

     Linux In a Linux profile, do not forget to assign a mount point for the new partition. To be valid, this mount point must reference an existing directory in the main image.
- To resize partitions with the sliders, grab the slider to the right of the partition and drag it.
- To update all other parameters, select a partition by clicking on it, and select **Edit partition** in the contextual menu.

Modified partitions are aligned on megabytes rather than on cylinders. The following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

If you want to use the same system profile with two different partition schemes, you can also duplicate a system profile by right-clicking the profile name and selecting **Duplicate profile**. The copy shares the same image files, but can have a different partition layout.

Updating device mapping

Device mapping can be updated to force logical disks to point to specific physical devices.

Note: Updating device mapping from both the **Profile details** page and the **OS configuration details** page can lead to incorrect OS configurations and prevent OS deployment. Depending on your particular needs, choose one or the other entry point, and then perform all your changes from that entry point.

1. Go to **Server** → **OS deployment** → **System profiles** → **Profile details** or to **Server** → **OS deployment** → **System profiles** → **Profile details** → **OS configuration details**, **Disks** tab.
2. Click **Modify device mapping**.
3. Select to which physical device you want to map your logical disk. The column starting with **Disk 0** corresponds to an automatic detection of the first to the eighth disk, the column starting with **/dev/hda** corresponds to standard device names.

   Note: Spanning a logical disk on several physical disks is not currently available.
4. Click **OK**.
5. Repeat step 2 to step 4 for each logical disk for which you want to update device mapping.

If the new device mapping you selected is incorrect, you receive a warning message.

OS configurations and fixed common parameters

A system profile is the partition layout and list of files to deploy, while OS configurations are operating system parameters.

At the very bottom of the **Profile details** page, there is a list of the OS configurations that correspond to your profile.

You can define several OS configurations for each system profile and duplicate them. These copies share the same image files, and the same partition layout, but can have different target parameters. You must then assign new values to some of the OS configurations parameters to make the original OS configuration and its copies distinct.

If you want to automate the assignment of parameters to targets, you can view and edit the OS configuration you are about to deploy by clicking on its name in the **Profile details** page. You are now in the **OS configuration details** page. The information is divided into panels, each displaying sets of parameters. You can modify the parameters either through the web interface or by using a parameter file.

Editing OS configuration parameters in the web interface for VMWare

The web interface displays a number of OS configuration parameters divided into panes. These parameters can be edited in the web interface.

To edit parameters:

1. Click a tab to select the corresponding pane.
2. Click **Edit** on the banner of the section where you want to update parameters.
3. Modify the values.
4. Click **OK**.

Editing a VMWare parameter file

You can modify OS configuration parameters by editing a file. This option allows you to modify parameters that are not displayed in the web interface. However, you must be experienced to use this option advantageously, because Tivoli Provisioning Manager for OS Deployment does not provide any syntax checking of the file. Information about the file format and syntax can be found in the documentation of the operating system itself.

1. Click **Edit custom 'ks.cfg'** to edit the file to modify the size of the VMFS and VMKcore partitions if needed and to define a custom partitioning scheme when installing VMWare with scripted installation.
2. Type the parameters and their values in the syntax requested by the operating system, or copy and paste it from another editor.
3. Click **OK**.

Tivoli Provisioning Manager for OS Deployment merges the information of the edited file with the information provided on the web interface (default file). Unless otherwise specified, parameters specified in the default file override the content of the custom file.

In the following example, the following partitions are created:

- A `ext3` partition of 900 KB on the `sda` disk.
- A `vmfs3` partition of 50 MB is created on the `sda` disk.

- A vmkcore partition of 94 KB on the sda disk.

```
part /var --fstype ext3 --size=900 --ondisk sda
part None --fstype vmfs3 --size=50000 --grow --ondisk sda
part None --fstype vmkcore --size=94 --ondisk sda
```

Troubleshooting:

If the OS configurations in the deployed operating system are not what you expected, you must examine carefully the parameter files. They are the result of the merge between the custom file and the default file created. See the log file Windows/Panther/unattendGC/setupact.log for problems in the file merge.

Note: Ensure you specify the full paths for the commands you use in the unatted.xml file.

To troubleshoot OS configuration parameters after a failed deployment, see the /tmp/anaconda.log file.

Task templates

Task templates group together elements that can be customized on a target. These elements are mostly screen layouts, which condition the appearance of the target screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment.

A deployment scheme is a specific type of task template. Together with the target display screen layout, it contains other parameters for customizing a deployment on a target.

Note: Starting with Fix Pack, version 5.1.0.2, deployment schemes are considered to be a subset of task templates. The functions of deployment schemes have not been altered. To access deployment schemes, go to the **Task template** page, and select the **deployment scheme** folder.

The task template page of the web interface contains a task template tree in the left pane with seven folders. The content of the selected folder is displayed in the right pane.

There are seven task template folders in the tree. They are described here.

Deployment Schemes
Deployment schemes contain parameters that indicate how an OS configuration must be deployed on your target. The **deployment Schemes** folder contains at least the **Default** scheme.

Idle layout
The idle layout defines what is shown on the target when there is no pending task. The **Idle Layout** folder contains at least the **Idle state** layout.

Menu Layout
The menu layout defines how deployment menus are shown to the users. Menus are used when an OS configuration and on deployment CDs. The **Menu Layout** folder contains at least the **Menu** layout.

OS Detection Layout
The operation system detection layout defines the target display when a target is busy detecting the currently installed operating system. It is used

when creating a cloning profile from the web interface. The **OS Detection Layout** folder contains at least the **Detect operating system** layout.

Profile Creation Layout
The system profile creation layout defines the target display when a target is busy creating a new system profile. It is used when creating a cloning profile from the web interface. The **Profile Creation Layout** folder contains at least the **Creating cloning profile** layout.

Profile Restoration Layout
The system profile restoration layout defines the target display during the manual restoration of a system profile by the administrator.

Note: A system profile restoration is always performed as-is and must not be confused with an automated deployment resulting in a fully configured operating system installation.
The **Profile Restoration Layout** folder contains at least the **Default OS Restoration** layout.

State Capture Layout
The state capture layout defines the target display when a target is saving the operating system state for future redeployments.

State Restoration Layout
The state restoration layout defines the target display when a target is redeploying an operating system from a saved state.

When a task template is selected in the right hand pane, the bottom of the web interface contains a link to **Customize GUI**. Follow this link to modify the look of your target screen . When the selected task template is a deployment scheme, there are additional links to view and edit the current scheme.

Customizing a screen layout

You can customize the screen layout of a target.

To customize a screen layout:

1. Select the layout that you want to customize in the right pane of the **Task Templates** page of the web interface page.

   Note: An actual layout must be selected and not a layout folder (left pane)
2. At the bottom of the page, the screen layout is shown in reduced size. Click **Customize GUI** to open the screen layout editor.
3. The editor is composed of a left column, containing instructions, a *What-You-See-Is-What-You-Get* (WYSIWYG) view of the screen being edited and a bottom banner with action buttons.
4. Click on the action buttons or directly on the items that you want to modify to see their editable properties displayed in the left column. Make the wanted changes and then click **Save** to keep your new screen design. Return to the **Task Templates** page by clicking **Back**.

Creating and editing deployment schemes

By customizing your deployment schemes, you can adapt the way in which your predefined OS configurations are installed onto targets.

1. The easiest way to create a new deployment scheme is to run the deployment scheme wizard.Go to **Server** → **OS deployment** → **Task templates**. and click **New deployment scheme**.

2. Alternatively, you can modify an existing scheme by editing its parameters. To do this, select a scheme and click **View deployment parameters** and then use **Edit** in the banner on top of each parameter section. Some advanced deployment scheme features are available only in this mode and not through the wizard.
3. If you prefer using a wizard to edit your scheme, click **Edit parameters using a wizard**.

   The following parameters apply for simple one-time deployments and for redeployment operations.

   Description

   The first step is to enter a name for this deployment scheme. Make it explicit enough so that you can pick it easily when starting a deployment (the web interface does not show the settings in a deployment scheme, so the choice must be made by name only). Because deployment schemes determine how the computers are installed (and not what is being deployed), use a description such as `Multicast 50 targets` or `On-site` rather than the name of a OS configuration or of a group of computers.

   When the deployment starts

   You must determine if Tivoli Provisioning Manager for OS Deployment requires user interaction during deployment (to edit individualtarget parameters) or runs completely unattended.Select:

   Always edit target-specific parameters
   to have the opportunity to change the target parameters at every deployment. The parameter edition can be made either directly on the target computer or by double-clicking the target icon in the Target Monitor.

   Edit parameters for targets that are not yet in the database
   to be prompted only during the first installation of each target. Subsequent deployments for the same targets run unattended. This is the default choice.

   Never edit parameters
   to have unattended deployments if all BOMs have been previously entered in the database. Any missing entry leads to a red banner on the target and cancels the deployment for this target.

   You must also select how Tivoli Provisioning Manager for OS Deployment behaves when the model of the computer being deployed does not match the model of the computer on which the image was created. This feature requires DMI for hardware detection.Select:

   No if you know that all your system profiles are fully hardware-independent or for deploying universal images.

   Yes, display a warning
   if you want to see all possible OS configurations for a computer, but want to avoid mistakes. This choice can require user interaction and is therefore not appropriate for a fully unattended deployment.

Yes, abort the deployment
if you want to prevent anyone from using an OS configuration on a computer different from the one for which it was designed.

Use 'BIOS fallback MBR' to start PXE is used when PXE activation (the process of enabling PXE when booting on the hard-disk) does not work.

The PXE boot code manages the multiple reboots needed to install a computer. To manage these reboots, the PXE boot code must intercept the boot process of the computer at every boot.

- If the computer is configured to always start on the network (LAN device first in the list of boot devices), there is nothing to do, because Tivoli Provisioning Manager for OS Deployment is loaded into memory at every boot.
- If the computer is configured to start on the hard-disk, you can change the MBR of the hard-disk and make it point to the work partition at the end of the hard-disk. Tivoli Provisioning Manager for OS Deployment is then loaded from the hard-disk when the computer starts up, instead of loading the operating system. The disadvantage of this method is that, because the computer did not use the network card to boot, PXE is not available. To enable network access, PXE is activated with a special function in the PXE card that makes it behave as though the computer had booted on the LAN. However, this is not documented in PXE, and does not work on every network card. If the network does not support this, an error is raised, and access to the OS deployment server fails (the message **Network started**, followed by an error).

  When PXE activation does not work, you can write a special MBR telling the BIOS that the hard-disk is not a valid boot device. By default, the BIOS falls back to the next device in the list, which in most computers is the network. As a result, the computer boots on the network and has full access to the network. This is the purpose of the **Use 'BIOS fallback MBR' to start PXE** check box.

Data collection

By default, Tivoli Provisioning Manager for OS Deployment automatically populates the database with an inventory of the hardware setups of all deployed computers. For Windows, a software inventory can also be populated based on the registry. If you are not interested in using those inventories, or if your computers do not comply with any of the hardware detection standards, you can disable these features. Be advised that running the hardware or software inventory on thousands of computers can produce a huge database. This inventory is performed on locked screen.

Tivoli Provisioning Manager for OS Deployment centrally reports the status of the deployment of targets to the OS deployment server and to the server database. Additionally, if you want to keep the deployment logs and the list of software modules on each deployed target, you can specify a local path where the log files are to be stored. The path that you specify is relative to the root of the operating system on the target, for example, `/` on UNIX operating systems or `c:/` on Windows operating systems. In the deployment scheme details, the label of this field is **Save deployment log to**.

When the deployment is completed

When the deployment process is finished, you can select if you want to:

- Turn off the computer automatically (if supported)
- Boot the operating system automatically (this value might not make sense with some values of the previous setting)
- Display a green banner and wait for a manual shutdown

Network usage

Note: Multicast is available only if:

- The targets have an Intel x86 or x86-64 architecture
- Multicast is selected in the deployment scheme
- The subnet supports multicast traffic
- Multicast is not disabled in the boot options of the target
- The target is not a VMWare 3.0 guest.

Depending on the number of targets that you are deploying simultaneously on your infrastructure, you must select one of the following networking modes:Select:

Unicast
to deploytargets one by one, or if you cannot use multicast. When deploying several targets simultaneously in unicast, the deployment time increases dramatically with the number of targets, as the result of network saturation.

Multicast, without explicit synchronization
to use soft-synchronized multicast protocol. Using this protocol, every target independently starts downloading images at soon as it is ready, and continues with the deployment as soon as it has downloaded all required material. When two or more targets (using the same deployment scheme) are downloading files in parallel, they automatically share the same bandwidth. The fastest target has the priority for the choice of the next shared files to be sent by the server, but the slower targets can receive them if they need them. This is a scalable solution that allows for a rolling deployment scenario.

Multicast, with the following synchronization parameters
to use a classical replicated multicast method. This mode is adequate for installing computers in batches. Enter the replication parameters (for example, the number of targets to wait for before starting the download, and the maximum timeout before starting in any case). Tivoli Provisioning Manager for OS Deployment multicast protocol can accept new download targets even after the initial replication period is over, and integrate them seamlessly into the transfer.

Note: In the first stage of an OS deployment, there are two target synchronization stages. Therefore it might seem that the maximum timeout that was set before starting the deployment is doubled.

Vista 2008 Windows 7 You can decide to use a network share on the server to download the files to the targets , rather than

downloading the whole image to the hard disk of the target. Using a network share provides a shorter installation time. To use a network share:

- Select **Use network share** in the deployment scheme.
- Share the `files\global\partition` directory and provide at least read-only access to it.
- Enter the relevant server parameter in the **Network share module** section. Go to **Server** → **Server parameters** → **Configuration**. (See "Network share module" on page 314).

On-site deployment

If you are running a one-time deployment in a deployment center and do not want to use redeployment, leave the check box blank and click **Next**.

If you are running an on-site deployment, or if you plan to use redeployment, you can enable the advanced feature.

Redeployment option

Indicate if you want to keep the deployment image in a protected partition and the size of this partition. These options are valid only to configure the deployment scheme for redeployment. More information is available in deploy/tosd_redeplscheme.dita.

Note: The following parameters cannot be modified using the wizard. You must edit your deployment scheme parameters.

Request user confirmation

This parameters allows you to ask for user confirmation before running a deployment.

Unbind OS configuration at the end

This parameter, located in the **General settings** section, is set to **No** by default. Setting this parameter to **Yes** unbinds the OS configuration from the target at the end of the deployment. This OS configuration is not proposed the next time the target boots and, if no other OS configuration is bound, the target presents a locked screen to the user. Review the section on network boot scenario of the deployment process topic.

Unbind software module at the end

This parameter, located in the **General settings** section, is set to `No` by default. Setting this parameter to `Yes` unbinds the software module at the end of the deployment. This software module is not proposed and installed the next time a deployment is performed.

Vista 2008 Windows 7 **Disable user interaction during deployment**

This parameter, located in the **General settings** section, is set to `Yes` by default. If you set this parameter to `No`, you can obtain a command prompt by pressing Shift-F10 on the target computer during a deployment to modify deployment files.

Note: This parameter is relevant for Windows Vista/2008/7 deployments only.

Send mail at end

This option can be set only if a *sendmail* tunnel has been created.

Send mail to
This parameter is available only if **Send mail at end** is not set to **No**. Use this field to enter the e-mail address to which information must be sent at the end of the deployment.

Configure Network
This parameter, located in the **Network settings** section, is set to **Before software installation** by default. The network setting of the target are set before software modules are installed, enabling the product to use the network settings during the installation of the software modules. Set the parameter to **After software installation** if you want the network settings to be applied after software modules are installed, for example if you intend to physically move the target after deployment and want it to be configured with the network settings for its final location.

Disable cancel button
This parameter, located in the **Client Display** section, allows you to prevent users from aborting a deployment by removing the cancel button. Set the parameter to **Yes** if you want to remove the cancel button from the client display

You can use the newly created deployment scheme to deploy a system profile.

To delete a deployment scheme, select the scheme, then right-click it, and click **Delete**.

Creating media for deployment for VMWare

You can create deployment media such as CDs, DVDs, or USB drives to install machines without connecting them to the OS deployment server.

You can use this kind of deployment when there is no connection or connection to the OS deployment server is very slow.

Some typical situations are small branch offices with slow links and no local deployment server, isolated computers with no connection to an internal network, laptop users currently away from LAN or connected using a modem.

If the data you want to use does not fit on a single CD or DVD, use a USB drive.

Note:

- You must create the deployment media from an OS deployment server or a web interface extension installed on a computer with the same byte order (little endian or big endian) as the one on which you want to use the deployment media.
- Offline deployment media are not supported on virtual machines.

Creating an OS deployment USB drive with the wizard

Tivoli Provisioning Manager for OS Deployment can automatically generate deployment USB drives that replay the deployment process for a given system profile or for any kind of software modules available.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must be formatted as FAT32 or NTFS.

Note: SuSE Linux Enterprise Desktop cloning is not supported on USB drive deployments.

Refer to the *Troubleshooting and support* set of topics for information about problems or limitations related to deployments using a network boot USB drive, and to the product release notes or the readme file provided with the fix pack for the most up-to-date information related to problems or limitations.

The deployment USB drive is self-contained and can be used instead of a CD or DVD to provision a target entirely offline, without using the OS deployment server. These deployment USB drives can also be used to deploy computers without a PXE-compliant network adapter.

To create OS deployment USB drives:

1. Perform one of the following operations:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.
2. Click **Generate Media** or select **Create deployment media** in the contextual menu.
3. Select **Create a deployment USB key** to start the USB key wizard. Click **Next**.
4. Specify the operating system for which to build the CD or DVD. Select **Windows** to load a WinPE deployment engine, **Linux** to load a Linux deployment engine environment, or **Both** to load both.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. (*Optional*) Change settings for targets running the USB key that you are creating.

   Included objects

   When selecting objects to be included, be aware that:
   - The wizard displays all the deployment schemes, system profiles, and software modules currently stored on your OS deployment server.
   - At least one system profile and exactly one deployment scheme must be included in your image.
   - The software application order is automatically included.
7. If your USB key has already been used as a deployment media, you might choose to keep a shared repository of previous data to improve data regeneration speed. If you keep the existing shared repository, you might use more space on the USB key.
8. Plug your USB key into a machine running the web interface extensionand specify its address.
9. Choose the drive matching your USB key.
10. Click **Finish** to close the wizard.

Use the USB drive to deploy a given system profile or any kind of software module.

Creating an OS deployment USB drive with command lines

You can create an OS deployment USB drive that Tivoli Provisioning Manager for OS Deployment can use when a target cannot boot from the network.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must have boot capabilities and a FAT32 or NTFS filesystem. The drive must be already formatted; existing files on the partition are not deleted. USB keys already filled with a bootable operating system might not work.

Note: Refer to the *Troubleshooting and support* set of topics for information about problems or limitations related to deployments using a network boot USB drive, and to the product release notes or the readme file provided with the fix pack for the most up-to-date information related to problems or limitations.

The command line must be used only when the web interface is either inappropriate or unavailable.

Use this command line:

- On Windows operating systems:

```
rbagent.exe -s <OSD_server_ip_address>:<OSD_server_password>
rad-usbget <drive>
keepshared|delshared preferwpe|prefermcp nodes
```

Where:

OSD_server_ip_address
Is the IP address of the OS deployment server.

OSD_server_password
Is the password for the administrative user (typically `admin`) on your OS deployment server.

drive Is a drive letter of the Windows target where you run the `rbagent` command. The rad-usbget command adds requested files to the FAT32 or NTFS partition and makes it bootable. The drive must be already formatted. Existing files on the partition are not deleted.

keepshared
Keeps a shared repository of previous data to improve data regeneration speed. If you keep the existing shared repository, you might use more space on the USB drive.

delshared
Deletes a shared repository of previous data.

preferwpe|prefermcp
Defines if a Linux deployment engine or WinPE is loaded from the USB drive, when a target boots from this USB drive, without accessing the network. Only when Linux deployment engine or WinPE is running, does the target connect to the network and try to contact an OS deployment server. If you deploy only Linux, specify `prefermcp` to skip WinPE. You can specify `preferwpe` only if there is a WinPE deployment engine on the OS deployment server.

nodes Defines the deployment settings with a space-separated list of objects. Specify at least `DEPLSET:Default` for the deployment schema, and `PROFILE:SystemID` for the system profile.

You can now boot the target using the OS deployment USB drive instead of the network card. To use the PXE emulation USB key, insert the USB key into the drive

and restart the target. If your machine does not boot from the USB key, check the BIOS boot list to see if your optical drive is included in the boot sequence and is listed before the hard disk. Most machines also allow you to select the temporary boot device without changing the boot sequence in BIOS.

Creating OS deployment CD and DVD

Tivoli Provisioning Manager for OS Deployment can automatically generate deployment CDs and DVDs that replay the deployment process for a given system profile or for any kind of software modules available. You can use this feature to create OS deployment CDs and DVDs that can be easily sent through the Internet or by e-mail, to refresh a computer back to its initial working state after installation.

These CDs and DVDs can also be used to deploy computers without PXE compliant network adapter. The creation of DVDs and media spanning is supported. These media can be protected using an activation code preventing unauthorized personnel from using it.

To create OS deployment CD and DVD:

1. Perform one of the following operations:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.
2. Click **Generate Media** or select **Create deployment media** in the contextual menu.
3. Select **Create a deployment CD or DVD** to start the CD and DVD wizard. Click **Next**.
4. Specify the operating system for which to build the CD or DVD. Select **Windows** to load a WinPE deployment engine, **Linux** to load a Linux deployment engine environment, or **Both** to load both.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. Follow the wizard instructions to create an ISO image.

   Included objects

   When selecting objects to be included in the ISO image, be aware that:
   - The wizard displays all the deployment schemes, system profiles, and software modules currently stored on your OS deployment server.
   - At least one system profile and exactly one deployment scheme must be included in you image.
   - The software application order is automatically included.

   Hardware options

   In the hardware options settings some boot options can be customized. By default the options are unchecked but some special cases can require changes. In particular, if the CD or DVD is to be used on a USB drive or as a secondary drive, it might be necessary to specify the option **use BIOS for CD or DVD ROM access**. When this option is selected, on some hardware it might also be necessary to select **disable enhanced disk access** (for IDE CD or DVD) or **disable USB** (for USB CD or

DVD) to ensure that Tivoli Provisioning Manager for OS Deployment use of other IDE or USB devices does not interfere with the BIOS access to the CD or DVD. In addition, deploying from the second CD or DVD drive of a target only works if you can ensure that subsequent boots keeps booting on the same CD or DVD drive.

Security issues

For security issues, you might want to protect deployment from the CD with an activation code. When your computer boots on the CD, the activation code is required for the deployment to proceed.

You might also want to hide the content of the ISO image that contains sensitive information such as product keys. To do this, select **Hide the content of CD or DVD** in the CD or DVD Wizard. If you then try to access files in your ISO image, you see the content as `CDROM_content_hidden`.

Size of the ISO file

The wizard allows you to choose the size of the ISO images.

a. Enter the maximum size in the field displayed.

b. Click **Next** and the wizard starts to precompute the ISO file size.

The wizard displays the results for the number of disk images and the size required. You then have the option to:

- Download it directly from the server.
- Use the web interface extension
- Generate it on the server itself in the import directory.
- Generate it on another computer running the web interface extension

Note:

- When creating the ISO files, all objects of type *single file to copy*, *image headers*, and *WIM images* (which includes Windows Vista/2008/7 unattended setup profiles), are put on the first CD or DVD. Therefore, the first ISO file might grow larger than the requested spanning size if the total size of the files to be put on the first ISO requires it.

  For example, if you try to create an OS deployment DVD containing both Windows Vista/2008/7 unattended setup profiles, both profiles must be contained on the first ISO, but their total size is larger than 4 GB. Therefore, the ISO cannot be burned into a single layer DVD. In this case, either use a double layer DVD, or transfer the ISO without burning it.
- When deciding where to generate the ISO image, be aware that:
  - If the estimated size is bigger that 2 GB, do not use the link to download directly from the server, because of limitations of web browsers. An exception to this rule is Mozilla Firefox on Linux, which can extract files as large as 4 GB or more.
  - Because of file system limitations, do not extract files bigger than 4 GB on FAT32 partitions.

Use a CD creation tool to burn the ISO image onto disks.

Note: Vista 2008 Windows 7 Windows Vista/2008/7 unattended setup profiles contain at least one file larger than 1 GB which cannot be split. Therefore, ISO files containing Windows Vista/2008/7 unattended setup profiles must be burned on a DVD.

If you encounter problems when deploying from this CD or DVD on a virtual machine, make sure that the CD drive comes after the hard disk in the boot order.

Setting up an activation code

For security issues, you might want to protect deployment or booting from the CD with an activation code. When your computer boots on the CD, the activation code is required for the deployment or the network boot to proceed.

To prevent being asked several times for the activation code during deployment:

- The deployment scheme included on your deployment CD must have the network setting **Use 'BIOS fall back MBR' to start PXE** set to **No**.
- The boot order of your target must be set to hard disk first and you must boot on the CD manually the first time.
- To set up an activation code for the first time, when creating the deployment CD:
  1. Select **Include activation code protection** in the deployment media wizard.
  2. Enter and confirm the chosen password. You must remember this password if you want to obtain other activation codes for this CD.
  3. Set a password expiration date under **Valid until**.
- To obtain a new activation code, for example, if you must use the CD after the current activation code expiration date:
  1. Click **Generate Media** on the Profiles page to start the deployment media wizard.
  2. Select **Generate a new activation code**.
  3. Click **Next** and follow the wizard instructions to obtain your new activation code. You must remember the password given when creating the first activation code for this CD.

The wizard provides you with the generated activation code that you need when using the CD.

Deploying VMWare

A deployment is the process of installing an operating system on a target, and configuring the operating system for a specific user.

The deployment process

In Tivoli Provisioning Manager for OS Deployment, a deployment is made up of several steps that are automatically run in sequence without user interaction:

1. Hardware configurations are optionally deployed, for example, to create RAID volumes.
2. Partitions are created on the hard disk, and then formatted according to information contained in the system profile.
3. All deployment objects (system profiles, partition files, and software modules) are downloaded to a temporary storage location on the hard disk.

4. Operating system files are written in the hard disk partitions, creating a bootable operating system with files and applications configured by database bindings between the *target* and *software modules*.
5. Target-specific configuration, such as the *host name* or the *product key* are gathered from the database to create a textual configuration file used by the system preparation tool.
6. The operating system is started, allowing Sysprep or LinPrep to configure the operating system according to information stored in the Tivoli Provisioning Manager for OS Deployment database.
7. Additional software is optionally installed, if it must be installed after the operating system.
8. The temporary storage location is cleaned. Installation files are removed.
9. Tivoli Provisioning Manager for OS Deployment takes control again when Sysprep or LinPrep has completed and rebooted the target, and displays a message indicating the status of the deployment.

When the deployment is complete, the operating system is installed and ready to be used by the user defined for this target in the database.

Network boot scenarios

Depending on the number of OS configurations bound to a specific target, a target behaves differently when it boots on the network:

- If no OS configuration is bound to the target (for example, when a target starts for the first time and has not been configured), a special screen is displayed that asks the administrator to configure an OS configuration binding for this target on the OS deployment server. Deployment is not possible until an OS configuration is bound to the target.
- If one or more OS configurations is bound to this target, but no deployment has been scheduled on the server, a screen is displayed with a list of all the OS configurations bound to the target. Clicking on an item in the list starts an interactive deployment for the selected OS configuration, using either the **Default** deployment scheme (if no deployment scheme has been configured for this target), or the deployment scheme used during the last deployment.
- If one or more OS configurations are bound to this target, and a deployment has been scheduled on the server for a specific OS configuration, the target immediately starts the deployment without requiring any user intervention.

Deployment requirements

To start a deployment on a target, several elements must be present in the database.

The following elements are required:

- A *deployment* scheme associated with the target to deploy. The deployment scheme determines how to deploy the operating system on the target. If there is no association between a deployment scheme and the target to deploy, Tivoli Provisioning Manager for OS Deployment automatically uses the **Default** deployment scheme.
- An operating system *configuration* that is used to select which operating system to install. If there is no OS configuration associated with the target to deploy, the deployment does not start.

- Optional *software modules* to install in addition to the operating system during the deployment process. If there is no software module associated with the target to deploy, the operating system image is deployed without modification.

The OS configuration and the software modules can be considered to be the *content* of the deployment, while the deployment scheme is the *how* of the deployment.

The database keeps information about associations (*bindings*) between targets and deployment schemes, between targets and OS configuration, and between targets and software modules. These bindings can be configured manually or with binding rules (for example, `deploy configuration windows XP on targets whose model name starts with Dell`).

The minimal binding required to start a deployment is an OS configuration. If no configuration is bound to a target, the deployment does not start. In practice, however, Tivoli Provisioning Manager for OS Deployment always asks for an OS configuration and deployment scheme when beginning a deployment.

Tools to start and configure deployments

Bindings between targets and deployment elements are necessary to perform a deployment. You can create and edit these bindings in the **OS configurations** panel of the **Target details** page.

The Target Monitor provides functions to prepare a deployment, start a deployment, follow the progress of a deployment, and organize targets.

Binding rules, used to create permanent implicit bindings between targets and deployment elements without having to explicitly create the binding for each target, are created using the web interface. OS configurations and software modules contain a specific section at the very bottom of the **Details** page for creating automatic binding rules.

Starting a one-time deployment

You start deployments in the web interface by indicating on which target or targets the deployment must occur.

VMWare ESX 3.x

It is recommended for the target on which you deploy VMWare ESX3.x to have two physical disks. On the first disk, the deployment installs the ESX operating system. On the second disk, the OS deployment server installs VMFS and VMKcore partitions.

If your target does not have two physical disks, a network storage must be used as the second disk.

VMWare ESX 4.0

The DHCP server must be configured to give a new IP address to the target on which you deploy the VMWare ESX 4.0 system profile, different from the one used for network booting. This second IP address is used by the console virtual machine within the hypervisor. You can use an open DHCP server, or a DHCP server with a range of *free* IP addresses in the pool, for example.

It is not possible to use reservations, as the MAC address of the console virtual machine cannot be known in advance.

To start a deployment:

1. Select a single target or multiple targets on the Target Monitor page. To select multiple targets or deployment, select an administrative group, a custom list, a subnet, or click on individual target names while holding down the Ctrl key.
2. Select **Deploy now** in the contextual menu.
3. In the first screen of the deployment wizard, you can choose to use the same deployment parameters as the previous deployment.
4. Select **Simple deployment** and click **Next**
5. Follow the deployment wizard instructions to select a deployment scheme, an OS configuration and optionally software modules, and to set up deployment options.

   Note: The Wake on LAN options are available only on Intel targets.

During the deployment of VMWare ESX 4.0, soon after the first boot into the ESX operating system, errors may show on the screen of the target.

```
[ERROR] open: no such file or directory
[ERROR] rtkh_open: cannot open /etc/rbotmp/rbagent.trc
[ERROR] rtkh_open: cannot open /etc/rbotmp/rbagent.log
```

These errors are normal and can safely be ignored.

When the deployment is complete, the server either displays a green banner on the target, boots in the operating system, or powers the target off, depending on how the deployment scheme is configured.

After deployment, the following warning message might appear in the logs and can be safely discarded. `Warning: partition x does not end at a cylinder boundary`. Partitions are aligned on megabytes rather than on cylinders. Aligning on megabytes is recommended by virtualization companies because it is safer when you deploy on both physical targets and virtual machines. The only drawback is an incompatibility with DOS.

Monitoring deployments

There are several ways available to monitor the deployment progress.

Monitoring deployment progress with theTarget Monitor

You can use the Target Monitor to monitor deployments remotely. Information is located on the **Target Monitor** page and on several tabs of the **Target details** page.

On the **Target Monitor** page, the target color changes during the deployment. When PXE is activated, targets are monitored on a regular basis. The color of the icon is updated as soon as the status changes. By pointing to the target icon, you can get a description of the target status.

Note: A successfully deployed computer can continue to have a yellow icon (indicating that the deployment is still in progress). This reflects a PXE activation problem. The computer, having booted on the hard disk, is not using the network to inform the OS deployment server of its status. To remedy this, select the **Use 'BIOS fallback MBR' to start PXE** check box in the deployment scheme wizard. This forces the computer to boot through the network first.

If the deployment scheme used is configured to collect inventory information about target hardware (which is the default), you can see information about target

hardware in the **Inventory** panel of the **Target details** page for that target (double-click on the target to go to the details page).

At the end of the deployment, the target icon shows either a green screen (success), or a red screen (failure). The deployment logs stored on the OS deployment server provide information about the deployment process. They are particularly useful in case of deployment failure to track its cause. To access the logs, double-click the wanted target. This opens the **Target details** page. Select the **Logs** tab to display a list of logs. To view a specific log, click its description. To download it, click **download** immediately after the log description.

Note: Logs are text files with UTF-8 formatting. If you are using a Windows operating system, you can view log files adequately by opening them in Microsoft WordPad.

There is only one log file for each deployment. This log file contains information about the different stages of the deployment process, including reboots and information provided by the operating system being deployed.

If any log information needs to be propagated to the OS deployment server outside of any task, an *idle* log file is created to store this information. The idle log file is created on demand and does not therefore exist for all the targets.

Another place of interest for information about a current deployment or another current task is the **Task history** tab, where each task of the target is listed. For each current task, the following information is provided:

- Description
- Status
- Scheduled date
- Start date
- Progress rate
- End date
- Download link to the log file
- Download link to the task file
- Download link to the `bom` file

Note: You must scroll to the right of the **Task history** tab to see all the fields.

The log file contains the target log. The task file contains all the parameters of the task. The `bom` file contains target-specific parameters for the given task.

The log file, the task file, and the `bom` file are needed by the development team to fix defects. Make sure that you download these three files if you suspect the presence of a defect in the software.

To cancel or destroy a task, select the task and select **Cancel target task** or **Destroy target task** in the contextual menu. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on running tasks, because the task destruction can cause strange behavior.

To view tasks scheduled for a later time, go to the **Tasks** page.

Monitoring deployment progress with the Tasks page for VMWare

The **Tasks** page is also a useful source of information to monitor a deployment (and other tasks). You can also cancel tasks from there.

Go to **Server** → **Server history** → **Tasks**. .

The description field of each deployment in the **Tasks** page is headed by the keyword *Deploy* for easy retrieval. The information provided includes

Description
Is headed by specific keywords, indicating the type of task. *Deploy* is the keyword for deployment tasks.

Execution
Is the scheduled date and time for the execution of the task.

State Uses icons to represent if the task is pending, in progress, completed, and so on. If in doubt to the meaning of a state icon, browse over it to get a state name.

Progress
Indicates the rate of completion of the task as a percentage.

Expire Indicates when the task information is going to be removed from the page.

Tasks are expandable by clicking their **+** sign. An expanded task displays information about its targets. The target information fields are:

- IP address
- Hostname
- Start date and time of the task
- State
- Progress rate
- Status date

If, for any reason, you want to cancel a running or scheduled task, you can easily do so by following these steps:

1. Expand the task
2. Select the target for which you want to cancel the task
3. Select **Cancel task** from the contextual menu
4. It is also possible to *destroy* tasks. When you destroy a task, all its records and files are permanently deleted. Use this option with caution, especially on a running task, because its destruction can cause strange behavior. To permanently delete tasks:
   a. Select one or several tasks. To select multiple tasks, use the Shift key for a range of tasks and the Ctrl key for individual tasks.
   b. Select **Destroy task** from the contextual menu

Receiving an e-mail notification

To receive an e-mail notification at the end of a deployment, you must configure a TCP tunnel called *sendmail*.

To receive an e-mail notification at the end of a deployment, you must have configured a *sendmail* TCP tunnel.

Note: The OS deployment server supports only sendmail servers without authentication.

There are two options to configure a deployment to receive an e-mail notification:

- You can edit the deployment scheme used for deployment to include the notification information.
  1. Go to **Server** → **OS deployment** → **Task Templates**.
  2. Select **Deployment schemes** and double-click a specific deployment scheme name to edit it.
  3. Click **Edit** in the **General settings** section.
  4. Under **Send mail at end:**, select the type of notification that you want. You can choose among:
     - **No**
     - **Whatever the notification is**
     - **If deployment is successfully completed**
     - **If deployment failed**
  5. If you selected a notification, you must now enter a valid e-mail address to which the notification is sent, under **Send mail to:**.
- You can modify the settings of the deployment scheme in the deployment wizard. Step deploy/tosd_sendmail.dita#Receivinganemailnotification/first and possibly step deploy/tosd_sendmail.dita#Receivinganemailnotification/second are available.

Depending on your selection, you will receive an e-mail notification at the end of the deployment.

The notification e-mail looks like this:

The target with MAC *xxxx* / IP **xxx** has completed an activity *activity description*.

You can now deploy targets with the edited deployment scheme and receive e-mail notification at the end of the task.

Creating a sendmail TCP tunnel:

A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Note: OS deployment server supports only sendmail servers without authentication.

1. Go to **Server** → **Server parameters** → **Predefined channels**.
2. Click **New tunnel**.
3. In the TCP tunnel information screen enter,
   a. The name of the tunnel. In this case the name is `sendmail`.
   b. The host name or IP address of the target.
   c. The TCP port of the target.
4. Click **Save**.

You can now configure deployment schemes to send an e-mail notification at the end of a deployment.

Bindings created during deployment

The Target Monitor creates a binding between the OS configuration chosen for the deployment and the targets being deployed. This binding is added into the database and can be later removed using the Target Monitor.

Because at least one configuration binding now exists, targets that have been deployed no longer show the locked screen. They show a boot menu with a list of the OS configurations that are bound to the target. This allows the target user to manually restart the deployment of an already deployed OS configuration by clicking on the corresponding line in the menu.

You can remove, add, or modify OS configurations and software bindings using the Target Monitor.

Chapter 9. Multiple server architecture

A key with parent and child servers is keeping them replicated in order for individual targets to be deployed with the appropriate content.

When building a multiple server architecture, there are two main design choices

- Several OS deployment servers connected to a single, centralized database.
- Several OS deployment servers connected to several databases.

Setting up the architecture and then replicating the information from one server to another depends on these options. More details on the two different architectures and how to set them can be found in the Tivoli Provisioning Manager for OS Deployment Installation Guide, in appendix A.

It is important to remember the following points when performing replication online:

- Each subordinate server needs to download files from its parent server. This means that the parent server must be up and running during the whole replication process.
- Replication can be scheduled for a specific time and repeated at an interval specified in days. These settings are set on the subordinate server. When set, the replication process becomes autonomous and can be performed without human intervention. Drawbacks to this are the relative loss of control over the process, and network and processor usage.
- Server replication is performed by copying files from the parent server to the subordinate server. A selection can be performed on the kind of information that must be replicated. Files that have been modified are copied over.

Note: If you use the `config.csv` configuration file to setup your server architecture and its options, do not use the links provided on the **Server → Server parameters → Server replication** page to create your architecture. Contradicting information provided on the web interface and through the `config.csv` file leads to unexpected behavior.

Server roles

In a multiserver architecture, roles are associated with OS deployment servers.

The server role of an OS deployment server in a multiserver hierarchy depends on whether the server has its own database and whether targets are replicated.

Server role information is displayed in **Server → Server parameter → Server replication** when a server is selected in the hierarchy.

Single database architecture and multiple database architecture must be considered separately. In the diagrams, servers are grouped according to the database they use.

© Copyright IBM Corp. 2006, 2010

Single database architecture

In a single database architecture, there are only two possible server roles as exemplified in Figure 5:

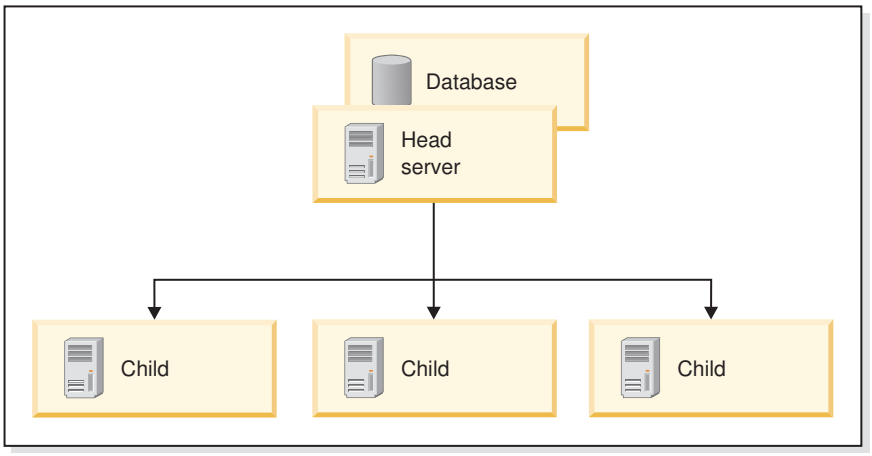


Figure 5. Single database architecture

Head server
The server at the top of the hierarchy.

Child All other servers.

Multiple database architecture

The structure of a multiple database architecture is more complex and more roles can enter into play, as exemplified in Figure 6 on page 267.

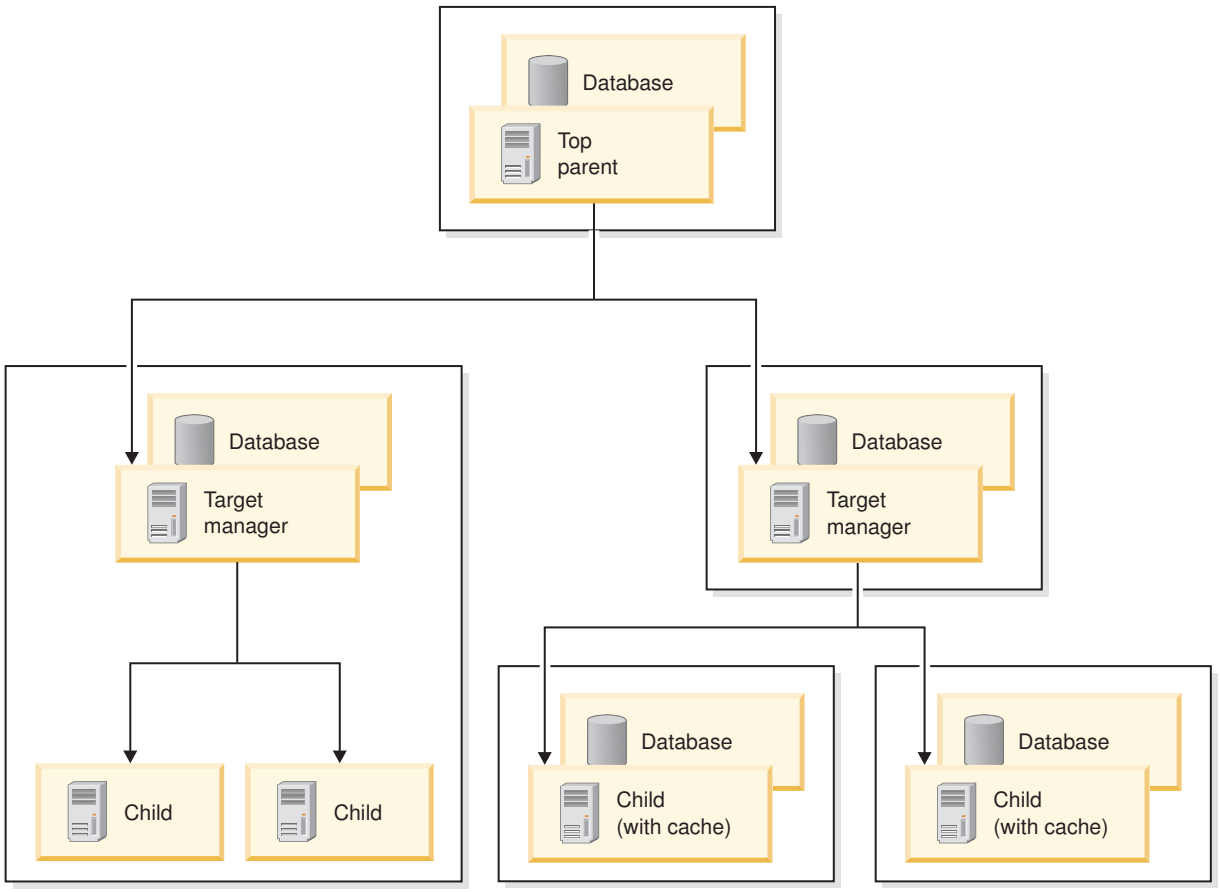


Figure 6. Multiple database architecture

Top parent
The server at the top of the hierarchy. It manages its database and does not have a parent database.

Target manager
A replicated server, with its own database, and which manages its targets, that is, it does not report target status to a parent server.

Child A server without its own database, as in the single database architecture.

Child (with cache)
A server with its own database, with full replication, and at the bottom of the hierarchy.

OS deployment server replication

Replication is the means to keep databases, files, and information up-to-date from parent to child servers. It can be performed through several mechanisms.

Replicated objects

The objects that are replicated from a parent to a child OS deployment server are the following:

- Deployment schemes
- Hardware configurations
- Software modules

- System profiles

Targets and server tasks are not normally replicated.

Main steps in server replication

Database replication for multiple database architecture only
A verification is performed on a regular basis between parent and child databases. Any discrepancy in the databases indicates that the files on the child server are not up-to-date. The child database must be updated. The default interval between two verifications is one minute, by default, but this can be configured in the `config.csv` file.

File checking against the database
A verification is performed between the files on the OS deployment server and the database. The web interface reflects any need of file replication.

File replication
Once the database verification has uncovered file discrepancies, the new and updated files must be downloaded from the parent server to the child server.

Server replication techniques

There are several ways to perform server replication depending on whether you use a single database or multiple databases, the type of network connectivity between the servers and the databases, and if you keep a strict top-down hierarchy between your servers.

Online, with a single database
If you have a single database and a good network connection between your servers and your database, you can opt for one of the following replication mechanisms:

- Automatically each time needed, with a `config.csv` file
- Automatically at scheduled times
- Manually using the web interface or a command-line (web interface extension).

Online, with multiple databases
If you have multiple databases and good network connectivity between your servers, you can set up replication through the `config.csv` file.

Offline, with the web interface extension
If you cannot ensure that your servers are always connected, you can replicate with the web interface extension and a specific package called, `sync.pak`. This method assumes a strict top-down hierarchy between your servers. All the deployment objects located on the parent server are replicated down to the children.

Offline, with RAD files
If you cannot ensure that your servers are always connected, but want to replicate only some objects from one server to another, you can export RAD files from one server and import them onto another.

Online, one time replication
In some cases, you can perform a one-time replication between servers, for example, if you want to change the database, or if you want to make a copy of a production server to perform tests on the copy. This requires reliable network connectivity.

Online, with the Java API

To replicate specific objects from one OS deployment server to another, you can use the Java API. See the documentation on using the Java API in the product in the section"Java API" on page 357.

Multiple server architecture considerations

If you delete an object from a child server, the parent server is not aware of the deletion. When you replicate, the object is recreated on the child server.

Replication is performed top-down, from parent servers to child servers.

Tasks must be started at the appropriate level in the server architecture. A task cannot normally be initiated on a parent to be performed on a child server.

To keep production servers in a clean state, it is recommended to create and test all replicated objects on a dedicated test server. When an object is ready for production, export it to a RAD archive and import it at the right level of the production hierarchy.

Multiple database architecture considerations

When an object is created, modified, or deleted on a parent server, one *PollInterval* should elapse to allow for the propagation of the changes to the database before replication is triggered. If the database is updated during the replication, the replication tasks fails.

When a child server is stopped for longer than three times the *PollInterval* (*PollInterval* is 1 minute by default), it performs a full refresh, checking the database, all objects, and all files when restarting.

The value of *PollInterval* can be updated in `config.csv`.

If you delete an object from a child server, the parent server is not aware of the deletion. When you replicate, the object is recreated on the child server.

Targets known to a child server are unknown from its parent because target information is stored in separate databases and is normally not replicated. This implies that you cannot capture an image or clone a system profile from a parent server if the reference target is known only by the child server.

Building a hierarchy between two or more servers with heterogeneous databases

When you build a hierarchy between two or more servers with heterogeneous databases, such as a Windows parent with a DB2 database and a Linux or UNIX child with an Apache Derby database, ensure that the database replication is performed using the JDBC gateway instead of the ODBC gateway. Use the ODBC gateway only if all servers run on Windows.

Moreover, all databases must use the same collation.

Exceptions to the replication flow and to the replicated objects

Under specific circumstances, it is possible to replicate some target information from parent to child servers, and from child to parent servers. In the same circumstances, it is also possible to replicate a deployment task. For this, you must do the following:

- Set up replication for a **child (with cache)** server. This server is at the bottom of a multiple database architecture, it has its own database which it does not share with another server.
- Use a text file (`config.csv`) to configure the replication.
- Set flag `h` in AutoSync, in the configuration text file.

In this particular configuration, target information is replicated top-down and bottom-up. A target added to a child server can therefore be known to its parent server. Moreover, a deployment task can be started on a parent server to be run by a child.

Replicating OS deployment servers with a schedule

To replicate your OS deployment servers regularly, you can set up a replication schedule, indicating the frequency of the replication

If you have a hierarchy of more than two levels of parent and child servers, the scheduling must match the hierarchy. Top servers must be replicated first, and child servers after.

- With a multiple database infrastructure, edit the `config.csv` file to include the new schedule.
- With a single database infrastructure, use the web interface to set up a replication schedule. For each child server in the hierarchy
  1. Go to **Server** → **Server parameters** → **Server replication**.
  2. Click **Set up a replication schedule**.
  3. Enter the start date and time, and the frequency of the replications. As child servers must be replicated after their parents, you must use the same or a lower frequency than the replication schedule on the parent server.
  4. Click **OK**.

Replicating an OS deployment server once manually

Replicating OS deployment servers can be done manually with the web interface or with the web interface extension.

- With the web interface:
  1. Go to **Server** → **Server parameters** → **Server replication**.
  2. Select the OS deployment server you want to replicate.
  3. Click on the link to replicate the server. The exact wording of the link depends on whether the server needs to replicate, and on the position of the server in the hierarchy.
- With a command line and the web interface extension:
  1. On the child server, open a command line shell.
  2. Go to the directory where rbagent is located.
  3. Run `rbagent rad-srvsync`.

Now, you can replicate the children of this OS deployment server. You can also setup a replication schedule.

Replicating offline with the web interface extension

Replication with the web interface extension (RbAgent) relies on the `sync.pak` package. This package must be located with the other `.pak` files, in `C:\Program Files\Common Files\IBM Tivoli\packages` for Windows, on both parent and child servers.

The main concept behind this replication process is to keep a list of important parent server states and to create differentials between states. These differentials can then be transferred from parent to child to update the child server.

Attention: Between the creation of a checkpoint and the end of the creation process of the corresponding differential, steps 1 and 2 of the procedure, the objects on the parent server must not be modified. It is prohibited to create or modify any deployment object including, but not limited to, system profiles, software modules, deployment schemes, and hardware configurations.

To perform replication with RbAgent:

1. Create a new checkpoint on the parent server when it is in a stable state. A new checkpoint must be created after major changes on the parent server. Checkpoint `0` (zero) refers to the initial state of the server and is always present.
2. Create a differential between a chosen checkpoint state and the latest checkpoint state of the parent server. This builds a `.rad` file (or several `.dat` files if you have indicated a file size limit) in the `TPMfOS Files\import` directory.

   You can perform this step synchronously (RbAgent waits until the task is complete before returning control) or asynchronously (RbAgent returns control immediately). In the asynchronous mode Tivoli Provisioning Manager for OS Deployment prevents you from launching two `.rad` file creation processes concurrently.

   Note: If changes have been made on the parent server since the last checkpoint, you cannot create a differential with the last checkpoint as endpoint. You must first create a new checkpoint reflecting the current state of the parent server.
3. Transfer the `.rad` file from the parent server to the child server. Tivoli Provisioning Manager for OS Deployment does not interfere in this transfer process.
4. To replicate your child server, copy your differential file from its current location (either the parent server or a local directory) to the specific `TPMfOS Files\import\auto` directory. This directory is automatically created when the `sync.pak` package is present. Tivoli Provisioning Manager for OS Deployment checks for changes in the `TPMfOS Files\import\auto` directory automatically. Whenever a new file is found, it is checked for coherence (if it is a `.rad` file), or recomposed as a `.rad` file (if it is a series of `.dat` files). The file is renamed with a `.ok` extension if the process succeeded, or with a `.err` extension in case of error.
5. The contents of the `.rad.ok` file are automatically replicated with the shared repository if the checking process is successful.

The checkpoint-based replication ensures that server files are up to date. This is enough if both OS deployment servers share the same database. If the OS deployment servers are using separate databases, it is necessary to replicate the

database records as well. This can be achieved through export files as well if needed, using the web interface extension command-lines rad-exportdb and rad-importdb.

You can customize the files that are replicated by indicating which folders are concerned. To do so, edit the [RSyncConf] section of the TPMfOS Files\global\serverstate\sequence.ini file where the list of folders has been initially populated. Subfolders are recursively and automatically included.

Replication with the web interface extension

The replication process has been redesigned for improved performances in branch office scenarios. Instead of file copies, replication of shared repository files is possible with the web interface extension (RbAgent) and a specific package implementing specialized command-lines for RbAgent (sync.pak).

The load on the parent server is reduced. Control over the performance of the replication process is split into operations on the parent and those on the subordinate server. The parent server does not must be running when a subordinate server is replicating itself.

Specific RbAgent commands

The sync.pak package implements several RbAgent commands that you must use for this specific replication process. With these commands, you can export and import a database content, create new checkpoints, list existing ones, and create .rad files.

Database replication commands

rad-exportdb filename
: Exports a RAD file named filename. This command dumps a single file of all database records that describe the deployment objects in a server at a given time. The file name is the only argument that is required to export the database to. The database is exported to global/rad/.

rad-importdbfilename
: Imports a RAD file. This command imports the differential.rad file generated by sync.pak and the database exportfile into a remote server so that the remote server can be upgraded to the exact same content. The filename is the only argument that is required to import the database from. The file must be located in TPMfOS Files/global/rad/.

File replication commands

sync-seqidlist
: Returns the list of all valid checkpoints. These checkpoints are extracted from the server file system. The command typically exits with the status 0. If the command exits with status 1, an error has occurred and is described in the standard output.

sync-newseqid new-sequence-id | auto [force] [TaskID=n Description=d]
: Creates a new checkpoint. new-sequence-id is a string identifying the new checkpoint; auto is the keyword that generates a new sequence ID automatically; force is an optional keyword that overrides an existing checkpoint; n is an unsigned 64-bit integer in decimal form used for status reports; d is a freely usable string, used for status reports. The command typically exits with the status 0. If the command exits with the status 1, an error has occurred and is described in the standard output. Checkpoint information is stored in TPMfOS Files/global/serverstate.

sync-radget newdiff.rad from-seqid | 0 [-split n] [TaskID=m Description=d]
Synchronously creates a differential RAD file. `newdiff.rad` is a RbAgent URL. For example, `local://root/c$/temp/diff-0-1.rad`; `from-seqid` is the reference checkpoint from where files can be omitted; 0 is the initial checkpoint; `-split n` optionally forces splitting the file into fragments of `n` MB. `m` is an unsigned 64-bit integer in decimal form used for status reports; `d` is a freely usable string, used for status reports. The command typically exits with the status `0`. If the command exits with the status 1, an error has occurred and is described in the standard output. The command creates a `newdiff.rad` file. With option `Split`, several files can be created. They are automatically renamed. For example, `newdiff.rad` becomes `newdiff-rad-x-of-y.dat`. Each fragment finishes with an MD5 and a signature (20 bytes). With the option `Split`, `newdiff-rad.dsc` is a description of the fragments. The command cannot start if the server files do not match the last checkpoint. Running `sync-newseqid` before `sync-radget` is a prerequisite.

sync-srvradget newdiff.rad from-seqid | 0 [Split=n] [TaskID=m Description=d]
Asynchronously creates a differential RAD file. This is the asynchronous version of the `sync-radget` command. Another important difference is in the definition of the parameter `newdiff.rad` which is here a path relative to `c:\TPMfOS Files\import`. If the command returns after several minutes, the OS deployment server is not responding. Although asynchronous, two or more `sync-srvradget` commands cannot run concurrently.

Replicating one time in command line

In some cases, you must replicate your server once, for example if you want to change your server hardware or your database, or if you want to make a copy of a production server to run tests on it.

One time replication is available from a parent version equal or lower than the child version, namely

- from version 5.1.1 interim fix to version 5.1.1 interim fix,
- from version 5.1.1 interim fix to version 7.1
- from version 7.1 to version 7.1.
- from version 7.1 to version 7.1.1
- from version 7.1.1 to version 7.1.1
- from version 7.1.1 to version 7.1.1.1

You must be aware that one time replication deletes the content of the server on which the operation is performed. Take all precautions to ensure that you have nothing valuable on the OS deployment server from which the command is run or on its database.

The command line must be run on the receiving (child) server.

1. Open a command line shell.
2. Go to the directory where rbagent is located.
3. Run `rbagent rad-replicate <parent-ip>` where `<parent-ip>` is the IP address of the parent server you want to replicate.

Now, you can use your newly replicated server to run tests without impacting your production server, or to replace obsolete hardware.

Server replication status and logs

You can see the server replication status from the Web interface. Go to **Server** → **Server parameters** → **Server replication**. The icons on this page inform you visually of the replication status of your servers. Logs also contain information about the replication process.

Status

- On the lower left hand-side of a server icon, the **up/down indicator** is displayed. The status indicator can take two different values:
  –

    A blue circle with a light center, indicating that the OS deployment server is up and running.
  –

    A black dot, indicating that the OS deployment server is down.
- On the lower right side of a child server icon, the **replication status indicator** is displayed. The status indicator can take three different values:
  –

    A cross in a red dot indicates that the selected child server is not up-to-date with its parent. Files are missing on the child server; the child server must be replicated with its parent before any action is performed.
  –

    A yellow triangle with an exclamation point indicates a warning. A discrepancy was discovered between the child and the server files. Some files can have been updated or added on the parent server.

    Click **Object version** to view which deployment objects are not up-to-date. If you plan to use any of these objects, you must replicate your server first. This page contains yellow triangles if SSL is not disabled or if the servers are temporarily unresponsive.
  –

    A green dot with a white check mark indicates that the child server is up-to-date with its parent.

Note: When a server is down, it keeps the replication status indicator it had when it was last running. Replicating while a server is down is not possible.

Logs

Whether server replication is activated manually (using the replication link in the web interface or using the web interface extension), through scheduling in the web interface, or with the `config.csv` file, server replication corresponds to a set of tasks. Several logs are available to monitor the replication process and these tasks.

Sync log file or `files/logs/sync.log`
This log file contains information specific on replication: checking files,

finding them or not, copying them, and so on. Its content can be viewed in the **Server log files** page of the web interface.

files/global/hostactivities
This directory contains the list of all target tasks. The content is merged with the information provided by `activities.log` and can be viewed from the **Tasks** page of the **Server history**: select the task and choose **Show log file**.

Switching from an ODBC to a JDBC gateway

In a server hierarchy with heterogeneous databases, the replication process works only if you use the JDBC gateway instead of the ODBC gateway.

To switch from an ODBC to a JDBC gateway, you must perform the following steps:

1. Create a database and ODBC source.
2. Install Tivoli Provisioning Manager for OS Deployment.
3. Run `net stop remboserver` to stop both server and database gateway.
4. Run regedit to modify the registries:
   a. Delete the **SubService** entry that starts the ODBC gateway. An example is:
   ```
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemboServer\Parameters\
   _SubServices]"RemboODBC"=dword:00000000
   ```
   b. Add the **SubRun** entry that automatically starts the JDBC gateway. An example is:
   ```
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemboServer\Parameters\
   _SubRun]
   "dbgw"="\"C:\\Program Files\\Java\\jre1.5.0_11\\bin\\java.exe\" -Xrs
     -cp \"C:\\Program Files\\fichiers communs\\IBM
     Tivoli\\dbgw.jar;C:\\Program
     Files\\derby\\db-derby-10.2.2.0-bin\\lib\\derbyclient.jar\"
     -Djdbc.drivers=org.apache.derby.jdbc.ClientDriver
     com.rembo.dbgw.Dbgw"
   ```
5. Create or update the `config.csv` file to use the JDBC syntax.
6. Copy all jars needed for accessing the remote databases into the `..\\Common Files\IBM Tivoli` directory.
7. Run `net start remboserver` to start both server and database gateway.
8. Check that the JDBC gateway is started automatically according to the **SubRun** registry entry.

Removing an OS deployment server from the hierarchy

Removing an OS deployment server from a multiserver hierarchy depends on whether you are using a single database or a multiple databases infrastructure.

- In a single database infrastructure

  Note: Removing a running OS deployment server from the database in a single database infrastructure can lead to unknown side-effects.

  1. Make sure the OS deployment server you want to remove does not have any child. If it has any, assign it a new parent.
  2. Go to **Server** → **Server parameters** → **Server replication**.
  3. Select the child to be removed.
  4. Click **Make this OS deployment server a standalone OS deployment server**.

5. On the former child, run the `rbagent rad-resetscope` command to set the scope of all the objects on the server to local. Otherwise all replicated objects remain read-only.

- In a multiple database architecture, with a running OS deployment server to be removed
  1. Edit the `config.csv` file of the OS deployment server you want to remove. Change the *MasterIP* value to `SELF` and *AutoSync* to an empty string. Do not change the value of *MasterDbName*.
  2. Restart the OS deployment server. The OS deployment server sees the changes in its `config.csv` files and updates the database of its former parent to indicate that it is not its child anymore.
  3. On the parent server, go to **Server** → **Server parameters** → **Server replication**.
  4. Select the child being removed and click **Make this OS deployment server a stand-alone OS deployment server**. The child moves to the **Standalone OS deployment servers** section.
  5. Select the child again and click **Remove this OS deployment server from database**.
  6. On the former child, run the `rbagent rad-resetscope` command to set the scope of all the objects on the server to local. Otherwise all replicated objects remain read-only.

  The OS deployment server is now detached from its parent but it keeps any child it might have had.
- In a multiple database architecture, when the OS deployment server to be removed is not working (if it crashed, for example)
  1. Make sure the OS deployment server you want to remove does not have any child. If it has any, assign it a new parent.
  2. Go to **Server** → **Server parameters** → **Server replication** on an OS deployment server parent to the one that you want to remove.
  3. Select the OS deployment server to be removed and click **Make this OS deployment server a stand-alone OS deployment server**. The child moves to the **Standalone OS deployment servers** section.
  4. Select the OS deployment server again and click **Remove this OS deployment server from database**.

  The information about the OS deployment server is removed from the database of its parent.

If the OS deployment server you have removed does not have any child, you can safely shut it down or uninstall it.

Chapter 10. Security

This section provides the user with information regarding security issues.

Security roles and access to the Web interface

Security roles allows you to create groups of users with restricted privileges to access the web interface, thus enhancing the overall security of your OS deployment server.

Administrator name and password

There is a unique administrator name and password for each OS deployment server. These name and password must be used first to create an authentication domain and create security roles. Afterwards, they must be stored in a safe place for reference. Instead, users must type their own user names and passwords to connect themselves to the web interface, as defined in the *HTTP* authentication domain.

Security roles

Security roles allows you to define groups of users with specific privileges on the OS deployment server. For each role, you can define which pages of the web interface they can view, which administrative groups of targets they can act upon, and which tasks they are allowed to perform. Preexisting roles are *Administrators* and *Operators*. However, any role can be created. Each user must be assigned to one or several security roles.

Note: Users belonging to several roles cumulate their privileges. You must therefore edit the two predefined roles which give overall control to all users.

HTTP authentication domain

Authentication domains determine how user and password information are verified, either locally or remotely. Users authenticated in the *HTTP* authentication domain and which belong to a security role gain access to the web interface, according to the privileges of the role. The specific *HTTP* authentication domain is a prerequisite to create security roles.

Connections to the web interface

You can monitor who is connected to the web interface on the page **Server** → **Server status** → **Network connections** page, under **Web interface sessions**.

Example

- *John* is a user of the computer on which the OS deployment server is installed.
- The *HTTP* authentication domain is set to local. Therefore, users trying to connect to the web interface must be users of the computer on which the OS deployment server is installed.
- A security role called *Rome Operators* with restricted privileges has been created. Members of this role can view all pages of the web interface, but only the *Rome*

© Copyright IBM Corp. 2006, 2010

administrative group. Moreover, they are denied any action which would change deployment objects or server parameters.

- *John* is assigned as a member of the security role *Rome Operators*.

In this configuration, user John can log into the web interface using his local password. Once logged in, he can deploy targets from the *Rome* administrative group. But he needs to ask an administrator if he wants to create a new software module and bind it to a profile, as he does not have the necessary privileges.

Creating an HTTP authentication domain

The *HTTP* authentication domain is a prerequisite for using security roles to control access to the OS deployment server through the web interface.

1. Go to **Server** → **Server parameters** → **Predefined channels**.
2. Click **New auth. domain**.
3. Type `HTTP` as domain name. Case matters.
4. Select the type of domain you want. There are three possibilities: local, remote NT and RADIUS.
5. Optionally, enter a user group to restrict access only to the users of this group.
6. Click **Save**.

You can now create security roles.

Creating security roles

Creating security roles allows you to provide access to the web interface for users besides the administrator, to restrict access to some pages and some features, and monitor who is currently logged in.

Before you can create valid security roles, you must have created an *HTTP* authentication domain.

1. Go to **Server** → **Server parameters** → **HTTP console security**.
2. Click on **New security role**.
3. Provide a name for the new role.
4. Edit the role parameters.
   a. Deselect web interface pages to which role members must not have access.
   b. Deselect administrative groups to which role members must not have access.
   c. Select features that you want to deny access to.
5. Click the available links to remove and add members to the role. When entering a user or group name, it must correspond to a name which can be identified in the HTTP authentication domain.

   Note: Users who are members of several roles cumulate the privileges of all their roles.
6. Click **Save**.

You can now log into the web interface using the username and password of a role member.

To edit the role at a later time, to change privileges, or to add or remove members, go to **Server** → **Server parameters** → **HTTP console security** and click on the role name.

Backups of server files

Tivoli Provisioning Manager for OS Deployment operating system images and other files are stored in a folder, and are accessible using file browsing tools. All of these files are stored in the data directory (typically `C:\TPMfOS Files` for Windows).

This directory contains regular files and control files, with `.md5`, `.dir` and `.inodes` extensions. These special files contain information about the file system structure, including the internal file number used by the provisioning server to identify a file.

When backing up files, it is important that you include both regular files and the provisioning server special files. When restoring files (or adding individual files), the provisioning server automatically detects new files and creates associated control files. Adding a file does not necessarily mean that the file will be usable by the provisioning server. A database entry is typically needed to describe what the file is used for. Therefore, it is also crucial to backup the database at the same time you backup the server files.

If you only want to back up a specific deployment scheme, system profile, or software module, it is easier to use RAD files.

Importing and exporting RAD files

Tivoli Provisioning Manager for OS Deployment allows you to export and import different types of objects if has created.

Some objects are exported and imported imbedded in a *RAD file*, others in a *target list*.

RAD files (with a `.rad` extension) can contain a single object or multiples objects. A RAD file can contain:

- WinPE deployment engines
- deployment schemes
- hardware configurations
- software modules
- system profiles

With RAD files, you can:

- move objects between OS deployment servers that do not have a good network connection between them
- archive objects without a running OS deployment server

Note: A RAD file can only be imported on a computer with the same byte order (little endian or big endian) as the computer on which the OS deployment server or web interface extension used for the export was installed.

Importing RAD files

To import RAD files in an OS deployment server:

1. Click **RAD Import** on the **Task templates**, **Profiles**, or **Software modules** page of the web interface.
2. Follow the instructions of the RAD Import Wizard.

Note:

a. When selecting the objects you want to import from the RAD file, you have the option to import the software application order (**Software stages**). Use this option carefully as the imported software application order overwrites the order present on your OS deployment server.

b. When importing deployment objects from a RAD archive, the byte ordering of the importing server must be the same as the one used by the exporting server. To be able to import a RAD archive created on a server using a byte ordering different than the importing server, you must perform the import using the web interface extension, running on a platform with the original byte ordering.

c. When importing a Windows 2008 or Windows Vista system profile in a RAD file created with version 7.1.1.2 of the the product, you also need to import the corresponding WinPE 2 ramdisk. If you do not, your system profiles cannot be upgraded and your system profiles are tagged as *too old*. If the necessary WinPE 2 ramdisk is not present in the RAD file, you need to export it again from your 7.1.1.2 OS deployment server.

d. When importing a WinPE deployment engine or a system profile, you also import the associated driver bindings. A check is then performed to find the associated driver software modules.

   The driver software module is present in the RAD file
   The binding is fully restored.

   The driver is found on the OS deployment server
   If the driver software module is not in the RAD file, a search is performed on the OS deployment server to look for the software module.

   The search is performed on the software module ID. Therefore, a match occurs only if the software module was exported from the same server. In some rare cases, the driver software module cannot be found although it is present because its ID has changed.

   If the driver software module is located at the time the WinPE deployment engine or the system profile is imported, the binding is fully restored.

   This implies that driver software modules must be imported before, or at the same time as, WinPE deployment engines and system profiles for the bindings to be restored.

   The driver is not found
   If the driver software module is found neither on the RAD file, nor on the OS deployment server, the driver binding cannot be restored.

Exporting RAD files

To export a RAD file:

1. Click **RAD Export** on the **Task templates**, **Profiles**, or **Software modules** page of the web interface.
2. Follow the instructions of the RAD Export Wizard.

- When exporting a RAD file, the software application order is automatically included.
- When exporting a RAD file, driver bindings associated with exported WinPE deployment engines and system profiles are automatically exported. However, the software modules associated with these bindings are not exported, unless they have been individually selected.

- When deciding where to generate the RAD file, be aware that the option to **download it directly from the server** is not available if the estimated size of the `.rad` file is bigger than 2GB, because of web browsers limitations.

  Note: If you export a RAD file by running the rbagent rad-radget command from a remote machine different from the OS deployment server and with the web interface extension installed, ensure that the local temporary directory, where the RAD file is temporarily copied, has enough space. For example, if the rbagent runs on UNIX, and the `/tmp` directory does not have enough space to contain the RAD file, then the rad-radget command fails. To avoid this problem you can either add more space to your temporary directory or change the temporary directory, as follows:

  1. Stop the OS deployment server.
  2. Define a new temporary directory. For example on UNIX, enter: `export TEMP=/root/temp`.
  3. Start the OS deployment server and enter the rbagent rad-radgetcommand again.

Importing and exporting targets lists

A target list file is a text file with comma separated values, with a `.csv` extension. Importing a target list is useful for adding large numbers of targets to the OS deployment server without having to start them up individually on the network. You can also import a PCI inventory for a single target in an `.ini` file.

Target list

Before you can import a target list, you must either export one or create a new one.

Information about each target in a target list is a collection of more than seventy items, including, :

- MAC address
- IP address
- User parameters
- Motherboard information
- Processor information

To view the complete list of items, export a target list, open it and read the beginning of the `.csv` file.

For the OS deployment server to successfully import targets in a list, you must fill at least one of the following items in:

- Serial number
- MAC address
- UUID
- IP address

The filled-in item can vary from target to target. Other items can remain empty.

Target lists above 1GB in size (about 1000 targets) cannot be imported into an OS deployment server, because of browser limitations. Therefore, you cannot use target lists for more than about 1000 targets.

Note: Do not use target lists to backup target information. To backup target information, you must backup the database used with an

appropriate tool. Lists of targets are not as complete as the database. In particular, target lists do not include some crucial target information found in the database, among which

- Bindings
- Disk inventory
- PCI inventory
- Deployment history

PCI inventory
A PCI inventory is exported on a USB key or floppy disk when this media is inserted in a target, booted through a network boot media, but which does not have network drivers.

- **Importing a target list**
  1. Go to the**Target Monitor page** in the web interface.
  2. Click **Import targets**.
  3. Indicate the location of the `.csv` file.
  4. Click **Ok**.
- **Exporting a target list**
  1. Go to the**Target Monitor** page in the web interface.
  2. Click **Export targets**.
  3. Click **Save**. The default file name (`hostexport.csv`) and saving location can be changed.
- **Importing a PCI inventory**
  1. Go to the**Target Monitor page** in the web interface.
  2. Click **Import targets**.
  3. Indicate the location of the `newhost.ini` file.
  4. Click **Ok**.

Exporting and loading configurations

You can export the configuration of the OS deployment server or load configuration settings that you have previously exported.

Exporting a configuration

To export the current configuration of your OS deployment server, click **Export configuration**. The button and the contextual menu item are present on the following pages of the web interface:

- **Server > Server parameters > Configuration**
- **Server > Server parameters > HTTP Console Security**
- **Server > Server parameters > Predefined channels**

Loading a configuration

To load a server configuration stored in the `rembo.conf` file, follow these steps:

1. Stop the OS deployment server.
   - On Windows operating systems, type `net stop remboserver` in a DOS window.
   - On UNIX operating systems, see "Startup scripts" in the Installation Guide.
2. Load the new configuration by typing

   ```
   rembo -d -c rembo.conf -exit
   ```

3. Restart the database gateway and OS deployment server.
   - On Windows operating systems, type `net start remboserver` in a DOS window to start both services.
   - On UNIX operating systems, see the Installation Guide.

Fault tolerance

A system is fault-tolerant if it can continue to perform despite parts failing. Fault tolerance helps to make your remote-boot infrastructure more robust.

In the case of OS deployment servers, the whole system is fault-tolerant if the OS deployment servers back up each other. When a server fails, other servers handle the requests from the down server.

Implementing fault tolerance at the Tivoli Provisioning Manager for OS Deployment level does not mean that your whole network infrastructure is fault-tolerant. You can implement fault-tolerances at all levels:

- At the physical level, by having redundant power sources (if all OS deployment servers are out of power at the same time, fault-tolerance at the product level is useless)
- At the network level, by having backup network links, and backup active elements (the backup server must be able to reach remote-boot targets)
- At the network operating system level, by having multiple network domains, or by running OS deployment servers outside of your domain architecture (OS deployment servers should not be all linked to the same NT PDC, or the same NFS server)
- At the DHCP level, by having multiple DHCP servers on the same subnet
- At the Tivoli Provisioning Manager for OS Deployment level, by implementing the fault-tolerance instructions.
- At the operating system level. If Tivoli Provisioning Manager for OS Deployment is able to survive to a severe problem, but then the operating system cannot find its network server, fault tolerance is useless

The following sections present information about how to implement fault tolerance at the DHCP and Tivoli Provisioning Manager for OS Deployment levels. Other levels are beyond the scope of this document.

Fault tolerance at the DHCP level

The DHCP protocol allows the implementation of fault tolerance and load-balancing very easily. If you connect two DHCP servers to the same IP subnet, and both servers are configured to serve IP addresses on this subnet, the protocol handles all conflicts between the two servers. A system is fault-tolerant if it can continue to perform despite parts failing. Load balancing specifies the maximum number of DHCP/BINL requests to a OS deployment server in one minute.

When a remote-boot target requests an IP address, the request packet is sent to the local broadcast address, that is, to all targets connected to the same IP subnet as the remote-boot target. If one or more DHCP servers are connected to the subnet, they send a DHCP offer packet to the remote-boot target, containing an IP address that has either been allocated in the server pool or administratively assigned to the remote-boot target (in case of statical binding between the hardware address and

an IP address in the DHCP configuration, also called *a reservation*). If more than one DHCP offer packet is received by the remote-boot target, only the *most informative* offer is kept by the target.

When the remote-boot target has selected a valid offer, it replies to the server from where the offer originated with a broadcast packet. This packet is received by all the targets connected to the local subnet, including the DHCP servers. This packet is used by DHCP servers to know if their offer was accepted or refused by the remote-boot target. If the target accepts, the IP address is locked in the DHCP server database, and the DHCP process can continue in unicast mode between the remote-boot target and the DHCP server. If the reply is for another offer, the server releases the IP address for its offer (which has been ignored by the target), and locks the IP address seen in the offer reply, to mark the IP address as used on this local subnet (even if the IP address has been allocated by another DHCP server).

Because of this, you can implement fault-tolerance by configuring multiple DHCP server for the same subnet. If the DHCP servers are identically configured, then the remote-boot targets always select the offer coming from the fastest server. Use this offer to implement load-balancing at the same time as fault-tolerance: the fastest server is always selected, and if the fastest server becomes overloaded, another server can send its offer first, and then it becomes the fastest server.

Fault tolerance at the Tivoli Provisioning Manager for OS Deployment level

Fault tolerance helps to make your remote-boot infrastructure more robust. A system is fault-tolerant if it can continue to perform despite parts failing. Fault tolerance at the product level is implemented with two configuration parameters: `Backup` and `BootReplyDelay`.

The boot process is made of several phases:

- DHCP discovery
- PXE discovery
- MTFTP download
- The product

Understanding these phases is key to understanding fault tolerance. Fault tolerance at the DHCP level is described in the previous section. You can implement fault tolerance at the PXE discovery level by using multipleOS deployment servers in Proxy DHCP mode (there is no OS deployment server on the DHCP server target, but all OS deployment servers are connected to the subnet). In Proxy DHCP mode, OS deployment servers send PXE reply packets to DHCP discovery packet initially sent by the remote-boot target. Because DHCP discovery packets are sent to the broadcast address, all OS deployment servers receive the discovery, and all send a reply packet, with the following considerations:

- The remote-boot target must be known by the OS deployment server (either by being a member of a target group in the server OS configuration, or if the OS configuration allows unknown targets to connect);
- The server does not answer immediately if the parameter `BootReplyDelay` is set.

You can use `BootReplyDelay` to introduce a preference order between the OS deployment servers on a same subnet. The server with the lowest `BootReplyDelay` is the first to answer DHCP discovery packets. All remote-boot targets are

redirected to this server. If this server fails, the server with the second lowest value for `BootReplyDelay` then answers, and so on. Fault tolerance at the PXE discovery level is in place.

If several OS deployment servers have the same value for the `BootReplyDelay` parameter, they all send the PXE reply at the same time, and the remote-boot target selects the fastest server. This specific environment implements load-balancing at the product level.

When the remote-boot target has selected its DHCP and PXE servers, the product bootstrap downloads from the PXE server (OS deployment server), and the target side is started. You can implement fault tolerance inside the product by using the `Backup` parameter for specifying a backup server. This value is sent to the remote-boot target during the initial startup of the target computer, and is used as a backup server if the primary server fails. The internal network protocols used in the product have been designed to enable the target to switch from the primary to the backup server in the middle of a file transfer. This only works under the following considerations:

- Files opened in write mode (upload to the product) cannot switch to a backup server. This could corrupt data on the OS deployment server, because one part of the file could be written on the primary server, and the other part on the backup server.
- The file system structure on the primary and backup servers must be strictly identical (that is, the same content under the `files` directory of the server).

Use backup servers at the product level (with the `Backup` parameter) when you stabilize your system (hard disk images are built, scripts are ready). After you stabilize the primary server, copy the `files` directory from primary to backup server, and set the `Backup` parameter on primary server.

Network security constraints

In many enterprise environments, an administrator must consider network security constraints.

For example, some ports can be unavailable to secure network traffic in and out of the enterprise.

By default, Tivoli Provisioning Manager for OS Deployment uses the following ports on the OS deployment server for communication:

- *DHCP* : port 67 *UDP*
- *PXE BINL* : port 4011 *UDP*
- *TFTP* : port 69 *UDP*
- *MTFTP* : port 4015 *UDP*
- *NBP* : port 4012 *UDP*
- *FILE* : port 4013 *UDP & TCP*
- *MCAST* : port 10000-10500 *UDP* Address: 239.2.0.1-239.2.255.255
- *HTTP (Web Interface)* : port 8080 TCP
- *HTTPS* : 443 TCP
- *Database gateway* : port 2020 TCP

On *targets*, the default ports are:

- *DHCP* : port 68 *UDP*

- *MTFTP* : port 8500-8510 Address: 232.1.0.1 *UDP*
- *MCAST* : port 9999 *UDP*
- *Remote control (web interface extension)* : port 4014 *UDP*

All of these ports can be modified, with the exception of port 69 for TFTP. Port 69 is part of the PXE specification, independent from Tivoli Provisioning Manager for OS Deployment, and cannot be modified. Any product using PXE boot needs to have this port open to permit PXE boot. This port needs to be open only on the OS deployment server, not on the target computers.

If you must modify ports (server or target) to conform to your network security constraints, you can use the web interface or edit the `rembo.conf` configuration file (and stop and restart your OS deployment server with option `-c rembo.conf`).

Note: If you do not want to use PXE to remote boot targets, you can create a network boot media.

Avoiding new security breaches

After you have installed the OS deployment server on your network while taking into account network security constraints, you still must ensure that using Tivoli Provisioning Manager for OS Deployment does not create new security breaches.

1. Protect your network against rogue PXE servers that can have access to your network. Otherwise, target computers can boot on the rogue server instead of the legitimate PXE server.
2. Prevent unwanted target computers from booting on your PXE server, unless you want to risk transferring sensitive information to unsecure computers.

Rogue PXE servers

A rogue PXE server is a server on a network which is not under the administrative control of the network staff.

By default, the PXE protocol is not protected against rogue PXE servers when it is working in boot discovery mode. There are ways to prevent this type of breach.

The target sends broadcast packets to the network requesting a PXE answer. The first PXE server to respond to the request takes control of the target computer. A rogue PXE server answering the request faster than the legitimate Tivoli Provisioning Manager for OS Deployment PXE server can take control of computers booting onto the network.

Using PXE in boot discovery mode is a well known security breach, independent from Tivoli Provisioning Manager for OS Deployment. While DHCP discovery must broadcast requests (the target does not yet possess any network information), there are ways to prevent the PXE security breach and permit only authorized PXE servers to answer requests from targets.

Using DHCP options to close the breach

Deactivate boot discovery mode for PXE targets. After this is done, computers trying to contact a PXE server must know the specific address and can no longer send broadcast packets. Information is transferred at the DHCP stage, by using options 60 and 43. Using these options, the DHCP server returns the target its IP

address and the IP address of the authorized PXE server. If necessary, option 43 can contain several IP addresses for backup servers.

information about how to configure the DHCP server is located in the Tivoli Provisioning Manager for OS Deployment7.1.1.1 Installation Guide, Chapter 4.

Unwanted target computers

You must ensure that target computers are legitimate in order not to distribute sensitive information outside of appropriate computers.

To achieve this with Tivoli Provisioning Manager for OS Deployment, set the OS deployment server to closed mode. In closed mode, the OS deployment server does not accept new targets and sends boot information aboutly to the targets listed in its database. To activate this parameter,

1. Go to **Server** → **OS deployment** → **Task templates**.
2. Select **Idle Layout** and double click **Idle state**.
3. Click **Edit** for the section **Handling of unknown targets**.
4. Select **Completely ignore unknown targets (closed OS deployment server)**

In closed mode, the OS deployment server (PXE server) checks the MAC address and the IP address of potential targets and sends the Tivoli Provisioning Manager for OS Deployment bootstrap only if these addresses belong to a known target.

The bootstrap is sent by TFTP, which is a non-secured protocol. However, this bootstrap is very small (around 300 KB) and does not contain any critical information. After the bootstrap runs on the Tivoli Provisioning Manager for OS Deployment target, all other transfers are performed using secure protocols. When the bootstrap is in place, the OS deployment server checks the UUID and the serial number of the Tivoli Provisioning Manager for OS Deployment target for before transferring any other data. This ensures that the MAC and IP addresses to which the bootstrap was sent were not faked.

If no rogue server can interfere between your targets and the OS deployment server and no unknown target can boot from your OS deployment server without authorization, you have ensured that Tivoli Provisioning Manager for OS Deployment does not add security breaches to your network environment.

Security issues and the web interface

Sessions on web interface have been made as secure as possible. However, security relies also on users and the way they use the product.

To ensure the highest possible level of security using the web interface, the following features have been implemented:

- Connections are made using the encrypted HTTPS protocol.
- Sessions on the web interface are identified by a unique session identification number. If you need a second web interface, you must log on to a new one to ensure having distinct session identifiers. If you open a new window or cut-and-paste the URL without logging in again, it can result strange web interface behavior.
- Sessions expire after a given delay, automatically logging users out if they forget to click **Logout** at the end of their session. To modify the length of this delay, see “web interface parameters” on page 311.

Chapter 11. Booting targets without using PXE

If you do not want to use PXE on your network, you can use Tivoli Provisioning Manager for OS Deployment to create a network boot CD, DVD, or USB drive.

With network boot media, your target can boot and connect to the Tivoli Provisioning Manager for OS Deployment server in a PXE-less environment. Use this kind of deployment when it is not possible to use PXE to boot the target.

Some typical situations are network card without PXE support, firewalls preventing PXE traffic, non-allowed PXE boot, or an unavailable DHCP server. In particular use the original Windows PE-based network boot CD to minimize target hardware compatibility issues.

To create the network boot media, you can either use the wizard or run command lines from a computer with the web interface extension installed.

Note:

- Network boot media must be updated every time the OS deployment server is updated or upgraded to ensure compatibility with the OS deployment server.
- If your network boot media is optimized for Windows operating systems, you must create the media from an OS deployment server or a web interface extension installed on a computer with the same byte order (little endian or big endian) as the one on which you want to use the network boot media.
- You cannot capture hardware information or deploy a hardware configuration from a target started with a network boot media.
- Before you create your network boot media, make sure that the **Disable DHCP/BINL module** parameter is set to **no**.

Creating network boot USB drive with the wizard

Tivoli Provisioning Manager for OS Deployment can automatically generate bootable USB drives that connect the target to an OS deployment server, without using DHCP or PXE, to perform deployments.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must be formatted as FAT32 or NTFS. USB keys already filled with a bootable operating system might not work.

Note: SuSE Linux Enterprise Desktop cloning is not supported on USB drive deployments

These bootable USB drives can also be used to deploy computers without a PXE compliant network adapter.

To create OS bootable USB drives:

1. Perform one of the following steps:
   - Go to **Server** → **OS deployment** → **Task templates**.
   - Go to **Server** → **OS deployment** → **System Profiles**.
   - Go to **Server** → **OS deployment** → **Software modules**.

© Copyright IBM Corp. 2006, 2010

2. Click **Generate media**.
3. Select **Create a network boot USB key** to start the USB key wizard. Click **Next**.
4. Specify the operating system on which to boot the target. Select **Linux** to load a Linux deployment engine, **Windows** to load a WinPE deployment engine, or **Both** to have the two.

   Note:
   - If you use a network boot media to deploy Linux profiles, you cannot use the HTTP protocol. For this reason when you create a deployment scheme ensure you do not select the **Download files with a network share when applicable** option in the **Network settings** section.
   - If you use a network boot media and want to erase hard disk content, your media must contain a WinPE deployment engine. Therefore, you must select either **Windows** or **Both**.
5. If you have selected **Windows** or **Both**, and if you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. If you want to obtain the target IP address through DHCP, select **Dynamic IP address with DHCP**, and click **Next**.

   If you want to use a fixed IP address for your target instead of having it go to the DHCP server, select **Static IP address**, and click **Next**.
   a. Enter the target IP address, gateway, and network mask.
   b. (*Optional*) Select **Allow IP address override at runtime** to be able to modify the target IP address when starting up the target.
   c. Click **Next**.
7. Enter the IP address of the OS deployment server.
8. (*Optional*) Select **Allow server IP address override at runtime** to be able to modify the IP address of the OS deployment server when starting up the target.
9. Plug your USB key into a machine running the Web interface extension and specify its address.
10. Choose the drive matching your USB key.
11. Click **Finish** to close the wizard.

Use the USB drive to boot the target.

Creating a network boot CD or DVD with the wizard

1. Go to the **Task templates**, the **System Profiles**, or the **software modules** page.
2. Click **Generate media** at the bottom of the page.
3. Select **Create a network boot CD/DVD** and click **Next**.
4. Specify the operating system on which to boot the target. Select **Linux** to load a Linux deployment engine, **Windows** to load a WinPE deployment engine, or **Both** to have the two.

   Note:
   - If you use a network boot media to deploy Linux profiles, you cannot use the HTTP protocol. For this reason when you create a deployment scheme

ensure you do not select the **Download files with a network share when applicable** option in the **Network settings** section.

- If you use a network boot media and want to erase hard disk content, your media must contain a WinPE deployment engine. Therefore, you must select either **Windows** or **Both**.

5. If you have more than one WinPE deployment engine, select the target models on which you want to use your media. The WinPE deployment engines matching the selected target models are included in the media.
6. If you want to obtain the target IP address through DHCP, select **Dynamic IP address with DHCP**, and click **Next**.

   If you want to use a fixed IP address for your target instead of having it go to the DHCP server, select **Static IP address**, and click **Next**.

   a. Enter the target IP address, gateway, and network mask.
   b. (*Optional*) Select **Allow IP address override at runtime** to be able to modify the target IP address when starting up the target.
   c. Click **Next**.
7. Enter the IP address of the OS deployment server.
8. (*Optional*) Select **Allow server IP address override at runtime** to be able to modify the IP address of the OS deployment server when starting up the target.

   Note: When you create the network boot CD or DVD in a multiserver infrastructure, ensure that the OS deployment servers share the same password and port number. The network boot CD or DVD works only if you specify the IP address of a OS deployment server having the same password and port number of the OS deployment server that generated the ISO file.
9. Click **here** to download the ISO file.
10. Click **Finish** to close the wizard.

The generated ISO file can be burned to create the network boot CD.

To start a target over the network using your OS deployment server without booting through PXE, start the target on the network boot CD and the target automatically connects to the OS deployment server.

Creating an original WinPE 3.0 network boot CD or DVD with the wizard

Before creating the Windows PE-based network boot CD or DVD, ensure that you have configured your WinPE 3.0 deployment engines to match your target models and to contain the critical drivers for the specific target hardware.

Create this CD/DVD if you want to deploy Windows operating systems without using PXE, minimizing hardware compatibility issues.

1. Go to the **Task templates**, the **System Profiles**, or the **Software modules** page.
2. Click **Generate media** at the bottom of the page.
3. Select **Create an original Windows PE-based network boot CD/DVD** and click **Next**.
4. If you have more than one WinPE 3.0 deployment engine, select the target models on which you want to use your media. The WinPE 3.0 deployment engines matching the selected target models are included in the media.

5. If you want your WinPE 3.0 deployment engine to use a dynamic IP address through DHCP for your target during the provisioning, select **Dynamic IP address with DHCP**, and click **Next**.

   If you want your WinPE 3.0 deployment engine to use a fixed IP address for your target instead of having it go to the DHCP server, select **Static IP address**, and click **Next**.

   a. Enter the target IP address, gateway, and network mask.
   b. (*Optional*) Select **Allow IP address override at runtime** to modify the target IP address when starting up the target.
   c. Click **Next**.
6. Enter the IP address of the OS deployment server.
7. (*Optional*) Select **Allow server IP address override at runtime** to modify the IP address of the OS deployment server when starting up the target.

   Note: When you create the network boot CD or DVD in a multiserver infrastructure, ensure that the OS deployment servers share the same password and port number. The network boot CD or DVD works only if you specify the IP address of a OS deployment server having the same password and port number of the OS deployment server that generated the ISO file.
8. If your target startup sequence is first CD or DVD and second hard disk, make sure that **CD/DVD will boot at user request only** is selected to boot from the network boot CD/DVD only after user interaction. This is because the deployment flow must first start with a boot from the WinPE 3.0 deployment engine available in the network boot CD/DVD while successive boots must be performed from the hard disk where the WinPE 3.0 deployment engine was cached. For this reason, the first time that the target boots, you must press any key to boot from the CD/DVD and start the deployment. Successive unattended reboots fall back to hard disk until the deployment is completed.
9. Specify the path of the .ISO file that will contain the deployment engine.
10. After a few minutes the CD/DVD media is created. Click **Finish** to close the wizard. In the specified directory you can see the .ISO file.

The generated ISO file can be mounted on the target or burned to create the network boot CD.

To start a target over the network using your OS deployment server without booting through PXE, start the target on the network boot CD and the target automatically connects to the OS deployment server.

Using a network boot CD

When PXE network boot is not available in your network, use a network boot CD to start up your target.

From the OS deployment server create an ISO image of a network boot CD.

1. To boot virtual machines, mount the ISO image. To boot target computers, burn a CD/DVD from the ISO image and use it on your targets.
2. If you are using an original WinPE 3.0 network boot CD during the first target boot, press any key to boot from the network boot CD and run the WinPE 3.0 deployment engine.
3. The target connects to the network and contacts the OS deployment server. From the OS deployment server you can now submit any task on the targets.

Creating a network boot USB drive with command lines

You can create a network boot USB drive, which Tivoli Provisioning Manager for OS Deployment can use when a target cannot boot from the network.

Install the rbagent, also known as web interface extension, on a Windows target. The USB drive must be formatted as FAT32 or NTFS. Existing files on the USB drive are not deleted. USB keys already filled with a bootable operating system might not work.

The command line must be used only when the web interface is either inappropriate or unavailable.

- If you want to obtain the target IP address through DHCP, use this command line:
  - Windows On Windows operating systems

```
rbagent.exe -s <OSD_server_ip_address>:<OSD_server_password>
rad-mkbootusb <drive>
<USB_OSD_server_ip_address> <USB_OSD_server_password>
[allowsrvipoverload] [nowpe|preferwpe]
[bootopt nnn] [clearcmos]
```

where:

OSD_server_ip_address
Is the IP address of the OS deployment server.

OSD_server_password
Is the password for the administrative user (typically `admin`) on your OS deployment server.

drive Is a drive letter of the Windows target where you run the `rbagent` command. The rad-mkbootusb command adds the requested files to the FAT32 or NTFS partition and makes it bootable. The drive must be already formatted. Existing files on the partition are not deleted.

USB_OSD_server_ip_address
Is the IP address of the OS deployment server that the target must contact, when it boots from the USB drive.

USB_OSD_server_password
Is the password of the OS deployment server that the target must contact, when it boots from the USB drive.

allowsrvipoverload
Allows you to choose an OS deployment server later, from the target.

nowpe|preferwpe
Defines if a Linux deployment engine or WinPE environment is loaded from the USB drive, when a target boots from this USB drive, without accessing the network. Only when Linux deployment engine or WinPE is running, does the target connect to the network and try to contact an OS deployment server. If you deploy only Linux, specify `prefermcp` to skip the WinPE deployment engine. You can specify `preferwpe` only if there is a WinPE deployment engine on the OS deployment server.

bootopt *nnn*
Allows you to specify additional flags before the boot.

clearcmos
Resets the CMOS alarm fields if they are in an invalid state.

For example:

```
> C:\TPMfOSd Files\global\http\agents\rbagent.exe
-s 10.10.10.10:abcd rad-mkbootusb C: 10.10.10.10 abcd
```

- If you want to use a fixed IP address for your target instead of having it go to the DHCP server, use this command line:
  - On Windows operating systems:

    ```
    rbagent.exe -s <OSD_server_ip_address>:<OSD_server_password>
    rad-mkbootusb <drive>
    <USB_OSD_server_ip_address> <USB_OSD_server_password>
    fixed [fixed_ip_address] [fixed_netmask] [fixed_gateway_ip_address]
    [allowsrvipoverload] [nowpe|preferwpe]
    [allowipoverload] [bootopt nnn] [clearcmos]
    ```

    where:

    OSD_server_ip_address
    : is the IP address of the OS deployment server.

    OSD_server_password
    : is the password for the administrative user (typically `admin`) on your OS deployment server.

    drive
    : is a drive letter of the Windows target where you run the `rbagent` command. The rad-mkbootusb command adds the requested files to the FAT32 or NTFS partition and makes it bootable. The drive must be already formatted. Existing files on the partition are not deleted.

    USB_OSD_server_ip_address
    : Is the IP address of the OS deployment server that the target must contact, when it boots from the USB drive.

    USB_OSD_server_password
    : Is the password of the OS deployment server that the target must contact, when it boots from the USB drive.

    fixed_ip_address
    : Is the static IP address of the target you boot using the USB drive.

    fixed_netmask
    : Is the netmask of the target you boot using the USB drive.

    fixed_gateway_ip_address
    : Is the IP address of the gateway that the target uses.

    nowpe|preferwpe
    : Defines if a Linux deployment engine or WinPE is loaded from the USB drive, when a target boots from this USB drive, without accessing the network. Only when Linux deployment engine or WinPE is running, does the target connect to the network and try to contact an OS deployment server. If you deploy only Linux, specify `nowpe` to skip the WinPE software module. You can specify `preferwpe` only if there is a WinPE software module on the OS deployment server.

    allowipoverload
    : Allows you to define IP settings manually on the target.

    bootopt *nnn*
    : Allows you to specify additional flags before the boot.

    clearcmos
    : Resets the CMOS alarm fields if they are in an invalid state.

You can now boot the target using the network boot USB drive instead of the network card. To use the PXE emulation USB key, insert the USB key into the drive and restart the target. If your machine does not boot from the USB key, check the BIOS boot list to see if your USB drive is included in the boot sequence and is listed before the hard disk. Most machines also allow you to select the temporary boot device without changing the boot sequence in BIOS.

Creating a network boot CD or DVD with command lines

This mode must be used only when the web interface is either inappropriate or unavailable.

Note: When you create the network boot CD or DVD in a multiserver infrastructure, ensure that the OS deployment servers share the same password and port number. The network boot CD or DVD works only if you specify the IP address of a OS deployment server having the same password and port number of the OS deployment server that generated the ISO file.

- If you want to obtain the target IP address through DHCP, use these command lines:
  - UNIX Linux On UNIX and Linux operating systems

    ```
    #./rbagent -s <target_ip_address>:<target_password>
    rad-mkbootcd <full_path_to_boot_iso>
    <target_ip_address> <target_password>
    ```

  - Windows On Windows operating systems

    ```
    rbagent.exe -s <target_ip_address>:<target_password> rad-mkbootcd
    <full_path_to_boot_iso> <target_ip_address> <target_password>
    ```

    where:

    target_ip_address
    Is the IP address of the OS deployment server.

    target_password
    Is the password for the administrative user (typically `admin`) on your OS deployment server.

    full_path_to_boot_iso
    Is the full path to the `.iso` file you want to create on the target where you run the `rbagent` command.

    For example:

    ```
    > C:\TPMfOSd Files\global\http\agents\rbagent.exe
    -s 10.10.10.10:abcd rad-mkbootcd C:\boot.iso 10.10.10.10 abcd
    ```

    This creates a file called `boot.iso` in `c:\` which can be burned onto a CD.
- If you want to use a fixed IP address for your target instead of having it go to the DHCP server, use these command lines:
  - UNIX Linux On UNIX or Linux operating systems:

    ```
    #./rbagent -s <target_ip_address>:<target_password> rad-mkbootcd
    <full_path_to_boot_iso> <target_ip_address>
    <target_password> [fixed_ip_address]
    [fixed_netmask] [fixed_gateway_ip_address]
    ```

  - On Windows operating systems:

```
> rbagent.exe -s <target_ip_address>:<target_password> rad-mkbootcd
<full_path_to_boot_iso>
<target_ip_address> <target_password>
[fixed_ip_address] [fixed_netmask] [fixed_gateway_ip_address]
```

where:

fixed_ip_address
Is the static IP address of the target you boot using the CD.

fixed_netmask
Is the netmask of the target you boot using the CD.

fixed_gateway_ip_address
Is the IP address of the gateway the target uses.

The generated ISO file can be burned to create the network boot CD.

To start a target over the network using your OS deployment server without booting through PXE, start the target on the network boot CD and the target automatically connects to the OS deployment server.

Booting on the network when the target is missing network drivers

Using Tivoli Provisioning Manager for OS Deployment you can boot on the network even if your target is missing network drivers and its model is unknown to the OS deployment server.

You are attempting to boot, through a network boot media, a new target that has a model unknown to the OS deployment server. Your target is missing network drivers and cannot therefore boot on the network as intended. Because the target model is unknown to the OS deployment server, you cannot bind the necessary drivers in the network boot media.

The solution is to import the PCI inventory of the target on the OS deployment server, to bind the needed drivers to the model, to recreate a network boot media, and to use this media to boot the target on the network successfully.

1. Export the PCI inventory of the target on a USB key or on a floppy disk.
   a. Create a network boot media.
   b. Boot your target with this media.
   c. If there are any missing drivers, and if you do not have a USB key already plugged in, the target waits until you have inserted a floppy disk or a USB key.
      1) If a file called `newhost.ini` already exists on the media, it is renamed.
      2) A file called `newhost.ini` is created on the media containing the PCI inventory of the target.
2. Import the PCI inventory of the target on the OS deployment server.
   a. Insert your media in the server.
   b. Go to **Server** → **OS deployment** → **Target Monitor**.
   c. Click **Import targets**.
   d. Type in the location of the `newhost.ini` file and its name, or browse to locate it, and click **OK**.
3. Re-create you network boot media

a. If you have more than one WinPE deployment engine per architecture, check with which WinPE deployment engine your new target model matches.
b. Bind the necessary network drivers to this WinPE deployment engine.
c. Re-create a network boot media, selecting at least the WinPE deployment engine to which you have bound the drivers, and selecting **Optimized for Windows**.

4. Boot your target with the newly-created network boot media.

Because the target now has the appropriate drivers, it can connect to the network and contact the OS deployment server.

You can now use your target like any other target that is booted through a network boot media.

Chapter 12. Tools

This section provides information the disk content blanking feature and on the deprecated software snapshots.

Erasing hard disk content

Permanently erasing the content of a hard disk can prove necessary for confidentiality reasons.

To erase the content of a hard disk with the product, you must have a WinPE deployment engine on your OS deployment server. If you booted your target with a network boot media, the WinPE deployment engine must be present on the network boot media.

When a computer changes hands or purpose, you might have to make sure that the new user cannot recover data previously stored on the hard disk. To do so, the hard disk is completely written over with meaningless data, thus permanently erasing all previously stored data. The process can take up to a few hours, as every bit on the disk is written over.

Note: Erasing the content of the hard disk is a non reversible process which must be used with caution.

To erase the hard disk content of a target:

1. Go to the **Target Monitor** page
2. Select the targetor targets on which you want to erase the hard disk
3. In the contextual menu, select **Additional features**
4. In the **Additional feature wizard**, select **Destroy hard disk contents**
5. Follow the instructions of the wizard. It displays five disposal methods. Depending on the method you choose, you get a description of how the selected method erases the contents of the hard disk.

Software snapshots

Using software snapshots is strongly discouraged. Current versions of the product can redeploy software snapshots which were created with older versions of the product. Software snapshot redeployment is supported for compatibility with earlier versions only. Creation of software snapshots is deprecated.

Limitation of the technology

Installing software by software snapshot is not intended as a general alternative for software installers. It is only safe when used in the correct environment, and when the software snapshots have been created carefully.

The difference between a real installer and a software snapshot is that whereas the installer can be aware of the present state of the computer and can act accordingly, the software snapshot is applied blindly and will therefore only do the correct work if the computer is in a similar state as the reference image on which the software snapshot was created.

© Copyright IBM Corp. 2006, 2010

The good point with the use of software snapshots is that as they are applied as part of a complete installation process, the environment is precisely known and does not depend on any previous user interaction or any past action performed on the target. The initial computer state is completely under control. It is therefore possible to safely use software snapshots.

The purpose of creating software snapshots is obviously to reuse them in several different circumstances, or to combine them in several ways. However, when doing these combinations, you must keep in mind that if some of the software snapshots are not fully independent one of the other (which is the ideal case, but which is not always possible), you must apply them in the correct order so as to reproduce the same environment originally present when each software snapshot was created.

Special care must be taken if software snapshots are used to handle hardware-related components. The binding of hardware drivers into the operating system can be tricky, and installing the same device in two different computer models can lead to very different registry keys, which can in some case make it impossible to use a common software snapshot.

When using software snapshots, you must also be aware of the fact that NTFS security attributes associated with files are also part of the software snapshot. However, the definition of users is typically not part of the software snapshot, but of the reference image. Therefore, you must avoid creating software snapshots with special user-related permissions, as it might lead to permissions problems if the user does not exist in the system profile being used for the final deployment.

Restoring software snapshots

You can only restore software snapshot which were created with old versions of the product. This feature is maintained only for backward compatibility.

Before restoring a software snapshot, you must ensure that you have an operating system properly installed on your hard disk. You might want to restore a system profile first.

1. Go to **Server** → **OS deployment** → **Target Monitor**.
2. Select the target on which you want to restore the software snapshot.
3. In the contextual menu, select **Additional features**.
4. Select **Restore a profile** and click **Next**.
5. Select the software snapshot and click **Next**.
6. Follow the remaining instructions of the wizard.

Chapter 13. Migrating users

When an operating system needs to be upgraded to a newer version or when hardware needs to be changed to newer material, users like to keep their settings and files and to have them available on their renewed computer. This is the purpose of *user migration*.

Basic operations for user migration are:

1. Performing an inventory of computer settings;
2. Capturing user settings to be migrated;
3. Reinstalling (or installing on a bare-metal computer) the operating system;
4. Restoring the computer and user settings on the target.

Tivoli Provisioning Manager for OS Deployment facilitates steps 1, 3, and 4 of this migration process.

Capturing user settings

Data to be captured on a computer for user migration includes, among others, user settings (user name, time zone, keyboard information, ...), user files, and the list of installed software. The data needs to be retrieved and then stored outside of the source computer, typically on a network, in prevision of a future restoration.

Some of the needed data (including user accounts, users' files, desktop settings) can be captured with tools broadly available.

Other data (such as the list of installed software) can be captured through a command-line. The information thus obtained is then stored in the OS deployment server. The following command must be run on the source target.

```
rbagent -s serverIPaddress:NetPassword rad-hostinventory updatebom
```

where `serverIPaddress` is the IP address of the OS deployment server, `NetPassword` is the superuser password allowing the web interface extension to be connected to the OS deployment server. The password can naturally be provided in the same encrypted format as found in the configuration file `rembo.conf`. `rad-hostinventory` performs the inventory itself, while `uploadbom` modifies the target record on the OS deployment server.

Restoring user settings

Tivoli Provisioning Manager for OS Deployment enables you to restore previously captured user settings during a deployment task, with the help of software modules. The migration tool and the stored settings are included in software modules which are bound to a deployment scheme. The necessary software modules can be of several types, including registry changes and files

© Copyright IBM Corp. 2006, 2010

copies. A careful ordering of the software modules is necessary as they are interdependent: the migration tool must obviously be installed and the setting files copied before the restoration operation is performed on the target.

Chapter 14. Reference

This section contains reference information on the product.

Keyboard navigation on the target

When a target is controlled by the OS deployment server and is in the PXE kernel, you can navigate using the keyboard only.

The keys described in Table 10 are assigned navigational functions when the target is controlled by the OS deployment server and is in the PXE kernel. If the controlled target is in the PXE kernel, the word *kernel* appears in the lower right corner of the target screen.

Table 10. Keys and actions

Key or key combination	**Action**
Enter	Selects
Tab	Goes to the next control
Esc	Deletes content of a text field Exits from a menu
Ctrl+S and Ctrl-Esc	Opens the start menu
Alt+F4	Closes the window
Alt+Tab	Toggles between windows
Alt+Right Arrow	Scrolls right
Alt+Left Arrow	Scrolls left
Alt+Page Down	Scrolls down
Alt+Page Up	Scrolls up
Print Screen	Takes a screen snapshot and stores it under `files/global/snapshot.png`
Alt+Print Screen	Takes a window snapshot and stores it under `files/global/snapshot.png`
Ctrl-+Alt+Backspace	Reinitializes USB devices

If you have difficulties in getting the focus on a window when only one is open, open and close the start menu with **Ctrl+S**, followed by **Esc**.

The web interface

The web interface is the most common way of interacting with the provisioning server. It is an Web-based console viewed in a standard browser and does not have to run on the same computer as the provisioning server.

The supported browser versions are Internet Explorer 6 and above, Mozilla Firefox, and Apple Safari.

Note: Mozilla Firefox 2.0.0.2 does not work on Linux PowerPC 64-bits.

© Copyright IBM Corp. 2006, 2010

To reach the web interface, type the name of the computer on which the provisioning server is located (or `localhost` if the browser is on the same computer as the server), with port `8080`. With the provisioning server and browser on the same computer, type `http://127.0.0.1:8080` in your browser to open the web interface login screen.

To access the web interface, you need a valid user name and password. You entered these into the provisioning server during the product installation.

On the web interface login page, there is a link to navigational tips for first time web interface users. You are strongly encouraged to follow the link and read these tips.

The web interface is divided into five components:

- Server status
- Server parameters
- Server history
- OS deployment
- Server log files

You can access these components

- by clicking on their respective icons in the main frame of the web interface
- by using the menu provided in the left frame of the web interface

Server status

Server status information is further subdivided into four sections: General information, Installation check, Network connections and web interface extension.

General information

Items in the **General information** section are read-only and contain:

- General server information, such as the OS deployment server version, the number of active targets, and the last warning message
- OS deployment server statistics, including the number of successful boots, and ignored boots, and the current boot module load
- File server statistics, containing information about MCAST, unicast, and PCAST sessions, and TCP connections
- Web interface modules, including statistics on the HTTP connections and operations

Installation check

Items in the **Installation check** section are read-only. It provides a summary of all errors that have happened in the past 30 minutes. They are presented in distinct sections representing their associated log file.

If you are not familiar with Tivoli Provisioning Manager for OS Deployment, use **PXE Check** at the bottom of the screen. A wizard helps you understand and check all of the necessary steps to ensure that the target works with your OS deployment server.

Network connections

Displays the active connections onto the OS deployment server. They are presented with thee different links: *Unicast* connections, *Multicast* connections, and *HTTP* connections.

The numbers **Traffic IN** and **Traffic OUT** are in Kilobytes (total number of KB seen on this connection).

web interface extension

Displays the status of the web interface extension on the computer that is running the browser.

Server parameters

Server parameters are used in the OS configuration of the Provisioning server.

Server parameters are used to configure the OS deployment server. They are divided in to five sections:

OS configuration
provides read and write access to the main server OS configuration parameters.

HTTP console security
allows you to change the administrator name and password, and to create security roles to secure access to the OS deployment server through the web interface.

Predefined channels
is the location to define TCP tunnels and authentication domains.

Hardware handling
provides information about the known compatibility and incompatibility of hardware devices with Tivoli Provisioning Manager for OS Deployment.

Server replication
allows you to replicate parent and child servers. See "Multiserver infrastructure" in the Installation Guide for more information.

Server history

The Server history component helps you to follow the task of your OS deployment server.

It is divided into four sections:

Server statistics
There are three different pages that show graphically the load of the processor, the network and the number of targets working with the OS deployment server. The graphs are displayed for the past 24 hours, the past week and the past month.

Each of the **Network traffic** graphs has two scales. The left scale shows the total inbound and outbound traffic, and the right scale shows multicast traffic. Numbers are in Kilobytes per minute (KB/min).

Deployment statistics
Deployment statistics show you, through tables, when deployments were performed over the last 24 hours, the last month, or the last two years. You can select to view successful or failed deployments by OS configuration, by deployment scheme, by computer model, or by administrative group

Modification history
Keeps track of creations, deletions, and modifications on OS configurations, system profiles, deployment schemes, and software modules. For every

update, the page displays the date and time of the update, its author, its item description, and its new version number.

Tasks tasks include most actions performed from the OS deployment server on targets. Examples of tasks are deployment, creation of cloning profiles, detection of the operating system currently on a target.

Displays a description, an execution and an expiration date, current state, and progress rate. You can expand a task by clicking on its **+** sign to view its target or targets.

To cancel a task, select **Cancel task** from the contextual menu when a target is selected.

See also deploy/tosd_monactivity.dita.

OS deployment

The OS deployment component provides access to the main functions of Tivoli Provisioning Manager for OS Deployment, allowing you to manage targets and to prepare and run deployments.

It is divided in to four sections:

Target Monitor
to manage the targets

Task template
to manage screen layouts on the target and deployment schemes

Profiles
to manage the operating system profiles and OS configurations

Software modules
to manage software to be deployed with an operating system

Server log files

Server log files help in diagnosing problems and recording commands.

Log file description

There are two sets of files for the logs contained in the `TPMfOS Files` folder: `.log` and `.trc`. For example, there is a `vm.log` file and a `vm.trc` file. The .log files contains information understandable by all users. It is in the language in which the product was installed. The .trc files contain more detailed information and are designed for the support team. It is always in English. When sending logs to support, make sure to always include the `.trc` files.

There are several distinct log files:

`files/logs/activities.log`
This log allows you to view when a replication task was started. Its content cannot be viewed from the web interface.

`files/global/hostactivities.log`
This directory contains the list of all target tasks. Its content can be viewed from the **Tasks** page of the **Server history**, where it is merged with the information provided by `activities.log`. To send the logs of a given task to support, ideally select the task and select **Export debug data** in the contextual menu.

Boot log file or `files/logs/boot.log`
This log is useful when diagnosing DHCP/TFTP problems. Its content can be viewed in the**Server log files** page of the web interface.

File log fileor `files/logs/File.log`
This log is useful to diagnose multicast file transfer problems. Its content can be viewed in the **Server log files** page of the web interface.

HTTP log file or `files/logs/HTTP.log`
This log records requests made by the web interface. Its content can be viewed in the **Server log files** page of the web interface.

NBP log file or `files/logs/NBP.log`
This log records remote target commands such as authentication, tracking logs, and so on. Its content can be viewed in the **Server log files** page of the web interface.

TCP log file or `files/logs/TCP.log`
This log is used to diagnose unicast file transfer problems. Its content can be viewed in the **Server log files** page of the web interface.

VM log file or `files/logs/VM.log`
This log records events generated by background tasks running on the server. Its content can be viewed in the **Server log files** page of the web interface.

Sync log file or `files/logs/rsync.log`
This log file contains information specific on replication: checking files, finding them or not, copying them, and so on. Its content can be viewed in the **Server log files** page of the web interface.

Note: The `files` directory corresponds to the main data directory. Under Windows operating systems, it is `TPMfOS Files` by default.

Retrieving pertinent information in the logs

To help you retrieve pertinent information easily, the following features are available in the web interface:

- Logs are color coded. Blue indicates no warning or errors, yellow indicates warnings, red indicates errors.
- Logs are hierarchical. Clicking on the expand sign allows you to see more details.
- Logs are chronological. If you want to go to a specific date and time in a log file, click **Jump to ...**, enter the date and time, and click **Jump**. The log section with the given date and time expands. If no section has the exact given date and time, the section just before the given time and date expands.

Log file cleaning

From the **Server Log Files** page, you can also clean the contents of the log files. You can either delete the content of individual server log files or perform a partial cleaning of all the server log files. You can also delete or perform a partial cleaning of all target log files.

Partial cleaning of a log allows you to keep the most recent content only. You can keep information of the latest

- 30 minutes,
- 60 minutes,

- 6 hours,
- 12 hours,
- 24 hours,
- 1 week,
- 2 weeks,
- or 1 month.

Log data older than the selected time is deleted.

Database schema

The database schema documentation describes the most important items of the database schema used by Tivoli Provisioning Manager for OS Deployment to store and manage targets and images. Further details can be found in the reference database file, `AutoDeployDistrib.mdb`, in the program installation directory.

Database schema overview

In Tivoli Provisioning Manager for OS Deployment, database schemas are used to store and manage targets and images.

The database schema can be divided in two parts:

- The Bill of Material-related tables (`BOM`, `DiskInventory`, `DMIInventory`, `PCIInventory`, `PCIDescription`, `Deployment`, `ErrorLog`, `Settings`, `UserProfile`) that store all target and user-dependent information
- The OS configuration-related tables (`SoftwareProfile`, `SoftwareItem`, `GroupingRule`, `Groups`, `SystemProfile`) that retain information related to the software modules and operating system that can be deployed

Bill of Material-related tables

There is one Bill of Material (BOM) record for each remote-boot target that has been started on the OS deployment server. The `DeplCount` field is a counter that allows you to track the number of deployments that have been run for a particular target(the summary of each deployment can be found in table `deployment`). The field `Status` indicates the current state of this target:

- If the target was last used to create an image, or if the deployment is configured with no database update, the status is empty
- If the target is showing the parameter edition window and waiting for BOM information to be reviewed (in the database or on the target), the status is `editbom`
- If the target is determining its software configuration (at the beginning of the deployment), the status is `0/`
- During the deployment, the status is `x/y`, where `x` is the current stage number and `y` is the last stage number (each stage ends with a restart)
- After a successful deployment, the status is `ok`
- In case of error, the status is `error` and the error time and message can be found in the table `ErrorLog`

Most target-specific parameters are stored directly in the `BOM` table. User-specific parameters are stored in the `UserProfile` table (and identified by the `UserID` key). The `Locale` field stores the Microsoft locale code, and the `TimeZone` field stores the Microsoft time zone index (a list of which can be found in the Windows registry).

There are also four freely configurable user categories that can be used to store information regarding the user (such as position, department, and location), and that can be used in the software matching mechanism.

The hardware inventory is stored in the three tables on the left, at the beginning of the deployment, after the BOM edition stage. There is one record in the `DiskInventory` table per disk detected, one record per target in the `DMIInventory` table (see the database for the exhaustive list of fields) and one record per PCI adapter in the `PCIInventory` table.

The `Settings` table lists deployment schemes. The actual settings for each deployment scheme are stored in an `.ini` file on the OS deployment server, whose prefix is made using the `DeplSet` identifier.

OS configuration-related tables

Every OS configuration is stored as one row in the `Groups` table. Each OS configuration is bound to exactly one system profile, but for multiboot systems there can be several OS configurations for a single system profile.

All system profile-dependent parameters (such as partitioning scheme) are stored in the `SystemProfile` table. Primary partitions are stored in the `PrimPart` field and logical partitions in the `LogiPart` field, as a space-separated list of partition `type:size` (size is in MB). The column `OSPart` is the index of the partition on which the operating system shall be installed. The image fields store the name of various image files on the OS deployment server (`OSImage` is the prefix used for the OS partition, `DiskImage` is the prefix used for all other partitions). The `Model` field is optional and used to perform computer model checking/locking if requested, to ensure that the correct operating system is deployed on each model of computer (check is made as a substring, that is, a system profile flagged for `Supra` matches any computer model with `Supra` in its model name, as found in the DMI tables). There are four freely configurable system categories, that can be used to store information regarding the system profile such as preconfiguration, distribution, and that can be used in the software matching mechanism (see the information later in this section for the explanation of table `SoftwareProfile`).

The relation between the `BOM` table and the `Groups` table is not direct, but is done through the `GroupingRule` table. In the simplest case, a row of this table only includes the `BomID` of a target record and the `Group ID` of the selected configuration. If several OS configurations are bound to a target, several rows are used. In a more complex case (automatic binding rules), the table acts as a dynamic relational index and follows a kind of rule-matching behavior. For each record of the table, if all specified values are matching those of a target, the corresponding OS configurations will be bound to it. The field `Condition` can include an additional Rembo-C condition to check before in addition to the pattern matching mechanism.

The `SoftwareItem` table (on the bottom right) record information regarding every software modules that can be deployed. Each software is identified by its `SoftItemID`. There is typically one record per software module, but there can be more than one if a package involves for instance both a file copy and a command line (each of which is described by a record). For each record, the `type` field identifies the kind of item:

- `copy` items are used to copy a tree of files to the OS partition. The `Source` is the file archive on the server, the `Dest` is the destination path.

- `run` items are used to start a command-line, for instance to run an unattended setup. The `Dest` is the full command line to run.
- `ini` items are used to make changes to an `.ini` (or `.inf`) file on the OS partition. The `Source` is the name of the `.ini` file on the OS deployment server, the `Dest` is the destination path and file name of the file to update.
- `diff` items are used to install a system snapshot. The `Source` is the file archive on the server.
- `floppy` items are used to start a non-DOS virtual floppy-disk image (in a ramdisk), for instance to run a vendor-specific firmware upgrade utility. The `Source` is the raw image name on the server.
- `part` items are used to start a DOS virtual floppy-disk image (as a ramdisk or on a partition), for instance to run a BIOS upgrade utility. The `Source` is the partition archive on the server, the `Dest` is the destination partition and mode.

Other software module parameters are the `Pass` number at which the software must be applied, and a boolean `SysOnly` that prevents the software from being made visible to the user in the BOM edition stage when turned on.

The choice of what software is to be installed is controlled by the `SoftwareProfile` table. This table acts as a dynamic relational index and follows a kind of rule-matching behavior. For each record of the table, if all specified values in the first sixteen columns are matched, the software item specified by the identifier in `SoftItemID` will be applied. For instance, software modules can be associated to a specific target by adding a record with the `BomID` of the target and the `SoftItemID` of the software. In the same way, software can be bound to a specific system profile or group. More complex OS configurations can also be used to implement the software binding rules.

OS deployment server configuration

These topics describe the different parameters available to configure the OS deployment server, and on how to modify the values of these parameters.

Global parameters

Global parameters are the main server side OS configuration parameters that can be modified either on the web interface or by editing the `rembo.conf` file.

The parameters are divided into seven categories that mirror the sections found in the web interface. They are:

- Debug level information
- Base parameters
- Web interface parameters
- Boot module parameters
- Network boot module parameters
- File access module parameters
- Network share module parameters
- HTTP Console Security parameters
- New targets default parameters

Most parameters can be modified in the web interface. Go to **Server** → **Server parameters** → **Configuration**.

Base parameters

- *Debug level information*

  GlobalDebugLevel `number` specifies the level of details that must be placed in the log file.

- *IP address of the backup server*

  Backup `ip-addr` defines the IP address of a backup server that can be used by the target if the primary server fails. The backup server must have the same `Net Password` as this server. For the backup recovery system to work, the backup server must also contain an exact copy of the files stored on primary server.

- *Maximum size of the server log files*

  MaxLogSize `number` limits the size of the log file generated by the OS deployment server. The maximum log size is specified in bytes, and applies to all log files created by the OS deployment server. If you do not specify this parameter is not set, or set to `0`, then log files are not limited in size. If a limit is set, and the server reaches this limit, the current log file is backed up and a new file is created.

- **CountersInterval** `minutes` specifies the interval, in minutes, for the server statistics measures.

- *Network interfaces used by the server*

  Interfaces `ip-addr` specifies the list of network interfaces that the OS deployment server uses when sending and receiving packets to and from targets. If you leave this option unspecified, the server uses the network interface corresponding to the official host name of the server. The format is a list of IP addresses, separated by spaces.

web interface parameters

- *Disable the HTTP module*

  DisableHTTP disables the web interface. If you set this parameter, run the `-c` command-line option to change server parameters.

  Note:

  – Disabling the HTTP module prevents you from using the Java API.

- *Disable HTTP SSL encryption*

  DisableSSL disables encryption in the Web interface and enables unencrypted HTTP connections.

- *TCP port for HTTP requests*, *TCP port for encrypted HTTP requests (SSL)*

  HTTPServerPort `port` specifies the TCP port used by the Web interface when listening for unencrypted HTTP requests. You can set this parameter to `80` if you want the Web interface to be accessible by typing the server host name in your Web browser. To make the Web interface accessible by typing `https` followed by the server host name, set this parameter to `443`.

- *Inactivity timeout before a session expires*

  HTTPSessionTimeout `minutes` specifies the timeout (in minutes) before Web interface users are automatically logged out. A typical value is 5 minutes.

- *Disable HTML contextual menus*

  DisableContextualMenu disables the contextual menu. When you use a right-click in the web interface, you will see the regular contextual menu of your browser.

- *Disable HTML drag-and-drop*

DisableDragAndDrop disables web interface drag-and-drop operations on selected icons. Drag-and-drop works on most common browsers, but it can be made unavailable in case of incompatibility.

- *Disable javascript drop-down lists*

  DisableJSDropDown disables enhanced menus in favor of standard menus. Standard menus do not recognize DHTML layers under Internet Explorer, therefore this parameter can cause problems.

- *Disable HTTP transfer optimization*

  DisableHTTPOptim disables the optimization of HTTP transfer in the web interface. Set this parameter if you do not want the web interface to group JavaScript™, stylesheets, and images into large files to reduce HTTP traffic and enhance the cache process. You must disable this optimization if you are creating console pages.

Boot module parameters

- *Disable DHCP/BINL module*

  DisableBOOT disables the PXE component of the OS deployment server.

  When this parameter is set to **Yes**, it is not possible to create network boot media.

  To prevent targets booting into the OS deployment server without disabling network boot media generation, you have two options:

  – Put the server in *closed server* mode by setting **Completely ignore unknown targets (closed OS deployment server)** to **yes** in the **Idle state** task template

  – Temporarily **disable DHCP proxy functionality**, which prevents any non-directed PXE request

- *Disable the DHCP proxy functionality*

  DisableDHCPProxy disables the DHCPProxy service on the OS deployment server. The server does not send extended DHCP replies (PXE replies) to DHCP discovery packets sent on the network by remote-boot target. When the DHCP proxy is unavailable the server must be configured to provide a complete PXE answer to PXE targets, including option 60 and option 43. DHCP proxy mode must be made unavailable if you have more than one OS deployment server operating on the same subnet.

- *Disable the multicast BINL functionality*

  BootNoMulticastDiscovery disables PXE multicast discovery support on the OS deployment server. When PXE multicast discovery is unavailable, target computers with multicast discovery configured in DHCP option 43 are not able to find the OS deployment server.

- *UDP port for TFTP requests*

  TFTPPort is the port used on the OS deployment server to receive TFTP request packets from PXE targets. The default value is 69. Note that unless your network equipment is automatically routing TFTP packets to a non-standard part, you must not change this setting because PXE always sends TFTP requests to port 69, without any possibility to change port.

- *Max. TFTP Segment Size*

  MaxTFTPSegSize `bytes` is the maximum size in bytes for TFTP segments. The default value is 512 but a smaller one can be given if the network configuration requires it.

- *Seconds to wait before starting a MTFTP stream*

  MTFTPStartDelay `secs` specifies the time, in seconds, to wait before starting a MTFTP transfer. This period of time is used by target computers to replicate

together. The higher the value is, the more replicated the deployment engine is. However, latency is introduced when booting target computers individually. The PXE standard default value is `2`.

- *IP address used by targets for MTFTP*

  MTFTPClients `port [:target-port ]` specifies the multicast IP address and port used by the OS deployment server when sending MTFTP data to target computers.

- *UDP port for MTFTP requests*

  MTFTPPort `port` specifies the UDP port used by the OS deployment server to receive MTFTP request packets from target computers. The default value is `4015`.

- *Seconds to wait before sending DHCP/BINL answers*

  BootReplyDelay `secs` specifies the number of seconds to wait before answering DHCP/BINL request packets sent by target computers. If you have more than one OS deployment server on the same subnet, target computers use the server with the lowest shortest reply delay. The default value is `0` (no delay).

- *Max. number of requests per minute (load balancing)*

  BootReplyLimit `number` specifies the maximum number of DHCP/BINL requests to an OS deployment server in one minute. The server discards boot requests coming from target computers when the specified limit is reached. A number of `0` indicates unlimited requests. This parameter can be used to implement load balancing over multiple servers.

- *UDP port for DHCP/BINL requests*

  BootDiscoveryPort `port` specifies the UDP port that the OS deployment server uses when listening for BINL boot requests. The default port used for a standard target computer is `4011`.

- *Multicast IP address used for BINL requests*

  BootDiscoveryAddress `ip-addr` specifies the multicast IP address that the OS deployment server listens to when waiting for BINL multicast requests coming from target computers. This value must match the value set in the DHCP option `43` sent to targets.

- *No-boot schedule*

  NoBootSchedule specifies the frequency at which the OS deployment server stops providing a PXE boot service. It is used in conjunction with `NoBootDuration`. Modify this parameter using the web interface only.

- *No-boot duration in minutes*

  NoBootDuration specifies the duration in minutes during which the OS deployment server stops providing a PXE boot service if a no-boot schedule is specified. It is used in conjunction with the parameter `NoBootSchedule`. Modify this parameter using the web interface only.

Network boot parameters

- *Disable the NBP module*

  DisableNBP disables the NBP component of the OS deployment server. Because NBP is used by target computers when booting, target computers are not able to boot if NBP is unavailable.

- *UDP port for NBP requests*

  NBPServerPort `port` specifies the UDP port used by the OS deployment server to receive NBP requests from target computers. The default value is `4012`. Only change this value if the default port is already used by another application on your computer.

Note: It is also possible to remote boot targets without using PXE. See Chapter 11, "Booting targets without using PXE," on page 289.

File access module parameters

- *Disable the FILE module*

  DisableFILE disables the FILE (NetFS, MCAST) component of the provisioning server. If you disable the FILE server module, target computers are not able to boot on this server.

- *Base directory for server files*

  BaseDir `string` specifies the base directory for the provisioning server. All paths included in the OS configuration are relative to this base directory. This is a mandatory parameter, and has no default value (if you are using file-based configuration mode, you must set it before starting the provisioning server for the first time).

- *Relative directory for the shared repository*

  SharedDir `path` specifies the path relative to the server file directory for storing the shared repository internal files. By default, this parameter is set to `shared`.

- *First multicast IP address used for MCAST transfers*

  FileMCASTAddress `ip-addr:port` specifies the multicast IP address used when sending packets to target computers. The OS deployment server can generate different multicast IP addresses using this parameter as the base for the first address in a range.

- *Number of multicast addressed to use for MCAST transfers*

  FileMCASTAddrCount `number` specifies the number of multicast addresses that the provisioning server can use when sending multicast data to target computers. This parameter sets the size of the range of multicast addresses that the server can use. This range begins with the address set by the parameter `FileMCASTAddress`.

- *Maximum number of PCAST connections*

  FileMaxPCASTSessions indicates the maximum number of PCAST sessions that the provisioning server accepts simultaneously. This parameter is relevant if the server is running in UNICAST mode and deploying a group with the UNICAST setting.

- *Port for FILE requests*

  FileServerPort `port` specifies the UDP port used by the provisioning server to receive transfer protocol control requests. The default value is 4013, but you can change this parameter if port `4013` is already used by an application on your computer.

- *Disable the TCP module*

  DisableTCP disables the TCP component of the provisioning server. Target computers might not be able to boot on this server if you disable the TCP server module.

- *Directory for internal files*

  DataDir `path` specifies the path relative to the provisioning server base directory for storing files accessible to the target computer. By default, this parameter is set to `files`.

Network share module

- *Network UNC path*

  NetworkShare `path` specifies the path of the shared partition directory of the OS deployment server. This parameter can be used to increase the deployment

speed of some operating systems. You need to have set the partition as read-only for the user entitled to access the network share.

- *User entitled*

  NetworkUser "string" is the name of the user entitled to access the network share. If the user is part of a domain, use the syntax Domain/User.

- *Password of the user*

  NetworkPasswd "string" is the password used to access the network share. The password is stored in an encrypted format in the rembo.conf file.

Note: If the Network UNC path, the User entitled, or the Password of the user are not correct, you get an error when attempting deployment using the network share. If the Network UNC path is empty, the deployment does not try to use a network share.

New targets default parameters

Targets are assigned a set of default parameters.

The following parameters can be modified in the web interface, or in the rembo.conf file.

- *Completely ignore unknown targets/ Let unknown computers contact another PXE server/ Make unknown computers boot on their hard disk/ Make unknown computers boot on their hard disk when there is no pending task*

  DefaultPXEBootMode { normal | hdboot | ignore | otherPXE | BootIfIdle} When the OS deployment server becomes aware of an unknown PXE target trying to boot from the network, it can decide whether to add the target automatically to its internal database. To add the new target, include normal in your argument list. To exclude unknown targets, add ignore in your argument list. You can leave them alone to discover another PXE server, force them to boot on their hard-disk, or force them to boot on their hard disk when no task is pending. Add otherPXE, HDBoot, or BootIfIdle in your argument list to obtain the wanted behavior.

- *Tivoli Provisioning Manager for OS Deployment kernel options*

  DefaultOptions [autoboot] [, nousb] [, noautousb] [, novesa] [, noapm] [, unicast] [, noudma] [, noprotpart] [, realmodedisk] [, realmodePXE][, routepxeirq] The argument of this parameter is a list composed of the following elements, separated by commas: autoboot reboots automatically on unrecoverable errors, NoUSB disables USB devices, noautousb enables USB devices only when no PS/2 connection is detected, NoVESA disables the graphic interface, NoAPM disables the APM, Unicast disables multicast, NoIGMP2 disables IGMP version 2, NoUDMA disables Ultra-DMA, NoProtPart disables ATA-5 features, RealModeDisk disables enhanced disk access, RealModePXE disables enhanced PXE access, andRoutePXEIRQ reroutes network IRQ separately from the disk controller IRQ .

- *Human interface locking*

  DefaultLockOut [none] [, mouse] [, keys] [, screen] [, all] The argument of this parameter is a list composed of the following elements, separated by commas: Mouse disables the mouse, Keys disables the keyboard, Screen disables the screen, and ALL disables the mouse, keyboard, and screen.

Other parameters for new targets can be configured only by editing the rembo.conf file.

- *Preferred screen resolution*

DefaultResolution `"string"` specifies the default screen resolution for new targets.

- *Authentication domain*

  DefaultAuthDomain `"string"` specifies the default authentication domain to use when a target wants to check credentials. This domain can be overridden individually for each target.

- *Server for files services*

  DefaultFileServer `ip-adr:port` specifies the default file server, given by its IP address (and port number), for new targets.

- *Boot redirection servers*

  DefaultBootRedirection`ip-adr [, ip-adr]` specifies the IP address of an OS deployment server to which boot requests of the targets must be redirected. Optionally, a second redirection server can be indicated, in which case you must separate the two IP addresses by a comma.

HTTP Console Security parameters

In the web interface, these parameters can be modified on the **Server** → **Server parameters** → **HTTP Console Security** page.

- *Super-user login*

  HTTPAdminName `"name"` specifies the superuser login of the OS deployment server administrator, in order to access all OS configuration parameters of the server. There is only one superuser login.

- *Super-user password*

  NetPassword `"string"` specifies the password used by the main OS deployment server administrator This password is used to protect the server files against illegitimate remote access.

- *New security role*

  HTTPRole `"name" {Members "stringlist" AllowPages "stringlist" AllowGroups "stringlist"}` allows new members to access the web interface by specifying which pages are available to them and which administrative groups these new members can manage.

Authentication domains

An authentication domain is a group of parameters related to the authentication of users by Tivoli Provisioning Manager for OS Deployment.

The term *domain* has nothing to do with Windows NT domains, or NIS+ domains. Parameters contained in an authentication domain define how user and password information entered on a target workstation are checked by the OS deployment server. For example, they can be checked against the local server database of users, or through a remote authentication device. Additionally, Tivoli Provisioning Manager for OS Deployment allows you to restrict the search for matching users to a single user group for greater flexibility.

Note: In order to setup security roles, you must create an authentication domain named `HTTP`.

The OS deployment server supports several authentication protocols, depending on the platform of the server:

- On Windows, a user identity can be verified with the local user database, or a remote user database (but still on a Windows server)

- On UNIX, the user identity is verified with the standard user database functions, which can be configured to use local files, NIS or NIS+. PAM is used if the server is running on Linux
- On both platforms, the OS deployment server can use the authentication standard *Radius* to perform the authentication with a device supporting the Radius protocol, or with a Radius gateway for Netware (NDS).

  Note:

  1. On Windows, authentication can be done only if the user running Tivoli Provisioning Manager for OS Deployment has the `Act as part of the Operating System` privilege. By default, the user account used for services has this privilege (the `SYSTEM` account).
  2. If you are using NT remote authentication, the OS deployment server must have this privilege on the remote computer. You must start the IBM Tivoli OS Deployment Server (*remboserver*) and IBM Tivoli Web Interface Extension (*remboagent*) services with domain accounts that have the NT remote authentication privileges.
  3. Authentication with redeployment does not work if the target is offline (the target has no network connection and boots from the hard disk). A message warns the user.

If you are modifying the server configuration in the web interface, you can add a new authentication domains by clicking **New auth. domain** in **Server → Server parameters → Predefined channels** and entering the appropriate information.

If you are maintaining your server configuration directly in the text file `rembo.conf`, you must add authentication domains in the configuration file. Each domain starts with `AuthLocalDomain`, `AuthNTDomain` (on Windows only), or `AuthRadiusDomain`, and is followed by the name that you want to attribute to this domain. For example:

```
AuthLocalDomain HTTP {}
```

There are three types of authentication domains:

- *Local domains* uses the local user database to authenticate a user. The optional `UserGroup` parameter can be used to restrict the verification to a specific group of users. This type of domain is supported by both Windows NT and UNIX.
- *Remote NT domains* forwards authentication requests to the NT server specified by the mandatory parameter `AuthServerName`. The parameter `UserGroup` can be used to restrict the verification to a specific group of users. This type of domain is supported only in Windows.
- *Radius domains* forwards authentication requests to the Radius-compliant device specified by the parameter `AuthServerAddress`. The value of the parameter `RadiusSecret` is used as the password for the Radius communication, and must match the password stored in the configuration of the Radius device for the protocol to work.

TCP tunnels

A TCP tunnel is a way to provide TCP connectivity to target computers.

- In Tivoli Provisioning Manager for OS Deployment, you must have an `ODBC` tunnel to access the database. This tunnel is present by default.
- A *sendmail* TCP tunnel is mandatory to receive e-mail notification at the end of a deployment.

Creating a new hardware rule

Hardware rules are useful to work around problems related to some specific hardware.

At boot on the Tivoli Provisioning Manager for OS Deployment, a target will be compared to the hardware rules and if there is a match within the list, some specific features (potentially making problems for that hardware) will be unavailable, for instance: "enhanced PXE access", "ATA-5 features", and so on. When Tivoli Provisioning Manager for OS Deployment is shipped there is a built-in default hardware rule list that is a list of the known hardware making problems at the time of the build. However, new hardware coming out is not listed here and if any new hardware causes problems it can be resolved by manually disabling specific features.

At remote-boot the hardware of a target is immediately detected and compared to the hardware rule list, the selected features will then be unavailable if necessary.

To create a **New hardware rule**:

1. Go to **Server** → **Server parameters** → **Hardware handling**.
2. Click **New hardware rule**.
3. Select one or several of the offered options.

   Specific PCI device present:

   PCI (Peripheral Component interconnect) devices are peripheral components connected on the PCI bus of the system board. These are typically: network cards, sound cards, Disk/USB controllers and so on.You can specify the PCI device that the rule is to be based on and enter information in the following fields:

   - Vendor ID (4 hex digits)
   - Device ID (4 hex digits)
   - Revision (optional, 2 hex digits)

   Note: To view Vendor/Device/Revision information,go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details. Look in the **Inventory** tab, under **PCI devices**.

   Specific Model name:
   You can specify a model name for the rule it is based on. For example: eServer™ xSeries®.

   Specific Linux distribution
   You can specify the Linux distribution name for the rule it is based on. For example: SUSE , Red Hat.

   Specific type of chassis
   You can specify the Type of chassis for the rule it is based on. For example: Single Chassis , Multi Chassis.

   Specific OS Architecture
   You can specify if you are working with a 32–bit or a 64 bit architecture.
4. Depending on your selections, the wizard provides steps with easy-to-follow instructions to create the new hardware rule.
5. Select what the rule will do if chosen criteria are met.

- Change Tivoli Provisioning Manager for OS Deployment kernel flags. The options available are:
  - Reboot on unrecoverable errors
  - Disable USB
  - Disable Auto USB
  - Disable graphic interface
  - Disable APM
  - Disable multicast
  - Disable IGMP version 2
  - Disable Ultra-DMA
  - Disable ATA-5 features
  - Disable enhanced disk access
  - Use BIOS for CD/DVD ROM access
  - Disable enhanced PXE access
  - Try to optimize IRQ
- Change Linux kernel parameters. Changing the Linux kernel parameters option requires a command line parameter to be entered that will be used when Linux is deployed and rule criteria are met. Enter the command line in the Linux kernel parameters field.
- Enable remote power management feature, using either Intel Active Management Technology or by performing the following, Go to **Server** → **Server parameters** → **Hardware handling** → **New hardware rule** . Here you can define a rule that will show the hardware where Remote power management is available. In the Host details page if there is a matching hardware rule for remote power management then **Switch on** and **Switch off** buttons will be visible.

  Note: These targets can also be switched on by checking the " Try to wake up targets using management interface" in the deployment wizard.

6. Follow the wizard instructions to complete the rule.

At remote-boot the hardware of a target is immediately detected and compared to the hardware rule list, the selected features will then be unavailable if necessary.

Default port numbers

During the installation process, you can choose to accept the default port numbers proposed or you can specify alternate unused port numbers.

The following table lists the default port numbers that are used by Tivoli Provisioning Manager for OS Deployment.

Table 11. Default port numbers used by the server

Name of port	Default port number
DHCP	67 UDP
PXE BINL	4011 UDP
TFTP	69 UDP
MTFTP	4015 UDP
NBP	4012 UDP
FILE	4013 UDP and TCP

Table 11. Default port numbers used by the server (continued)

Name of port	Default port number
MCAST	10000-10500 UDP Address: 239.2.0.1 - 239.2.1.1

Table 12. Default port numbers used by the web interface extension (rembo agent)

Name of port	Default port number
Listening port	4014 UDP
The web interface extension (rbagent) connects to ports 4012 and 4013 on the OS deployment server to retrieve information and files. Port 4012 is normally used when sending commands, while 4013 is used when dealing with file transfers. The source port of the agent when connecting to 4012 and 4013 is random.	

Table 13. Default port numbers used by the client PXE

Name of port	Default port number
DHCP	68 UDP
MTFTP	8500-8510 UDP Address: 232.1.0.1
MCAST	9999 UDP
Remote control (agent)	4014 UDP

Configuring OS deployment server with a text file

A single text file (config.csv) can be used to configure all the OS deployment servers in a complex environment.

OS deployment servers can be configured using a file named `config.csv` (comma separated values). The advantage of this configuration file over other configuration methods is that the same file can be used to supply different configuration parameters to a large number of OS deployment servers installed in a big organization.

You must place the file `config.csv` in directory `<datadir>/global/rad`, for instance `c:\TPMfOS Files\global\rad` on a Windows server.

The same configuration file can be used for all the OS deployment servers in the organization. But it must be deployed on all the OS deployment servers. The reason is that each OS deployment server reads the configuration file from its local file system, and filters out all irrelevant information. The file can also be different for different parts of the organization, or even for each OS deployment server. It is up to the operator to select the most convenient management strategy.

The file is read automatically at server startup, or on demand, using the web interface extension command rad-configure. Some of the changes automatically trigger a server restart to be properly applied.

Each line in the file describes one OS deployment server. The fields are separated by a semicolon **;**. Fields can be optionally quoted with double quotes ("). The double quotation mark itself must be repeated to differentiate it from a quotation mark. Fields containing a semicolon or a double quotation mark must be quoted (see Table 14 on page 321).

Table 14. Code

Field	Interpretation
abcd1234	abcd1234
"abcd1234"	abcd1234
"abcd;1234"	abcd;1234
"abcd""1234"	abcd"1234
abcd""1234	abcd1234
abcd";"1234	abcd;1234

Note: The first line of the `config.csv` file must contain the label of the fields.

This file format can be conveniently imported and exported by Microsoft Excel and OpenOffice.

Field description

Within one line, each field is interpreted according to its position. The interpretations are detailed in the information shown later in this section.

HostName

This can be either the host name (for example: rembomaster) or the fully qualified name (rembomaster.rembo.com) of the target targeted by this line of the file.

There are two very important constraints on this field.

1. All the letters in the name must be lowercase.
2. The name must be immediately followed by the semicolon.

If one of these two constraints is not true, the script which parses the file will not be able to recognize the line and the server will not be configured properly.

As a side effect, all the lines which do not start with a valid host name are considered to be a comment. The prefix ";RBO " is reserved for internal use (for example: ";RBO;DB2_BIN_DIR;")

When an OS deployment server has recognized its HostName, the parameters found in the line overrides the values that were supplied earlier, for instance when running the setup program or by the means of a rembo.conf file.

Interfaces

The list of IP addresses used by this OS deployment server. If the server has several interfaces, PXE remote boot is available only for the interfaces listed in this field. Addresses must be given in the dotted notation (for example: 192.168.168.16), and separated by one space character.

DbName DbUser DbPass

Connection information for the server database used by this OS deployment server. Typical value for DbName is "AutoDeploy". The password can be hidden using the web interface extension command rad-hidepassword.

Note: If *MasterDbName* is used, then *DbName* is mandatory.

DbName can be either the name of an existing ODBC source, or a valid string for SQLDriverConnect, enclosed in square brackets.

Example (MS SQL Server)

```
[DRIVER={SQL Server};SERVER=<hostname>;UID=<user>;PWD=<pass>;
DATABASE=<db>[DRIVER={SQL Server};]
```

Example (DB2®)

```
[DRIVER={IBM DB2 ODBC DRIVER};SERVER=<hostname or IP>;DATABASE=<db>;
UID=<user>;PWD=<pass>;]
```

With this syntax, which is not natively recognized by DB2, the product will automatically use or create a DB2 Node that matches <hostname or IP>, and a catalog entry that matches <db>.

Additional fields (DB2 database)

Some optional parameters can be added to the db2 command:

- - REMOTEPORT: this parameter will be set to 50000 if not present
- - REMOTE_INSTANCE: this parameter will be added if present
- - SYSTEM: this parameter will be added if present
- - OSTYPE: this parameter will be added if present

Example:

```
[DRIVER={IBM DB2 ODBC DRIVER};SERVER=srvkludge-1.rembo.private;
DATABASE=GwA;UID=administrator;PWD=poiuz;OSTYPE=AIX; REMOTEPORT=1288;
REMOTE_INSTANCE=devtinv4; SYSTEM=rtxidb2;]
```

will generate the command db2 catalog tcpip node SRVKLDGR remote srvkludge-1.rembo.private server 1288 REMOTE_INSTANCE devtinv4 SYSTEM rtxidb2 OSTYPE AIX

MasterIP

There are three choices for this field:

- "" (empty string) indicates that the server is a stand-alone server and that no replication is necessary.
- "SELF" indicates that the server is a parent at the top of the hierarchy. Some replication tasks are performed, but only in the "downward" direction.
- IP address of the *parent* OS deployment server server which is used for replication.

MasterDbName MasterDbUser MasterDbPass

Connection information for a second optional OS deployment server database. These fields must be left empty if the field MasterIP is "" or "SELF". Otherwise, they must lead to the same database as the parent's DbName. The password can be hidden using the web interface extension command rad-hidepassword.

Note: When MasterDBName is included, DbName becomes mandatory.

BinDir

This can only be used for radtcm.pak.

The directory where the wapmrpt.exe is located. Using forward slashes as separators, for example: `c:/temp`

Description

This can only be used radtcm.pak.

The Tivoli endpoint identifier of this server.

Reporting

This can only be used for radtcm.pak.

A string of letters controlling which reporting to wapmrpt must be performed by this OS deployment server. The letter **u** stands for **updates** and activates the reporting of branch update events, expected after the web interface extension command sync-validate .The letter **d** stands for **deployments** and activates the reporting of deployment events, expected after web interface extension command rad-deployhost. The order of letters in the string is irrelevant. If the string is empty, no reporting is done. The letter **k** stands for **keep files** and can be used for debugging.

AutoSync

A string of letters controlling server replication. The order of letters in the string is irrelevant. If the string is empty, no replication is done. The letters have the following interpretation:

f Replicates files: when this letter is present, the files on this OS deployment server are copied from the parent periodically. File replication is the mechanism to retrieve the "missing" files for the objects stored in the local database of the server. The *SyncSchedule* flag performs the same operation at a given time. Therefore, the *f* flag must not be set if another mechanism is used for file replication, for example: sync.pak or *SyncSchedule*.

c Replicates OS configurations: when this letter is present, the OS configurations in this server database are copied from the parent's database periodically. This should be set if sync.pak is used to synchronize the files, but the server has its own database (for example: on spokes).

s Replicates software modules: same as **c**, but for software modules.

d Replicates deployment schemes: same as **c**, but for deployment schemes.

h Replicates targets: when this letter is present, the targets in this server's database will be copied to/from the parents's database periodically. This should be set at the branch level, to provide target management from the spokes.

l Light target replication: when this letter is present, do not copy parent's targets which are known to belong to a different branch of theOS deployment servers hierarchy. Copy only new targets and targets which have booted on this server, or one of its children. With this letter, **h** is automatically added.

b Replicates blacklist: when this letter is present, the hardware compatibility blacklist is copied from the parent periodically.

The new values are unconditionally propagated to other servers using the same DbName.

PollInterval

The interval in minutes between two server replications. The default value is 1.

DebugLevel

The required global debug level for this server.

DB2BinDir

This can only be used for DB2

The directory where the command db2cmd is located. Using forward slashes as separators, for example: `c:/db2/bin` The default location is `c:/Program Files/IBM/SQLLIB/BIN`. It can be overriden globally for all servers with a special comment line ;RBO;DB2_BIN_DIR;<path>, and also individually for each server with this column.

SyncSchedule

You can use it only if the flag *f* in *AutoSync* and the sync.pak package are not used. Replication occurs according to the timestamp of the different servers. For example if in config.csv, the replication time is set to 9.00 PM for a level 2 server, the replication occurs at 9.00 PM of level 2 server and not 9 PM of the parent server.

The schedule for file replication. Tells the server when it should copy the files from its parent. This should not be set if another mechanism is used for file replication, for example: sync.pak.

Example: synchronize once on 2006-02-22, at 3PM:

```
type=once;date=2006-2-22;time=15:00
```

Example: synchronize every day, at 3PM:

```
type=daily;period=1;time=15:00
```

Example: synchronize every week, every day except Thursday, at 3PM ("days" is a bit mask, where Monday=1, Tuesday=2, Wednesday=4, Thursday=8, Friday=16, Saturday=32, Sunday=64):

```
type=weekly;period=1;days=119;time=15:00
```

Example: synchronize on the first day of each month, at 3PM:

```
type=monthly;day=1;time=15:00
```

If flag **f** is used in `AutoSync`, scheduling a file replication here is not necessary.

SyncBandlim

This can only be used when sync.pak is not used

When SyncSchedule is used, SyncBandlim can be used to limit the network bandwidth used for file replication. The syntax is a number followed by a white space and one of "Mbits/s" or "Kbits/s". For instance "15 Mbits/s" or "192 Kbits/s". It is also possible to give just one integer, in this case the units are kilobytes per second.

APISecret

This can be used for the Java API, since version 5.1.0.2

The password that must be supplied when connecting to this server through the Java API. Must be hidden using the web interface extension command rad-hidepassword.

APITrusted

This can only be used for the Java API, since version 5.1.0.3

The list of trusted targets that can use the Java API when there is no APISecret. If APISecret is defined, this field is not used and any Java target that knows the secret can create a new session. If APISecret is empty or undefined, the server checks that the target IP address is in this list before creating a new session. Addresses must be given in the dotted notation (for example: 192.168.168.16), and separated by one space character

TPMReporting

This can only be used for TPM 7.1 and Director 6.1

A string of letters controlling events reported to TPM. The order of letters in the string is irrelevant. If the string is empty, no event is reported. The letters have the following interpretation:

b bare-metal targets : when this letter is present, targets without Description are announced to TPM. An attempt is made every time the OS deployment server is restarted. Fields TPMUser, TPMPass, TPMPort and TPMBinDir are required when this letter is present.

TPMUser

This can only be used for TPM 7.1 and Director 6.1

User name for thesoapcli command line.

TPMPass

This can only be used for TPM 7.1 and Director 6.1

Password for the soapcli command line, hidden in the same way as APISecret.

TPMPort

This can only be used for TPM 7.1 and Director 6.1

HTTP port for thesoapcli command line. Defaults to 4080.

TPMBinDir

This can only be used for TPM 7.1 and Director 6.1

The directory where soapcli.cmd or soapcli.sh is located. Using forward slashes as separators, `c:/program files/ibm/director/tpm/tools/soapclient`

Mandatory fields

The only mandatory field in `config.csv` is HostName.

However, a typical configuration for a multidatabase architecture also includes

- DbName
- MasterIP
- MasterDbName

- AutoSync

Moreover, the fields DbUser DbPass MasterDbUser MasterDbPass are often needed.

For a use with the JavaAPI, a typical configuration contains only HostName and APISecret.

Example

Typical settings for a large organization

HUB:

Single server at the top of the production servers hierarchy. Contains the latest version of all the deployment objects but not of the targets for scalability reasons.

- C) DbName: name of an ODBC source that points to the database "AutoDeployHUB"
- F) MasterIP: "SELF"
- G) MasterDbName: "", no master
- L) Reporting: "", no event reporting
- M) AutoSync: "", this server owns the up-to-date database

SPOKE:

Main management point for a large unit in the organization. Synchronizes its deployment objects from the HUB. Stores target data for all the servers in the list but doesn't propagate it upwards for scalability reasons.

- C) DbName: name of an ODBC source that points to the database "AutoDeploySPOKE"
- F) MasterIP: HUB's IP address
- G) MasterDbName: name of an ODBC source that points to the database "AutoDeployHUB"
- L) Reporting: "ud", this is the right place to report to TCM Tivoli Provisioning Manager for OS Deployment plugin
- M) AutoSync: "csd" to synchronize deployment objects (with sync.pak), "fcsd" to synchronize files as well (without sync.pak)

PROXY:

OS deployment server for a medium unit in the organization. Has its own database to perform deployments even when not connected to the SPOKE.

- C) DbName: name of an ODBC source that points to the database "AutoDeployPROXY"
- F) MasterIP: SPOKE's IP address
- G) MasterDbName: name of an ODBC source that points to the database "AutoDeploySPOKE"
- L) Reporting: "", no event reporting
- M) AutoSync: "csdh" to synchronize deployment objects and all targets (with sync.pak), "fcsdh" to synchronize files as well (without sync.pak), "csdl" or "fcsdl" to synchronize only a subset of targets

BRANCH:

OS deployment server for a small unit in the organization. Requires good connectivity with the PROXY.

- C) DbName: name of ODBC source that points to the database "AutoDeployPROXY"
- F) MasterIP: PROXY's IP address
- G) MasterDbName: "", no DB replication is needed because we use the same database as the PROXY
- L) Reporting: "", no event reporting
- M) AutoSync: "" if file replication is done using sync.pak, **f** if Tivoli Provisioning Manager for OS Deployment must synchronize files periodically

Parameter reference

Parameter reference contains syntax information on OS deployment server parameters.

Global parameters

The following list provides detailed explanations of the parameters used in the `rembo.conf` file and the web interface.

Backup

Name
Backup — IP address of backup server

Synopsis
Backup `ip-addr`

Location in the server configuration

- *web interface* > **Server Parameters > Server configuration: Base parameters**
- *rembo.conf:* Beginning of the file

Description
This parameter defines the IP address of a backup server for this OS deployment server. The backup server IP address is sent to all target computers connected to this server. If this server fails, the targets detect that the primary server has failed, and automatically switch to the backup server.

Here are some considerations regarding the OS deployment server running at the IP address specified by this parameter (*the backup*):

- The backup must be configured to use the same UDP ports as this server for the NBP, FILE, MCAST and PCAST services (that is for all specific services);
- Its file system must be strictly identical to the file system of this server, so that boot agents can switch from one server to the other in the middle of a file transfer;
- Ideally, the backup must be configured to answer DHCP discovers and PXE discovers for the same targets as this server, but with a higher delay (see (BootReplyDelay). This ensures that the remote-target boots on the backup server on the next boot.

BaseDir

Name
BaseDir — Directory for all OS deployment server files

Synopsis
BaseDir " directory"

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: File access module**
- *rembo.conf:* Beginning of the file

Description
By default under Windows, BaseDir points to c:\Program Files\Common Files\IBM Tivoli.

You must set this parameter to the directory where you have installed the OS deployment server executable program. If you are using the Windows version the BaseDir parameter is automatically configured during the setup process.

BaseDir contains a subdirectory named packages which holds the .pak server extensions.

Note: Always use forward slashes (/) in all path names you enter in the text-based configuration file.

Examples
```
BaseDir "c:\Program Files\Common Files\IBM Tivoli"
BaseDir "/usr/local/tpmfos"
```

BootDiscoveryAddress

Name
BootDiscoveryAddress — Multicast address used for PXE discovery requests

Synopsis
BootDiscoveryAddress ip-addr

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description
If PXE multicast boot discovery is enabled on the OS deployment server (see (BootNoMulticastDiscovery)), this option must be set to the multicast IP address used by the remote-boot targets when performing PXE discovery.

If this option is not set, Tivoli Provisioning Manager for OS Deployment uses the IP address 232.2.0.1.If the OS deployment server is in the same subnet as the DHCP server, PXE discovery is not required because the remote-boot targets know that the PXE server is on the same computer as the DHCP server (option 60 set to PXEClient). If the OS deployment server and the DHCP server run on different computers, but are on the same IP subnet, PXE discovery is still not required because the OS deployment server intercepts DHCP requests and provides PXE information at the DHCP stage. You only require

PXE discovery if the remote-boot targets are not on the same subnet as the OS deployment server, or if your existing PXE infrastructure is already setup for PXE discovery.

Example

```
BootDiscoveryAddress 232.5.6.7
```

BootDiscoveryPort

Name

BootDiscoveryPort — IP port used when listening to PXE discovery requests

Synopsis

BootDiscoveryPort `port`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description

This parameter must be set to the port used by remote-targets when sending PXE boot requests (multicast boot discovery, or BINL discovery). The default value works with PXE bootroms which have not been modified, therefore there is no reason to change this value unless you know what you are doing.

If this option is not set, Tivoli Provisioning Manager for OS Deployment uses the IP port `4011`.

Example

```
BootDiscoveryPort 5011
```

BootNoMulticastDiscovery

Name

BootNoMulticastDiscovery — Disables PXE multicast discovery support

Synopsis

BootNoMulticastDiscovery `yes/no`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description

If the parameter `BootNoMulticastDiscovery` (without argument) is present in the configuration file (rembo.conf config), then PXE multicast discovery is disabled on the server. The OS deployment server does not answer PXE multicast packets received from remote-boot targets.

PXE multicast discovery is used by remote-boot targets if your existing PXE infrastructure is configured to use PXE discovery. Multicast discovery is not needed if the remote-boot targets and the OS deployment server are on the same subnet, because the targets use other mechanisms to discover the OS deployment server (DHCP proxy, or DHCP option 60).

If you know that you do not need PXE multicast discovery, you can disable it by setting this parameter to true. This is suggested if you are in a large company or institution where other PXE targets can be configured to use multicast discovery on other subnets, but you do not want your server to reply to these targets.

Example

```
BootNoMulticastDiscovery
```

BootReplyDelay

Name

BootReplyDelay — Sets the delay before answering discovery requests

Synopsis

BootReplyDelay `secs`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description

This parameter sets the number of seconds to wait before answering a PXE discovery request (PXE discovery enabled), or a DHCP request (ProxyDHCP enabled). The default value is 0 (no delay).

The only reason to delay a PXE reply is when two or more OS deployment servers are configured to run on the same subnet, with the same list of targets. The parent server can be configured with a delay of 0 seconds (answer immediately), and other servers with a delay of 2 seconds. If the parent server fails, backup servers handle remote-boot target requests.

Example

```
BootReplyDelay 2
```

DataDir

Name

DataDir — Directory for internal files

Synopsis

DataDir `" path"`

Location in the server configuration

- *web interface* > **Not available**
- *rembo.conf:* Beginning of the file

Description

The server uses the directory specified by `DataDir` to store its internal files, in particular all the files stored on its file system.

You can use this parameter to specify a new directory for the internal files of the server, for example if you want to store the server files on a new storage device with more space, but you want to keep the executable files and configuration files on the original location.

All the files stored in the directory specified by `DataDir` must not be modified, or the OS deployment server might not work correctly. If the specified directory does not exists, the server tries to create it.

Examples

```
DataDir "/mnt/morespace/rembo"
DataDir "C:/TPMfOS"
```

DefaultAuthDomain

Name
DefaultAuthDomain — Default authentication domain

Synopsis
DefaultAuthDomain " domain"

Location in the server configuration
- *web interface* > **:Not available**
- *rembo.conf:* Beginning of the file

Description
This is the name of the default authentication domain to use when a target sends an authentication request to the server to identify a user. This name must match the name of an existing authentication domain as defined in *Authentication domains*.

Example

```
DefaultAuthDomain "HTTP"
```

DefaultFileServer

Name
DefaultFileServer — Defines a new server for FILE services

Synopsis
DefaultFileServer ip-addr

Location in the server configuration
- *web interface* > **Not available**
- *rembo.conf:* Beginning of the file

Description
This parameter can be used to specify the default IP address of the target serving the NETfs and MCAST protocols. If you want to use the same OS deployment server for all services, you do not have to specify this parameter.

The target defined by this parameter must be configured to use the same network password (NetPassword) as the server containing this parameter.

This parameter is applied to all the targets contained in the group.

Example

```
DefaultFileServer 172.16.8.9
```

DefaultGroup

Name
DefaultGroup — Default group for unknown targets

Synopsis
DefaultGroup " name"

Location in the server configuration
- *web interface* > **OS deployment >Target Monitor: Default group**
- *rembo.conf:* Beginning of the file

Description

This parameter defines the default administrative group for unknown targets.

The "name " argument must be an existing administrative group.

Example

```
DefaultGroup "MyNewHosts"
```

DefaultLockOut

Name

DefaultLockOut — Locks peripherals on unknown targets

Synopsis

DefaultLockOut [none] [, mouse] [, keys] [, screen] [, all]

Location in the server configuration

- *web interface* >**OS deployment > Target Monitor: Default group**
- *rembo.conf:* Beginning of the file

Description

Use this parameter to lock peripherals on the remote-boot targetsduring the time Tivoli Provisioning Manager for OS Deployment is active on the unknown target. You can for example lock the mouse or/and the keyboard so that the user cannot interrupt an automatic image restoration process.

Peripherals are automatically unlocked when Tivoli Provisioning Manager for OS Deployment gives the control to the operating system (with a HDBoot, LXBoot, RDBoot or DeviceBoot).

Example

```
DefaultLockOut keys, mouse
```

DefaultOptions

Name

DefaultOptions — Startup options for the remote target computer

Synopsis

DefaultOptions [admin] [, autoboot] [, nousb] [, novesa] [, noapm] [, unicast] [, noigmp2] [, noudma] [, noprotpart] [, realmodedisk] [, realmodepxe] [, routepxeirq]

Location in the server configuration

- *web interface* >**OS deployment > Target Monitor: Default group**
- *rembo.conf:* Beginning of the file

Description

Twelve different options can be set for targets. These options are used when unknown targets run under Tivoli Provisioning Manager for OS Deployment:

- Admin: forces the target to run in administrative mode. Administrative mode adds options in the start menu on the target: Interact, to run the interactive evaluator, and Purge cache to purge the local disk cache. Additionally, functions defined in the admin.rbx plugins are available (File Manager, TextEdit,...).
- Autoboot: tells the target to automatically reboot on unrecoverable errors. Unrecoverable errors are low-level errors which cannot be intercepted with the exception mechanisms. By default

unrecoverable errors are displayed on the screen and the computer is halted. You can define the `autoboot` option if you want to reboot on unrecoverable errors.

- `NoUSB:` disables USB devices on the remote-boot target.
- `NoVESA:` disables all the graphical interface on the remote-boot target.
- `NoAPM:` disables advanced power management on the remote-boot target.
- `Unicast:` uses unicast instead of multicast when downloading files from the OS deployment server.
- `NoIGMP2:` disables IGMP version 2 to force using IGMP version 1 on the remote-boot target.
- `NoUDMA:` disables UltraDMA support for all IDE hard-disks on the remote-boot target. This can solve problems on poorly designed hardware or buggy chipsets, but at a high cost in terms of performance.
- `NoProtPart:` disables ATA-5 features.
- `RealModeDisk:` disables enhanced disk access.
- `RealModePXE:` disables enhanced PXE access.
- `RoutePXEIRQ:` reroutes network IRQ separately from the disk controller IRQ.

Example

```
DefaultOptions admin
```

DefaultPXEBootMode

Name

DefaultPXEBootMode — Selects default PXE boot behavior

Synopsis

DefaultPXEBootMode { `normal | hdboot | ignore` }

Location in the server configuration

- *web interface* >**OS deployment > Target Monitor: Default group**
- *rembo.conf:* Beginning of the file

Description

This option defines how the OS deployment server answers PXE boot requests coming from unknown targets. When mode is `normal`, the OS deployment server processes the boot request. When mode is `hdboot`, the OS deployment server tells the unknown target to immediately boot on the hard disk. When mode is `ignore`, the OS deployment server does not answer the boot request, thus giving the opportunity for another PXE server to answer.

Example

```
DefaultPXEBootMode normal
```

DefaultResolution

Name

DefaultResolution — Default screen resolution for new targets

Synopsis

DefaultResolution " `path` "

Location in the server configuration

- *web interface* > **Not available**

- *rembo.conf:* Beginning of the file

Description
This parameter defines the default screen resolution for target on a new target.

Example

```
DefaultResolution "800x600"
```

DisableBOOT

Name
DisableBOOT — Disables PXE services

Synopsis
DisableBOOT

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description
Use this option to disable the PXE component of the OS deployment server. Targets are not able to boot on PXE anymore if this option is included in the configuration file. Use this option if you want to use the OS deployment server for remote file access only (for example when setting up the server).

Not included in the default configuration.

Example

```
DisableBOOT
```

DisableContextualMenu

Name
DisableContextualMenu — Disables the specific contextual menus in the web interface

Synopsis
DisableContextualMenu

Location in the server configuration

- *web interface* > **Server Parameters >server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description
Add this option to the configuration file if you want to disable the specific contextual menus in the web interface. This can be useful if you want access to the regular contextual menu of your Web browser, in order, for example, to view the source page.

Not included in the default configuration.

Example

```
DisableContextualMenu
```

DisableCookies
This parameter is deprecated from version 7.1.1.3 onwards. The web interface does not use cookies anymore.

DisableDHCPProxy

Name

DisableDHCPProxy — Disables answers to DHCP discovers (DHCPProxy)

Synopsis

DisableDHCPProxy `yes/no`

Note: In rembo.conf configuration, the keyword `DisableDHCPProxy` is used without argument. If the keyword is present, it is assumed to be set to the value `yes` automatically.

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description

If this parameter is set to `yes`, the OS deployment server does not send PXE replies to DHCPDISCOVER packets sent by remote-boot targets. You can set this parameter to `yes` if you do not need this feature because either the DHCP server and the OS deployment server are on the same target, or you are using PXE multicast discovery.

The default value is `false` (that is DHCP Proxy is enabled by default).

Example

```
DisableDHCPProxy
```

DisableDragAndDrop

Name

DisableDragAndDrop — Disables the drag-and.drop feature of the web interface

Synopsis

DisableDragAndDrop

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description

Add this option to the configuration file if you want to disable the drag-and-drop feature of the web interface. This can be useful in case of severe incompatibility with unsupported browsers. Note that the feature in known to work on most common browsers.

Not included in the default configuration.

Example

```
DisableDragAndDrop
```

DisableFILE

Name

DisableFILE — Disables FILE components of the OS deployment server

Synopsis

DisableFILE

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: File access**
- *rembo.conf:* Beginning of the file

Description

Add this option to the configuration file if you want to disable the FILE component of the OS deployment server. This can be useful if you are accessing the server from TCP targets.

Note: Do not set this option if targets are booting on this OS deployment server. If you set `DisableFile`, you must also set `DisableTCP`

Not included in the default configuration.

Example

```
DisableFILE
```

DisableHTTP

Name

DisableHTTP — Disables the console

Synopsis

DisableHTTP

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description

Add this option to the configuration file if you want to disable the HTTP component of the OS deployment server, that is the web interface. Disabling HTTP is not recommended as it forces you to run the server in command-line only. If you set `DisableHTTP`, you must run the `-c` command line option to change server parameters.

Note: A full server restart is necessary when this parameter is modified.

Not included in the default configuration.

Example

```
DisableHTTP
```

DisableHTTPOptim

Name

DisableHTTPOptim — Disables the web interface

Synopsis

DisableHTTPOptim

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: deployment servers**
- *rembo.conf:* Beginning of the file

Description

Add this option to the configuration file if you want to prevent the web interface to group JavaScript, stylesheet, and image files into large files to reduce HTTP traffic.

Note: If you are creating console pages, you must set this parameter to disable optimization.

Not included in the default configuration.

Example

```
DisableHTTPOptim
```

DisableJSDropDown

Name

DisableJSDropDown — Disables JavaScript menus in the web interface

Synopsis

DisableJSDropDown

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description

Add this option to the configuration file if you want to disable enhanced menus in the web interface, and use standard drop-down selectors instead. Note that standard drop-down selectors do not respect DHMTL layers under Internet Explorer, resulting in strange effects.

Not included in the default configuration.

Example

```
DisableJSDropDown
```

DisableNBP

Name

DisableNBP — Disables NBP services

Synopsis

DisableNBP

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Network boot module**
- *rembo.conf:* Beginning of the file

Description

Including this option in the configuration file causes the NBP component of the OS deployment server to stop answering requests coming on the NBP port. Do not set this option if your server is used by targets.

Not included in the default configuration.

Example

```
DisableNBP
```

DisableSSL

DisableSSL — Disables SSL encryption

Synopsis
DisableSSL

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description
Add this option to the configuration file if you want to disable the SSL encryption and use unencrypted HTTP connections. This can be useful if you want a faster download time, but it is not as safe.

Note: A full server restart is necessary when this parameter is modified.

Not included in the default configuration.

Example
```
DisableSSL
```

DisableTCP

Name
DisableTCP — Disables TCP services

Synopsis
DisableTCP

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: File access**
- *rembo.conf:* Beginning of the file

Description
This option disables the TCP component of the OS deployment server. You can include this option with caution if you are not using server replication

Note: Do not set this option if targets are booting on this server. If you set `DisableTCP`, you must also set `DisableFile`.

Not included in the default configuration.

Example
```
DisableTCP
```

FileMCASTAddress

Name
FileMCASTAddress — Multicast address used for the MCAST protocol

Synopsis
FileMCASTAddress `ip-addr:port`

FileMCASTAddrCount `number`

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: Not available**
- *rembo.conf:* Beginning of the file

Description

`FileMCASTAddress` defines the destination multicast address and destination port used by the OS deployment server when sending multicast UDP datagram to targets requesting a file with the MCAST protocol. The default value is `239.2.0.1:10000`.

`FileMCASTAddrCount` defines the upper limit of the multicast address range used by the server (that is the number of different multicast addresses used). When combined with `FileMCASTAddress`, this parameter can control the lower and upper limits of the range of multicast addresses used by the server. The default value is `256` (that is use no more than 256 multicast addresses).

The MCAST protocol is used by targets to download large files from the server. This protocol is a proprietary protocol made using multicast UDP.

Example

To force the OS deployment server to use 1000 multicast addresses starting at `232.1.1.1`, use:

```
FileMCASTAddress 232.1.1.1
FileMCASTAddrCount 1000
```

FileMCASTEncrypt

Name

FileMCASTEncrypt — Encrypts MCAST datagrams

Synopsis

FileMCASTEncrypt

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Not available**
- *rembo.conf:* Beginning of the file

Description

If this parameter is present in a rembo.conf configuration file, or set to `yes` in the web interface, MCAST datagrams exchanged between the OS deployment server and the remote-boot targets are encrypted.

The MCAST protocol is used by targets to download large files from the server. This protocol is a proprietary protocol made using multicast UDP.

MCAST datagrams are not encrypted by default.

Example

```
FileMCASTEncrypt
```

FileServerPort

Name

FileServerPort — Port used on the server for NETfs and MCAST requests

Synopsis

FileServerPort `port`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: File access module**

- *rembo.conf:* Beginning of the file

Description

This value is used by the OS deployment server as the port for all file-related requests sent by targets. File-related requests include NETfs requests and MCAST requests.

The default value is `4013`. The value of this parameter is sent to targets during the NBP phase of the boot process. If you change this parameter, targets automatically use the new value on next boot.

Example

```
FileServerPort 5013
```

HTTPAdminName

Name

HTTPAdminName — User name of the main administrator of the OS deployment server

Synopsis

HTTPAdminName " `username` "

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: HTTP Console Security: Base Parameters**
- *rembo.conf:* Beginning of file

Description

This is the superuser name intended to be used only by the main OS deployment server administrator, in order to get access to the all configuration parameters of the server. There is only one superuser login.

Example

```
HTTPAdminName "Admin"
```

HTTPRole

Name

HTTPRole — Security roles with access to the web interface console

Synopsis

HTTPRole `"RoleName"{ Members "membername" [,...] Allowpages "pagename" [,...] Allowgroups "groupname" [,...] Policies "policyname" [,...] }`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: HTTP Console Security: Security Role list**
- *rembo.conf:* After global parameters

Description

An HTTP role allows specific members to access predefined pages on the web interface, as well as preselected administrative groups. One can also define security policies. Parameters are a single string or a list of strings separated by commas. The star `*` means *all*, without restriction.

Role members can be either individuals or groups defined by the authentication authority. An individual, who does not belong to any role, trying to log on to the web interface is denied access to the

console. An individual, belonging to several roles, cumulates the page access rights and administrative group access rights of all the roles s/he belongs to. Security policies are also cumulative, resulting in a more restrictive policy when an individual belongs to several roles.

The security policies include:

- *CONF_RO* to deny changes to the server configuration
- *HOST_RO* to deny addition and removal of targets

To use `HTTPRole`, you must first have set a local authentication domain named `HTTP` with `( )AuthLocalDomain`.

Example

```
HTTPRole "RestrictedAccess" {
    Members "rembo"
    Allowpages "*"
    AllowGroups "local1", "local2"
    Policies "HOST_RO"
}
```

HTTPServerPort

Name

HTTPServerPort — TCP port for HTTP requests

Synopsis

HTTPServerPort `port`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description

This is the TCP port used by the web interface when listening for unencrypted HTTP requests. You can set this parameter to `8080` if you want the web interface to be accessible by typing the server host name in your Web browser. The OS deployment server always listens on this port even if connections are encrypted, in order to redirect unencrypted connections to the encrypted TCP port.

Note: A full server restart is necessary when this parameter is modified.

Example

```
HTTPServerPort 8080
```

HTTPSServerPort

Name

HTTPSServerPort — TCP port for encrypted HTTP requests (SSL)

Synopsis

HTTPSServerPort `port`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description

This is the TCP port used by the web interface when listening for

encrypted HTTP requests. You can set this parameter to 443 if you want the web interface to be accessible by typing the server host name in your Web browser, prefixed with the https URL syntax.

Note: A full server restart is necessary when this parameter is modified.

Example

```
HTTPSServerPort 443
```

HTTPSessionTimeout

Name

HTTPSessionTimeout — Time in minutes before a web interface session expires.

Synopsis

HTTPSessionTimeout minutes

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: web interface**
- *rembo.conf:* Beginning of the file

Description

This is the inactivity timeout (in minutes) before web interface users are automatically logged out. A typical safe value is 5 minutes, but you might want to make it longer if you receive the message Session timed out too often and are sure that you will never forget to log out when you leave your computer.

Example

```
HTTPSessionTimeout 30
```

Interfaces

Name

Interfaces — List of network interfaces used by the OS deployment server

Synopsis

Interfaces ip-addr1 [ip-addr2] [ip-addr3...]

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Base parameters**
- *rembo.conf:* Beginning of the file

Description

You will find this option very useful if you are running on a multihomed computer (that is a target with more than one network card, or with a network card and a dial-up adapter).

This option lets you specify the list of network interfaces used by the OS deployment server when receiving and sending packets to targets. If you leave this option unset, the server uses the network interface corresponding to the official host name of the server.

You must specify a list of IP addresses to use. Each IP address must correspond to the IP address to one of the network interfaces of the OS deployment server. The list must contain at least one address.

Note: You are strongly encouraged to set this option if your computer is multihomed. Otherwise the server might not receive the network packets not originating from its official interface.

Examples

```
Interfaces 192.168.1.1
Interfaces 192.168.1.1 192.168.10.1 10.1.1.1
```

MaxLogSize

Name

MaxLogSize — Maximum size of log files created by the OS deployment server

Synopsis

MaxLogSize `size_in_bytes`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Base Parameters**
- *rembo.conf:* Beginning of the file

Description

This parameter can be used to limit the size of the log file generated by the OS deployment server. The maximal log size must be specified in bytes, and apply to all the log files created by the server (file, nbp, http, tcp, VM, and boot). If you do not specify this parameter, or set the limit to `0`, then log files are not limited in size.

Example

To limit the size of logs to 10MB:

```
MaxLogSize 10000000
```

To disable log size limit:

```
MaxLogSize 0
```

MaxPCASTSessions

Name

MaxPCASTSessions — Maximum number of simultaneous PCAST sessions

Synopsis

MaxPCASTSessions `number`

Location in the server configuration

- *web interface* > **:Not available**
- *rembo.conf:* Beginning of the file

Description

This is the maximum number of PCAST sessions that the OS deployment server accepts simultaneously. This parameter is mostly relevant if the server is running in UNICAST mode and deploying a group with UNICAST setting.

The default value is 8. Because each session uses around 16MB of server memory, higher values must be avoided on servers with no more than 256Mb.

Example

```
MaxPCASTSessions 8
```

MaxTFTPSegSize

Name
MaxTFTPSegSize — Maximum size of a TFTP segment

Synopsis
MaxTFTPSegSize `number`

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description
This is the maximum size of TFTP segment in bytes. This parameter can be used to reduce the size of packets sent by the OS deployment server if required by the underlying network. This may be useful for instance when network traffic is encrypted by routers.

The default value is `512`.

Example

```
MaxTFTPSegSize 256
```

MTFTPClients

Name
MTFTPClients — The destination address and port used by the MTFTP server

Synopsis
MTFTPClients `ip-addr` [: `port`]

Location in the server configuration
- *web interface* > **Server parameters > Server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description
This is the destination IP address and port used by the server when sending multicast MTFTP datagrams to boot agents. The IP address must be a valid multicast address and must match the address sent to the target during the DHCP phase. If the deployment engine has received its PXE parameters from the OS deployment server during the DHCP phase, this address is automatically included so that the deployment engine always uses the same address/port as the server.

The default value is `232.1.0.1:8500`.

Example

```
MTFTPClients 232.8.7.6
```

MTFTPPort

Name
MTFTPPort — The port used by the MTFTP server

Synopsis
MTFTPPort `port`

Location in the server configuration
- **web interface > server configuration: Boot module**
- *rembo.conf:* Beginning of the file

Description

This is the port used by the server when listening to MTFTP requests. If the remote-boot targets are using the OS deployment server to get their PXE parameters, this value is sent in the DHCP/PXE reply so that the remote-boot target always uses the correct port.

The default value is `4015`.

Example

```
MTFTPPort 5015
```

MTFTPStartDelay

Name

MTFTPStartDelay — The initial delay on target computers before sending first MTFTP request

Synopsis

MTFTPStartDelay `secs`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Boot Module**
- *rembo.conf:* Beginning of the file

Description

This is the delay used by remote-boot targets before sending the first MTFTP request to the OS deployment server. This delay is used by the target to listen to an existing MTFTP transfer. If the MTFTP file they are about to request is already being sent by the server on the multicast address, then the target does not request the file and listens to the packets. The longer this delay is, the more probability you have that many targets reuse the same MTFTP channel instead of requesting the file. But if you want to have a fast boot process, you must set this value to 1 (the minimum value).

The default value is `2` (seconds).

Example

```
MTFTPStartDelay 1
```

NBPServerPort

Name

NBPServerPort — Port used on the OS deployment server for NBP requests

Synopsis

NBPServerPort `port`

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Network Boot Module**
- *rembo.conf:* Beginning of the file

Description

This value is used by the OS deployment server as the port for all NBP requests sent by remote-boot targets. NBP is a protocol used by target computers to get their startup parameters (startup page, groupname,...) and to send authentication requests to the server.

The default value is 4012. The value of this parameter is sent to targets as part of the last MTFTP packet when the server sends the initial part to the PXE bootrom. If you change this parameter, targets automatically use the new value on next boot.

Example

```
NBPServerPort 5012
```

NetPassword

Name

NetPassword — Network password for remote file access

Synopsis

NetPassword " password "

Location in the server configuration

- *web interface* > **Server Parameters > HTTP Console Security: Base parameters**
- *rembo.conf:* Beginning of the file

Description

This is the password used by the OS deployment server to accept or deny network requests coming from a target, the web interface or NetClnt.

You must change this option when you install a new OS deployment server and select a password to protect your files against unpermitted access. If you are using the Windows version of the server, you are asked to enter the password during the setup.

If you are using the web interface, or NetClnt to access the files stored in your OS deployment server, the password you enter in NetClnt (or the web interface) must match the word set in NetPassword.

Note: This password is an important piece of your security. If you require optimal security, you are strongly encouraged to protect the configuration of the server. By default, this file is created by the installer and the password is stored encrypted in it.

Example

```
NetPassword "mypassword"
```

NetworkShare

Name

NetworkShare — UNC path of the shared partition directory of the OS deployment server.

Synopsis

NetworkShare "path "

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Network share module**
- *rembo.conf:* Beginning of the file

Description

This is the path of the shared partition directory of the OS deployment server. This parameter can be used to increase the deployment speed of some operating systems. The target computer access the network share

directly to retrieve the necessary installation files. You need to have set the partition as read-only for the user entitled to access the network share.

Example

```
NetworkShare "provisioningServer/partition"
```

NetworkUser

Name

NetworkUser — name of a user entitled to access the network share.

Synopsis

NetworkUser "name "

Location in the server configuration

- *web interface* > **Server parameters > Server configuration: Network share module**
- *rembo.conf:* Beginning of the file

Description

The name of the user entitled to access the network share, which is given in `NetworkShare`. If the user is part of a domains, you must use the syntax `Domain/User`. This user name is used by the targetcomputer to access the network share on the OS deployment server.

Example

```
NetworkUser "ClientComputer"
```

NetworkPasswd

Name

NetworkPasswd — Network password used by the target computer to access the network share

Synopsis

NetworkPasswd "password "

Location in the server configuration

- *web interface* > **> Server parameters > Server configuration: Network share module**
- *rembo.conf:* Beginning of the file

Description

This is the password used by the target computer to access the network share located on the server, using the `NetworkUser` user name. The password is stored in the `rembo.conf` file.

Example

```
NetworkPasswd "clientPassword"
```

TCPServerPort

Name

TCPServerPort — Defines the TCP port used

Synopsis

TCPServerPort `port`

Location in the server configuration

- *web interface* > **Server Parameters > HTTP Console Security: File access Module**

- *rembo.conf:* Beginning of the file

Description

This option defines the TCP port used by the TCP component of the OS deployment server when listening to requests coming from other servers.

The default value is `4013`

Example

```
TCPServerPort 2048
```

Authentication domains

The following information details the authentication domain references.

AuthLocalDomain

AuthLocalDomain

Name

AuthLocalDomain — Local authentication domain

Synopsis

AuthLocalDomain

AuthLocalDomain { UserGroup "group" }

Location in the OS configuration

- *web interface:* `Predefined channels/Authetication domain`
- *rembo.conf:* After global parameters

Description

A local domain uses the server local user database.

On UNIX servers, if PAM is correctly configured, a local domain uses PAM to authenticate users against any PAM-supported server (including LDAP, Smb, and so on).

Attention: Do not forget to configure PAM if you wan to use it.

Example

```
AuthLocalDomain {
    UserGroup "rembousers"
}
```

AuthNTDomain

AuthLocalDomain

Name

AuthLocalDomain — Remote authentication domain

Synopsis

AuthNTDomain { AuthServerName "hostname" UserGroup "group" }

Location in the OS configuration

- *web interface:* `Predefined channels/Authetication domain`
- *rembo.conf:* After global parameters

Description

A Remote NT domain (supported on Windows platform only) gets users from a given Windows server, for example a domain PDC.

Example

```
AuthNTDomain {
    AuthServerName "domain-pdc"
    UserGroup "rembousers"
}
```

AuthRadiusDomain

AuthLocalDomain

Name
AuthRadiusDomain — Radius authentication domain

Synopsis
AuthRadiusDomain { AuthServerAddress ip-addr RadiusSecret "secret" }

Location in the OS configuration

- *web interface:* `Predefined channels/Authetication domain`
- *rembo.conf:* After global parameters

Description
A Remote Radius domain gets the user list from a Radius compatible server.

Example

```
AuthRadiusDomain {
    AuthServerAddress 192.168.1.15
    RadiusSecret "testing123"
}
```

TCP tunnels

The following information details the TCP Tunnel references.

Remotetarget

Name Remotetarget — Host name or IP address of remote TCP target

Synopsis
Remotetarget "target"

Location in the OS configuration

- *web interface:* `Predefined channels/TCP tunnels`
- *rembo.conf:* After global parameters

Description
`RemoteHost` is a string representing the TCP remote targetto contact when a target opens this TCP tunnel. You can either specify a host name or an IP address for this parameter. Note that you need to use double quotation marks(") around the host name or IP address if you are using `rembo.conf`.

Examples

```
TCPTunnel finger {
    RemoteHost "finger.company.com"
    RemotePort 79
}

TCPTunnel ODBC {
    RemoteHost "127.0.0.1"
    RemotePort 2020
}
```

RemotePort

Name RemotePort — Numeric TCP port of remote connection

Synopsis

RemotePort port

Location in the OS configuration

- *web interface:* `Predefined channels/TCP tunnels`
- *rembo.conf:* After global parameters

Description

`RemotePort` is a number representing the TCP port to connect to when a target opens this TCP tunnel. You can get the complete list of valid port numbers in the file `/etc/services` on any UNIX computer, or in the file `/system32/drivers/etc/services` on a Windows computer.

Examples

```
TCPTunnel finger {
    RemoteHost "finger.company.com"
    RemotePort 79
}

TCPTunnel ODBC {
    RemoteHost "127.0.0.1"
    RemotePort 2020
}
```

OS deployment object parameters

This section contains information on the parameters of various OS deployment objects.

Target parameters and details

To view information on target parameters, go to **Server** → **OS deployment** → **Target Monitor**. Double-click on a target to view its details.

General target info

This section contains information used to identify the target, such as the serial number, the UUID and the MAC address (also called the NIC hardware address). It also contains the host name and a textual description for the computer and its hardware model name (only valid after a hardware inventory, during deployment). You can enter the name of your organization and an administrative password. You can also configure the time zone used by the operating system, and the locale. You can also decide to enable the IPv6 protocol.

Note: Linux Windows Setting **Enable IPv6 support** to false for Windows or Linux operating system disables the IPv6 kernel module even if, for some administrative tools, it might still appear enabled.

Common networking info and Advanced network settings

This section contains information used to configure the networking part of the operating system. **Common networking info** contains information that is compatible with earlier versions. **Advanced network settings** allows you to specify several interfaces for your target. A link in this section allows you to switch back and forth between the two modes.

Note:

- When switching back to basic IP settings mode, all manually created interfaces are lost.
- If your target has more than one network adapters, use **Advanced network settings**. Using **Common networking info** can result in incorrect values at deployment time.
- **Common networking info**
  - In the **TCP/IP** settings section, you can set the `TCP/IP mode` of the target being deployed. The default value for TCP/IP settings is **Undefined, use configuration parameters**. TCP/IP parameters can be set to use DHCP if DHCP is supported on the end-user's network. If the target has to be configured with a fixed IP address, **TCP/IP settings** has to be set to either **Static IP, import from DHCP server** or **Static, manual**. In the last case, you need to switch to **Advanced IP settings mode** to specify the value of the static IP address.

    Note: Option **Undefined, use configuration parameters** is incompatible with option **IP, Netmask and Gateway defined in target** in the OS configuration.

    Note: It is not possible to import DHCP settings on PowerPC and SPARC targets. Therefore, you must not use **Static, import from DHCP server** on these types of targets.
  - Default gateway, DNS servers 1 to 3, and DNS domain are some of the advanced TCP/IP parameters which can be entered manually if the TCP/IP mode above is set to manual.

    Note: The value of **DNS domain** is not used in Windows Vista/2008/7 deployments.
  - The **Domain Suffix Search Order** defines the order in which domains are searched for target information.
  - The last two fields are the WINS servers used for resolving a name.

  Note:
  - Any value provided in the OS configuration overrides information entered at the target level.
  - Values provided in the fields following **TCP/IP settings** are overridden by DHCP if DHCP provides any value. However, values entered and visible in these fields are used if DHCP does not provide these values, independently of the value of **TCP/IP settings**.
- **Advanced network settings**

  In this mode, you get an additional link on each interface, enabling you to remove the current interface. Moreover, on the most recently created interface, you have a link allowing you to set yet another new interface.

  After clicking on **Edit**, you get three additional fields on top of the ones described for the **Common networking info** section.
  - NIC Device, which is retrieved by theOS deployment server.
  - The MAC address, provided when the interface was created.
  - A connection name, that you can freely select.

  Moreover, it is now possible to change the IP address and the subnet mask, and the other fields, if **TCP/IP settings** is set to **Static, manual.**.

  Note:

- Any value provided in the OS configuration overrides information entered at the target level.
- Values provided in the fields following **TCP/IP settings** are overridden by DHCP if DHCP provides any value. However, values entered and visible in these fields are used if DHCP does not provide these values, independently of the value of **TCP/IP settings**.

Windows-specific info

Contains Windows-specific information used in the deployment:

- **Windows product key** is used to set the product key of your operating system. If you have a special license with Microsoft, and you have one single product key for all your computers, it is more convenient to set the product key as a fixed value in the OS configuration you are deploying on these targets, rather than setting the product manually for each target.
- If the computer is part of a workgroup, the field network type must be set to **Stay in a workgroup**. If the computer is part of a Windows NT domain, the network type must be set to either **Join a Windows domain during deployment**, or **Join a Windows domain on-site**, that is the first time the target boots after the deployment has been completed successfully.
- In **Workgroup name**, you can enter the workgroup or domain name to join.
- The domain administrator name and password corresponds to the user name and password to use to join the computer to the Windows NT domain. This is not used if the computer is configured to run in a workgroup.
- You can also configure video parameters such as horizontal and vertical resolution, the color depth, and the vertical refresh rate in Hertz. Typical values are 1024, 768, 32 bpp, and 60 Hz. These fields are not mandatory.
- It is also possible to enter an administrator name for your target.

Note: If a fixed property field is left empty, the value currently on the target is not altered by deployment. This is of particular importance for the administrator name, as an empty administrator name field does not mean that the administrator name will be null on the deployed target. The value of the administrator name must be checked in case of user login trouble.

UNIX-specific info

Contains UNIX-specific information used in the deployment:

- You must give a name resolution method. Select the type of service to use to convert target names to IP addresses (and vice-versa).

  Note: Some name resolution services might not be supported on all Linux distributions.
- If you are using NIS/NIS+, enter the name and IP address of the NIS server, in the form `name(ipaddress)`. If you are using LDAP, enter the IP address of the LDAP profile server. For DNS, leave this field empty (DNS servers are specified in the standard IP information panel).
- If you are using LDAP, provide the name of the LDAP profile to use.
- Kerberos is a standard way of ensuring network security in a UNIX environment. To use Kerberos, you must first configure a Kerberos server, providing security tokens for its realm. Then you can enter Kerberos parameters to automatically join newly deployed computers to

your Kerberos realm. The Kerberos administration server is the central repository for managing principals (targets, services, and so on) in the Kerberos environment.

- You must provide one to three key distribution centers that provides authentication tokens when a secured operation has to be performed.
- If you want to replicate the clock of the deployed targets to a reference server using the NTP protocol, enter the server IP address or target name.
- For Solaris, you might configure the default terminal emulation to use (see `/usr/share/lib/terminfo` for possible values). The default is `sun-cmd`.

User details

You can configure the registered owner of your operating system (valid for Windows operating systems) in this section. Enter the full name and the organization for the registered owner of the operating system.

You can also enter a user login name and a user domain name if you have selected the following options for the OS configuration you are deploying on this target:

- `Add domain user to local admin group` (uses the user login name and domain name to build the domain user name)
- `Create a local account for the user` (uses the user login name to determine the local account name to create)

It is possible to enter the values of the site-specific database fields as well. These values are used to store site-specific data, such as the localization of a PC or the name of an application server. These values are then be used during deployment when working with `.ini` updates packages.

Boot settings

The boot settings section provides information about the last known PXE OS deployment servers, on the PXE boot mode, on boot engine options, and on human interface locking.

- **PXE boot mode** Consists of three sub-options:
  - `Use alternate PXE server`
  - `Boot on hard-disk`
  - `Boot on hard-disk if idle`

  Set `Use alternate PXE server` if you do not want the remote-boot target to boot on the OS deployment server but want to force it to search for another PXE server. Set `Boot on hard-disk` to force the target to boot on its hard-disk. Set `Boot on hard-disk if idle` to force the target to boot on its hard-disk only if the target is idle and there is no pending task. For redeployment, `Boot on hard-disk if idle` must not be set.
- **Boot engine options** Consists of ten self-explanatory sub-options:
  - `Reboot on unrecoverable errors`
  - `Disable USB`
  - `Disable Auto USB`
  - `Disable graphic interface`
  - `Disable APM`
  - `Disable multicast`
  - `Disable IGMP version 2`

- Disable Ultra-DMA
- Disable ATA-5 feature
- Disable enhanced disk access
- Disable enhanced PXE access
- Try to optimize IRQ

- **User interface locking options** Consists of three sub-options:
  - Disable mouse
  - Disable keyboard
  - Disable screen

  You can use this parameter to lock a specific peripheral during the time the deployment engine is active on the remote-boot target.
- **Boot redirection server** and **Alternative redirection server** lets you specify one or two servers to which PXE requests from the target are automatically redirected, without needing to modify DHCP options.

Hardware details

If the target has been deployed at least once, and if the deployment scheme used for the deployment was configured to collect hardware information, then the **Target details** page shows information about hardware installed in the computer.

CPU, Memory and Disks provides information about the processor, the amount of main memory (RAM) and the size of the hard-disks found.

Note: This information is also available on Windows hypervisors, but not on Linux ones.

PCI devices provides a list of all the PCI devices found on the computer.

You can remotely turn on and off your targets using the **Switch on** and **Switch off** buttons at the bottom of the **Target details** page. Every time a wizard launches a task, it asks you for switching on the target using the **Management Interface Parameter** identifier.

OS configuration parameters

The OS configuration parameters are divided in the web interface along several tabs.

General

This tab is further divided in to **OS deployment** and **General settings**.

The **OS deployment** section contains operating system information including an operating system version, its architecture, its language, the root of the system installation and boot parameters.

In the **OS deployment** section, you can also specify (for a multi-partition image) partitions that must be excluded from one-time deployment and redeployment. In the text fields, enter the number of the partitions you want to exclude, keeping in mind that partitions with numbers 1 to 4 are primary partitions, while those with number 5 or above are logical partitions. If you want to exclude more than one partition, separate the partition numbers with commas. For example, `2,5` excludes the second primary partition and the first logical partition.

Note:

1. This option is valid only for the first physical disk.
2. This option is valid only for partitions defined in the system profile. The partition scheme defined in the system profile is always written on the disk of the deployed target. Therefore, if your target has two partitions but your system profile only one, the deployment always creates the one partition of the system profile and deletes the second partition of your target.
3. Partitions can be preserved only when the target has the same partition scheme as the one defined in the system profile. The way partition sizes are defined in a system profile makes it unlikely for a target not previously deployed with this system profile to have the same partition scheme.
4. For unattended setup profiles, all partitions must have the option **Must be deployed** set to **yes** for a successful deployment.

Values in **Fixed target properties** can contain special keywords that are replaced by dynamic information. For example, [IP] is replaced by the full IP address of the target being deployed, while [MAC] is replaced by the hardware address, also known as Media Access Control (MAC) address. To set target names based on the MAC address, you can enter the following value in the **target name** field: `pc[MAC]`. The computer with the MAC address 00:01:02:03:04:05 is named pc000102030405.

The following keywords are supported:

- `[IP]`: full IP address (received by DHCP)
- `[MAC]`: hardware address
- `[SN]`: serial number as found in DMI (SMBIOS)
- `[BOMID]`: unique target identifier in the OS deployment server database
- `[AT]`: DMI asset tag
- `[GRP]`: deepest administrative group name to which the target belongs
- `[DHCPNAME]`: target name as known to the DHCP server

Every keyword supports a *range* extension if you want to include only part of the dynamic information. The range starts at value `0`. [IP3] corresponds to the last byte of the IP address (in IP addresses, bytes are separated by dots. `pc-[IP3]` becomes `pc-12` if IP address is 192.168.0.12). [IP1-3] corresponds to bytes 1 to 3. [MAC3-5] is replaced by the last three bytes of the MAC address (MAC addresses are typically represented in hexadecimal, with colons to separate the bytes). For `AT`, `GRP`, and `DHCPNAME`, the range corresponds to a substring.

If you need more flexibility, and you want to use a program to change the values of the target properties, you must access the ODBC/JDBC database directly. The BOM table contains almost all of the items that are used by a target when filling the Sysprep answer file. If your program knows how to access an ODBC/JDBC database, then you can change everything without using the web interface.

Disks

You can manage partition schemes: you can change partition size, you can add new partitions, you can change mount points.

Targets

Note: In the **Fixed target properties** section, you can specify fixed parameters common to all targets using this OS configuration, to avoid specifying them for

each target individually. You can set the TCP/IP mode of the target being deployed. The value you can select are the following:

IP, Netmask and Gateway defined in target
The deployment uses the values of the IP, netmask, and gateway defined in the target page.

Note: This is not compatible with option **Undefined, use configuration parameters** in the **Target details** page.

Import all parameters from DHCP server
All the parameters are taken from the DHCP and set as fixed values.

Force all parameters as dynamic (DHCP)
All the parameters are set dynamically by the target requests to the DHCP server.

If a DHCP server is available on the end-user's network, you can set TCP/IP settings to use DHCP.

Note:

- Any value provided in the OS configuration overrides information entered at the target level.
- Values provided in the fields following **TCP/IP settings** are overridden by DHCP if DHCP provides any value. However, values entered and visible in these fields are used if DHCP does not provide these values, independently of the value of **TCP/IP settings**.

Note: The value of **DNS domain** is not used in Windows Vista/2008/7 deployments.

Users

In the **Fixed user properties** section, you can enter the organization name once and for all. Other required fields are a name, an organization, the time zone and the language.

Bindings

The **OS configuration binding rules** summarizes all the bindings linked to the current OS configuration

Operating system

The name of this section varies depending on the operating system of the OS configuration.

Windows **Windows**

Windows-specific info

Vista 2008 Windows 7 If you have a Volume License, set **Volume licensing** to **Yes** to avoid a product key entered on the target or on the OS configuration to be taken into account.

Note: Tivoli Provisioning Manager for OS Deployment supports the Key Management Service (KMS) key only. If you have a

Multiple Activation Key, select **Volume licensing, no product key required** on the product key screen of the wizard.

Windows You can enter a Windows product key if you have an Open License or a similar site license that uses the same product key for all targets. You can copy the key in its xxxxx-xxxxx-xxxxx-xxxxx-xxxxx format and paste it directly in the product key fields by first pressing and holding down Ctrl and then pressing V.

Note: This is not a regular Ctrl+V, therefore make sure to first press Ctrl and then V.

System customization
Two options are mutually exclusive: **Add domain user to the local admin group** and **Create a local user account**.

Add domain user to the local admin group
If set, the user is added as an administrator.

Create a local user account
If set, the user is added in a local account.

If either option is set, add the administrator name as an administrator of the domain.

The option **Force user to put a new administrator password** works only in conjunction with the option **Create a local user account**. It forces the user to change the password of the local user.

UNIX **UNIX**
Net boot device indicates the network card to be used when deploying PowerPC with multiple interfaces. You must set it to the network interface that was used when registering the target in theOS deployment server. If this value is incorrect for a given target, the installation switches to interactive mode.

Hardware compatibility list

The list of certified compatible hardware for this product is provided as a Tivoli Technote, located at http://www-01.ibm.com/support/docview.wss?rs=203&q1=TPM+for+OS+Deployment+7.1+hardware+compatibility+list&uid=swg21321438&loc=en_US&cs=utf-8&lang=en.

Java API

The Java API is a set of classes that map to readable and editable data structures in the server (OS deployment server).

Java classes are provided to interface directly with all objects and settings. Previously, Rembo-C scripts were used for these tasks. You can still make Rembo-C calls through the Java API but you can now write applications that use the technology without writing a single script in Rembo-C code.

Capabilities of the Java API

At a high level, the Java API offers the following capabilities:

- Booting targets with Wake-on-LAN technology within tasks

- Scheduling tasks
- Creating tasks. Tasks available with the Java API include:
  - Deployment of targets
  - Operating system restoration on targets
  - Image capture of targets (cloning)
  - Booting a preboot image on a targets: DOS, LinPE and WinPE are supported
  - Running customized Rembo-C scripts on a PXE booted computer, a web interface extension, or the OS deployment server itself
  - Object replication between OS deployment servers
  - Configuration of server parameters
  - Creation of software modules
  - Creation of unattended system profiles from the installation media
  - Creation of bootable CDs, which can be used to start the deployment engine on a target without using PXE
  - Exporting objects to RAD files (RAD export)
  - Importing RAD files (RAD import)
  - Blanking of hard disks on targets
  - Conversion of existing Rembo Toolkit images into system profiles
- Creation of *open* tasks. Every target (and every newly added target) runs an open task once and only once
- Monitoring tasks
- Cancelling tasks
- Managing server configuration
- Managing targets, such as adding targets, removing targets or changing target properties
- Managing system profiles
- Managing OS configurations
- Managing software modules
- Managing deployment template

Getting started with Java API

The `javaapi.zip` file contains the `rbapi.jar` file required for the Java API, a set of Java class examples that you can use, and Javadoc for the entire API.

When you installed the product a file called `javaapi.zip` was copied. This file contains the `rbapi.jar` file necessary to use the Java API. The `javaapi.zip` file is located in the top level folder in the directory. By default on OS deployment servers with a Windows operating system, this is `c:\Program Files\IBM\TPMfOSd`.

The communication protocol between the Java API and the OS deployment server is HTTP and XML. The Java API generates XML code which is passed to the OS deployment server over HTTP. This communication handles all the function calls from the Java API to the OS deployment server.

Configuring the OS deployment server to use the Java API

To work with the Java API, some configuration steps are required for the product.

1. By default, there is no password set up to access the OS deployment server through the Java API. When no password is set up, the OS deployment server accepts connections from trusted computers only. Therefore, you must either set

up your own password to connect to the OS deployment server through the Java API, or indicate a list of trusted computers.

- To set up your own password to connect to the server through the Java API, complete the following steps:
  a. Open a command prompt in the same directory that contains the `rbagent.exe` executable file. On Windows operating systems, the path is generally `C:\Program Files\Common Files\IBM Tivoli\rbagent.exe`.
  b. Run the `rbagent` command to encrypt your chosen password:

     ```
     rbagent.exe -d -s <ip_of_tpmfosd_server>:<tpmfosd_web_password>
     rad-hidepassword <your_new_password>
     ```

     Your new encrypted password is generated and can be found in the `Result` string. For example, an encrypted password can be `A846025095B4AA763231579210233951`.
  c. Create a file called `config.csv` that looks like this:

     ```
     HostName;APISecret
     <your_hostname>;<your_APISecret>
     ```

     where <*your_hostname*> is your Tivoli Provisioning Manager for OS Deployment host name and where <*your_APISecret*> is your encrypted password.

     This is an example of a `.csv` file:

     ```
     HostName;APISecret
     testhost;A846025095B4AA763231579210233951
     ```

     Note:

     1) <*your_hostname*> must be either a short host name or a fully qualified host name, it cannot be an IP address
     2) <*your_hostname*> must be written in lowercase only
     3) <*your_hostname*> must be immediately followed by the semicolon (`;`) without any intervening space
     4) Inside `config.csv`, the APISecret can either be in plain or encrypted format. However, do not use the hidden form on the Java side, only <*your_new_password*> of step 2.b.
  d. Copy the `config.csv` file to the`files/global/rad` directory. The full path of the directory is generally `C:\TPMfOS Files\global\rad`.
- To create a list of trusted computers, complete the following steps:
  a. Create a file called `config.csv` that looks like this:

     ```
     HostName;APITrusted
     <your_hostname>;<list_of_trusted_ip>
     ```

     where <*your_hostname*> is your product host name (<*your_hostname*> must be either a short host name or a fully qualified host name, it cannot be an IP address) and where <*list_of_trusted_ip*> is the list of IP addresses of the trusted computers. Trusted IP addresses must be given in the dotted notation (for example, 192.168.168.16) and separated by one space character.

     This is an example of a `.csv` file:

     ```
     HostName;APITrusted
     testhost;198.162.32.04 198.162.35.23 198.162.12.45
     ```
  b. Copy the `config.csv` file to the`files/global/rad` directory. The full path of the directory is generally `C:\TPMfOS Files\global\rad`.

2. Stop and start your OS deployment server.

The product is now configured to work with the Java API.

Note: There is a known problem when using HTTPS and the IBM JRE 1.5. Other versions of the IBM JRE work, as well as JRE 1.5 from other brands. If you are using IBM JRE 1.5, follow these steps to ensure the Java API works with HTTPS enabled:

1. Check if there is a `ibmjsseprovider2.jar` under `%JAVA_HOME%/jre/lib/ext`.
2. If it exists, edit the file `java.security` under `%JAVA_HOME%/jre/lib/security`.
3. In the section headed by *security.provider*, replace `security.provider.x=com.ibm.jsse.IBMJSSEProvider with security.provider.x=com.ibm.jsse2.IBMJSSEProvider2`, wherex is he numeric value indicating the order of the providers being called.
4. Save the `java.security` file and try again to use the Java API with HTTPS enabled.

Examples

The `javaapi.zip` compressed file contains many ready to use Java code examples for all kinds of tasks in the `examples` sub-folder.

Compiling and running examples

The example code in `javaapi.zip` can be compiled with any Java 1.5 or higher Java Developer's Kit (JDK).

To compile the Java code (on a Windows operating system, with `apidir` the directory where `rbapi.jar` is extracted), type the following commands :

```
c:\> cd apidir
c:\apidir> javac -classpath .;rbapi.jar examples\*.java
```

Where apidir is the directory where rbapi.jar is extracted

To run an example class, in this instance `TestRADImport`, with `10.10.10.10` as server IP address, on the default port (`8080`), with an API secret of `mypassword`, type:

```
c:\> cd apidir
c:\apidir> java -cp rbapi.jar;. examples.TestRADImport 10.10.10.10 8080 mypassword
```

Understanding the sequence of procedure calls

Understanding the sequence of procedure calls between the Java client application, its objects, the OS deployment server, the SQL database, and the web interface extension (rbagent) is necessary to use the Java API adequately.

`TestRADImport.java`, which imports a RAD file to the OS deployment server through the Java API, is a good example to detail the process. To get the most of this example, follow the source code of the example, provided in `javaapi.zip`, while you view the explanatory diagrams and read their legend.

RAD file import is a process in four sequential top-level steps. For each top-level step, there is a diagram containing items such as the Java client application, Java objects, and the OS deployment server, among others. These items are linked by lettered arrows representing procedure calls. Follow the arrows in alphabetic order. You can discover the meaning of each arrow in the list which follows each diagram.

1. Asking the OS deployment server about known targets

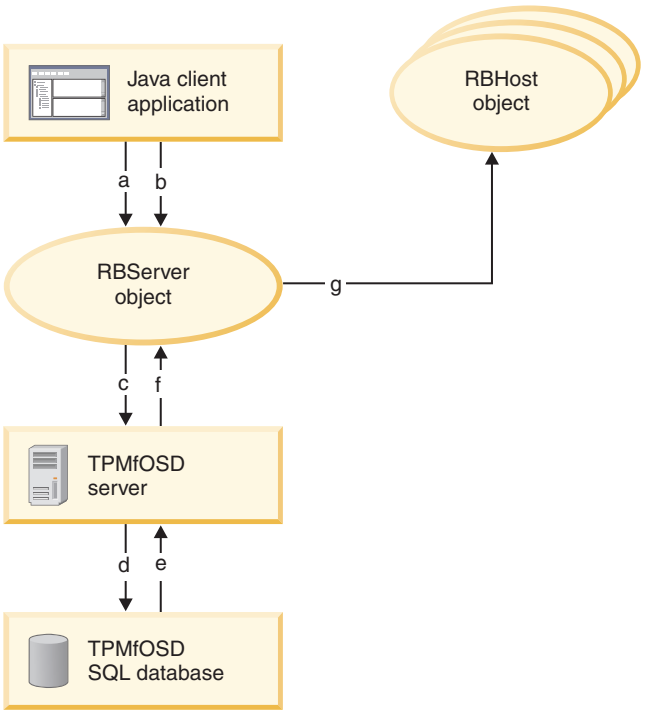


Figure 7. Step 1

a. The Java client application creates a new `RBServer` object, the OS deployment server is referenced by IP address and port
b. The Java client application calls the `RBServer.getHosts()` method
c. `RBServer` sends a remote procedure call to the OS deployment server (XML over HTTP)
d. The OS deployment server sends an SQL query to the database to get the list of targets-->
e. The database returns an array of targets is returned -->
f. The OS deployment server encodes the result in XML
g. The `RBServer` object creates one `RBHost` object for each targetID returned by the query

2. Checking the status of the web interface extension on the chosen target

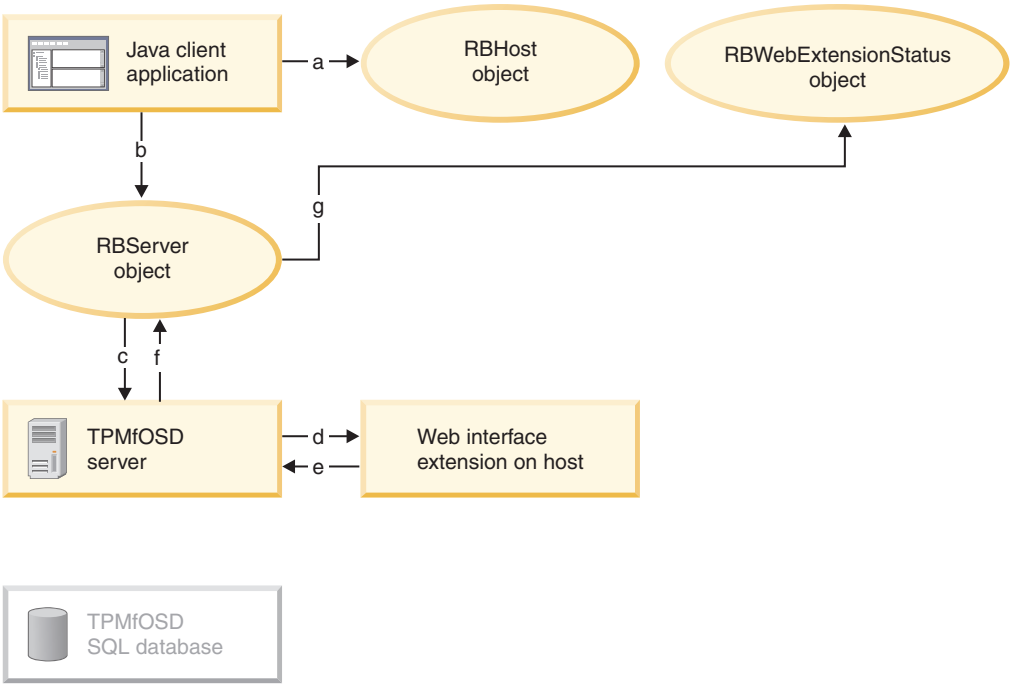


Figure 8. Step 2

a. The Java client application calls `RBHost.getIPAddress()` to lookup attributes of the target
b. The Java client application calls `RBServer.isWebExtensionReady(IP)` to check the status of the web interface extension (rbagent) on the target selected by the user
c. `RBServer` sends a remote procedure call to the OS deployment server
d. The OS deployment server sends a *probe* packet to the web interface extension
e. The web interface extension returns its status to the OS deployment server
f. The OS deployment server encodes the result in XML
g. The Java client application gets the status of the web interface extension in a new `RBWebExtensionStatus` object

3. Performing the RAD import task

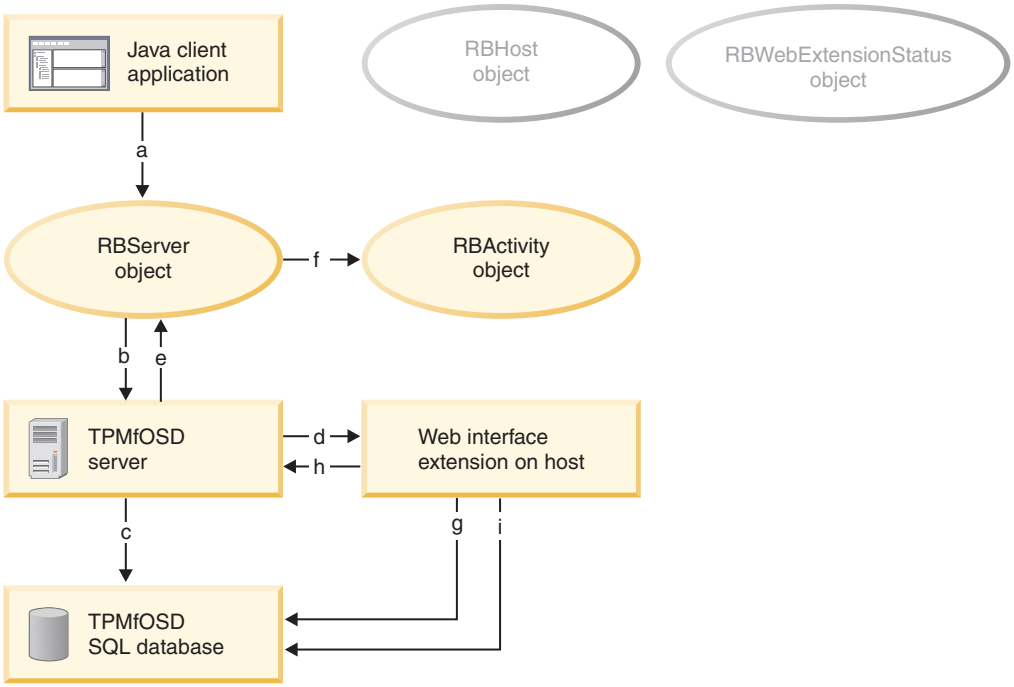


Figure 9. Step3

a. The Java client application calls `RBServer.registerRADImportActivity` with the name of the RAD file and the target `RBHost`
b. The `RBServer` object sends a remote procedure call to the OS deployment server
c. The SQL database records a new task with its type (RAD import) and parameters (file name, target, schedule, ...)
d. The OS deployment server sends a signal to notify the target that there is something to do
e. The OS deployment server sends, in XML, a reference to the new task to `RBServer`
f. The target receives a new `RBActivity` object
g. The web interface extension finds the new task to perform in the SQL database
h. The web interface extension uploads the RAD file content on the OS deployment server
i. The web interface extension reports the progress and the completion of the task to the SQL database

4. Monitoring task progress

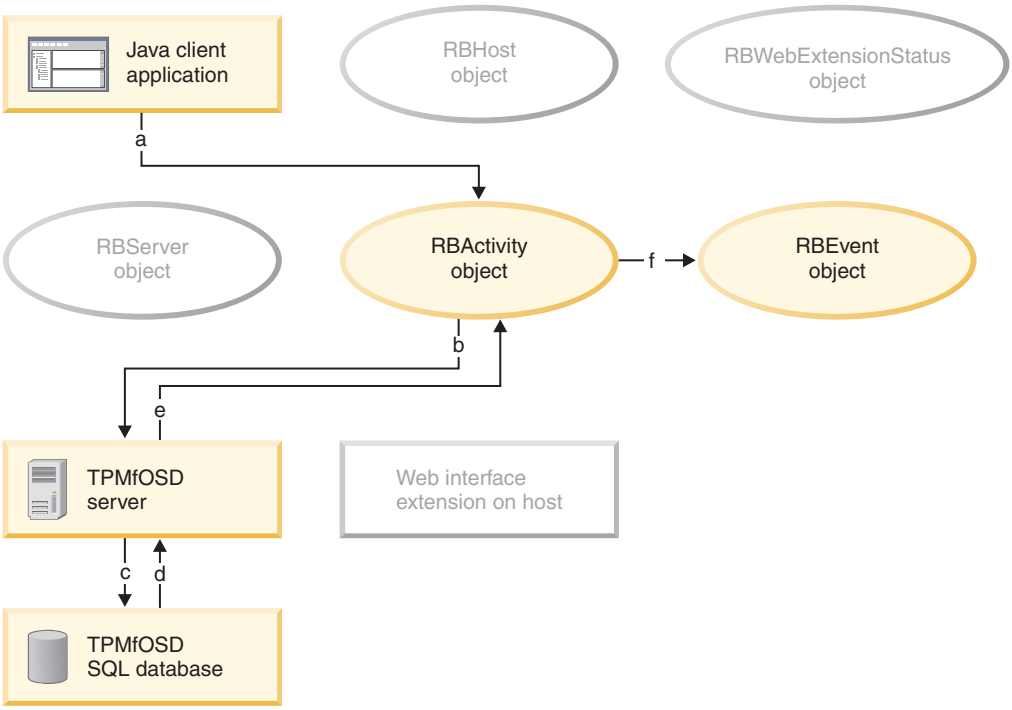


Figure 10. Step 4

a. The Java client application calls `RBActivity.getEventsFromServer` to check the task progress and status
b. `RBActivity` sends a remote procedure call to the OS deployment server
c. The OS deployment server uses an SQL query to get the list of events for this task
d. The SQL database send the task status to the OS deployment server
e. The OS deployment server encodes this task in XML
f. The Java client application receives a new `RBEvent` object that describes the current status of the task

Example classes

Here is the list of example classes present in `javaapi.zip` and a short description of what each one does.

TestAbortingActivities.java
Shows ways of cancelling a task that has been registered. This test shows both cancelling the whole task, and cancelling the task on only one specific task target.

TestAbortingActivitiesForHost.java
Shows an example of cancelling all tasks for a specific target.

TestActRepl.java
Shows how to run a task on a child, even if it has been created by connecting to the parent server. First a task template is created on the parent server and then it is replicated to the child workstation by performing an object copy task. This ensures that when running a task that uses the template, the child server has this information. Afterward a task is scheduled that uses the task template in question. If the target boots off the

child server, it will not succeed unless the object copy task of the task template to the parent server was successful.

TestAddRemoveHost.java
Shows a very basic example of adding or removing targets.

TestAgentScript.java
Runs a Rembo-C script on an target running the web interface extension.

TestBlackDevAndMod.java
Displays information that you can retrieve about black-listed PCI devices and models.

TestBootCDCreation.java
Creates a bootable CD from a target running the web interface extension. You can use the bootable CD to start the deployment engine on a target without using PXE.

TestCloneFromImageFile.java
This is an example showing capturing a cloned system profile from a reference image. The reference file can be a Windows Vista WIM image, or a Solaris Flash file. This is basically an example of using the Java API for what you can do using the new profile wizard if you select the option "Cloning from a reference file". Rembo toolkit files can also be cloned into system profiles, however for these it's best to look at the example TestToolkitProfileCreation

TestCloneHost.java
This example shows how to capture a cloned system profile from a reference workstation - Java API equivalent to running the new profile wizard and choosing "Cloning from a reference workstation". This is the preferred way to capture a system profile from a target, and should be used in place of the example class TestRBImageCaptureTemplate

TestImageManagement.java
Copies objects (task templates, system profiles, or software modules) from one OS deployment server to another. To run this test you must have multipleOS deployment servers working together in your environment.

TestLinuxPreboot.java
Boots a LinPE image on a target PXE target.

TestMultiServer.java
Provides information about how to setup a multi-server environment. Also shows a specific multi-boot server scenario of creating a task template, moves it to another server, and changes the scope so that the other server becomes the owner of the object.

TestOpenActivity.java
Runs an open task: this task runs once and only once on all targets that PXE boot off the OS deployment server.

TestPassword.java
This test class tries connecting to the OS deployment server using the Java API without supplying a password - typically this is not allowed, as an APISecret (java api password) needs to be setup in config.csv - however, if you use the method in config.csv of supplying "APITrusted" and list the targets that you trust, and then do not supply an APISecret - you can test that those specific targets can connect to the OS deployment server without a password with this example.

TestRADExport.java
Exports object(s) to a RAD file using the Java API.

TestRADImport.java
Imports a RAD file using the Java API.

TestRBCustomTemplate.java
Runs a custom Rembo-C script on a PXE target.

TestRBCustomTemplateUsingVariant.java
Similar to `TestRBCustomTemplate`, but also demonstrates the use of a variant to override parameters from the template during task registration without modifying the template itself.

TestRBDeviceBlankingTemplate.java
Performs military grade device blanking of a hard drive on a PXE target.

Note: Run this test with caution as it blanks disks.

TestRBEvents.java
Monitors events: you can run this test while other tests are running to monitor their progress.

TestRBHosts
Illustrates some properties of `RBHost` that can be gathered with the Java API.

TestRBImageCaptureTemplate.java
Refer to the TestCloneHost.java example for details.

TestRBLogFileLines.java
This class shows some examples of how to retrieve the contents of log files from the OS deployment server through the Java API.

TestCloneHost.java
Creates a system profile cloned from a reference computer (a PXE target).

TestRBOSDeployment.java
Deploys an OS configuration to a PXE target with some customization.

TestRBOSRestoreTemplate.java
Restores a system profile to a PXE target.

TestRBPrebootExecuteTemplate.java
Boots a DOS image on a PXE target.

TestRBServerConfig.java
Displays information about the OS configuration of the OS deployment server that you can retrieve and modify using the Java API.

TestRBServerReset.java
Restarts the OS deployment server through the Java API.

TestRBServerStatus.java
Gets the status of the OS deployment server through the Java API.

TestSoftwareComponentCreation.java
Creates software modules using the Java API. This example requires a target running the web interface extension.

TestStress.java
Runs many tasks and does inefficient polling to put a heavy load on the OS deployment server.

TestStressSmart.java
Shows a better method than `TestStress.java` for event polling.

TestSubnets.java
This test shows some of the subnet information available using the Java API, and how to probe the subnet again for multicast capability, change the multicast status and so on.

TestToolkitProfileCreation.java
Creates a system profile from an already existing Rembo Toolkit image.

TestUnattendedProfileCreation.java
Creates an unattended system profile from CD/DVD media.

TestWinPE.java
Boots a WinPE image on a PXE target.

TestWinPE2.java
Boots a WinPE2 image on a PXE target

TestWinPE2Update.java
This example shows how to update drivers in an existing WinPE2 image.

Deployment server configuration and maintenance

This documentation describes the initial classes that you will use, settings, tasks, and troubleshooting information.

Server connection and status

The first class that is used to establish a relationship with a OS deployment server is `RBServer`.

It can be instantiated by providing the IP address and port of the OS deployment server and a valid authentication token. When a method returns an object, see the information later in this section for details on the class.

```
RBServer(InetAddress ip, int port, String token)
```

There is another constructor for `RBServer` that takes an additional parameter, `java.util.logging.Level verbosity`. This parameter can be used to set a different level of details in the traces. See the "Controlling API traces" on page 378 for a longer discussion on traces.

The following methods should be used on instances of `RBServer` in order to get access to the various aspects of the server status:

- `bool isReady(int timeout)`: Probes the availability of the OS deployment server. The parameter `timeout` is specified in milliseconds.
- `RBServerStatus getStatus()`: Returns an object describing the current server status.
- `RBServerActHistory getActHistory(int StatType, Date start, long duration, long resolution)`: Returns an object describing the specified server task history.
- `bool isWebExtensionReady(InetAddress ip)`: Asks the OS deployment server to check the availability of the web interface extension. The variable `ip` is the address of the computer on which the web interface extension is required. The caller must expect negative answers the first couple of times they call this method because the answer of the agent never comes back immediately. The appropriate way to use this method is in a loop and with a few seconds delay between calls.

An instance of the class RBServerStatus represents a view on the server general status at the time it was instantiated, as displayed on the page **General Information** of the web interface. Individual methods such asgetVersion, getHostCount, and getSubnetCount provide access the individual values describing the state.

Deployment server settings

The following methods should be used on instances of RBServer in order to read and update the server OS configuration:

- RBServerConfig getConfig(): Returns an object describing the current server OS configuration
- void updateConfig(RBServerConfig config): Update the server OS configuration according to the object provided. This can trigger a server restart depending on the changes applied.

  Note: When this method returns the attributes of object, config can be different because the OS deployment server actually merges differences with its own records and sends back the resulting object. This remark applies to all update methods.
- RBBootServerRecs getBootServers(String filter, String order): Returns an object describing one or several server records in the server replication table.
- RBServerBlackDev[] getBlackListedDevices(): Returns the full list of black-listed PCI devices .
- RBServerBlackMod[] getBlackListedModels(): Returns the full list of black-listed target models.
- void restart(): Force a restart of all threads of the OS deployment service.

An instance of the class RBServerConfig represents a view on the current server OS configuration, exactly as it is available in the built-in Rembo-C variable RemboConf on the OS deployment server. Individual methods are provided to access the individual values of each setting.

The arrays of RBServerBlackDev and RBServerBlackmod instances represent the list of PCI devices and target models requiring special handling by the deployment engine, as displayed in “Hardware handling” on the Web interface. Individual methods are provided to access the individual values of each record.

Targets

At any time, the OS deployment server can know a large number of targets and the API to access them must remain efficient. For this reason, the API is based on a query and result set model.

Getting the attributes of a target or of a set of targets is, for scalability reasons, a two stage process:

1. Query the database about known targets matching a given search pattern with getHosts; a potentially large number of small records (only IDs) is created.
2. Query the actual attributes, a large record, of a small number of targets with elementAt or subList.

This is the approach used in the Web interface to respond in a timely manner to user actions. It is based on the fact that the console can display only a small amount of data at a time, so it is not necessary to fetch data that cannot be displayed. Additional records are fetched in a lazy way when they can really be shown to the console operator.

The following method should be used on instances of RBServer in order to query for known targets:

`RBHostRecs getHosts(String filter, String order)` builds a set of records matching the specified SQL filter condition, and sorted by the specified SQL ordering expression. Because SQL syntax is not exactly the same for all DBMS supported by the OS deployment server, only very simple operators should be used, for example, `<,>,=,like`, and `AND`. Field names which correspond to the attributes of `RBHost` must be enclosed in square brackets; this helps the server replace them with the actual field names expected by the database. For example,`getHosts("[IP] like '9. ", "[HostName] asc")`

A current description of all tables and columns in the SQL database can be downloaded directly from the server using a Web browser and the URL:

http://<server_name_or_IP>:8080/virtual/AutoDeployDistrib.ini.

The following methods can then be used on instances of `RBHostRecs`:

- `int size()`
- `RBHost elementAt(int index)`
- `RBHost[] subList(int fromIndex, int toIndex)` with the same parameter semantic as `java.util.Vector.subList` (element at `toIndex` not included in the list)

When several targets should be displayed simultaneously, use the `subList` method rather than making several individual `elementAt` calls.

A record of type `RBHost` includes all attributes editable on the target details page in the Web interface of the OS deployment server. Methods are provided to access them individually, for example, `getHostName` and `setHostName`. After making changes to a `RBHost` instance, call its `update()` method to propagate the change to the OS deployment server. Removing or deleting a target can be done with the following method:

- `void RBHost.remove()`: Removes the specified target record from the server database.

Manually adding a target is different, because it must handle the case where the target already exists in the database, which should not be considered as an error because the computer can have been rightfully discovered. This is handled by the target registration API, where each of the four parameters can be empty if unknown but at least one should be specified:

```
RBHost RBServer.registerHost(String IP, String MAC,
 String UUID, String Serial)
```

This returns the corresponding target record or creates a new one if none exists yet. The IP address must be given in decimal dotted format, for example, *192.168.2.34*. The MAC address must be given in hexadecimal, with or without colons, for example, 00:16:41:17:AF:DE or 00164117AFDE.

`RBHost` methods `getPxeBootMode` and `setPxeBootMode` use an additional class `PxeBootMode` that can take the following several values:

- `IGNORE`: the PXE server ignores all requests coming from this target.
- `HDBOOT`: the PXE server sends an answer to cancel the network boot and perform a harddisk boot.

- HDBOOT_IF_IDLE: the target is loaded and performs a harddisk boot if there is no activity for this target.
- NORMAL: the target is loaded and waits for tasks to perform

Deployment objects

The following objects are required to run an OS deployment:

- System profiles (disk images or installation CD images).
- The differentOS configurations that can be defined for each system profile.
- The automatic binding rules that can be defined for each OS configurations.
- Optional software modules (including drivers and DOS floppy images).
- The automatic binding rules that can be defined for each software modules.

Although these objects typically exist in small numbers only on a OS deployment server, a mechanism that allows the caller to select how many objects are loaded together is implemented. It is the same principle as for RBHosts.

The following method should be used on instances of RBServer to work on system profiles:

- RBSystemProfileRecs getSystemProfiles(String filter, String order): Returns an object that knows all system profiles matching the filter expression (same semantic as getHosts) and can fetch subsets from the server using the three methods:
  - int size()
  - RBSystemProfile elementAt(int index)
  - RBSystemProfile[] subList(int fromIndex, int toIndex) with the same parameter semantic as java.util.Vector.subList

There are three ways to create system profiles:

1. Capture an image from a reference computer.
2. Capture the contents of an installation CD or DVD.
3. Capture an image from a reference image file (a windows WIM image or a Solaris Flash Archive file).

The partition scheme used during a deployment is part of the system profile. It is not practical to change this value inside RBSystemProfile for each deployment, and not always possible to specify partition sizes as percentage of the target of the hard disk size. It is possible to select the partition scheme dynamically, and other target parameters as well, every time a deployment task is performed. See the documentation on RBDiskLayout.

RBConfiguration class

Configurations are used to automate the assignment of parameters to targets during the deployment of a system profile.

For a longer description of the properties, in the web interface, go to **Server → OS deployment → System Profiles**.

The Fixed_ prefix in the property names is a reminder that at deployment time, the value found in the RBConfiguration overrides values found in the RBHost. The values in RBHost are applied only if the corresponding Fixed_ value in RBConfiguration is empty. After the deployment, the values in RBHost reflect the actual values that have been used. Fixed_ values themselves can be over-ridden with values provided in the RBOSDeploymentTemplate in the argument params Map of RBServer.registerActivity(), or in the params Map of RBActivity.addTarget().

`RBConfigurationRecs getConfigurations(String filter, String order)` returns an object that knows all configurations matching the filter expressions, (same semantic as `getHosts`) and can fetch subsets from the server using the three methods.

`RBSystemProfile` also has a `getConfigurations` method to retrieve the configurations that belong to a specific system profile.

RBSoftwarePackage class

`RBSoftwarePackageRecs getSoftwarePackages(String filter, String order)` returns an object that knows all software modules matching the filter expression (same semantic as `getHosts`) and can fetch subsets from the server using the three methods:

- `int size()`
- `RBSoftwarePackage elementAt(int index)`
- `RBSoftwarePackage[] subList(int fromIndex, int toIndex)` with the same parameter semantic as `java.util.Vector.subList`.

`RBSoftwarePackage` provides methods to query individual attributes of the instances, including `getDescription` and `setDescription` to virtually rename the object, corresponding to the various fields displayed on the Web interface.

In regards to pass numbers, older versions used to put software modules in *passes* with reboots between the passes. The current version uses a more refined concept of *stages*. The reboot between stages is optional, therefore it is possible to install two software modules in a specific order, without having to reboot between them. When a software module is bound by an automatic binding rule to the target of a deployment task, the system uses `SoftSeqID` to decide when the package must be installed. The default uses the stage found in the `RBSoftwarePackage`.

For OS Deployments, an ordered list of software modules can be added to the `RBDeploymentParams`. After the OS is deployed, the software modules are installed based on the order of the software modules in the `softwareSequence` and passed to `setSoftwareToDeploySequence`. It is possible to specify reboots in the `softwareSequence` between the installation of two software modules.

RBBootServer class

By default, each OS deployment server automatically installs and uses its own private database. On Windows, this is done by creating and using Open DataBase Connectivity (ODBC) source AutoDeploy, linked to the file `c:\program files\ibm\tpmfosd\autodeploy.mdb`. You can force the server to use another existing ODBC source with MSI properties `ODBC_DSN`, `ODBC_USERNAME`, and `ODBC_PASSWORD`.

After installation, it is also possible to select a different source using the fields `DbName`, `DbUser`, and `DbPass` of config.csv. For more information, go to the Web address: http://www-1.ibm.com/support/docview.wss?uid=swg21247013. The advantage of using config.csv is that the same configuration file can be used for multiple servers, and that it can be updated or reloaded at any time.

MultipleOS deployment servers can use the same database. In this case, each server automatically adds a record in the Servers table. These records are available in the Java API as `RBBootServer` objects.

A sample config.csv, for three servers sharing a common MS SQL database can be found in the javaapi.zip file, inside examples/TestMultiServer.java.

RBBootServerRecs getBootServers(String filter, String order): Returns an object that knows all OS deployment servers registered in the database that match the filter expression (the same semantic as getHosts).

Deployment server tasks

A task is an operation that the user wants to perform on one or several targets.

Three classes are used for task management:

1. RBActivityTemplate: Defines what must be done, for example, deploy a given Windows XP cloning image in multicast, with no bandwidth limitation and power-off when done.
2. RBActivity: Defines the temporal aspect, for example, start not before than today 9 p.m. and not after tomorrow 2 a.m..
3. RBActivityTarget: Records one intended of the task and keeps track of progress and completion. All the instances are stored in the SQL database in order to persist when a target needs to restart, or when the daemon is restarted.

Task templates

RBActivityTemplate is an abstract class with several subclasses designed to represent different kind of tasks.

The role of a task template is to store a set of predefined parameters to use when instantiating future tasks. The constructors and methods reflect the characteristics of the task. The current list of subclasses includes

- RBPrebootExecuteTemplate is used for hardware configuration.
- RBprofileCreationTemplate creates system profiles from CD-ROM/DVD (unattended setup), from reference computers (cloning), or from image files.
- RBOSRestoreTemplate restores cloning profiles without customization.
- RBCustomTemplate is used for arbitrary Rembo-C code execution.
- RBOSDeploymentTemplate performs customized deployments.
- RBDeviceBlankingTemplate performs military-grade device blanking.

Note: For some categories of tasks, typically web interface extension tasks such as creating a software module or creating RAD files, there is no explicit task template. This is because instances of such individual tasks intrinsically need different parameters.

Depending on the template, the targets are either OS deployment servers, web interface extensions, for example, rbagent.exe, or PXE clients.

The instances of RBActivityTemplate have a unique internal ID and a longer textual description. They know their type and are managed using the RBServer method :

- RBActivityTemplateRecs getTemplates(String filter,String order)

Methods of RBServer are used to create new templates on the OS deployment server and get corresponding Java objects.

RBActivityTemplate has two methods, getEndAction and setEndAction to manage end actions. The end action indicates what the target must do when the action is complete:

- STAY: the target waits for more tasks.
- HDBOOT: the target performs a boot on harddisk.
- REBOOT: the target performs a warm reboot.
- POWEROFF: the target is switched off.

RBCustomTemplate:

RBCustomTemplate registerCustomTemplate(String description, String script, Map defaultParams) creates a new task template on the OS deployment server to run a custom script. The method will return an object that describes the template. With defaultParams, the caller can pass arbitrary pairs, key and value, to the script. Keys and values must be textual strings.

RBPrebootExecuteTemplate:

RBPrebootExecuteTemplate registerPrebootExecuteTemplate(String description, RBSoftwarePackage diskImage, String fsType, boolean unloadUndi, int size, RBFile[] filesToAdd, String[] filesToPatch, Map customParams) creates a new task template on the provisioning server for the execution of a Preboot image, for example, a DOS floppy image.

Before a preboot task boots the ramdisk it currently sets the progress to 100 and the status to COMPLETED.

RBOSDeploymentTemplate:

RBOSDeploymentTemplate registerOSDeploymentTemplate(String description, RBConfiguration config) creates a new task template on the OS deployment server for the deployment of the given OS configuration

To supply custom values at deployment time in a more flexible way than what the RBConfiguration allows, class called RBDeploymentParams was defined. This class can be instantiated by the target, using setter methods for relevant fields in RBHost.

The customization done at this level is applied to all targets of the task. The properties defined in the RBDeploymentParams overrides the same properties defined in the OS configuration and system profile. In addition, the partition scheme can be set for all of the task targets from the RBDeploymentParams. The partition scheme is first retrieved from the OS configuration with the getDiskLayout method which returns an RBDiskLayout object. This RBDiskLayout can be modified then set in the RBDeploymentParams with the method setDiskLayout. An ordered list of software modules can be specified in the RBDeploymentParams that is installed after the OS deployment.

The installation of software modules is always related to deployment tasks. For a given deployment, the list of software modules to install is built dynamically. The list contains packages selected by an automatic binding rule and packages that are manually selected.

Automatic binding rules are used to install a driver when a PCI device is present on the target, or to add an application every time a given operating system image is deployed.

For manual selection, all instances of RBOSDeploymentParams can have an optional attribute SoftwareToDeploy that contains an array of RBSoftwarePackage objects.

RBImageCaptureTemplate:

```
RBImageCaptureTemplate registerImageCaptureTemplate
(String description, boolean withCMOS,
boolean withProtectedPartitions, boolean forceRABCapture,
boolean allowNoSysPrep)
```

This registers a new OS image capture template on the OS deployment server. The task creates a new system profile, with all OS configurations that can be detected on the target.

When registering a task that uses this template, the call to RBServer.registerActivity() must have the following specified in the params parameter:

1. An entry with key profiledesc and value which is a String representing the textual description of the profile.
2. An entry with key profilecomment and value which is a String representing a comment for the profile.

These must be specified because the same template can be used to take several images; therefore the strings used to describe and identify the new profile are given at the task level.

RBOSRestoreTemplate:

```
RBOSRestoreTemplate registerOSRestoreTemplate
(String description, boolean restoreCMOS,
boolean restoreProtectedPartitions, boolean runSysPrep)
```

This creates a new task template for the deployment of a cloning profile without customization.

RBSoftwarePackageParams:

Parameters used for software module creation tasks are always different, therefore using a template does not make sense. Consequently, software module creation tasks use an implicit template and use the class RBSoftwarePackageParams to specify the task parameters directly at the time when the task is scheduled, similarly to what can be done using RBDeploymentParams.

Because the type of parameters relevant for the various types of software modules differ, static methods are provided to instantiate and preset most parameters. In any case, various methods enable you to read and update optional parameters, for example, to define static software ordering.

When a set of parameters is obtained, the RBServer method, probeSoftwarePackageCreation, can be called to simulate the creation of the software module with the specified set of parameters and return the effective values that would be completed by the software module creation process. For example, when a command line is automatically generated or when a default description is used. Subsequently, or alternatively, the RBServer method, registerSoftwarePackageCreationActivity, must be called to actually perform the software module creation. Progress monitoring can be done as with any other type of task.

All the software module creation tasks must use as the target a computer running the Web interface extension, `rbagent`. When the target is a OS deployment server, the source path must be visible under the `\import` directory (but symlinks are permitted, including under Windows where the -specific symlink files can be used). In Windows, file paths must be an absolute path preceded with the disk letter, for example, `C:\temp\...` or UNC paths, `\\file-server\share\...`. In order to have UNC path work correctly, the `rbagent` process must run as a domain user having privileges to access the network share.

The software module creation tasks started using the following parameters captures a diskette into a new software module, for the purpose of running the floppy in a RAM disk:

```
RBSoftwarePackageParams newFloppyCreationParams
(String pkgDescr, String comment, String sourcePath)
```

The software module creation tasks started using this parameter captures a WinPE CD into a new software module, for the purpose of running WinPE in a RAM disk:

```
RBSoftwarePackageParams newWinPECreationParams
(String pkgDescr, String comment, String sourcePath)
```

The software module creation tasks started using this parameter captures a Linux kernel and initial RAM disk into a new software module, for the purpose of running Linux in a RAM disk:

```
RBSoftwarePackageParams newLinuxPECreationParams
(String pkgDescr, String comment, String sourcePath,
String kernelName,
String ramdiskName)
```

The software module creation tasks started using the following parameters capture a Windows MSI installation file into a new software module, for the purpose of installing an application:

```
RBSoftwarePackageParams newMSICreationParams
(String pkgDescr, String comment, String sourcePath, String destPath,
String cmdLine)
```

The software module creation tasks started using these parameters captures a Linux RPM installation file into a new software module, for the purpose of installing an application:

```
RBSoftwarePackageParams newRPMCreationParams
(String pkgDescr, String comment, String sourcePath,
String destPath, String cmdLine)
```

The software module creation tasks started using the following parameters captures a Solaris PKG installation file into a new software module, for the purpose of installing an application:

```
RBSoftwarePackageParams newSolarisCreationParams
(String pkgDescr, String comment, String sourcePath,
String destPath, String cmdLine)
```

The software module creation tasks started using these parameters captures a Windows driver installation file into a new software module, for the purpose of installing a driver:

```
RBSoftwarePackageParams newWinDriverCreationParams
(String pkgDescr, String comment, String sourcePath,
String destPath, boolean createRules)
```

The software module creation tasks started using these parameters captures an arbitrary set of files, scripts, ini files, registry exports (in NT4 format), into a new software module to customize the deployments.

```
RBSoftwarePackageParams newFileSetCreationParams
(String pkgDescr, String comment, String sourcePath,
String destPath, String cmdLine,
boolean substKeywords, boolean removeFiles)
```

The software module creation tasks started using the following parameters capture a command line for execution during a deployment.

```
RBSoftwarePackageParams newCmdLineCreationParams
(String pkgDescr, String comment, String cmdLine)
```

Task variants

Each task template has a corresponding task variant class.

Variants are used to set or override parameters in the template at run time without modifying the template itself. Each task template contains the method `getVariant();` which returns an `RBActivityVariant` that corresponds to the template in question. Attributes of the variant are the same as the attributes of the template itself (in certain cases, variants also contain additional attributes that apply only at task registration time). This variant can then be modified and passed in when registering the task (with the `registerActivity()` method) instead of the template itself.

For example, if you have an `RBOSRestoreTemplate` that has an end action of `STAY` (the default), but for one particular operating system restore task, you want to change the end action to `HDBOOT` (to boot off the hard drive after restoring the operating system), you can follow these steps:

1. Get an `RBOSRestoreVariant` based on your template by calling the `getVariant()` method on the template
2. Modify the end action of `RBOSRestoreVariant` by calling `setEndAction(RBActivityTemplate.HDBOOT)`
3. Call the `RBServer.registerActivity()` method, but pass in the modified `RBOSRestoreVariant` instead of the `RBOSRestoreTemplate`

Task scheduling

The class `RBActivity` is used to schedule the start of a task. Instances are created using a method of `RBServer`.

Here are a couple of examples:

```
RBActivity registerActivity(String description, RBActivityTemplate template,
boolean suspendOnError, boolean doWakeOnLan, Map params, RBHost target)
```

It creates a new task of the given type on the OS deployment server and returns an object that describes it. This simple class method schedules the task to start immediately and have no expiration date.

```
RBActivity registerActivity(String description, RBActivityTemplate
template, boolean suspendOnError, boolean doWakeOnLan, Map params,
 RBHost[] targetHosts, Date execDate, Date doneDate, Date expireDate)
```

This method creates a new task of the given type and returns an object that describes it. It registers several targets, called `targetHosts`, and gives better control on the scheduling of the task. The parameters of this method, `execDate` and `doneDate` are used to define a window for the beginning of the task. Targets can finish performing the task after the end of the window, but it is too late to start.

The parameter execDate can also be used to select between multiple possible tasks. If an idle target has several choices, it takes the task with the smallest execDate first.

The expiration date, expireDate, is used to cleanup database records after a reasonable amount of time, therefore it is not necessary to manually remove them. SuspendOnError is used to prevent the start of other tasks by targets that have not successfully completed this one.

There are two methods in RBServer to stop the execution of tasks:

1. boolean[] abort(RBActivity[] rbActivities): Asks all of the targets of given tasks to skip or abandon the task.
2. boolean[] abort(RBActivityTarget[] rbActivityTargets): Asks all of the given targets to skip or abandon the task.

Task targets

Instances of RBActivityTarget represent the actual target performing the task. There are several ways to provide access to task targets.

- RBServer.registerActivity
- RBActivity.addTarget
- RBActivity.getActivityTargets returns an RBActivityTargetRecs object that gives access to RBActivityTarget instances.

When an RBActivityTarget generates a result, the method getActivityTargetResults() can be used to retrieve the resulting object. The class of the result depends on the type of task. For more information, see the class name and description in the related task templates.

Events

Monitoring events on OS deployment servers is based on the same HTTP communication channel as other API calls. A set of lightweight messages can be used to retrieve lists of events from the server.

The class RBEvent provides information about an event initiated from an RBActivityTarget. An event is generated when the status of a target changes and is kept in the SQL database until another event overrides it.

There are two methods to retrieve events from a given OS deployment server:

1. RBEvent[] RBServer.getEvents(Date since) performs a single call to the OS deployment server and returns all the events that have occurred on this server, regardless of the task. Only events more recent than *since* are returned. This parameter is given in the local time of the Java client, the API does the necessary conversions to the time of the server database.
2. RBEvent[] RBActivity.getEventsFromServer(Date since) performs a single call to the OS deployment server and returns all the events that have occurred on all the targets of this task. Only events more recent than *since* are returned.

The class RBEvent has the following methods to query event attributes:

- int getProgress()
- String getStatus()
- String getLastProgressMsg()
- Date getUpdateTime()

These methods fetch information locally.

The class RBEvent has methods to get fresh instances of related objects:

- RBActity getRBActivity()
- RBActivityTarget getRBActivityTarget()
- RBHost getRBHost()

These three methods perform additional server calls to retrieve current values of the objects.

It also has the following methods to retrieve related instances within the Java Virtual Machine (JVM):

- boolean belongsTo(RBActivity)
- boolean belongsTo(RBActivityTarget)
- boolean belongsTo(RBHost)

The purpose of these methods is to pass existing Java objects and to compare internal IDs in order to test for identity.

Controlling API traces

The logger reads $JAVA_HOME/lib/log.properties for the log settings. The user can use the system property, java.util.logging.config.file to specify the specific location of the log.properties file.

An example setting is -Djava.util.logging.config.file=c:/Program Files/IBM/common/logging/log.properties.

This is a sample of a log.properties that writes to both the console and to a file:

```
# Properties file which configures the operation of the JDK
# logging facility.

# The system will look for this config file, first using
# a System property specified at startup:
#
# >java -Djava.util.logging.config.file=myLoggingConfigFilePath
#
# If this property is not specified, then the config file is
# retrieved from its default location at:
#
# JDK_HOME/jre/lib/logging.properties

# Global logging properties.
# ------------------------------------------
# The set of handlers to be loaded upon startup.
# Comma-separated list of class names.
handlers=java.util.logging.FileHandler, java.util.logging.ConsoleHandler

# Default global logging level.
# Loggers and Handlers because override this level
.level=ALL

# Loggers
# ------------------------------------------
# Loggers are usually attached to packages.
# Here, the level for each package is specified.
# The global level is used by default, so levels
# specified here simply act as an override.

com.ibm.rbapi.level=ALL

# Handlers
# -----------------------------------------
```

```
# --- ConsoleHandler ---
# Override of global logging level
java.util.logging.ConsoleHandler.level=FINEST
java.util.logging.ConsoleHandler.formatter=java.util.logging.SimpleFormatter

# --- FileHandler ---
# Override of global logging level
java.util.logging.FileHandler.level=FINEST
java.util.logging.FileHandler.formatter=java.util.logging.SimpleFormatter

# Naming style for the output file:
# (The output file is placed in the directory
# defined by the "user.home" System property.)
java.util.logging.FileHandler.pattern=%h/java%u.log

# Limiting size of output file in bytes:
java.util.logging.FileHandler.limit=50000

# Number of output files to cycle through, by appending an
# integer to the base file name:
java.util.logging.FileHandler.count=1
```

The log level for the rbapi can be set in the log.properties file with the following:

```
com.ibm.rbapi.level=(SEVERE | WARNING | CONFIG | INFO | FINE | FINEST | ALL)
```

The first constructor for RBServer uses the settings defined in log.properties:

```
public RBServer(InetAddress ip, int port, String token)
```

This other constructor overrides the log level defined in log.properties:

```
public RBServer(InetAddress ip, int port, String token, Level verbosity)
```

There are three ways to modify these settings:

1. Restart the application.
2. Instantiate a new `RBServer`.
3. Modify log.properties and then make a call to `LogManager.readConfiguration();`.

Command-line interface

You can run OS deployment server command-line commands, as well as commands to interface with the OS deployment server.

NetClnt commands are deprecated.

NetClnt

NetClnt commands are deprecated.

NetClnt is a command-line utility, available for Windows, Linux and Solaris. You can use NetClnt to access the files stored on a provisioning server remotely, and to perform some maintenance tasks.

Here is a list of features included in NetClnt:

- Communications between NetClnt and the server are protected with a password and encrypted.
- You can use NetClnt in interactive mode (with an interface similar to a FTP client), or in batch mode.

- You can delete, rename, move, download, and upload files. You can also delete, rename, and move directories.
- You can extract shared files from the server repository, and upload them to the server repository. Archives including both regular files and related shared files can be created and reimported to the server.

Using NetClnt interactively

NetClnt commands are deprecated.

1. Run the file `netclnt` in your installation directory.

   You see the interactive prompt `NETFS>`.
2. Open a connection with a OS deployment server:
   a. Run the `open` command, followed with the IP address (or host name) of the server.
   b. Enter the same password as the one set in the `NetPassword` parameter of the remote OS deployment server.
3. Test the connection by issuing `adir` command. This command downloads and displays the content of the current directory on the remote OS deployment server.
   - If this command works (that is, you see some result), then your connection is working.
   - If you do not see any result, double-check the IP address and password you have entered with the `open` command.
4. You can now use any NetClnt command. For example:
   - Send a new file to the OS deployment server (`put`)
   - Change the current directory (`cd`)
   - Get a server file locally (`get`)

   When you want to send a file to the OS deployment server, the file must exist on your local computer, in the current directory (you can change the current local directory with `lcd`). Downloaded files are stored in the current local directory. See "NetClnt command reference" on page 382 for more information about commands. At any time, you can use the `help` command to get the list of valid commands, and a command name without argument to get a short summary of its syntax.

Using NetClnt in batch mode

You can use the batch mode of NetClnt to create scripts operating on remote files without any user intervention (for example, file upgrade).NetClnt commands are deprecated.

To use NetClnt in batch mode:

1. Create a text file containing the sequence of commands you want to run with `NetClnt`.
2. Run `netclnt` *`filename`*, where *`filename`* is the name of the text file containing the sequence of commands. The first command must be the `open` command, with a valid IP address and a valid password (password can be passed as the third parameter to `open`).

   You can pass parameters to your `netclnt` script, and use them in the script with the `#` prefix. For example, you replace every occurrence of `#1` in your script with the first parameter following the script name in the command line used to run `netclnt`. To use this new feature, call `netclnt` with the following syntax:

   ```
   netclnt -f scriptname parameters
   ```

where *scriptname* is the file name of the `netclnt` script to run, and parameters are the parameters passed to the `netclnt` script. You can use the new command `chkparams` to check that the number of parameters passed to the script correspond to what the script is expecting.

Example

Here is an example of batch file for use with NetClnt:

```
open 192.168.1.10 rembopwd
lock
mkdir /myfiles
cd /myfiles
put file1
put file2
put file3
exit
```

Here is an example with parameters:

```
netclnt -f script 192.168.1.10 rembopwd
```

```
chkparams 2 <server-ip> <rembo-password>
open #1 #2
dir
exit
```

Using NetClnt to manage the shared repository

NetClnt commands are deprecated.

Creating a duplicate repository

The commands `getshared` and `putshared` can be used to download and upload all shared files related to a given archive, to and from a local shared repository. This can be used to duplicate a part of the shared repository locally, for backup purposes or to transfer it to another provisioning server.

Note: The local shared repository uses the deployment engine format of index files, which is slightly different from the server format (the server format is ready to handle a large number of simultaneous targets, while the local format is optimized for local access only). Therefore, you should not copy a shared repository built using `getshared` directly onto a server, but use `putshared` instead.

Exporting shared files

A more convenient way to manipulate disk archives along with the related shared files is to use `.rad` file exports. A `.rad` file can embed several archive headers and other files, and all related shared files. It is a convenient way to backup and transmit archives in a self-contained format. See the description of the commands `radget`, `radput` and `radlist` in the "NetClnt command reference" on page 382.

Synchronizing shared repositories

When using several OS deployment servers, if archive files are copied manually or mechanically from one server to the other, you might need to replicate the shared repositories to ensure that the destination server has all necessary server files. The `sync`, `msync` and `rsync` commands can do that on a specified subset of server files. These commands ask the targeted server to directly contact the parent server and download any missing shared file. See the documentation for these commands in the reference section "NetClnt command reference" on page 382. Combined with the

command `xcopy`, `rsync` is the best way to replicate the content of a parent provisioning server on a child provisioning server.

NetClnt command reference

NetClnt commands are deprecated.

Here is the complete reference for NetClnt commands:

open, connect

```
open host [password]
connect host [password]
```

Sets up the connection information required for remote access to the OS deployment server. This is the first command to issue when starting NetClnt. If you do not specify the password in the command line, you are asked to enter the password.

close

```
close
```

Closes the current connection. This command must be issued if you want to issue a new `connect` command to the same server, but with a different password.

pwd

```
pwd
```

Displays the current remote directory.

dir, ls

```
dir [path]
ls [path]
```

Displays the content of a directory on the remote server. If you use `dir` without parameters, the content of the current directory on the server is displayed (you can use `cd` to change the current remote directory). If you use a parameter, the content of the directory specified by the parameter is displayed (the path is relative to the current remote directory).

cd, chdir

```
cd path
chdir path
```

Changes the current remote directory. When NetClnt starts, the current remote directory is the root directory of the server file system. The path is relative to the current remote directory.

md, mkdir

```
md path
mkdir path
```

Creates a new directory on the server. The path is relative to the current remote directory.

rd, rmdir

```
rd path
rmdir path
```

Deletes a directory on the server. The directory must be empty. The path is relative to the current remote directory.

deltree

```
deltree path
```

Deletes a remote directory and all its sub-directories. The target directory does not have to be empty. Use this command with caution.

ren, move

```
ren old-name new-name
move old-name new-name
```

Renames or moves a file on the server. Both the old file name and the new file name must be specified. If the new file name is in a different directory from the old file name, the file is moved. Paths are relative to the current remote directory.

del, rm

```
del path
rm path
```

Removes (deletes) a file on the server. The path is relative to the current remote directory. If the `path` is a globbing pattern (that is, it expands file names using a pattern matching notation), all matching files are deleted. Acceptable globbing wildcards are the asterisk and the question mark.

lcd

```
lcd path
```

Changes the current directory on the local target. This command has no effect on the server: it only modifies the current directory on the target which is running NetClnt.

!*localcommand*

```
!localcommand
```

Executes a command on the local target . For example, `!dir` displays the content of the current directory on the local target . `!edit myfile` starts the MS-DOS editor on the file `myfile`.

get, mget, rget

```
get remote-path [local-path]
mget pattern
rget pattern
```

Downloads the remote file `remote-path` from the current remote directory into the file `local-path` in the current local directory. If the second parameter is omitted, the file is stored under the name `remote-path`. The `mget` variant downloads all remote files that match the globbing `pattern` to the local directory. Acceptable globbing wildcards are the asterisk and the question mark. The `rget` variant recursively searches all directories matching the globbing `pattern` (all files within the directory entered are copied).

put, mput, rput

```
get local-path [remote-path]
mput pattern
rput pattern
```

Uploads the local file `local-path` from the local remote directory into the file `remote-path` in the current remote directory. If the second parameter is omitted, the file is stored under the name `local-path` on the server. The `mput` variant uploads all local files that match the globbing `pattern` to the

remote directory. The rput variant recursively searches all directories matching the globbing pattern (all files within the directory entered will be copied).

xcopy

```
xcopy [-d remote-server remote-source-dir local-dest-dir]
```

Copies files recursively, from a remote server to a local server. All the files stored in the directory remote-source-dir on the remote server remote-server are copied in the directory local-dest-dir on the local server. Tests are made on the source and the destination to skip files that have already been copied. This function is therefore used to replicate two servers daily, because it only copies files that have been modified, or added since the last xcopy operation.

By default, xcopy does not delete any file on the local server. But if -d is specified as the first parameter, xcopy then deletes every local file that has not been found on the remote server, creating an exact copy of the remote server directory in the local directory. The server password must be the same on both servers for this command to succeed, because xcopy uses the password provided to the connect when connecting to the remote server.

sync master, msync master, rsync master

```
sync master [remote-path]
msync master pattern
rsync master pattern
```

Orders the connected server to download all missing shared files for the archive specified by remote-path from another OS deployment server specified in parent. The server is specified by its IP address or by its DNS name. The msync variant triggers the transfer of shared files for all archives matching the globbing pattern in the remote directory. Acceptable globbing wildcards are the asterisk and the question mark. The **rsync** variant recursively enters into all directories matching the globbing pattern (all files within the directory entered are processed). A typical use for replicating all shared files from a given server is rsync master *. Note that the rsync command involves no file transfer to and from NetClnt, but only between the two servers.

getshared

```
getshared localarchive [localrepository]
```

Downloads all available shared files related to a given local archive to a local repository. By default, the repository is created in the target current working directory, but you can override the path by specifying a localrepository parameter. The localarchive can have been previously downloaded from the server, or copied by another mean.

putshared

```
putshared remotearchive [localrepository]
```

Uploads all locally available shared files related to a given server archive to the server shared repository. By default, files are read from a repository in the target working directory, but you can override the path by specifying a localrepository parameter. The remotearchive can have been previously uploaded to the server, or copied by another means. Both getshared and putshared optimize network traffic, in the sense that they only transfer files needed by the specified archive and already in the destination shared repository.

radget

```
radget radfile [file*]
radput radfile [file*]
radlist radfile
radcheck radfile
```

The radget command downloads all files specified on the command line (several filenames and globbing patterns are accepted), and all related shared files. The result is stored in a monolithic .rad export file, specified in radfile. You might need a significant amount of disk space on the deployment engine to perform this operation. The destination file is always overwritten (files are not added to an existing radfile). You can use radlist to list the content of a radfile export, radcheck to verify the internal consistency of a radfile export, and radput to reupload the files to the same or another server. If you have not specified a file pattern to the radput command, then all files in radfile are uploaded (including all shared files which are not yet known to the destination server). If you provide a list of files or patterns, only the matching files are uploaded. The radput works very well even if the source radfile is on a network share, because it reads only the shared files that are not yet present on the destination shared repository.

wake

```
wake hardware-address
```

Sends a wake packet to the specified hardware address. You can use this command to wake (remote power-on) targets. The target must be located on the same network as the computer running netclnt. Additionally, the target must be equipped with a Wake-On-LAN capable network card. The hardware address must be entered without any delimiters. For example, hardware address 00:10:4B:12:34:56 must be entered as 00104B123456.

sleep

```
sleep sleeptime
```

Waits sleeptime seconds. This command is useful in batch scripts. No command is executed during the waiting time.

lock

```
lock
```

Acquires a unique lock on the computer, which is automatically released when NetClnt exits. This mechanism can be used to ensure that there is only one instance of NetClnt running at a time on a target , to avoid conflicts on the server, when automated batch files are run.

getlock, releaselock

```
getlock lockname
releaselock lockname
```

Acquires or releases a server lock. A server lock is identified by a name (lockname) and can be locked by only one computer at a time. These two commands can be used to make sure that no computer is accessing a particular resource during an update. See the GetLock function.

start, stop, reload

```
start servername
stop servername
reload servername
```

Controls the Windows service on a remote computer (implemented in Windows `netclnt` executable file only). These commands start, stop or reload the OS deployment server on a remote computer, by using standard Windows remote service control. The user that is logged in when running `netclnt` must have administrator equivalence on the remote server.

timestamp

```
timestamp message
```

Displays a timestamp and a message to the output. This command can be used in batch scripts to output the current time on the screen, or in a log file.

chkparams

```
chkparams reqparamcount [usage]
```

Checks the number of parameters and displays a usage string if parameters are missing. Use this command only in scripts called with the `netclnt -f scriptname params` syntax. `Chkparam` verifies that there is at least `reqparamcount` parameters in the command line, and displays a usage string containing the message in the `usage` parameter if the number of actual parameters is less than the requested parameters count.

sqlinit

```
sqlinit dbgw-ip-addr odbcsource [username password]
sqlclose
sqlexec sql-query
sqlcopy source-dbgw-ip source-odbcsource table-name
        [source-username source-password]
```

This group of functions connects `netclnt` with a database gateway (dbgw) to perform SQL queries remotely. The remote target must be running both dbgw and a OS deployment server. Use `SqlInit` to initiate the connection with the remote database gateway. You must provide the name of the ODBC source to use on the remote target , and the ODBC user name and password if needed. Then, use `SqlExec` to run SQL statements on the remote target , or `SqlCopy` to copy the content of a table from a given database gateway, to the remote target (or local host if `sqlinit` was called to connect to `localhost`). When calling `SqlCopy`, the parameters `source-dbgw-ip`, `source-dobcsource`, `source-username` and `source-password` identify the ODBC source to use as the source of the copy operation, and `table-name` is the name of the database table to copy. Calling `SqlCopy` on a database table is equivalent to the running of a `select` statement on the source, and `delete + insert` statements on the destination.

RbAgent

RbAgent is running under the operating system and is often referred to as the web interface extension in this documentation.

The web interface extension, running either on the OS deployment server or on a deployed computer, can access local resources under the operating system that are unavailable through the web interface. Its built-in operations are oriented to make use of these local resources.

RbAgent command reference

rbagent [-d] [-v *loglevel*] [[-o] | [-f *iface*] [-s *srvip:password*]] [-arguments]

- `-d`: prints debug info to the standard output, do not run as a daemon (do not detach)
- `-v`: sets the logging verbosity level (from 1 to 6, default is 2)
- `-o`: runs in offline mode (no connection to the OS deployment server)
- `-f`: specifies the interface. `iface` is the IP address of the preferred interface/subnet to use
- `-s`: specifies the OS deployment server address. *srvip:password* is a server IP address and password, that can be plain text or MD5.
- `-arguments`: are optional built-in supported operations described in "Built-in RbAgent operations."

Built-in RbAgent operations

The following built-in agent operations are used as arguments of `rbagent`. Each operation identifier is followed by a short description and a usage example.

- `hostinfo`

  Displays general information about the computer.

  ```
  C:\rbo\bin\remboc\win32>rbagent hostinfo
  RbAgent 4.0 ($Revision: #12 $) - (c) Copyright 2004-2005 by IBM, Switzerland
  Connect 192.168.1.36 -> 192.168.1.36
  Starting Web interface extension
  model       : 2373FWG ThinkPad T42
  platform    : IAx86
  serial      : 99K5VHM
  uuid        : 261D6A81466E11CBABB7D0EB0A4468DB
  hwaddr      : 000D60D0602F
  ipaddr      : 192.168.1.36
  netmask     : 255.255.252.0
  gateway     : 192.168.1.254
  dhcpserver  : 192.168.1.5
  remboserver : 192.168.1.36
  Stopping Web interface extension

  C:\rbo\bin\remboc\win32>
  ```

- `rad-registerhost <IP|MAC|SN|UUID>= ... [HostName=...] [...]`

  Registers a new target identified by either its IP address, MAC address, serial number or UUID. You can assign values to the target database records of the new target , such as a target name. The database records include the BOM table and the User profile. View the information about keyword substitution in the user's guide for more information.

  ```
  rbagent rad-registerhost IP=192.189.34.32 HostName=Host32@location
  ```

- `rad-unregisterhost <IP|MAC|SN|HostName|Description>= ...`

  Removes from the OS deployment server a target identified by either its IP address, MAC address, serial number, target name, or description.

  ```
  rbagent rad-unregisterhost IP=192.189.34.32
  ```

- `rad-deployhost <IP|MAC|SN|HostName|Description>= ... [Scheme=...|SchemeName=...] [Config=...|ConfigName=...] [bind] [reboot|pxeboot] [wakeup]`

  Deploys the target specified by either its IP address, MAC address, serial number, target name, or description. You can specify a *scheme* and *OS configuration*. Use option *bind* to bind the OS configuration to the target . Option

reboot attempts to restart the target if it is currently undergoing deployment, on locked screen, or is running the web interface extension. Option *pxeboot* forces the computer to start on the network . Option *wakeup* tries to turn on target currently powered off using Wake-on-LAN technology.

```
rbagent rad-deployhost IP=192.189.34.32 Scheme=Default pxeboot
```

- `rad-radput` *`radpath.rad [force|ignore] [spdl=<kbytes/s>] [softstages]`*

  Uploads a `.RAD` archive to the OS deployment server. The *force* option adds new objects, even if there are pre-existing objects (profiles, OS configuration, schemes) with the same description on the server. *ignore* does not add new objects when there are pre-existing objects with the same description. *spdl* is the upload speed limit. If the option *softstages* is present, the software stages included on the RAD file override those of the server.

  ```
  rbagent rad-radput c:\temp\backup.rad force
  ```

- `radcheck` *`radpath.rad`*

  Verifies the consistency of a `.RAD` archive. `radcheck` not only checks the file format but tries to read all files in the `.rad` file to verify that they are properly encoded.

  ```
  rbagent radcheck c:\temp\backup.rad
  ```

- `rad-chksoft` *`sourcepath ["<attr>=value" ...]`*

  simulates the creation of a new software module. *attr* can take the values *descr, content, pkgname, dest, cmdline, pass, flags, dosubst, norules, nopcirules, OSType, OSVersion, OSArch, Model*

  ```
  rbagent rad-chksoft "c:\drivers\ahci"
  ```

- `rad-mksoft` *`sourcepath ["<attr>=value" ...]`*

  Creates a new software module. *attr* can take the values *descr, content, pkgname, dest, cmdline, pass, flags, dosubst, norules, nopcirules, OSType, OSVersion, OSArch, Model* Before using this command, try *rad-chksoft* to view default values and optionally modify them.

  ```
  rbagent rad-mksoft "c:\drivers\ahci" "descr=DriversAHCI" "OSType=Windows 2003%"
  "OSVersion=%Service Pack 1%" "OSArch=x86-64" "Model=1951 Thinkpad T60"
  ```

- `rad-mkbootcd` *`iso-image-path server-ip server-pwd [fixedip] [fixednetmask] [fixedgateway][allowipoverload][allowsrvipoverload]`*

  Creates a bootable CD to start without network boot. *iso-image-path* is the path of the iso file. *server-ip* is the IP address of the OS deployment server on which the target must start. *server-pwd* is the OS deployment server password. Options *fixedip, fixednetmask* and *fixedgateway* are used for target properties. Options *allowipoverload* lets the user to modify the IP address of the target , and *allowsrvipoverload* lets the target contact a OS deployment server at another IP address than the one specified by *server-IP*.

  ```
  rbagent rad-mkbootcd myimage.iso 192.168.2.34 mypassword
  ```

  Note: This command requires "PXE activation" which is not available on all hardware. The CD created with this command might not work on specific hardware. See also Chapter 11, “Booting targets without using PXE,” on page 289.

- `fallback-mbr`

  Installs a special MBR on the hard disk to force PXE booting at the next boot. Use this once; the original MBR is restored afterward with its typical boot order.

  ```
  rbagent fallback-mbr
  ```

- `cmdlines file`

  Runs the list of commands specified in `file`, waiting for the end of each command before starting the next one. Error codes returned by the commands are indicated.

  ```
  rbagent cmdlines c:\cmdfile.txt
  ```
- `joindomain domain adminuser adminpwd [joinou]`

  joins a Windows domain`joindomain /w workgroup [adminuser adminpwd]` joins a Windows workgroup `joindomain /s domain trustpwd` changes the trust account.

  ```
  rbagent joindomain mydomain myadmin mypassword
  rbagent joindomain /w myworkgroup
  rbagent joindomain /s mydomain mynewtrustpassword
  ```
- `checkdevices`

  Lists the devices (under Windows OS) which do not function correctly, with their full name.

  ```
  rbagent checkdevices
  ```
- `rad-osrestore <IP|MAC|SN|HostName|Description>= ... [Config=...|ConfigName=...] [bind][reboot|pxeboot][wakeup][switchon]`

  Performs the same as a manual operating system restoration from the web interface.

  ```
  rbagent rad-osrestore IP=192.168.1.19 configName="Windows XP(setup)" bind
  ```
- `rad-configlist`

  Returns the list of available OS configurations on the server.

  ```
  rbagent rad-configlist
  ```
- `rad-resetscope`

  Used when the server is renamed. All objects are marked in the database as belonging to the new name.

  ```
  rbagent rad-resetscope
  ```
- `rad-exportlogs destination=... [mode= 1|2|3|4 [compress]] [activityid=...]`

  Used to export debugging information. `destination` is the folder where the logs are copied. If `mode=1`, all the logs are placed in `destination`. If `mode=2`, all the logs are placed in `destination`, with only one file for all the tasks. If `mode=3`, all the logs are placed in `destination`, with one file for each task. If `mode=4`, all the logs are placed in `destination` in raw format. `compress` can be used only if `mode=4` and compresses the resulting log file. `activityid` provides the logs only for the specified activity.

  ```
  rbagent rad-exportlogs destination=c:\temp mode=1
  ```

RbAgent and access to remote files on Windows

If you want RbAgent to have access to remote files, for instance to create a software module, you must pay attention to whether RbAgent is running as a service or within a logon session:

- If RbAgent is running within a logon session (user application), the network shares can be accessed through mapped drive letters.
- If RbAgent is running as a service, mapped drive letters are not accessible because drive mappings are created for a specific logon session. However, RbAgent is able to access files on the network share using a UNC path (for example `\\192.168.168.17\c$\install\`).

RbAgent must naturally also have read privileges on the files. These privileges depend on the workgroup/domain account used to run RbAgent.

rembo command reference

Use this command to perform maintenance operations on the server shared repository.

Syntax

```
rembo [-d][-v loglevel][options][maintenance_options]
```

Parameters

-d
Specifies to run as a console application instead of as a service.

-v Sets the verbosity level. The default value is: 3.

Verbosity levels:

- 0: no output
- 1: log errors
- 2: log errors and warning
- 3: log errors, warnings, and informational messages
- 4: log errors, warnings, informational messages, and notice messages
- 5: log errors, warnings, informational messages, notice messages, and debug messages
- 6: log everything, including network packets.

[options]

-c Specifies the configuration file name. The default file name is: `rembo.conf`.

-cert Specifies the file name of a certificate to import.

-convert
Converts the server to the new native file system.

-delcerts
Deletes certificates that were either created automatically or by using the **-cert** option.

-exit Stops the server after processing **-cert** / **-c**

-force Forces conversion even if a conflict is found.

-readonly
The slave is no longer permitted, use the child instead.

-rehash
Reattribute hashed inode IDs to every file.

-sdb Specifies the server database file name. The default file name is: `server.db`.

[maintenance_options]
The following options determine the maintenance operation to be performed on the shared file repository.

-chkshared
Initiates a verification process on the server shared repository. Errors

are reported, but no corrective actions are applied. This operation can take from several minutes to several hours, depending on the size and fragmentation of the shared repository. During this time, a connection to the server cannot be established.

-fixshared
Initiates a repair process on the server shared repository. Errors are corrected by recomputing missing indexes if possible, or deleting corrupted records. The repair process is automatically triggered when the server is killed with unflushed changes to the shared repository. This operation can take from several minutes to several hours, depending on the size and fragmentation of the shared repository. During this time, a connection to the server cannot be established.

-statshared
Initiates a complete analysis on the server shared repository, and displays a summary of shared files usage. This operation can take from several minutes to several hours, depending on the size and fragmentation of the shared repository. During this time, a connection to the server cannot be established.

-sweepshared
Initiates a mark and sweep operation on the server shared repository. Shared files not used in any archives on the server (as reported by the -statshared option) will be cleaned out. This can be used to recover disk space on the server after deleting a significant number of disk images. This operation can take from several minutes to several hours, depending on the size and fragmentation of the shared repository. During this time, a connection to the server cannot be established.

-packshared
Initiates a mark and sweep operation on the server repository, then shrinks all shared repository files to their minimum size to recover any preallocated storage. Note that the use of this function may lead to fragmentation of the shared file repository, and its use is therefore not recommended unless disk space must be recovered for other applications. This operation can take from several minutes to several hours, depending on the size and fragmentation of the shared repository. During this time, a connection to the server cannot be established.

Example

To mark and sweep a set of files on the shared repository to decrease the size of the repository, run the command:

```
rembo -d -v 4 -packshared
```

dbgw command reference

Use this command to install and uninstall the dbgw service. You can also disable remote connections using this command. This command is available only for Windows operating systems. For Linux operating systems a `dbgw.jar` file is provided.

Syntax

```
dbgw [-d] [-v verbosity] [-f logfile] [-t idle-sec] [-l]

dbgw -i

dbgw -u
```

Parameters

-d Specifies to run as a console application instead of as a service.

-v verbosity
Sets the verbosity level. The default value is: 3.

Verbosity levels:

- 0: no output
- 1: log errors
- 2: log errors and warning
- 3: log errors, warnings, and informational messages
- 4: log errors, warnings, informational messages, and notice messages
- 5: log errors, warnings, informational messages, notice messages, and debug messages
- 6: log everything, including network packets.

-f logfile
Writes the output of the gateway to a file. By default, it is sent to standard output.

-t idle-sec
Specifies the timeout before the connection is automatically closed. The default value is 120.

-l Disables remote connections.

-i Installs the dbgw service.

-u Uninstalls the dbgw service.

Example

To run the gateway as a command from the command line and log all the SQL queries, submit the following command:

```
dbgw -d -v 4
```

Glossary

A

administrative group
A group of related computers. An administrator can create administrative groups to organize target systems into meaningful categories, and to facilitate deployment of software to multiple targets.

B

bare metal computer
A computer on which there is nothing reliable but the hardware. It can be

coming straight from factory without any data on its hard disk (out of the box) or it can contain a possibly damaged operating system.

Basic Input/Output System (BIOS)
The code that controls basic hardware operations, such as interactions with diskette drives, hard disk drives, and the keyboard.

BIOS See Basic Input/Output System.

blacklist
In Tivoli Provisioning Manager for OS Deployment, a list of PCI devices or of computer models which are known to raise issues, accompanied by hardware settings which must be used to work around the issues.

C

child An OS deployment server that is a subordinate of another OS deployment server in a replication tree structure. Only the top-level parent OS deployment server is not a child. See also parent.

clone To prepare a reference computer and create a system profile ready for deployment.

D

database server
The computer on which the database application and database are installed.

Deployment
A process which installs an operating system, and possibly other applications and files, on a target computer. During a deployment, data previously stored on the hard drives of the target is deleted.

Deployment scheme
A specific type of task template. A deployment scheme contains parameters for customizing a deployment on a target, and the target display screen layout. See also task template.

DHCP See Dynamic Host Configuration Protocol.

Dynamic Host Configuration Protocol (DHCP)
A communications protocol that is used to centrally manage configuration information. For example, DHCP automatically assigns IP addresses to computers in a network.

F

free-text condition
In Tivoli Provisioning Manager for OS Deployment, a condition written in Rembo-C; syntax, using variables and Java-like logical operators, and which evaluates to true or false.

H

hardware configuration
A set of parameters used to configure hardware before an operating system installation. It includes RAID settings, BIOS update information, BIOS settings, and custom hardware configuration parameters.

M

MCAST
A proprietary transfer protocol of Tivoli Provisioning Manager for OS Deployment computers using multicast. Contrast with unicast and PCAST.

MTFTP
See Multicast Trivial File Transfer Protocol.

multicast
Bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to many computers.

Multicast Trivial File Transfer Protocol (MTFTP)
Multicast TFTP.

N

network boot
The process of starting up a computer directly over the network rather than on a disk.

O

OS configuration
The operating system parameters of a system profile .

OS deployment server
The computer on which the Tivoli Provisioning Manager for OS Deployment application and files are installed.

P

parent An OS deployment server in a replication tree structure that has at least one dependent OS deployment server. See also child.

PCAST
A proprietary transfer protocol of Tivoli Provisioning Manager for OS Deployment that delivers non-identical sets of files to several target computers using multicast. Contrast with MCAST and unicast.

PCI See Peripheral Component Interconnect.

Peripheral Component Interconnect
A local bus that provides a high-speed data path between the processor and attached devices.

Preboot Execution Environment (PXE)
PXE is an industry standard target/server interface that allows networked computers that are not yet loaded with an operating system to be configured and booted remotely. PXE is based on Dynamic Host Configuration Protocol (DHCP). Using the PXE protocol, targets can request configuration parameter values and startable images from the server. The PXE process consists of the system initiating the protocol by broadcasting a DHCPREQUEST containing an extension that identifies the request as coming from a target that uses PXE. The server sends the target a list of OS deployment servers that contain the operating systems available. The target then selects and discovers an OS deployment server and receives the name of the executable file on the chosen OS deployment server. The target downloads the file using Trivial File Transfer Protocol (TFTP) and runs it, which loads the operating system.

PXE See Preboot Execution Environment.

R

RAD file
A file containing deployment objects such as task templates, system profiles, and software modules used to archive data or to transfer data between two OS deployment servers. A RAD file has a `.rad` extension.

RAID See Redundant Array of Independent Disks.

redeployment
The process of synchronizing a hard-disk content to its reference image stored on a hidden and protected redeployment partition.

redeployment preload
The process of creating a reference image of a computer at the end of a deployment, and saving this reference image into a protected redeployment partition (invisible to the user and to the operating system itself).

Redundant Array of Independent Disks (RAID)
RAID is a way of storing the same data in different places (thus, redundantly) on multiple hard disks. By placing data on multiple disks, I/O operations can overlap in a balanced way, improving performance. Multiple disks increase the mean time between failure (MTBF) and storing data redundantly increases fault-tolerance.

Rembo-C;
A programming language, descendant of the C language combined with traces of JavaScript and Java.

replicated server
An OS deployment server which shares data with one or several other OS deployment servers. The servers are hierarchically structured with a parent and child servers. A child child can act as parent to replicated servers further down in the hierarchy.

replication
The process of copying files from a parent server to a child server. A selection can be performed on the kind of information that must be replicated. Files that have been modified are copied over.

S

shared repository
In Tivoli Provisioning Manager for OS Deployment, a repository of server objects where each file is stored only once, even if it belongs to several objects. The shared repository reduces the storage space necessary to hold all server objects.

software module
A group of files, and potentially command lines, packaged together under one name. A software module can be installed on a target during a deployment.

software snapshot
A differential image of software installed on top of a running operating system. Software snapshot creation is deprecated. Any previously created software snapshots can be deployed for compatibility with earlier versions.

system profile
The partition layout and list of files for deployment of an operating system, either by unattended setup or by cloning. A system profile can have several configurations.

system snapshot
For Windows only. The partition layout and list of files for deployment of an operating system, created by cloning without using Sysrep. A system snapshot cannot be parametrized and can only be restored, not deployed.

T

target A computer that is known to an OS deployment server.

target list
A comma-separated-value list of targets used for adding large numbers of targets to the OS deployment server without having to start the targets up individually on the network.

task A set of actions designed to achieve a particular result. A task is performed on a set of targets on a specific schedule.

task template
A group of elements which can be customized on a target computer. These elements are mostly screen layouts which condition the appearance of the target computer screen during the different phases of its control by Tivoli Provisioning Manager for OS Deployment. See also Deployment scheme.

TCP tunnel
A way to provide TCP connectivity to target computers.

TFTP See Trivial File Transfer Protocol.

Trivial File Transfer Protocol (TFTP)
In Internet communications, a set of conventions that transfers files between targets using minimal protocol.

U

unattended setup
Operating system installation on a target, using original installation files and parameters contained in a script defined on the OS deployment server. Contrast with clone.

unicast
Transmission of data to a single destination. In Tivoli Provisioning Manager for OS Deployment, a transfer protocol that delivers a stream of files to a single target. Based on TCP, this protocol is faster when there are only a few target computers on the receiving end of the transfer. This protocol can also be used in networks where multicast traffic is not properly handled. Contrast with MCAST and PCAST.

universal image
A cloned system profile that has been prepared with all drivers for disk types and hardware abstraction layer variants encountered in the pool of targets to be deployed.

W

Wake on LAN
A technology that enables a user to remotely turn on systems for off-hours maintenance. A result of the Intel-IBM Advanced Manageability Alliance

and part of the Wired for Management Baseline Specification, users of this technology can remotely turn on a server and control it across the network, thus saving time on automated software installations, upgrades, disk backups, and virus scans.

Web interface
A user interface for one or more administrative tasks.

Web interface extension
An agent that allows the web interface to have access to the content of the target on which it is running. For example, to browse disks and read and write files.

Z

zone An IP range or domain that is used to logically group computers into regions. You can define one or more zones for each region.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Notice for Windows Automated Installation Kit (AIK)

Windows Automated Installation Kit (AIK) for Windows 7 in English is distributed by Microsoft and is available on the Microsoft website from the following link at the time of publication: http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=en.

The Windows AIK is licensed to you by the code's owner and not by IBM it is your responsibility to determine whether the license terms offered by the code's owner are acceptable to you.

YOUR USE OF THE WAIK AND ANY URL'S OR MATERIALS ON THIRD PARTY WEBSITES ("THIRD PARTY MATERIALS") IS "AS IS", WITHOUT WARRANTY FROM IBM OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. TO THE EXTENT PERMITTED BY LAW, IBM DISCLAIMS ALL LIABILITY FOR ANY CLAIMS ARISING OUT OF USE OF THE THIRD PARTY MATERIALS.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml

Adobe® is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Intel and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows , and Windows NT are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

Copyrights

© Copyright IBM Corporation 2009, 2010. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM web site pages may contain other proprietary notices and copyright information which should be observed.

Portions of third-party software included in this IBM product is used with permission and is covered under the following copyright attribution statements:

- Copyright (c) 1998-2005, The OpenSSL Project. All rights reserved.
- Copyright (c) 1995-2005 Jean-loup Gailly and Mark Adler, the ZLIB data compression library.
- Copyright 1994-2006, The FreeBSD Project. All rights reserved.

The MD5 Message-Digest Algorithm was developed by Ron Rivest. The public domain C language implementation used in this program was written by Colin Plumb in 1993. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, without any conditions or restrictions. This software is provided "as is" without express or implied warranty.

Portions include cryptographic software written by Eric Young (<(eay@cryptosoft.com)>). This product may include software written by Tim Hudson (<(tjh@cryptosoft.com)>).



Printed in USA