Tivoli Provisioning Manager for OS Deployment

# *Installation Guide*

Tivoli Provisioning Manager for OS Deployment

# *Installation Guide*

**Note**
Before using this information and the product it supports, read the information in "Notices" on page 73

This edition applies to IBM Tivoli Provisioning Manager for OS Deployment 7.1.1.3 and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2007, 2010. All rights reserved. US Government Users Restricted Rights – Use duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents


**Chapter 1. Installation requirements . . 1**
Upgrade paths to 7.1.1.5 . . . . . 1
Server system requirements . . . . . 1
Hardware requirements. . . . . 1
Operating system requirements . . . . . 2
DHCP server . . . . . 3
Java requirements . . . . . 3
Target system requirements . . . . . 4
Hardware requirements. . . . . 4
Operating system requirements . . . . . 5
File systems . . . . . 6
Network requirements . . . . . 6
Web interface requirements . . . . . 7

**Chapter 2. Installing Tivoli Provisioning Manager for OS Deployment on Windows operating systems . . . . . 9**
Installation prerequisites . . . . . 9
Installation steps on Windows operating system . . 10
Silent installation . . . . . 12
Uninstalling Tivoli Provisioning Manager for OS Deployment on Windows operating systems . . . 14
Full build - complete uninstallation . . . . . 15
Full build - partial uninstallation . . . . . 15
Fix pack or interim fix uninstallation . . . . . 15
Upgrading the OS deployment server on Windows 16
Upgrading Tivoli Provisioning Manager for OS Deployment . . . . . 16
Upgrading in a multiserver infrastructure . . . 17
Upgrade of OS deployment server database . . 17
Installation and uninstallation logs on Windows . . 18

**Chapter 3. Installing Tivoli Provisioning Manager for OS Deployment on UNIX and Linux systems . . . . . 21**
IBM AIX specific prerequisite . . . . . 21
Database prerequisite . . . . . 21
Web interface prerequisite . . . . . 21
Installing an OS deployment server with an Apache Derby database . . . . . 22
Installing with other databases . . . . . 22
MySQL 4.1 example . . . . . 22
DB2 on AIX example . . . . . 24
Oracle 11i on Linux example . . . . . 24
Silent installation . . . . . 25
Advanced features . . . . . 27
Startup scripts . . . . . 27
Starting the OS deployment server . . . . . 28
PAM configuration (optional) . . . . . 29
Accessing the user interface with Active Directory domain users . . . . . 29
Uninstalling Tivoli Provisioning Manager for OS Deployment on UNIX . . . . . 31
Upgrading Tivoli Provisioning Manager for OS Deployment on UNIX . . . . . 31
Upgrading Tivoli Provisioning Manager for OS Deployment . . . . . 31
Upgrading in a multiserver infrastructure . . . 32
Upgrade of OS deployment server database . . 33

**Chapter 4. DHCP server configuration 35**
Configuring the DHCP server . . . . . 36
Adding Tivoli Provisioning Manager for OS Deployment to an existing Boot Discovery infrastructure . . . . . 37
DHCP option 60 . . . . . 37
Adding DHCP option 60 to Windows 2003 DHCP server . . . . . 37
Adding DHCP option 60 to a host with ISC DHCP server . . . . . 37
DHCP option 43 . . . . . 38
Setting DHCP option 43 . . . . . 38
Example: option 43 for PXE servers on different subnets . . . . . 39
Example: option 43 to create a PXE boot menu 40
Additional Linux cloning options . . . . . 41
Additional SUN and IBM PowerPC options . . . 41
dhcpd.conf example . . . . . 41

**Chapter 5. Prerequisites for provisioning Windows . . . . . 45**

**Chapter 6. Prerequisites for provisioning Solaris . . . . . 47**
Jumpstart and the OS deployment server . . . . 47
Solaris install server . . . . . 47
Preparing a Solaris install server for operating system content . . . . . 48
Preparing a Solaris install server for Flash Archives . . . . . 49

**Appendix A. Integrating Tivoli Provisioning Manager for OS Deployment in a corporate environment . . . . . 51**
Using an alternative database on Windows for Tivoli Provisioning Manager for OS Deployment . . 51
Example: Installing with Oracle on Windows 64-bit. . . . . 52
Password protecting a Microsoft Access database 54
Multiserver infrastructure . . . . . 54
Installing a multiserver infrastructure with a centralized database . . . . . 56
Installing a multiserver infrastructure with multiple databases . . . . . 59
Working with Tivoli Provisioning Manager for OS Deployment locally . . . . . 65

**Appendix B. Installing and uninstalling the web interface extension. . . . . 67**
Status of the web interface extension . . . . . 67
Installing the web interface extension on Windows operating systems . . . . . 68
Uninstalling the web interface extension on Windows operating systems . . . . . 68
Installing the web interface extension on UNIX operating systems . . . . . 68
Uninstalling the web interface extension on UNIX operating systems . . . . . 69

**Appendix C. Federal Information Processing Standards Compliance . . 71**

**Notices . . . . . 73**
Trademarks . . . . . 75
Copyrights . . . . . 76

# Chapter 1. Installation requirements

This *Installation Guide* contains the information you need to install Tivoli® Provisioning Manager for OS Deployment. Before you use the product, ensure that you meet all installation requirements.

## Upgrade paths to 7.1.1.5

The following table describes the upgrade paths to Tivoli Provisioning Manager for OS Deployment Version 7.1.1.5:

*Table 1. Tivoli Provisioning Manager for OS Deployment upgrade paths:*

| Upgrade from | Windows upgrade (*) | UNIX upgrade (**) |
|---|---|---|
| 5.1.1 | Yes | No |
| 7.1 | Yes | No |
| 7.1.1 | Yes | Yes |
| 7.1.1 fix pack 1 | Yes | Yes |
| 7.1.1 fix pack 2 | Yes | Yes |
| 7.1.1 fix pack 3 | Yes | Yes |
| 7.1.1 fix pack 4 | Yes | Yes |

(*) An upgrade is the installation of a full build. The upgrade requires the partial uninstallation of the current full build.

(**) An upgrade is a full build file replacement after having stopped the dbgw, rembo, and rbagent daemons.

## Server system requirements

To use Tivoli Provisioning Manager for OS Deployment, you can either use a network boot CD or set up a *DHCP server* and an *OS deployment server*. Both servers can be on the same computer. ODBC or JDBC support must be available on the target running the OS deployment server.

### Hardware requirements

The minimum and recommended configurations for the OS deployment server includes:

*Table 2. System requirements for the OS deployment server*

| | Processor type | Processor speed | RAM | Free disk space |
|---|---|---|---|---|
| **Minimum** | Dual-Xeon | 2 GHz | 1 GB RAM | 10 GB |
| **Recommended** | Quad-core or two dual-core | 2 GHz | 2 GB RAM | 100 GB |

You must store Tivoli Provisioning Manager for OS Deployment files on a large hard disk if you plan to create many hard-disk images, and you might want to use

a fast processor to minimize the time spent creating these images. The OS deployment server is multithreaded and benefits from computers with multiple processors.

## Operating system requirements

Compatible operating system and architecture for Tivoli Provisioning Manager for OS Deployment servers are shown in Table 3:

*Table 3. Tivoli Provisioning Manager for OS Deployment server operating systems*

| Operating system | Architecture |
|---|---|
| Windows Server 2003 and Windows Server 2003 R2 | x86-32 and x86-64 |
| Windows Server 2008 and Windows Server 2008 R2 | x86-32 and x86-64 |
| SUSE Linux Enterprise Server (SLES) 10 | x86-32 and x86-64 |
| SUSE LinuxEnterprise Server (SLES) 10 | IBM® PowerPC® 64 (iSeries® and pSeries®) |
| SUSE Linux Enterprise Server (SLES) 10 | IBM System z® |
| SUSE Linux Enterprise Server (SLES) 11 | x86-32 and x86-64 |
| SUSE Linux Enterprise Server (SLES) 11 | IBM PowerPC 64 (iSeries and pSeries) |
| SUSE Linux Enterprise Server (SLES) 11 | IBM System z |
| Red-Hat Enterprise Linux Server (RHEL) 5 | x86-32 and x86-64 |
| Red-Hat Enterprise Linux Server (RHEL) 5 | IBM PowerPC 64 (iSeries and pSeries) |
| Red-Hat Enterprise Linux Server (RHEL) 6 | x86-32 and x86-64 |
| Red-Hat Enterprise Linux Server (RHEL) 6 | IBM PowerPC 64 (iSeries and pSeries) |
| Red-Hat Enterprise Linux Server (RHEL) 6 | IBM System z |
| IBM AIX® 6.1 | IBM PowerPC 64 (iSeries and pSeries) |
| IBM AIX 7.1 | IBM PowerPC 64 (iSeries and pSeries) |
| Solaris 10 Update 6 | SPARC 64-bits |

Table 4 shows which databases can be used together with specific operating systems installed on the OS deployment server.

*Table 4. Databases and server operating systems*

| Database | Operating systems |
|---|---|
| Microsoft SQL Server 2005 SP2, 2008 SP1 | Windows |
| Microsoft Access Driver | Windows |
| IBM DB2® Enterprise 9.1 FP4a and 9.5 FP3b | Windows and UNIX |
| Apache Derby 10.2.2 , 10.3.1.4, 10.5.5.1, 10.6.1.0 | UNIX |
| MySQL 4.1 | UNIX |
| Oracle 10g and 11g R1 | Windows and UNIX |

**Note:**

1. When installing Tivoli Provisioning Manager for OS Deployment on a UNIX computer with a MySQL database, use MySQL version 4.1 exclusively.
2. Tivoli Provisioning Manager for OS Deployment does not support MySQL onIBM System z.

## DHCP server

There is no special product requirements for the DHCP server, but it must be configured appropriately.

If the DHCP server and the OS deployment server are running on the same computer, the DHCP server must support the definition of the `Class identifier` DHCP option (option 60). Tivoli Provisioning Manager for OS Deployment can work with almost any RFC compliant DHCP server, including Windows 2003 DHCP server, Windows 2008 DHCP server, ISC DHCP server, Solaris DHCP server and the Netware 5 DHCP server.

**Note:** Because of the nature of PXE, you cannot run two PXE servers on the same computer. If you have installed another PXE boot tool such as Microsoft ADS, you must disable it before installing Tivoli Provisioning Manager for OS Deployment.

## Java requirements

You need a Java Virtual Machine, preferably Sun version, on your OS deployment server

- To connect to the database on UNIX, and on Windows if you use JDBC instead of ODBC.
- To use the Java API.

If your operating system does not include a Java virtual machine, you can download J2SE SDK at http://java.sun.com. Versions 1.5 and above are compatible.

**Note:**

1. GNU Java (GCJ) is not supported.
2. Before proceeding with the product installation, ensure that the Java run time version is correct. Determine which Java version is in use by typing `java -version`. Sun Java (*supported*) reports something similar to:

   ```
   java version "1.5.0_06" Java(TM) 2 Runtime Environment,
   Standard Edition (build 1.5.0_06-b05) Java HotSpot(TM)
   Client VM (build 1.5.0_06-b05, mixed mode, sharing)
   ```

   GNU Java (*unsupported*) reports something similar to:

   ```
   gij (GNU libgcj) version 4.0.0 20050519 (Red Hat 4.0.0-8)
   Copyright (C) 2005 Free Software Foundation, Inc. This is free software;
   see the source for copying conditions.  There is NO warranty;
   not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
   ```
3. If you install Sun Microsystems Java on top of an existing version included in the distribution, the `PATH` variable needs to be prepended with the Sun Java bin directory and export the variable, for example:

   ```
   export PATH=/usr/java/jre1.5.0_12/bin:$PATH
   ```

   The `PATH` export can be added to the startup configuration file (`.bashrc`, `.bash_profile`, and so on.)
4. If no version of Java is installed, the Sun Java bin directory can be either prepended or appended to the `PATH` variable.

# Target system requirements

Make sure your targets conform to the requirements for a better experience with the product.

## Hardware requirements

x86 and x86–64 target requirements are:

- Minimal processor: Pentium type level.
- Minimal RAM memory: 512 MB. Recommended RAM memory: at least 1 GB.

  **Note:** Some targets require more than 512 MB of RAM to run WinPE 3.0.
- VESA compliant (release 2.0 or later) Video BIOS to get "high" resolution (VGA fallback is always possible in case of incompatibility) and to have a user interface on the target.

  **Note:** Tivoli Provisioning Manager for OS Deployment can also work on headless computers, in which case all operations have to be performed from the web interface.
- Either a legacy ATA drive (with Ultra DMA support if speed is required) or a BIOS-supported hard drive.
- Vista 2008 Windows 7 To deploy an unattended setup of a Windows Vista/2008/7 operating system, you need 10GB of disk space for a 32-bit operating system and 20 GB for a 64-bit operating system.
- DMI support for collecting hardware information, such as model and serial number.

**Note:** Disks with a size equal to or larger than 1 TB are not supported on targets running Linux and on virtual images.

To make full use of the Tivoli Provisioning Manager for OS Deployment features, remote-boot x86 and x86–64 targets must be equipped with a PXE-compliant bootrom, either version 2.00 and above. Most recent computers with on-board network adapters have built-in PXE support. The network cards that have been shown to work best with Tivoli Provisioning Manager for OS Deployment are *Intel* adapters.

For computers without built-in PXE support, you might consider using a PXE emulation floppy available from various third party sources, creating network boot media, or working offline with deployment media.

SUN SPARC targets need a firmware that supports the SUN wanboot method for network booting. Wanboot is the only supported network boot method for SUN SPARC.

Tivoli Provisioning Manager for OS Deployment can also deploy operating systems to VMWare virtual machines. Use of the Intel e1000 adapter on VMWare virtual machines requires VMWare ESX 3.0.2 with February 2008 patches, VMWare ESX 3.5 or later, VMWare Workstation 6.0.3 or later, VMWare Server 2.0 or later.

Microsoft Virtual PC and Microsoft Virtual Server are not supported.

## Operating system requirements

Compatible operating system, version, and architecture for OS deployments on Tivoli Provisioning Manager for OS Deployment targets are shown in Table 5:

*Table 5. Tivoli Provisioning Manager for OS Deployment target operating systems*

| Operating system | Version | Architecture |
|---|---|---|
| Windows Server 2008 | GA, R2 | x86-32 and x86-64 |
| Windows 7 | GA | x86-32 and x86-64 |
| Windows Vista | GA and SP2 | x86-32 and x86-64 |
| Windows 2003 Server | SP2/R2 | x86-32 and x86-64 |
| Windows XP Professional | SP3 | x86-32 and x86-64 |
| Windows 2000 Server | SP4 | x86-32 |
| SUSE Linux Enterprise Server (SLES) | 11, GA and SP1 | x86-32 and x86-64 |
| SUSE Linux Enterprise Server (SLES) | 10, GA and SP 1, 2 , 3 | x86-32 and x86-64 |
| SUSE Linux Enterprise Desktop (SLED) | 10, GA and SP 1, 2 , 3 | x86-32 and x86-64 |
| Red-Hat Enterprise Linux (RHEL) Server | 6, GA | x86-32 and x86-64 |
| Red-Hat Enterprise Linux (RHEL) Server | 5, GA and Update 1, 2, 3, 4, 5 | x86-32 and x86-64 |
| Red-Hat Desktop | 5, GA and Update 1, 2, 3, 4, 5 | x86-32 and x86-64 |
| Red-Hat Enterprise Linux (RHEL) Server | 4, Update 8 | x86-32 and x86-64 |
| Red-Hat Desktop | 4, GA version only | x86-32 and x86-64 |
| VMWare ESX Server | 4.1 | x86-64 |
| VMWare ESX Server | 4.0 GA, Update 1 and 2 | x86-32 |
| VMWare ESX Server | 3.5 Update 4 | x86-32 |
| VMWare ESX Server | 3.0.2 | x86-32 |

**Note:** RedHat Enterprise Linux Server 4.x is not supported on Hyper-V.

Limited support is provided for the following non x86 operating systems:

*Table 6. Tivoli Provisioning Manager for OS Deployment non x86 target operating systems*

| Non x86 operating system | Version | Architecture |
|---|---|---|
| SUSE Linux Enterprise Server (SLES) | 11, GAand SP1 | IBM PowerPC 64 (pSeries) |

*Table 6. Tivoli Provisioning Manager for OS Deployment non x86 target operating systems (continued)*

| Non x86 operating system | Version | Architecture |
|---|---|---|
| SUSE Linux Enterprise Server (SLES) | 10, GA and SP 1, 2 , 3 | IBM PowerPC 64 (pSeries) |
| Red-Hat Enterprise Linux (RHEL) Server | 5, GA and Update 1, 2, 3, 4, 5 | IBM PowerPC 64 ( pSeries) |
| Solaris | 10 Update 6 | SPARC 64 bits |
| Solaris | 10 | SPARC 64 bits |
| Solaris | 9 (cloning only) | SPARC 64 bits |
| IBM AIX 5L™ | 5.3 (setup only) | IBM PowerPC 64 (pSeries) |
| IBM AIX 6L | 6.1 (setup only) | IBM PowerPC 64 (pSeries) |

## File systems

Support for Windows Vista/2008/7 is restricted to NTFS partitions. FAT 32 and exFAT partitions are not supported.

Supported file systems include:

- The Third Extended File system (Ext3). Tivoli Provisioning Manager for OS Deployment can natively write files on Ext3. This means that the product can create and format Ext3 partitions, and can add, delete, and modify files on these partitions.
- Partitions UUID and labels
- LVM file system

**Note:**

- The Second Extended File system (Ext2) is not supported anymore. Ext2 partitions are automatically transformed to Ext3 during deployment.
- The product cannot clone, capture an image, or perform direct migration from a target with unformatted partitions, or partitions formatted using a proprietary file system that it does not support. Such partitions should be either deleted, or formatted using a supported file system before cloning, capturing an image, or migrating to another computer.

# Network requirements

To ensure taking full advantage of your network speed when using Tivoli Provisioning Manager for OS Deployment, you must set your network switches to *auto-negotiate*.

If you cannot configure your network, PXE, or DHCP is the manners suggested in the installation guide, you must use network boot media to ensure connection between your targets and your OS deployment server.

## Web interface requirements

To use the web interface, you require a Web browser. Supported Web browsers are:

- Microsoft Internet Explorer versions 6, 7 and 8
- Mozilla Firefox
- Apple Safari

**Note:** Mozilla Firefox 2.0.0.2 does not work on Linux PowerPC 64–bits

# Chapter 2. Installing Tivoli Provisioning Manager for OS Deployment on Windows operating systems

This section describes how to install, uninstall, or upgrade your OS deployment server for standard operations under Windows 2003, and Windows 2008.

## Installation prerequisites

To perform an installation, gather the material you need, ensure that the system environment is set up correctly, and install the prerequisite applications.

For the installation, you need:

- The *installation executable*: The `.exe` file that you received when you purchased Tivoli Provisioning Manager for OS Deployment.

  **Note:** The installation executable must be run as a local Administrator, or as a user with equivalent privileges.
- Either a network boot CD or a *DHCP server* and a computer on which to install Tivoli Provisioning Manager for OS Deployment. Both servers can be on the same computer if wanted.
- To meet the *Database requirements*: Tivoli Provisioning Manager for OS Deployment requires a database (ODBC source) to store information about system profiles and OS deployments.

  **On Windows 32-bit**

  You must use the 32-bit installer. Tivoli Provisioning Manager for OS Deployment provides a database in Microsoft Access format. Even if Microsoft Access is not installed on your system, the necessary drivers (`.mdb` files) should already be present. You can also install an alternative database.

  **On Windows 64-bit**

  You can use either the 32-bit installer, or the 64-bit installer.

  **With the 32-bit installer**

  You must use the ODBC Data Source Administrator 32-bit which must be run with the command `c:\Windows\SysWOW64\odbcad32.exe`. Tivoli Provisioning Manager for OS Deployment provides a database in Microsoft Access format. Even if Microsoft Access is not installed on your system, the necessary drivers (`.mdb` files) should already be present. You can also install an alternative database.

  **Note:** To create a WinPE 3.0 deployment engine, stop the 32-bit web interface extension (rbagent) and start the 64-bit web interface extension.

  **With the 64-bit installer**

  You must use the ODBC Data Source Administrator 64-bit. Tivoli Provisioning Manager for OS Deployment does not provide the database. Before you install the product, you must:

  1. Set up a database.

2. If necessary, install the 64-bit ODBC drivers for your database.
3. Create an `AutoDeploy` source using 64-bit ODBC drivers.

Otherwise the installation will fail.

- Adequate disk space to install the OS deployment server. The folder that will contain all of the disk images managed by Tivoli Provisioning Manager for OS Deployment requires a disk partition of a minimum of 10 Gigabytes.
- At any time, there must be at least 200 MB free on the partition holding the disk images for the OS deployment server to work properly.
- On Windows Server 2008 64-bit, User Access Control must be disabled during installation.
- For FIPS 140-2 compliance, you need to install IBM Global Security Kit version 7d (GSKit 7).
- If you plan to provision Windows operating systems you must have a WinPE 3.0 deployment engine on your OS deployment server. For this reason, install Windows Automated Installation Kit (AIK) for Windows 7 in English on the machine where you want to install the OS deployment server. In this way during the OS deployment server installation, the WinPE deployment engines are automatically created as part of the OS deployment server configuration.

## Installation steps on Windows operating system

On Windows 32-bit operating system, use the 32-bit installer. On Windows 64-bit operating system, you can use either the 32-bit or the 64-bit installer.

To install the OS deployment server:

1. Log on as the local Administrator (or as a user with equivalent privileges).
2. Run the installation executable file. The installation executable is a self-extracting archive. By default, it expands itself in a temporary folder under `c:\install\`. This executable file installs Tivoli Provisioning Manager for OS Deployment. It also performs basic OS configuration tasks from the answers provided, such as setting the administrator name and password.

   **Note:** You cannot use National Language Version (NLV) characters.
3. Follow the setup wizard instructions. Some additional information is available to help you answer the questions of the setup wizard:

   **Language selection**
   Language selection is valid for both the setup wizard and Tivoli Provisioning Manager for OS Deployment.

   **Custom Setup**
   Click **Reset** if you have deselected some components and want to select them all again.

   Click **Disk Usage** to check that you have enough space on your disks to install the product.

   **Database configuration**
   There are three main kinds of configurations:

   - If the Microsoft Access database provided by default (with the 32-bit installer) is applied, there is nothing specific to do. However, if the account is not a local service, some other parameters might be necessary (such as **Database account** and **Gateway account**).

- If the OS deployment server is configured to point to an SQL (trusted) database, the account from the database Gateway (dbgw) is used for authorization. You must fill in **DBGW Account** and **DBGW Password**.
- When the server points to an SQL (untrusted) database, you must fill in the ODBC Database account parameters **Username** and **Password**.

**Data folder**
The default location for the data folder, also known as *data directory*, is `c:\TPMfOS Files\`

**Note:** You cannot use National Language Version (NLV) characters.

**HTTP server console settings**
The default HTTP console port is `8080`. The default HTTPS console port is `443`. The port parameters proposed by default are standard and must be changed only if required.

**DHCP Server OS Configuration**
Installing Tivoli Provisioning Manager for OS Deployment and a DHCP server on the same computer requires some parameterization. If the installer detects that a DHCP server is running, and that changes in the OS configuration are necessary, the panel is displayed. Selecting **Yes** in this dialog allows the installer to perform all the changes described in Chapter 4, "DHCP server configuration," on page 35.

When the installation process is finished, you must have the following new items on your computer:

- A new service called `IBM Tivoli OS Deployment Server`: this is the main component of Tivoli Provisioning Manager for OS Deployment, that offers the PXE remote-boot functionality. This service is automatically started and stopped with the operating system.
- A new service called `IBM Tivoli TCP to ODBC gateway`: this service allows Tivoli Provisioning Manager for OS Deployment targets to use an ODBC source to retrieve their OS configuration and store inventory information in a database. This service is automatically started and stopped when the IBM Tivoli OS Deployment Server service is started and stopped.
- A new service called `IBM Tivoli web interface extension`: this service gives access to different OS components through the web interface of the Tivoli Provisioning Manager for OS Deployment. The web interface extension is automatically started and stopped with the operating system.
- An ODBC source (System DSN) called `AutoDeploy`: this source points to the database used by Tivoli Provisioning Manager for OS Deployment to store information about the OS deployment process, operating system and other software images and computer inventory. This was created only if it did not exist before the installation.
- A new group called `IBM Tivoli Provisioning Manager` in your start menu, with one icon for launching the web interface of the Tivoli Provisioning Manager for OS Deployment.

When the installation is complete, you can start the web interface of the Tivoli Provisioning Manager for OS Deployment by logging onto `http://localhost:8080`, where *localhost* is the hostname of the computer. After logging in, go to **Server > Server status > Installation check** on the web interface. It provides you with a summary of errors encountered during the installation process. If no errors were

encountered during installation, you see messages indicating that *No significant event found in the xxx file in the past 30 min.*

If it is the first time you are working with a OS deployment server, click **Check PXE boot** to become more familiar with the concepts behind Tivoli Provisioning Manager for OS Deployment.

# Silent installation

Tivoli Provisioning Manager for OS Deployment is available as a Microsoft Software Installer (MSI) and can thus be installed silently, using `MSIEXEC`.

## MSIEXEC options

To obtain the complete list of options that you can use with `MSIEXEC`, type `MSIEXEC` in a DOS window. Here are described the three options for installing (or uninstalling) Tivoli Provisioning Manager for OS Deployment silently:

**/qb** Enables msiexec to run quietly with the basic (minimal) user interface. With this option, the setup process does not interact with the user, and either succeeds or fails if all the mandatory parameters are not present.

**/i** Product installation

**/x** Product uninstallation, useful for upgrades.

## Parameters

The parameters described in Table 7 allow you to install Tivoli Provisioning Manager for OS Deployment silently. You can enter these parameters in any order.

*Table 7. Setup parameters*

| Parameter | Default value | Meaning |
|---|---|---|
| `INSTALLDIR` | `C:\Program Files\Common Files\IBM Tivoli` | Server executable files directory |
| `DATADIR` | current directory/`files` | Directory for storing OS deployment files. You need at least 10 GB of free space. |
| `LANG_ID` | `en` | Installation language. For a list of language values, see Table 8 on page 14. |
| `LANG_PROMPT` | Yes | If set, the user is asked to select the language. If not set, the language in `LANG_ID` is used. |
| `HTTP_PORT` | `8080` | Server web interface HTTP port |
| `HTTPS_PORT` | `443` | Server web interface HTTPS port |
| `DISABLE_SSL` | `No` | If set, disables SSL encryption. |
| `ADMIN_NAME` | `admin` | OS deployment server superuser name.<br>**Note:** You cannot use National Language Version (NLV) characters. |
| `NET_PASSWORD` | | OS deployment server superuser password.<br><br>This value is *mandatory*.<br>**Note:** You cannot use National Language Version (NLV) characters. |
| `SERVER_IP` | `127.0.0.1` | IP address of the OS deployment server |

*Table 7. Setup parameters (continued)*

| Parameter | Default value | Meaning |
|---|---|---|
| CONFADDON | | Path to a file that is inserted at the end of the configuration file during installation |
| FORCE_FRESH | No | If set to yes, forces a fresh install by deleting any existing server.db, rembo.conf, and rbagent.conf files before installing.<br><br>When used for web interface extension installation, only rbagent.conf is deleted, as the other files are never used. |
| SERVER_ACCOUNT | | Windows user name for the IBM Tivoli OS Deployment server service<br><br>This user must have **Act as part of the operating system** privileges. Using the System account is often a good choice. |
| SERVER_PASSWORD | | Password to be used for the IBM Tivoli OS Deployment server service |
| ODBC_DSN | AutoDeploy | Name of the ODBC DSN.<br><br>If you want to use an ODBC DSN with a name other than AutoDeploy, you must create the ODBC DSN before you install Tivoli Provisioning Manager for OS Deployment.<br><br>If the ODBC link provided by ODBC_DSN exists (whether AutoDeploy or a custom name), the installer uses it. If the installer cannot find the link given by ODBC_DSN, the installation fails. |
| ODBC_USERNAME | | Name of a user of the ODBC DSN with sufficient privileges to create tables, columns, and so on (database owner). Its format depends on the database used.<br><br>Do not use if you have an SQL trusted database. With an SQL trusted database, use DBGW_ACCOUNT. |
| ODBC_PASSWORD | | Password of the user given in ODBC_USERNAME<br><br>Do not use if you have an SQL trusted database. With an SQL trusted database, use DBGW_PASSWORD. |
| DBGW_ACCOUNT | | Windows user name to be used for the IBM Tivoli TCP to ODBC Gateway service<br><br>Use only if you have an SQL trusted database. Otherwise, use ODBC_USERNAME.<br><br>Select a user name with sufficient access rights to the database. |
| DBGW_PASSWORD | | Password for the IBM Tivoli TCP to ODBC Gateway service<br><br>Use only if you have an SQL trusted database. Otherwise, use ODBC_PASSWORD. |
| RBAGENT_ACCOUNT | | Windows user name for the IBM Tivoli Web Interface Extension service.<br><br>The user should be part of the domain. However, the user does not need to have **Act as part of the operating system** privileges. |
| RBAGENT_PASSWORD | | Password to be used for the IBM Tivoli Web Interface Extension service. |
| DHCP_CONFIG_UPDATE | No | If set, updates the Microsoft DHCP server OS configuration. |

*Table 7. Setup parameters (continued)*

| Parameter | Default value | Meaning |
|---|---|---|
| CREATE_CONSOLE_LINK | Yes | Creates a menu item in the Windows start menu pointing to the login page of the web interface. If set to No the menu item is not created. |
| REMOVE_USER_FILES | No | For *unistallation* only.<br><br>If set, deletes all images on the OS deployment server. |

*Table 8. Language values*

| Language | Value |
|---|---|
| English | en |
| Spanish | es |
| French | fr |
| German | de |
| Italian | it |
| Portuguese | ptBR |
| Korean | ko |
| Japanese | ja |
| Simplified Chinese | zhCN |
| Traditional Chinese | zhTW |

### Installation with command-line examples

To use the following examples, change rbversion-build.msi with the actual name of the Tivoli Provisioning Manager for OS Deployment installation file.

- To perform a silent installation that indicates which .msi files to start and the administrator password, type:

  ```
  MSIEXEC /qb /i "c:\install\rbversion-build.msi" NET_PASSWORD="abcd"
  ```

- To perform a silent installation that indicates which .msi file to open, and a specific superuser name and password and updates the DHCP OS configuration, type:

  ```
  MSIEXEC /qb /i "c:\install\rbversion-build.msi" NET_PASSWORD="abcd"
  ADMIN_NAME="superuser" DHCP_CONFIG_UPDATE="Yes"
  ```

- To completely *uninstall* Tivoli Provisioning Manager for OS Deployment, also removing the images stored on the server, type:

  ```
  MSIEXEC /qb /x "c:\install\rbversion-build.msi" REMOVE_USER_FILES="Yes"
  ```

## Uninstalling Tivoli Provisioning Manager for OS Deployment on Windows operating systems

This section describes how to uninstall your Tivoli Provisioning Manager for OS Deployment software.

You can uninstall a product full build using either the *complete* or *partial* uninstallation option.

**Full build - complete uninstallation**
A *complete* uninstallation removes everything from your system, including the ODBC sources, disk images and registry information.

**Full build - partial uninstallation**
A *partial* uninstallation removes the program and services, but leaves the database and the disk images on your system for future use.

**Note:** Removing a full build without having removed an installed fix pack or interim fix may result in strange side effects.

## Full build - complete uninstallation

To perform a complete uninstallation, you must first uninstall any installed fix pack or interim fix as described in "Fix pack or interim fix uninstallation."

1. From your computer desktop, open **Start > Control Panel > Add/Remove Programs**.
2. Select **IBM Tivoli Provisioning Manager for OS Deployment**
3. Click **Change**. The program `setup.exe` starts.
4. Select the language and click the arrow.
5. Click **Next** in the welcome page.
6. In some cases, Windows asks you to restart the computer.
7. Click the trash icon next to **Remove** from the **Program Maintenance** page.
8. Select **Remove IBM Tivoli Provisioning Manager for OS Deployment completely, including user data** and click **Remove**.

   **Note:** If you are using databases other than Apache Derby, you must manually delete the AutoDeploy entry in the system DSN.
9. Click **Finish** to exit the wizard.

The application and all related information are deleted from your computer.

## Full build - partial uninstallation

To perform a partial uninstallation, you must first uninstall any installed fix pack or interim fix as described in "Fix pack or interim fix uninstallation."

1. From your computer desktop, open **Start > Control Panel > Add/Remove Programs**.
2. Select **IBM Tivoli Provisioning Manager for OS Deployment**
3. Click **Remove**.

In some cases, Windows asks you to restart the computer.

## Fix pack or interim fix uninstallation

To uninstall a fix pack or an interim fix:

1. From your computer desktop, open **Start > Control Panel > Add/Remove Programs**.
2. Select **IBM Tivoli Provisioning Manager for OS Deployment FixPack**
3. Click **Remove**.

In some cases, Windows asks you to restart the computer.

## Upgrading the OS deployment server on Windows

To install a new version, release, fix pack, or interim fix of Tivoli Provisioning Manager for OS Deployment, you must first uninstall, partially or completely, the existing one, due to some operating system limitations related to Universal Unique Identifiers (UUID). If you perform a partial uninstallation, all your data is preserved and available with the new version.

During an upgrade process from one build to another, some fields and some tables can be modified in the database. This modification is performed automatically. Results from this modification are logged and can be viewed in the files `infos.txt` and `details.txt`. Check these files if you want to know if the operation succeeded and which changes were made to your database.

An *upgrade* is the installation of a full build. The upgrade requires the partial uninstallation of the current full build. Due to some operating system limitations, to upgrade to Tivoli Provisioning Manager for OS Deployment 7.1.1 from previous versions, you must uninstall the previous version of the product.

**Note:** If you upgrade the OS deployment server after a partial uninstall of the existing one and you plan to provision Windows operating systems, recreate the WinPE 3.0 deployment engines to include the partitioning WMI support library.

## Upgrading Tivoli Provisioning Manager for OS Deployment

Upgrading your OS deployment server enables you to have the latest available features of the product.

From version 7.1.1.3 of the product, WinPE2 is no longer used. It is displayed in the WinPE 2 Deployment engine folder of the **Software modules** page only for downgrade compatibility. If you plan to provision Windows operating systems you must have a WinPE 3.0 deployment engine on your OS deployment server. To create it automatically, install Windows Automated Installation Kit (AIK) for Windows 7 in English on the machine before upgrading the OS deployment server. In this way during the upgrade, the WinPE 3.0 deployment engines are automatically created.

**Note:** If you upgrade the OS deployment server after a partial uninstall of the existing one and you plan to provision Windows operating systems, recreate the WinPE 3.0 deployment engines to include the partitioning WMI support library.

If you have Linux cloning system profiles 64-bit with LVM partitions, or Linux cloning system profiles with LVM partitions for which you need to preserve logical volume names, you must recreate these profiles. The specific procedure contains steps to be performed before the server upgrade and others to be performed after.

To upgrade your OS deployment server to a new version or release:

1. Perform a partial uninstallation of the product.
2. Run the installation executable file for the new release or version. The installation executable is a self-extracting archive. By default, it expands itself in a temporary folder under `c:\install\`. This executable file installs Tivoli Provisioning Manager for OS Deployment and performs basic installation tasks.
3. Follow the setup wizard instructions.
4. It is recommended to restart your computer.

### Upgrading Linux cloning system profiles with LVM partitions

In some cases, Linux cloning system profiles with LVM partitions, created with version 7.1.1.2 or earlier, need to be recreated during the upgrade of the product to version 7.1.1.3 or later.

If you have

- Linux 64-bit cloning system profiles with LVM partitions
- Linux 32-bit cloning system profiles with LVM partitions and you want to preserve logical volume names

you need to recreate these cloning profiles as part of the upgrade process.

1. Before you upgrade the product:
   a. Export your cloning system profiles into RAD files for backup purposes.
   b. Deploy your cloning system profiles on targets.
   c. Delete these cloning system profiles from the OS deployment server.
2. Upgrade your OS deployment server to version 7.1.1.3.
3. After the upgrade of the OS deployment server, capture the cloning system profiles again from your targets with version 7.1.1.3 of the product.

You can now deploy your recreated Linux cloning profiles with LVM partitions.

## Upgrading in a multiserver infrastructure

When you upgrade OS deployment servers to a newer version of the product in a multiserver infrastructure, you must pay particular attention to the order in which the OS deployment servers are stopped, upgraded and restarted.

Upgrading a multiserver infrastructure in a production environment requires careful planning because all the OS deployment servers need to be stopped during the whole upgrade process. This prevents replication attempts between servers running different versions of the product.

1. Stop all the OS deployment servers, starting from the child servers which are at the bottom of the hierarchy and moving upwards to the top-most parent.
   - To stop the OS deployment servers on Windows operating systems, open a DOS window on each server and type `net stop remboserver`.
2. Upgrade all the OS deployment servers to the same product level, following the upgrade procedure provided for the specific operating system.
3. Restart the OS deployment servers, starting from the top-most parent server and working downwards to the lowest child servers. Before starting a child server, make sure its parent server is ready, for example, the web interface is available.
   - To start the OS deployment servers on Windows operating systems, open a DOS window on each server and type `net start remboserver`.

You have now upgraded all the servers in the multiserver infrastructure. You can start performing tasks on them again.

## Upgrade of OS deployment server database

When you install a new version, release, fix pack or interim fix of Tivoli Provisioning Manager for OS Deployment, all your data is preserved and available with the new version.

During an upgrade the following data is automatically upgraded in the OS deployment server database:

**Deployment engines**
If you plan to provision Windows operating systems you must have a WinPE 3.0 deployment engine on your OS deployment server. To create it automatically, install Windows AIK for Windows 7 in English on the machine before upgrading the OS deployment server. In this way during the upgrade the WinPE 3.0 deployment engines are automatically created.

**Windows Vista/2008/7 setup upgrade**
TheWindows Vista/2008/7 system profiles are introspected and automatically updated.

**System profiles introspection**
The existing system profiles are introspected and automatically upgraded.

**Driver software modules introspection**
The drivers of the existing software modules are introspected and automatically upgraded.

Results from this modification are logged and can be viewed in the files `infos.txt` and `details.txt`. Check these files if you want to know if the operation succeeded and which changes were made to your database.

**Note:**

The OS deployment server is not able to update Windows Vista/2008/7 64-bit unattended setup profiles created with version 7.1.1.1 and lower of the product if the corresponding WinPE2 64-bit ramdisk software module is not present on the server. In this case the icon of the profile is changed to  , the warning message **Profiles are too old.** is issued , and the profile cannot be deployed anymore. To solve this issue:

1. Create a new system profile.
2. Copy the configurations from the old system profile to the new system profile.
3. Delete the old system profile.

## Installation and uninstallation logs on Windows

If the installation or uninstallation of Tivoli Provisioning Manager for OS Deployment on Windows operating systems does not succeed, you can create logs to help you or the appropriate support channel to solve the issue.

To create an installation log, you must try to install Tivoli Provisioning Manager for OS Deployment in command line mode. If you need an uninstallation log, replace `/i` by `/x` in the `msiexec` command line.

1. Open a command prompt window.
2. Navigate to where your MSI file is located.
3. Type

   ```
   msiexec /i <rbversion-build.msi> /l*v install.log
   ```

   where `/i` indicates you are performing an installation, `rbversion-build.msi` is the name of the Tivoli Provisioning Manager for OS Deployment MSI file (it is similar to `rb5.1.03-025.23.msi`), `/l*v` logs all information in verbose mode, and `install.log` is an arbitrary path and file name for the installation log file.

Have this log ready if you need to contact your IBM Software Support representative for installation or uninstallation troubles.

# Chapter 3. Installing Tivoli Provisioning Manager for OS Deployment on UNIX and Linux systems

This section describes how to install, uninstall, or upgrade your OS deployment server for standard operations under UNIX and Linux.

## IBM AIX specific prerequisite

On IBM AIX, the user resource limits are low by default for the root user. Before installing the Tivoli Provisioning Manager for OS Deployment, you might need to raise these limits with the `ulimit` command.

Run

```
ulimit -m unlimited
ulimit -d unlimited
ulimit -s unlimited
```

Tivoli Provisioning Manager for OS Deployment uses UTF-8 for all output. If your locale is not UTF-8 compliant, before running setup, you must either change the character encoding of your terminal session to UTF-8 using the **`xterm -u8`** command line or change your session parameters to a UTF-8 compliant language encoding.

## Database prerequisite

A OS deployment server uses database support to store data and to offer the possibility for other applications to work with those data. The following components should be installed:

**Java** Because Tivoli Provisioning Manager for OS Deployment database gateway is a Java-based application, you need Java installed on the OS deployment server.

**A database**
The suggested database is Apache Derby. Other possible databases are listed in Table 4 on page 2.

## Web interface prerequisite

Name resolution is not always properly set on Linux computers. When trying an HTTP connection to a running OS deployment server on a Linux computer, whether locally or remotely, the web interface could not appear.

OS deployment server redirects an HTTP connection made using an IP address to the name of the running OS deployment server. If the server name cannot be found on the local computer, the connection fails. For instance `HTTP://127.0.0.1` is usually redirected to `HTTP://localhost`.

To solve the problem, you must edit, on the computer running the Web browser, the file containing name resolution information and add the IP address and the name of the OS deployment server you want to connect to. The location of the name resolution file depends on the operating system:

**Linux** `/etc/hosts`

**Windows XP, Windows 2003, and Windows Vista/2008/7**
C:\WINDOWS\system32\drivers\etc\hosts

**Windows 2000 server**
C:\WINNT\system32\drivers\etc\hosts

## Installing an OS deployment server with an Apache Derby database

1. Install and start an Apache Derby server:
   a. Go to the Apache Derby home page (http://db.apache.org/derby), click the **Download** tab, and download the `bin distribution` of the latest version.
   b. Copy the downloaded file into `/usr`, decompress and extract the file contents.
   c. Start the Apache Derby server with the following commands. You might have to adapt filenames and paths depending on your own path and version of the Apache Derby database.

```
export DERBY_HOME=/usr/db-derby-10.2.1.6-bin

startNetworkServer &
```

   **Note:**
   a. Be aware that the Apache Derby database server is not installed by default as a service and that Tivoli Provisioning Manager for OS Deployment *does not* provide scripts to start an Apache Derby database server as a service. Every time you shut down and restart the computer on which the Apache Derby database server is located, you have to restart the database server manually before you can start the OS deployment server.
   b. Ensure that you always start the Apache Derby server from the same directory, otherwise the Apache Derby server might not be able to find back the database required for Tivoli Provisioning Manager for OS Deployment.
2. Decompress and extract the content of the Tivoli Provisioning Manager for OS Deployment `tar.gz` file in `/usr/local`.
3. Run `./setup` in the `/usr/local/tpmfos` directory and follow the setup program instructions.

   `./setup` can be used with parameters. Using parameters when launching `./setup` allows you to provide mandatory and optional parameters in the command line and to run the setup process without user interaction, that is silently.

## Installing with other databases

If you want to install Tivoli Provisioning Manager for OS Deployment with a database other than Apache Derby, follow the instruction in this section.

You can use one of several databases in conjunction with Tivoli Provisioning Manager for OS Deployment, such as MySQL 4.1 and IBM DB2. The list of available databases is presented in Table 4 on page 2.

### MySQL 4.1 example

1. Configure your MySQL 4.1 database:
   a. Run the MySQL database post-installation step:

```
./bin/mysql_install_db --user=mysql
```

   b. Check access rights. The file owner must be the user `mysql`:

```
cd /var/lib/mysql/mysql
chown mysql *
```

c. Start the MySQL server. In case of problems, check `/var/lib/mysql/mysqld.log` file for errors. You can use the startup script typically provided in `/etc/rc.d`:

```
mysqld_safe --user=mysql &
```

**Tip:** The `mysqld.log` file is not in the `/var/lib/mysql` directory for all Linux distributions.

d. Secure the installation by removing the anonymous user, and setting the root password (`xyz`):

```
mysql -u root
delete from mysql.user where user='';
update mysql.user set password=password('xyz');
flush privileges;
quit;
```

e. Require a password for connecting to the database:

```
mysql -u root -pxyz
```

f. (*Optional*) Allow remote access from another computer identified by its IP:

```
grant all on *.* to 'root'@'192.168...' identified by 'xyz';
flush privileges;
```

g. Create a new database for Tivoli Provisioning Manager for OS Deployment

```
create database tpmfosd character set utf8;
```

2. Decompress and extract the content of the Tivoli Provisioning Manager for OS Deployment `tar.gz` file in `/usr/local`. A directory named `tpmfos` is created.
3. Set up your database gateway.

   `dbgw.jar` is a Java implementation of the TCP-to-ODBC database gateway, using JDBC instead of ODBC for database access.

   **Note:** To use ODBC for MySQL database access, ensure you download MySQL Connector/ODBC 3.51.

   a. Download and extract MySQL Connector/J, which you can find at http://dev.mysql.com/downloads/connector/j/3.1.html. The `mysql-connector-java-3.1.8-bin.jar` file is now in the `Connector/J` directory.

   b. Create a symbolic link in the Tivoli Provisioning Manager for OS Deployment installation directory:

```
cd /usr/local/tpmfos
ln -s XXX/mysql-connector-java-3.1.8-bin.jar mysql.jar
```

   c. In another terminal, check the database connectivity:

```
telnet 2020
use mysql://127.0.0.1/tpmfosd?useUnicode=true&characterEncoding=UTF-8,
root,<password>
```

4. Run `./setup` in the `/usr/local/tpmfos` directory.
5. Follow the setup program instructions.

`./setup` can be used with parameters. Using parameters when launching `./setup` allows you to provide mandatory and optional parameters in the command line and to run the setup process without user interaction, that is silently.

## DB2 on AIX example

**Note:** Some paths and parameter values depend on your DB2 installation. Paths and values provided here are only examples.

1. Verify that your AIX 5.3 kernel is 64-bit enabled:

   ```
   bootinfo -K
   ```

   If there are no errors, all the prerequisites are met.
2. Extract the contents of the `.tar.gz` archive for Tivoli Provisioning Manager for OS Deployment on AIX in `/usr`. A directory named `tpmfos` is created under `/usr`.
3. Create a DB2 database. In the remainder of this example, we assume the database is named *tpmfosd*.

   **Note:** If you want to support multiple languages when creating the DB2 database, you must specify a UTF-8 compliant code set. When installing the database on Windows with ODBC, set `DB2CODEPAGE` to 1208: `DB2CODEPAGE=1208`.
4. Create two links in `/usr/tpmfos` for the files `db2jcc.jar` and `db2jcc_license_cu.jar`:

   ```
   cd /usr/tpmfos
   ln -s /opt/IBM/DB2/V9.1/java/db2jcc.jar db2jcc.jar
   ln -s /opt/IBM/DB2/V9.1/java/db2jcc_license_cu.jar db2jcc_license_cu.jar
   ```
5. From the directory where you have extracted the build of Tivoli Provisioning Manager for OS Deployment, start the Java database gateway issuing the following command from a shell prompt:

   ```
   java -cp dbgw.jar:db2jcc.jar:db2jcc_license_cu.jar \
   -Djdbc.drivers=com.ibm.db2.jcc.DB2Driver com.rembo.dbgw.Dbgw -d
   ```
6. In another terminal, verify the database connectivity using:

   ```
   telnet localhost 2020
   use db2://127.0.0.1:50000/tpmfosd,db2inst1,<password>
   ```
7. Stop the `dbgw` process. If you do not, you will encounter errors when running setup.
8. Go to the `/usr/tpmfos` directory.
9. To install Tivoli Provisioning Manager for OS Deployment, start ./setup in the current directory and follow the setup program instructions. Mandatory parameters specific to using the product with DB2 are `CLASSPATH`, `JDBCDRIVER`, and `JDBCURL`. Default values for these parameters may not fit your particular installation of DB2. Make sure to provide appropriate values to the installer to have a working OS deployment server. In this example, you should use the following values:

   ```
   CLASSPATH = /usr/tpmfos/db2jcc.jar:/usr/tpmfos/db2jcc_license_cu.jar
   JDBCDRIVER = com.ibm.db2.jcc.DB2Driver
   JDBCURL = db2://127.0.0.1:50000/tpmfosd
   ```

## Oracle 11i on Linux example

**Note:** Some paths and parameter values depend on you Oracle installation. Paths and values provided here are only examples.

1. Extract the contents of the `.tar.gz` archive for Tivoli Provisioning Manager for OS Deployment on Linux in `/usr/local`. A directory named `tpmfos` is created under `/usr/local`.
2. Create an Oracle database. In the remainder of this example, we assume the database is named *tpmfosd*.

3. From the directory where you have extracted the build of Tivoli Provisioning Manager for OS Deployment, start the Java database gateway, issuing the following command from a shell prompt:

   ```
   java -cp \
   dbgw.jar:/u01/app/oracle/product/11.1.0/db_1/jdbc/lib/ojdbc5.jar:\
   /u01/app/oracle/product/11.1.0/db_1/jdbc/lib/ojdbc6.jar \
   -Djdbc.drivers=oracle.jdbc.driver.OracleDriver com.rembo.dbgw.Dbgw -d
   ```

   If there are no errors, all the prerequisites are met.

4. In another terminal, verify the database connectivity using:

   ```
   telnet localhost 2020
   use oracle:thin:@<hostname>:1521:tpmfosd,system,<password>
   ```

   where <hostname> must be replaced with the actual hostname and <password> with the actual password of the system user.

5. Stop the dbgw process. If you do not, you will encounter errors when running setup.
6. Go to the /usr/local/tpmfos directory.
7. To install Tivoli Provisioning Manager for OS Deployment, start ./setup in the current directory and follow the setup program instructions. Mandatory parameters specific to using the product with Oracle are CLASSPATH, JDBCDRIVER, and JDBCURL. You also need to provide a user name to access the database. Default values for these parameters may not fit your particular installation of Oracle. Make sure to provide appropriate values to the installer to have a working OS deployment server. In this example, you should use the following values:

   ```
   CLASSPATH = /u01/app/oracle/product/11.1.0/db_1/jdbc/lib/ojdbc5.jar:/u01/app/\
   oracle/product/11.1.0/db_1/jdbc/lib/ojdbc6.jar
   JDBCDRIVER = oracle.jdbc.driver.OracleDriver
   JDBCURL = oracle:thin:@<hostname>:1521:tpmfosd
   ```

   where <hostname> must be replaced with the actual hostname.

## Silent installation

You can perform a silent installation of Tivoli Provisioning Manager for OS Deployment by providing setup parameters to the process.

- "Parameter description"
- "Silent installation examples" on page 27

### Parameter description

The setup parameters are described in Table 9, following the order of the questions asked by the interactive setup process. However, you can enter these parameters in any order when used in a command line, such as in a silent installation.

*Table 9. Setup parameters*

| Parameter | Default value | Meaning |
|---|---|---|
| ALWAYSCONTINUE | 0 | If set, setup runs quietly without stopping to ask questions.<br><br>If not set, setup interactively asks for missing parameter information. |
| LANG | en | Installation language. For a list of values, see Table 8 on page 14. |
| INSTALLDIR | current directory | Server executable files directory |

*Table 9. Setup parameters (continued)*

| Parameter | Default value | Meaning |
|---|---|---|
| DATADIR | current directory/files | Directory for storing OS deployment files<br><br>You need at least 10 GB of free space.<br>**Note:** You cannot use National Language Version (NLV) characters. |
| HTTPPORT | 8080 | Server web interface HTTP port |
| USESSL | 1 | Enables SLL on the web interface. |
| HTTPSPORT | 443 | Server web interface HTTPS port |
| ADMINNAME | admin | OS deployment server superuser name<br>**Note:** You cannot use National Language Version (NLV) characters. |
| ADMINPASS | | OS deployment server superuser password<br><br>This value is *mandatory*.<br>**Note:** You cannot use National Language Version (NLV) characters. |
| JAVABIN | Value returned by "which java" | Path to the Java executable |
| various database parameters | | See tables below for parameters specific to the chosen database, either Apache Derby (see Table 10), MySQL 4.1 (see Table 11), or other databases, including DB2 (see Table 12 on page 27). |
| DBUSER | root | Username to connect to the database |
| DBPASS | | Password to connect to the database |
| CREATESCRIPTS | 1 | If set, creates automatic startup scripts |
| STARTSERVICES | 1 | If set, starts services when setup is completed |

*Table 10. Apache Derby database parameters*

| Parameter | Default value | Meaning |
|---|---|---|
| DERBYHOME | | Directory where you can find the Apache Derby .jar files.<br><br>This value is *mandatory*. |
| DBIP | 127.0.0.1 | Apache Derby server IP address |
| DBPORT | 1527 | Apache Derby server port |
| DBNAME | tpmfosd | Apache Derby database name |

*Table 11. MySQL database parameters*

| Parameter | Default value | Meaning |
|---|---|---|
| MYSQLCONNECTORJAR | | Path to the MySQL Connector/J.jar file<br><br>This value is *mandatory*. |
| DBIP | 127.0.0.1 | MySQL server IP address |
| DBPORT | 1527 | MySQL server port |

*Table 11. MySQL database parameters (continued)*

| Parameter | Default value | Meaning |
|---|---|---|
| DBNAME | tpmfosd | MySQL database name |

*Table 12. Other database parameters*

| Parameter | Default value | Meaning |
|---|---|---|
| CLASSPATH | | Classpath to any kind of JDBC connector<br>This value is *mandatory*. |
| JDBCDRIVER | | Full class name of the JDBC driver<br>This value is *mandatory*. |
| JDBCURL | | Full JDBC URL to use to connect to the database<br>This value is *mandatory*. |

### Silent installation examples

- To silently install Tivoli Provisioning Manager for OS Deployment with an Apache Derby database, use the following command line, with only the mandatory parameters:

  ```
  ./setup ALWAYSCONTINUE ADMINPASS="xxxx" DERBYHOME="/usr/local/Derby
  DBPASS=password"
  ```

- For another silent installation with an Apache Derby database and additional parameters, use the following command line:

  ```
  ./setup ALWAYSCONTINUE ADMINPASS="xxxx" DERBYHOME="/usr/local/Derby" LANG="en" \
   DBPASS="yyy" INSTALLDIR="/usr/local/tpmfosd"
  ```

  The language is now set to English explicitly, a password has been given for the database, and the installation directory has been changed from the current directory to `/usr/local/tpmfosd`.

- To silently install Tivoli Provisioning Manager for OS Deployment with a MySQL database, use the following command line:

  ```
  ./setup ALWAYSCONTINUE ADMINPASS="xxxx" MYSQLCONNECTORJAR="/usr/local/
  tpmfos/mysql.jar"
  ```

## Advanced features

Some more advanced features are provided under this heading.

## Startup scripts

Startup scripts for the Java database gateway, for the OS deployment server, and for the web interface extension are provided for Linux (SuSE, RedHat), AIX and Solaris. They are located in the `tpmfos/scripts` directory. For Linux and AIX, the scripts are copied during installation to the startup scripts directory of the platform. You might have to adapt variables in these scripts to adapt them to your OS deployment server installation paths.

**Note:** Scripts generated for SLES 10 are valid only if you have installed the latest updates for this operating system.

Scripts for the Apache Derby database server *are not* provided and you need to create them yourself in order to completely automate the startup process of your OS deployment server. If you do not create startup scripts for the database server, the OS deployment server (for which scripts have been created during setup) does not start correctly after a reboot or a shutdown/restart of the computer on which it is installed. You need to stop your OS deployment server and start or restart the services in the order described in Table 13.

*Table 13. Correct order for starting services*

| Starting order | Service | Script provided |
|---|---|---|
| 1 | Apache Derby database server (or your other database server) | No |
| 2 | Java database gateway | Yes |
| 3 | OS deployment server | Yes |
| 4 | Web interface | Yes |

To start the services using the scripts provided on a RedHat server, for instance, type

```
/etc/init.d/dbgw start
/etc/init.d/rembo start
/etc/init.d/rbagent start
```

To stop the same services, use the reverse order. For instance, using the scripts provided on a RedHat server, type

```
/etc/init.d/rbagent stop
/etc/init.d/rembo stop
/etc/init.d/dbgw stop
```

## Starting the OS deployment server

**Important:** Your database server and your Java database gateway must be running.

You can start the OS deployment server from the command prompt:

- In daemon mode (the process is detached. The log file can be found in the server log directory), type:

  ```
  ./rembo
  ```

  This is the standard mode.
- In debug mode (with debug log on the web interface), type:

  ```
  ./rembo -d -v 3
  ```

  This mode must be used if you are experiencing difficulties and need to troubleshoot them, or report information to support, or both.

You can automate the starting and stopping of services.

## PAM configuration (optional)

The OS deployment server gives you ways to authenticate users on specific authentication domains. Tivoli Provisioning Manager for OS Deployment is PAM enabled. You can use any PAM module according to your authentication scheme. If you do not configure PAM, Tivoli Provisioning Manager for OS Deployment will fall back to a classic UNIX authentication scheme (read `/etc/passwd` and `/etc/shadows`). To enable PAM, add a file named `rembo` in `/etc/pam.d` (This location can change depending on your UNIX distribution.)

For authentication on a local UNIX computer, the `rembo` file can contain the following lines, for example:

```
auth    required    /lib/security/pam_unix_auth.so
account required    /lib/security/pam_unix_acct.so
```

For authentication through a remote LDAP server, the `rembo` file can contain the following lines, for example:

```
auth    required    /lib/security/pam_ldap.so
account required    /lib/security/pam_ldap.so
```

For SuSE Linux Enterprise Server, use:

```
auth    include common-auth
account include common-account
```

## Accessing the user interface with Active Directory domain users

This procedure is used to authenticate users against the Active Directory server when the OS deployment server server is installed on a Linux computer.

Perform the following steps on the Linux computer where the OS deployment server is installed to authenticate Tivoli Provisioning Manager for OS Deployment users on a domain located on the Active Directory server. The following procedure was verified on a Red Hat Enterprise Linux 5 computer.

1. Verify that the Linux computer points to the DNS used by Active Directory. If not, edit the *nameserver <DNS IP>* parameter in the`/etc/resolv.conf` file.
2. Ensure that the time settings on both the Linux and Windows computers are aligned. If the time settings differ more than five minutes, then the Linux computer will not be able to join the Active Directory domain.
3. Ensure that the following packages are installed on the Linux computer:
   - pam
   - nss
   - samba-client
   - samba-common
4. Launch the Linux configuration wizard system-config-authentication.
5. Ensure that the **Enable Winbind Support** option is selected on both the **User Information** and **Authentication** tabs.
6. Click **Configure Winbind** on either one of the tabs and set the following options:
   - **Winbind Domain**: The name of the domain you want to bind.
   - **Security Model**: ads
   - **Winbind ADS Realm**: DNS domain name of the AD domain.

- **Winbind Domain Controllers**: The name of a Domain Controller against which you want the Linux computer to authenticate (or * if it has to be retrieved from DNS).
- **Template Shell**: The login shell to access the Linux computer with Active Directory users.

7. To create a user home dir on the Linux computer at the first login of the Active Directory user, add the following line before the last line in the `/etc/pamd./system-auth` file:

   ```
   session optional map_mkhomedir.so skel=/etc/skel umask=0644
   ```

8. Edit the Global Settings section in the `/etc/samba/smb.conf` file as follows:
   - To remap the Windows users' SID to the Linux users' UID (so that each Active Directory user has the same UID on any Linux computer), add the following line:

     ```
     idmap backend = rid
     ```

   - To define the user's home dir, add the following line:

     ```
     template homedir = /home/%U
     ```

   - To use the default domain, modify the following parameter as follows:

     ```
     winbind use default domain = true
     ```

9. Launch the following command to add the Linux computer to the domain:

   ```
   net ads join -U AD <administrator username>
   ```

   where <*AD administrator username*> is the user name of an account that has privileges to join a machine to the domain. After successful completion of the join, run the following commands:
   - `wbinfo -t`: to test the trust relationship to the domain
   - `wbinfo -u`: to list all the users in the domain
   - `wbinfo -g`: to list all the groups in the domain

   Active Directory users, as well as local operating system users, can now authenticate on the Linux computer.
10. To enable these same users to log in to the Tivoli Provisioning Manager for OS Deployment user interface, copy the entire contents of, `/etc/pam.d/system-auth`, to the path, `/etc/pam.d/rembo`, to extend the Pluggable Authentication Module (PAM) to the Tivoli Provisioning Manager for OS Deployment application. If you only want authentication for Active Directory users, then entries related to the "`pam_unix.so`" library must be removed from the rembo file.
11. Create an HTTP authentication domain and then create the appropriate security roles for users.
    a. Create an HTTP authentication domain on the OS deployment server to allow authentication of local and Active Directory users on the Tivoli Provisioning Manager for OS Deployment user interface. See the topic in the information center about "*Security roles*".
    b. Create the security roles for the specific local or Active Directory users to allow authorization of Active Directory users on the Tivoli Provisioning Manager for OS Deployment user interface.

Specific local and Active Directory users can now access the Tivoli Provisioning Manager for OS Deployment user interface.

## Uninstalling Tivoli Provisioning Manager for OS Deployment on UNIX

For complete uninstallation of Tivoli Provisioning Manager for OS Deployment on a UNIX computer, perform the following steps.

**Note:** Unlike on Windows, there is no partial uninstallation on UNIX.

1. Stop all daemons or services related to the product
2. Delete the directory in which the Tivoli Provisioning Manager for OS Deployment files are located
3. Delete the database.
   - To delete an Apache Derby database, use the following command:

     ```
     rm -rvf $DERBY_HOME
     ```

     where `$DERBY_HOME` is the home directory of the Apache Derby server.
   - To delete a MySQL database, use the following command:

     ```
     mysql -u username -ppassword
     drop database tpmfosd;
     quit;
     ```

     where `username` and `password` are the user name and password required to access the database named `tpmfosd`.

**Note:** If you are using databases other than Apache Derby, you must manually delete the AutoDeploy entry in the system DSN.

## Upgrading Tivoli Provisioning Manager for OS Deployment on UNIX

An *upgrade* is a full build file replacement after having stopped the dbgw, rembo, and rbagent daemons.

**Note:** If you upgrade the OS deployment server after a partial uninstall of the existing one and you plan to provision Windows operating systems, recreate the WinPE 3.0 deployment engines to include the partitioning WMI support library.

### Upgrading Tivoli Provisioning Manager for OS Deployment

From version 7.1.1.3 of the product, WinPE2 is no longer used. It is displayed in the WinPE 2 Deployment engine folder of the **Software modules** page only for downgrade compatibility. If you plan to provision Windows operating systems you must have a WinPE 3.0 deployment engine on your OS deployment server. To create it automatically, install Windows Automated Installation Kit (AIK) for Windows 7 in English on the machine before upgrading the OS deployment server. In this way during the upgrade, the WinPE 3.0 deployment engines are automatically created.

**Note:** If you upgrade the OS deployment server after a partial uninstall of the existing one and you plan to provision Windows operating systems, recreate the WinPE 3.0 deployment engines to include the partitioning WMI support library.

If you have Linux cloning system profiles 64-bit with LVM partitions, or Linux cloning system profiles with LVM partitions for which you need to preserve logical volume names, you must recreate these profiles. The specific procedure contains steps to be performed before the server upgrade and others to be performed after.

To install a Tivoli Provisioning Manager for OS Deployment upgrade on UNIX computers, follow these steps:

1. If they are running, stop the rbagent, rembo and dbgw daemons.
2. Extract the upgrade `.tar.gz` file on top of the previous version of Tivoli Provisioning Manager for OS Deployment.
3. Restart the dbgw, rembo and rbagent daemons.

**Note:** *Do not* run `./setup` again to prevent losing your `rembo.conf` configuration.

### Upgrading Linux cloning system profiles with LVM partitions

In some cases, Linux cloning system profiles with LVM partitions, created with version 7.1.1.2 or earlier, need to be recreated during the upgrade of the product to version 7.1.1.3 or later.

If you have

- Linux 64-bit cloning system profiles with LVM partitions
- Linux 32-bit cloning system profiles with LVM partitions and you want to preserve logical volume names

you need to recreate these cloning profiles as part of the upgrade process.

1. Before you upgrade the product:
   a. Export your cloning system profiles into RAD files for backup purposes.
   b. Deploy your cloning system profiles on targets.
   c. Delete these cloning system profiles from the OS deployment server.
2. Upgrade your OS deployment server to version 7.1.1.3.
3. After the upgrade of the OS deployment server, capture the cloning system profiles again from your targets with version 7.1.1.3 of the product.

You can now deploy your recreated Linux cloning profiles with LVM partitions.

## Upgrading in a multiserver infrastructure

When you upgrade OS deployment servers to a newer version of the product in a multiserver infrastructure, you must pay particular attention to the order in which the OS deployment servers are stopped, upgraded and restarted.

Upgrading a multiserver infrastructure in a production environment requires careful planning because all the OS deployment servers need to be stopped during the whole upgrade process. This prevents replication attempts between servers running different versions of the product.

1. Stop all the OS deployment servers, starting from the child servers which are at the bottom of the hierarchy and moving upwards to the top-most parent.
   - To stop the OS deployment servers on UNIX operating systems, use the scripts provided.
2. Upgrade all the OS deployment servers to the same product level, following the upgrade procedure provided for the specific operating system.
3. Restart the OS deployment servers, starting from the top-most parent server and working downwards to the lowest child servers. Before starting a child server, make sure its parent server is ready, for example, the web interface is available.
   - To start the OS deployment servers on UNIX operating systems, use the scripts provided.

You have now upgraded all the servers in the multiserver infrastructure. You can start performing tasks on them again.

## Upgrade of OS deployment server database

When you install a new version, release, fix pack or interim fix of Tivoli Provisioning Manager for OS Deployment, all your data is preserved and available with the new version.

During an upgrade the following data is automatically upgraded in the OS deployment server database:

**Windows Vista/2008/7 setup upgrade**
TheWindows Vista/2008/7 system profiles are introspected and automatically updated.

**System profiles introspection**
The existing system profiles are introspected and automatically upgraded.

**Driver software modules introspection**
The drivers of the existing software modules are introspected and automatically upgraded.

Results from this modification are logged and can be viewed in the files `infos.txt` and `details.txt`. Check these files if you want to know if the operation succeeded and which changes were made to your database.

**Note:**

The OS deployment server is not able to update Windows Vista/2008/7 64-bit unattended setup profiles created with version 7.1.1.1 and lower of the product if the corresponding WinPE2 64-bit ramdisk software module is not present on the server. In this case the icon of the profile is changed to  , the warning message **Profiles are too old.** is issued , and the profile cannot be deployed anymore. To solve this issue:

1. Create a new system profile.
2. Copy the configurations from the old system profile to the new system profile.
3. Delete the old system profile.

# Chapter 4. DHCP server configuration

The DHCP server is used by the PXE bootrom to get its IP address and other basic networking information (including subnet mask, and default gateway). Using Tivoli Provisioning Manager for OS Deployment can require changes to your DHCP configuration. These changes can typically be performed automatically by the Tivoli Provisioning Manager for OS Deployment installer. However, in some cases, you might want to perform the changes manually, or to verify them.

You can configure your DHCP server for one of the three following situations:

- The DHCP server and the OS deployment server *are not* running on the same host
- The DHCP server and the OS deployment server *are* running on the same host
- You already have a PXE 2.0 infrastructure with PXE Boot Server discovery installed and you want to add Tivoli Provisioning Manager for OS Deployment to the list of servers to discover

**Note:** If you have previously configured your DHCP server for another PXE bootstrap, do not reuse your existing DHCP configuration. Remove DHCP options 43 & 60 for the hosts on which you want to run Tivoli Provisioning Manager for OS Deployment and follow the instructions given in this section (if you are running Tivoli Provisioning Manager for OS Deployment on the same host as the DHCP server, you need to set option 60 again).

**Note:** There are also cases where you must set both DHCP options 43 & 60, including when you have two different OS deployment server .

## DHCP server and OS deployment server on different targets, without information on PXE server location

Actions to perform:

- If DHCP options 43 and 60 are set, remove them.
- If the DHCP server *is not* running on the same computer as the OS deployment server, the DHCP configuration does not change. The OS deployment server detects DHCP packets sent over the network by PXE bootroms and offers PXE parameters without disturbing standard DHCP negotiation process. This behavior is called DHCPProxy.

## DHCP server and OS deployment server on different targets, with information on PXE server location

Actions to perform:

- Set option 60 (Class identifier) to "PXEClient" to inform the target that the location of the PXE server is known.
- Set option 43 to indicate that the PXE server does not reside on the same computer as the DHCP server and to precise the location of the PXE server.

## Configuring the DHCP server

You can configure your DHCP server in several different ways.

The following options are available:

- The DHCP server and the OS deployment server *are not* running on the same host
- The DHCP server and the OS deployment server *are* running on the same host
- You already have a PXE 2.0 infrastructure with PXE Boot Server discovery installed and you want to add Tivoli Provisioning Manager for OS Deployment to the list of servers to discover

**Note:**

- If you have previously configured your DHCP server for another PXE bootstrap, do not reuse your existing DHCP configuration. Remove DHCP options 43 & 60 for the hosts on which you want to run Tivoli Provisioning Manager for OS Deployment and follow the instructions given in this section (if you are running Tivoli Provisioning Manager for OS Deployment on the same host as the DHCP server, you need to set option 60 again).
- There are also cases where you must set both DHCP options 43 & 60, including when you have two different OS deployment server .

Select the appropriate situation and then perform the configuration steps:

- DHCP server and OS deployment server on different computers, without information on PXE server location

  Actions to perform:

  – If DHCP options 43 and 60 are set, remove them.

  – If the DHCP server *is not* running on the same computer as the OS deployment server, the DHCP configuration does not change. The OS deployment server detects DHCP packets sent over the network by PXE bootroms and offers PXE parameters without disturbing standard DHCP negotiation process. This behavior is called DHCPProxy.
- DHCP server and OS deployment server on different computers, with information on PXE server location.

  Actions to perform:

  – Set option 60 (Class identifier) to "PXEClient" to inform the target that the location of the PXE server is known.

  – Set option 43 to indicate that the PXE server does not reside on the same computer as the DHCP server and to precise the location of the PXE server.
- DHCP server and OS deployment server on the same computer.

  Actions to perform:

  – If DHCP option 43 is set, remove it

  – Set option 60 (Class identifier) to "PXEClient" to inform the target that the location of the PXE server is known.

  If you are planning to run the DHCP server and the OS deployment server on the same computer, you must tell your DHCP server to send DHCP option 60 (Class identifier) to the targets. When option 60 is set to `PXEClient` the DHCP server knows where the PXE server is. If option 43 is not set, the PXE server has the same IP address as the DHCP server.

## Adding Tivoli Provisioning Manager for OS Deployment to an existing Boot Discovery infrastructure

If your network is already configured for PXE 2.0 Boot Server Discovery, just add Tivoli Provisioning Manager for OS Deployment to your boot menu. The identifier of the Tivoli Provisioning Manager for OS Deployment Server is `15`. If you want to use multicast discovery, use the multicast IP address `232.1.0.1`.

**Note:**

1. By default, the OS deployment server responds to all requests, including those originating from unknown targets.
2. If the flag *Let unknown computers contact another PXE server* is set, the server only responds to discovery requests originating from known targets.

## DHCP option 60

Option 60 (Class identifier) allows you to inform the target that the location of the PXE server is known.

### Adding DHCP option 60 to Windows 2003 DHCP server

By default, option 60 is not set on Windows 2003. If the OS deployment server is running on the same host as the DHCP server, you have to add this option and set its value to `PXEClient` in order to tell PXE clients where to find the OS deployment server.

Follow these steps to add option 60 on Windows 2003:

1. Open a command window (select **Start > Run > cmd**)
2. Type `netsh`
3. Type `dhcp`
4. Type `server \\<servername>` or `server <ip_address>`
5. You then see a command prompt that says: `dhcp server>`
6. Type `add optiondef 60 PXEClient STRING 0 comment=option added for PXE support`
7. Type `set optionvalue 60 STRING PXEClient`
8. To confirm that everything has been set correctly, type `show optionvalue all`

### Adding DHCP option 60 to a host with ISC DHCP server

If you are using the ISC DHCP server 2.0, you can add the DHCP option 60 to a group of targets or to a single target by adding the statement `option dhcp-class-identifier "PXEClient";` to a section of the configuration file. If you were using option 43 (vendor-encapsulated-options) for another PXE application, remove it for Tivoli Provisioning Manager for OS Deployment targets.

The modifications to perform on a ISC DHCP server 3.0 are the same as for a 2.0 server, but the names differ:

- Add `vendor-class-identifier "PXEClient";` for the targets running Tivoli Provisioning Manager for OS Deployment
- Remove any occurrences of `option space PXE;` if you were running another PXE application

**Note:**

1. The OS deployment server responds to all requests, including requests originating from unknown targets.
2. If the flag **Completely ignore unknown targets** is set for the server, it only responds to discovery requests originating from known targets. You can specify either the IP address or the Ethernet address in the target list. At this stage the IP address of the remote-boot target is known.

# DHCP option 43

Option 43 is a binary buffer, containing a list of sub-options. Sub-options are packed in the binary buffer in no specific order. Most sub-options are optional.

An exhaustive list of sub-options can be found in the PXE specifications. Only sub-options of interest to specify the IP address of the PXE server are described here. Other configurations, like multicast discovery, multiple unicast servers, or multiple choices, are not explained in this section.

**PXE option 6: PXE_DISCOVERY_CONTROL**
: This option specifies how the PXE client contacts PXE servers, using either unicast, multicast or broadcast. The format of this option is one byte.

**PXE option 8: PXE_BOOT_SERVERS**
: A list of IP addresses, each address corresponding to one PXE server (when `discovery_control` is unicast). A PXE server is identified by a number (the standard value for Tivoli Provisioning Manager for OS Deployment is 15) and its IP address. The format of this option is two bytes for the server type (15 for Tivoli Provisioning Manager for OS Deployment), one byte for the number of servers to list (1 in our example), and four bytes per server address.

**PXE option 9: PXE_BOOT_MENU**
: This option contains a list of choices to prompt on the screen at boot time. You need to have a PXE boot menu even if it is not used. The format of this option is two bytes for the server type, one byte for the length of the string to display, and the string to display on screen. The total length of this option is be 5 bytes.

**PXE option 10 (0A): PXE_BOOT_TIMEOUT**
: This option is required to specify how long (in seconds) the boot menu is displayed, and the text of a prompt that is displayed during waiting time. The format of this option is one byte for the timeout value, followed by the prompt text.

**PXE option 255 (FF): PXE_END**
: To be valid, the binary buffer of DHCP option 43 must end with `FF`.

## Setting DHCP option 43

If your DHCP server is running on Windows NT, you can enter the suboption values one at a time in option 43, by selecting hexadecimal input.

If your DHCP server is ISC DHCP (version 2.x), then you can enter the suboption values as provided in the examples (bytes separated with colons) for parameter `vendor-encapsulated-options` (the exact name depends on the version you are using).

If your DHCP server is ISC DHCP (version 3.x), then you can use the explicit syntax to describe the PXE options, as follows:

```
# In the global section:
   option space PXE;
   option PXE.discovery-control code 6 = unsigned integer 8;
   option PXE.boot-server code 8 = { unsigned integer 16,
                                     unsigned integer 8,
                                     ip-address };
   option PXE.boot-menu code 9 = { unsigned integer 16,
                                   unsigned integer 8,
                                   text};
   option PXE.menu-prompt code 10 = { unsigned integer 8, text };

# In the scope/host section:
   option dhcp-parameter-request-list = concat(option dhcp-parameter-request-list,60,43);
   option vendor-class-identifier "PXEClient";
   vendor-option-space PXE;
   option PXE.discovery-control 7;
   option PXE.boot-menu 15 5 "Rembo";
   option PXE.menu-prompt 0 "Rembo";
```

## Example: option 43 for PXE servers on different subnets

In this example, you want to have targets A and B boot on server 192.10.10.10, and targets C and D boot on server 192.10.11.10, which is on a different subnet (a valid gateway must be setup for C and D in order to locate the PXE server on a different subnet).

**Note:** Server type 15 is translated into 00:0F in hexadecimal. IP address 192.10.10.10 is translated into C0:0A:0A:0A, and 192.10.10.11 is translated into C0:0A:0B:0A. Letters R and B are translated into 52 and 42.

Here are the options that one must insert in the binary buffer and their values:

**PXE option 6, length 1, value=07**
: Value 7 means use PXE_BOOT_SERVERS list, disable multicast and broadcast discovery

**PXE option 8, length 7, value = 00:0F:01:C0:0A:0A:0A**
: (targets A and B) Only one IP address is used, the address of the PXE server for the target which receives these DHCP options.

**PXE option 8, length 7, value = 00:0F:01:C0:0A:0B:0A**
: (targets C and D) Only one IP address is used, the address of the PXE server for the target which receives these DHCP options.

**PXE option 9, length 5, value = 00:0F:02:52:42**
: There is only one line, displaying RB, and which goes to server type 15 (Tivoli Provisioning Manager for OS Deployment).

**PXE option A, length 2, value=00:52**
: The timeout is set to 0 seconds, meaning that one wants to boot using the first line of the boot menu ,and the prompt is set to R.Because the timeout is 0, the prompt text is not displayed, but it must be at least one character long.

**PXE option FF**
: This closes the buffer

The format of the binary buffer is similar to what is used for the DHCP packet itself. The buffer is a list of options, each option starting with its option number (one byte), followed by its length (one byte), and its value (a list of bytes).

Here is the transcription of the above example, for targets A and B :

```
Option 43 =
06:01:07:08:07:00:0F:01:C0:0A:0A:0A:09:05:00:0F:02:52:42:0A:02:00:52:FF
```

And for targets C and D :

```
Option 43 =
06:01:07:08:07:00:0F:01:C0:0A:0B:0A:09:05:00:0F:02:52:42:0A:02:00:52:FF
```

## Example: option 43 to create a PXE boot menu

The administrator wants to display two lines in the PXE boot menu, pointing to two separate OS deployment servers. The two servers must use different PXE server type numbers or they will be seen as only one server by the PXE network card.

In addition to the standard PXE server type 15, OS deployment server accept any number between 33008 (80F0 in hexadecimal) and 33280 (8200 in hexadecimal). These new PXE server type numbers are used to differentiate multiple OS deployment servers in the BOOT_SERVERS sub-option of DHCP option 43.

In this example, the first server has the address 192.168.1.4 (C0:A8:01:04 in hexadecimal), and the second server, 192.168.1.5 (C0:A8:01:05 in hexadecimal).

1. Assign an OS deployment server type to each of the servers. OS deployment servers accept server type 15, and server types from 33008 to 33280. For this example, 33008 (80F0) is used for the first server, and 33009 (80F1) for the second server.
2. The sub-options of DHCP option 43 must then be configured as follows:

   **PXE option 6, length 1,value = 07**
   Value 7 means use PXE_BOOT_SERVERS list, disable multicast and broadcast discovery

   **PXE option 8, length 14 (0E), value = 80:F0:01:C0:A8:01:04:80:F1:01:C0:A8:01:05**
   Option 8 defines the two PXE servers. The first server is defined by the first 7 bytes, starting with the OS deployment server type (80:F0, 33008), followed by one IP address: C0:A8:01:05 (192.168.1.4). The second server is defined in the following manner, using OS deployment server type 80:F1 (33009).

   **PXE option 9, length 22 (16), value = 80:F0:08:53:65:72:76:65:82:20:41:80:F1:08:53:65:72:76:65:82:20:42**
   Option 9 defines the boot menu that is displayed at boot time. The first 11 bytes correspond to the first line, for the first server. It shows the string Server A, associated with type 80:F0 (first server). The second line shows the string Server B, associated with type 80:F1 (second server).

   **PXE option A, length 6, value =0F:53:65:6C:65:63:74**
   Option 10 (0A) specifies a 15 second timeout and shows the string Select as the boot menu prompt.

**PXE option FF**
To close the buffer.

The full option 43 reads as follow:

```
06:01:07:08:0E:80:F0:01:C0:A8:01:04:80:F1:01:C0:A8:01:05:
09:16:80:F0:08:53:65:72:76:65:82:20:41:80:F1:08:53:65:72:76:65:82:20:42:
0A:06:0F:53:65:6C:65:63:74:FF
```

When your boot menu is displayed on your target screen, press F8 to make your selection.

## Additional Linux cloning options

To be able to clone Linux targets, you need to have the following options in your DHCP server configuration

**Option 1 (also known as `subnet-mask` under ISC DHCP server)**
Set this option to your subnet mask.

**Option 6 (also known as `domain-name-servers` under ISC DHCP server)**
Set this option to the IP address of your domain name server.

**Option 15 (also known as `domain-name` under ISC DHCP server)**
Set this option to an non-empty string indicating your domain name.

Without them, the targets can boot into PXE, but a Linux cloning profile cannot be created.

These options must be manually added in `dhcpd.conf` for ISC DHCP servers.

## Additional SUN and IBM PowerPC options

To boot SUN and PowerPC targets using Tivoli Provisioning Manager for OS Deployment, you can use DHCP instead of the old-fashioned RARP/bootparams method, which does not cross routers.

You need to add the following options in your DHCP server configuration:

**Option 66 (known as `next-server` under ISC DHCP server and `tftp-server-name` under Solaris DHCP server)**
Set this option to the IP address of your OS deployment server.

**Option 67 (known as `filename` under ISC DHCP server and `bootfile-name` under Solaris DHCP server)**
Set this option to the boot file name `rembo.fcode`.

For Microsoft DHCP server, you also have to setup

- Vendor class `DHCP Standard Options`
- User Class `Default BOOTP class`

For ISC DHCP server, you also have to setup

- `allow booting`
- `allow bootp`

## dhcpd.conf example

Here is an example for a configuration file for an ISC DHCP server 3.0.

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "example.org";

option domain-name-servers ns1.example.org, ns2.example.org;
default-lease-time 600;
max-lease-time 7200;

option subnet-mask 255.255.255.0;

option space PXE;
option PXE.discovery-control code 6 = unsigned integer 8;
option PXE.boot-server code 8 = {
       unsigned integer 16, unsigned integer 8, ip-address};
option PXE.boot-menu code 9 = { unsigned integer 16, unsigned integer 8, text};
option PXE.menu-prompt code 10 = {unsigned integer 8, text};

# if you do not use dynamical DNS updates:
#
# this statement is needed by dhcpd-3 needs at least this statement.
# you have to delete it for dhcpd-2, because it does not know it.
#
# if you want to use dynamical DNS updates, you should first read
# read /usr/share/doc/packages/dhcp-server/DDNS-howto.txt
ddns-update-style none; ddns-updates off;


# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

allow booting;
allow bootp;


subnet 10.0.0.0 netmask 255.255.255.0 {
option subnet-mask 255.255.255.0;
option broadcast-address 10.0.0.255;
default-lease-time 6000;
max-lease-time 6000;

# this is the section for the x86
host ibmx3655  {
hardware ethernet 00:14:5E:5A:31:57;
fixed-address 10.0.0.200;
option dhcp-parameter-request-list 1,6,15,60,43;
option subnet-mask 255.255.255.0;
option domain-name-servers 10.0.0.20;
option domain-name "site";
option vendor-class-identifier "PXEClient";
  vendor-option-space PXE;
    option PXE.discovery-control 7;
    option PXE.boot-server 15 1 10.0.0.20;
    option PXE.boot-menu 15  15 "Tpm for OSd 7.1";
    option PXE.menu-prompt 0 "Tpm for Osd";
}
```

```
# this is the section for the x86
host x41  {
hardware ethernet 00:0a:e4:2f:66:38;
fixed-address 10.0.0.201;
option subnet-mask 255.255.255.0;
option domain-name-servers 10.0.0.20;
option domain-name "site";
  option vendor-class-identifier "PXEClient";
  vendor-option-space PXE;
    option PXE.discovery-control 7;
    option PXE.boot-server 15 1 10.0.0.20;
    option PXE.boot-menu 15  15 "Tpm for OSd 7.1";
    option PXE.menu-prompt 0 "Tpm for Osd";
}

# this is the section for the pSeries
host ibmpseries  {
hardware ethernet 00:09:6b:ab:0e:f2;
fixed-address 10.0.0.141;
 next-server 10.0.0.20;    # IP address of the OS deployment server
filename "rembo.fcode";    # the name is not important, but it must not be empty
}

# this is the section for the SUN
host sunTarget1  {
hardware ethernet 00:03:ba:92:92:f0;
fixed-address 10.0.0.142;
option routers 10.0.0.15;
 next-server 10.0.0.20;   # IP address of the OS deployment server
 filename "rembo.fcode";
}
 range 10.0.0.100 10.0.0.220;
}
```

# Chapter 5. Prerequisites for provisioning Windows

Since version 7.1.1 of the product, a WinPE deployment engine is required to create or to deploy any Windows system profile. The current required version of WinPE is WinPE 3.0.

**Note:** From version 7.1.1.3 of the product onwards, WinPE2 is not supported anymore.

The WinPE 3.0 deployment engine is a group of files extracted from Windows Automated Installation Kit (AIK) for Windows 7 in English. It can be viewed as a subset of Windows Vista/2008/7.

You need at least one WinPE 3.0 32-bit deployment engine and one WinPE 3.0 64-bit deployment engine.

WinPE 3.0 deployment engines are objects of their own category in the OS deployment server. They are stored under **Server** > **Advanced features** > **Deployment engines**.

## Uses of the WinPE 3.0 deployment engine

**Formatting disks and partitions**
When you format disks and partitions with the product, the WinPE 3.0 deployment engine is sent to the target and started. It is the WinPE 3.0 deployment engine which performs the actual formatting.

**Cloning targets**
When you clone a target, the WinPE 3.0 deployment engine is sent to the target and started. It is the WinPE 3.0 deployment engine which sends the files necessary to create the cloning system profile to the OS deployment server.

**Deploying a system profile**
When you deploy a system profile, whether unattended setup or cloning, the WinPE 3.0 deployment engine is sent to the target and started. It is the deployment engine which performs the actual copying of files and operating system installation.

These operations are performed faster and more reliably using the WinPE deployment engine than in previous versions when files were transferred using the BIOS. WinPE uses network and disk drivers which are provided by the target vendor and which are optimized for speed.

## Create your WinPE 3.0 deployment engine

From version 7.1.1.3 of the product, you cannot use WinPE2 anymore. You must have a WinPE 3.0 deployment engine to be able to continue provisioning Windows operating systems.

The OS deployment server can automatically create a WinPE 3.0 deployment engine when its service starts, if

- The OS deployment server runs under a Windows operating system.

- The OS deployment server and the Windows operating system under which it runs share the same architecture, either 32-bit or 64-bit.
- The web interface extension is also running.
- You have installed Windows AIK for Windows 7 in English on the same server.
- You do not yet have a WinPE 3.0 deployment engine on your OS deployment server.

If one of these conditions is not fulfilled, you have to create your WinPE 3.0 deployment engine explicitly.

# Chapter 6. Prerequisites for provisioning Solaris

There are a few additional prerequisites to provision Solaris operating systems.

If you intend to deploy Solaris operating systems, you must understand the interaction between Jumpstart and the OS deployment server and be aware of a few additional prerequisites.

- Additional DHCP options.
- A Solaris install server

## Jumpstart and the OS deployment server

Tivoli Provisioning Manager for OS Deployment builds on the standard mechanism developed and maintained by SUN to install the Solaris operating system. What Tivoli Provisioning Manager for OS Deployment provides is:

- A web-based management console for managing Solaris OS deployments, integrated with similar functions for other operating systems such as Linux and Windows.
- A simplified means of configuring target-specific parameters, stored in a central database, without having to play commands on a specific OS deployment server and to edit obscure configuration files.
- A simplified means of selecting the operating system to deploy centrally, without having to run commands on a specific OS deployment server.
- A simplification of the DHCP options configuration requirements, removing the need to change the DHCP configuration when changing the OS to install.
- A clean way of automating unattended OS deployments and operating system changes
- A hardware inventory management tool "on the fly"

Tivoli Provisioning Manager for OS Deployment is compatible with Solaris 9 and 10. It provides:

- A web interface for defining system profiles and target configurations, managing and scheduling OS deployments. This console is compatible with Netscape x, Mozilla x, Firefox 1.x and IE 6
- An OpenBOOT compatible bootstrap code for enhancing SUN built-in mechanism. This code is compatible with OpenBOOT versions 3.25 and higher (available in all SUN Ultra)
- An web interface extension running under Solaris to automate the provisioning of Jumpstart configuration files (`sysidcfg` and `profile`) from a central configuration database, and upload the logs back to a central location

## Solaris install server

Tivoli Provisioning Manager for OS Deployment does not require a full Solaris install environment (with JumpStart) to be able to provision Solaris. The only requirement is to have the corresponding operating system image files available via an NFS share to perform Solaris unattended setup and flash imaging. The Solaris install server can be configured on any standard Solaris computer using the steps below. Only after the system profiles are created, the Solaris install server can

also be moved on another UNIX computer with an NFS server compatible with Solaris targets. Although it is common to use the OS deployment server as Solaris install server, you can use multiple Solaris install servers closer to the Solaris computers to deploy than the OS deployment server if required.

To configure your Solaris install server, you must

1. Configure a network share and load the installation content of Solaris operating systems.
2. Configure a network share for Solaris Flash Archives.
3. Download and run the web interface extension on the Solaris install server.

   **Note:** If the Solaris install server resides on the OS deployment server, the web interface extension is already installed and running.

## Preparing a Solaris install server for operating system content

This section describes how to make available the installation content of Solaris operating systems on the Solaris install server.

**Note:** For the system profiles you intend to create from the files stored on the Solaris install server to be fully usable, you must

- Copy the files from the actual operating system version you want to create a profile for. If you want to deploy Solaris 10, then you should have the full content of the Solaris 10 CDs on your Solaris install server. Reusing files from another version might prevent proper OS deployment.
- Use the latest available versions of the operating systems to ensure you have the most up to date fixes.

To make available the installation content of Solaris operating systems on the Solaris install server, perform the following steps:

1. Create a root directory on a volume with enough disk space. For instance `mkdir /export/install`
2. For each version of Solaris to be deployed by the OS deployment server, create a subdirectory of the root directory. For instance, `mkdir /export/install/sol10`
3. Insert the appropriate installation CD or DVD.
4. Go to the CD/DVD directory, normally `/cdrom/xxxxx` where `xxxxx` is either `cdrom0` (a symbolic link to the actual media directory) or media title such as `sol_10_606_sparc`.

   ```
   cd /cdrom/cdrom0
   ```

   **Note:** A directory listing must show entries of the form `s0`, `s1`, `s2`, and so on. If the installation files are on multiple CDs, there is only one directory `s0`.
5. Copy the content of the CDs/DVD into the Solaris install server directory.
   a. Go to `/cdrom/cdrom0/Solaris_10/Tools`
   b. Make sure you have at least 5 GB available, for example, in `/export/install` on your Solaris install server.
   c. Run `./setup_install_server -w /export/install/sol10-miniroot /export/install/sol10`. This operation can take 15 to 30 minutes.
   d. If you want to deploy Solaris 10 Update 6 or higher, you must modify the resulting wanboot directory as follows:
      - 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 - )
```

**Note:**

a. The paths given in the substeps are for a Solaris computer which hosts both the Solaris install server and the OS deployment server.

b. These substeps must be performed at least once on a Solaris computer to create the `miniroot` file. This file can then be copied to any NFS shares.

6. Verify that NFS is started by making sure that `nfsd` is running. If needed, start `/etc/init.d/rpc` and start `/etc/init.d/nfs.server` (for Solaris).
7. Export and share the installation root directory as read-only for everyone, with root equivalence. On the Solaris install server, edit `/etc/dfs/dfstab` and add:

```
share -F nfs -o ro,anon=0 /export/install
```

8. Refresh shares with the command `shareall`.
9. Check that the export succeeded with the command `showmount -e`.

## Preparing a Solaris install server for Flash Archives

It is recommended to create a separate directory on the Solaris install server for Solaris Flash archives because this directory needs to provide write permission during the creation of the flash archives.

1. Create an export directory for Flash Archives: `mkdir /export/flars`.
2. Share this new directory with read and write permissions:

```
share –F nfs –o rw,anon=0 /export/flars
```

You can now create Flash Archives in this directory.

For more information, see “Creating Flash archives” in the information center.

# Appendix A. Integrating Tivoli Provisioning Manager for OS Deployment in a corporate environment

This section provides advice regarding the integration of Tivoli Provisioning Manager for OS Deployment into corporate environments and with other applications.

## Using an alternative database on Windows for Tivoli Provisioning Manager for OS Deployment

If installed straight out of the box on Windows, Tivoli Provisioning Manager for OS Deployment sets up a Microsoft Access database that is used for storing all parameters and target inventory. You do not need MS Access to use it, the MDAC component of Windows 2003/2008 (and freely available for other versions of Windows from the Microsoft Web site) is sufficient for Tivoli Provisioning Manager for OS Deployment to work.

Although this database is sufficient for using Tivoli Provisioning Manager for OS Deployment with a few hundred targets, you may need to customize or convert the database for integration into a corporate environment. This is in particular the case if you set up a multiserver infrastructure requiring a database allowing concurrent access (which Microsoft Access does not). Tivoli Provisioning Manager for OS Deployment database access supports custom databases, and the databases listed in Table 4 on page 2.

If you want to use your own ODBC source, ensure that it was created as a System DSN (not a User DSN), because it has to be usable by the IBM Tivoli OS Deployment Server service.

To install the product with an alternative database on Windows, follow these steps:

1. Create an empty database. The OS deployment server automatically populates the database with the necessary tables.
2. Create an ODBC system DSN from the computer on which you want to install the OS deployment server to your database. You must name the DSN `AutoDeploy` (case sensitive).

   **Note:** If you install Tivoli Provisioning Manager for OS Deployment before creating the ODBC DSN, or if you fail to name the DSN `AutoDeploy`, the Tivoli Provisioning Manager for OS Deployment installer will create a local database for this OS deployment server.
3. Run the installation wizard.
4. At the very end of the installation process, you are prompted for an account to use the ODBC gateway (as shown in Figure 1 on page 52). The account information that must be given at this stage is for access to the database. It does not need to correspond to the login for access to the OS deployment server or to the web interface.

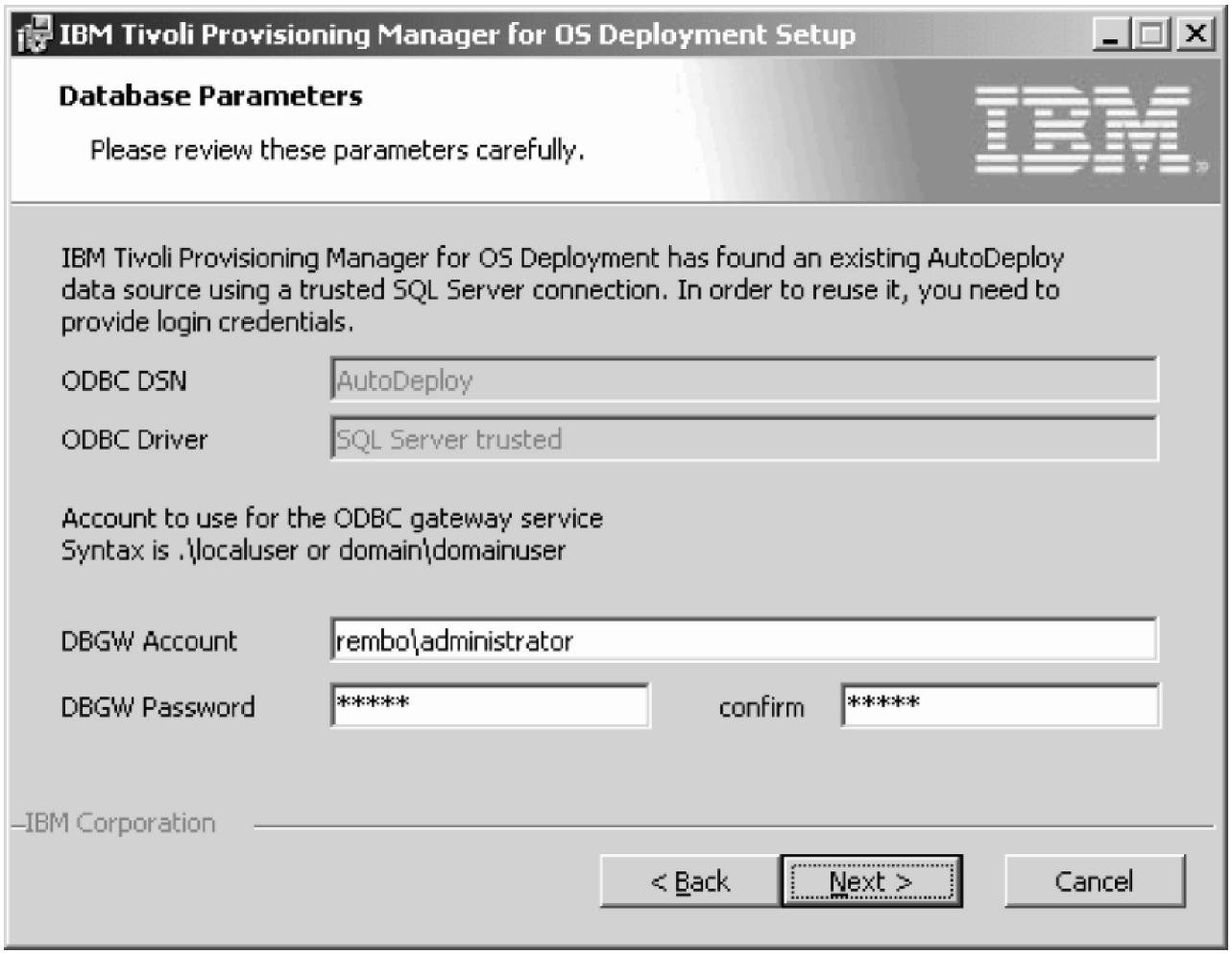


*Figure 1. ODBC account*

5. If your specified user account does not have the right to log on as a service, update your local security settings. To update your settings on a Windows 2003/2008 server, click **Control Panel > Administrative Tools > Local Security Policy > Local Policies > User Rights Assignments > Log on as a service**.

## Example: Installing with Oracle on Windows 64-bit.

To work correctly with a Tivoli Provisioning Manager for OS Deployment server on a Windows 64-bit box (i.e. w2k3 64-bit) connected to a remote Oracle 64-bit database server (i.e. on a Linux 64-bit ) , it is required to install and customize on the Oracle target machine both instantClient 32-bit and 64-bit with the appropriate Oracle version. This enables `rembo.exe` (32-bit application) to work correctly with Oracle 64-bit via a database client connection.

Here are the steps to follow on the Oracle target machine where Tivoli Provisioning Manager for OS Deployment server is installed (the scenario in this case is for Oracle 11g):

1. Unzip the InstantClient packages in a folder instant32
   - instantclient-basic-win32-11.1.0.6.0.zip
   - instantclient-odbc-win32-11.1.0.6.0.zip
2. Unzip the InstantClient packages in a folder instant64

- instantclient-basic-win-x86-64-11.1.0.6.0.zip
- instantclient-odbc-win-x86-64-11.1.0.6.0.zip

3. Add the paths

   ```
   c:\instant64
   ```

   and

   ```
   c:\instant32
   ```

   to the PATH variable.
4. Run the installer `odbc_install.exe` (from the folder instant32)
5. Run the installer `odbc_install.exe` (from the folder instant64)
6. Create a text file `tnsnames.ora` in the folder instant32 with all the information concerning the Oracle 64 bit server like:

   ```
   line1 DBORA64 = (DESCRIPTION = (ADDRESS_LIST =
   line2 (ADDRESS = (PROTOCOL = TCP)
   line3 (HOST = ip_address)
   line4 (PORT = listener_port)))
   line5 (CONNECT_DATA = (SERVER = DEDICATED)
   line6 (SERVICE_NAME = listener_service_name)))
   ```

   Where ip_address = the ip address of the machine on which the Oracle instance is running.

   listener_port = the port on which the Oracle listener is listening. The default value is 1521.

   listener_service_name = the name of the service corresponding to the instance you want to connect to. If you don't know it, then issue `lsnrctl status` on the machine running the instance and you will see which services are managed by the listener.

   For example:

   ```
   line1 DBORA64 = (DESCRIPTION = (ADDRESS_LIST =
   line2 (ADDRESS = (PROTOCOL = TCP)
   line3 (HOST = 192.168.1.13)
   line4 (PORT = 1521)))
   line5 (CONNECT_DATA = (SERVER = DEDICATED)
   line6 (SERVICE_NAME = jaf15mai)))
   ```
7. Set the TNS_ADMIN variable with

   ```
   c:\instant32
   ```
8. Use the odbc administrator (the 64bit one) to create a link to the Oracle 64 bit DBORA64 with the name oracle64 .
9. Click **Test** to check the connection.
10. Proceed with the installation of Tivoli Provisioning Manager for OS Deployment and fill the "database" fields with the appropriate parameters (userid and password).

The dbgw process may not have the right to run as a service (by default running as LocalAdmin). The error message will be *TNS could not be solved*. In this case, change the login of that service and verify that the problem is solved.

Running `dbgw` in command line also solves this problem.

## Password protecting a Microsoft Access database

If your product configuration is based on a Microsoft Access database, you can optionally choose to password protect the database.

The default Microsoft Access database is not password protected because it is intended to be used for evaluation purposes only. If you want to password protect your database, perform the following steps:

1. Stop the OS deployment server.
2. Set the password for the Microsoft Access database. The Microsoft Access database is located by default in the path `C:\Program Files\IBM\TPMfI\AutoDeploy.mdb`.
   a. Create a visual basic script named, `chgpwd.vbs`, to set the password and copy the sample script specified in "Sample chgpwd.vbs script" to the `chgpwd.vbs` file. Ensure you customize `dbSource`, `newPassword`, and `oldPassword` according to your environment.
   b. Run the script.
3. Edit the radb.ini file located in the path `C:\TPMfOS Files\global\rad\radb.ini` with the new password to be used by the server. Create this file in the path if it does not already exist as follows:

   ```
   [Settings]
   ODBC_Source=AutoDeploy
   ODBC_Username=admin
   ODBC_Password=<new_password>
   ```

4. Start the OS deployment server.

### Sample chgpwd.vbs script

```
file chgpwd.vbs  (customize dbSource, newPassword, oldPassword):
dbSource ="C:\Program Files\IBM\TPMfI\AutoDeploy.mdb"
'use back-quotes `` for empty password
newPassword ="<new_password>"
oldPassword ="``"
set conn = CreateObject("ADODB.Connection")
conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & dbSource & ";
  Persist Security Info=False;Mode=12;Jet OLEDB:Database Password=" & oldPassword
conn.Execute "ALTER Database Password " & newPassword & " " & oldPassword & ";"
conn.close
```

# Multiserver infrastructure

When working with replicated OS deployment servers, Tivoli Provisioning Manager for OS Deployment can use a single centralized database or multiple databases.

If you plan to have more than 10000 targets, the suggested multiserver configuration is with multiple databases.

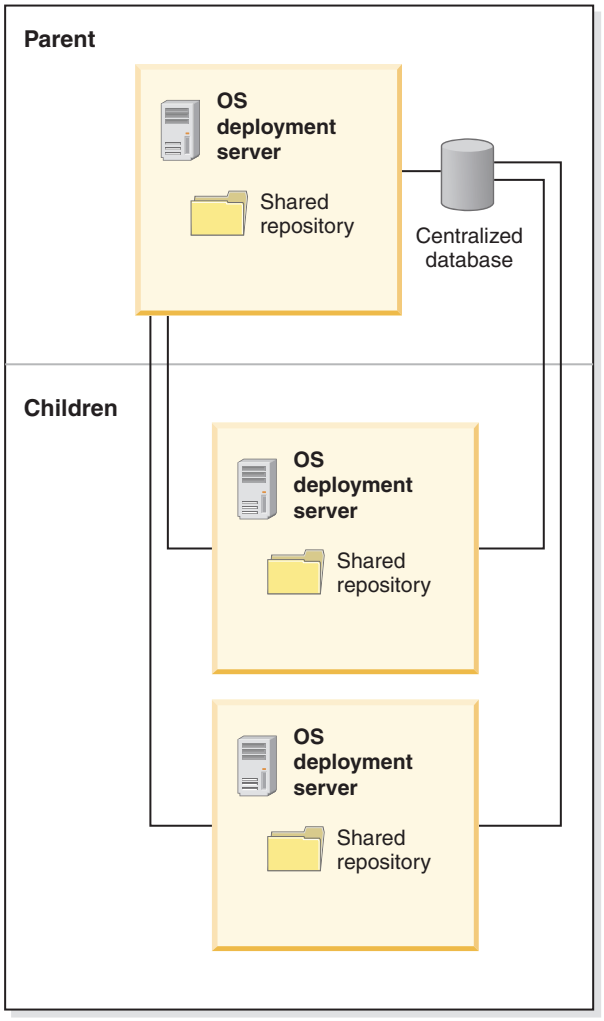


*Figure 2. Single database architecture*

The main prerequisite to use a single database is a good network connections between all your OS deployment servers and the database, as they need to connect on a regular basis. A single database does not require you to administer several databases located over the world, with the implied costs. However, you should certainly plan on a database backup because if your single database is down for whatever reason, none of your OS deployment servers is operational. An advantage of the single database infrastructure is the use of web interface to build the server hierarchy with a few simple clicks, to view all the servers in the hierarchy from any of its servers, and to perform all actions pertaining to the multiserver infrastructure.

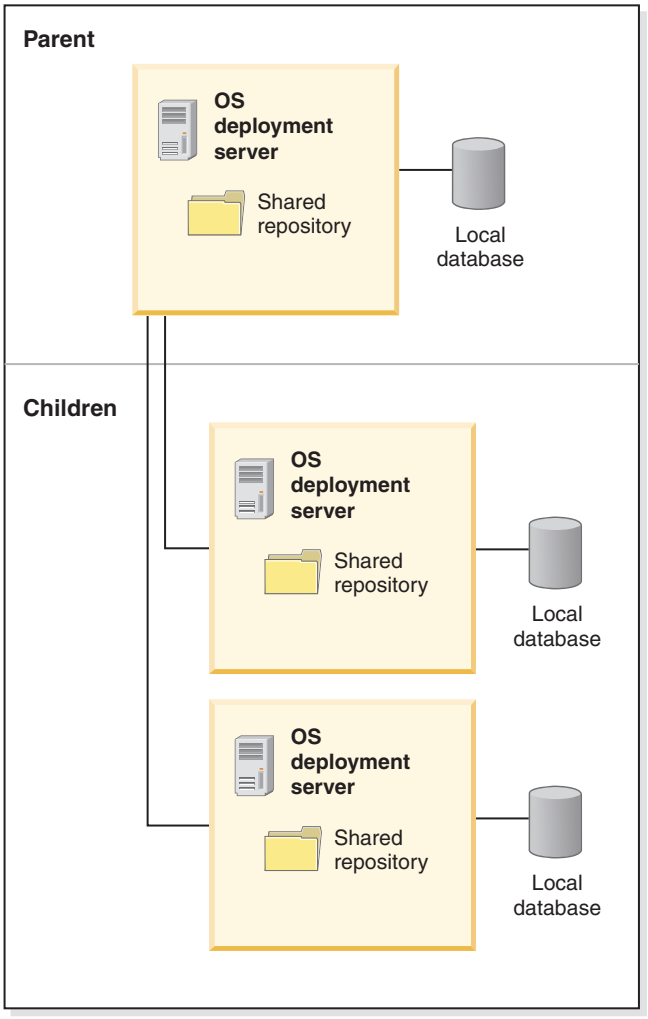


*Figure 3. Multiple databases architecture*

Use multiple databases for your multiserver infrastructure if your network infrastructure does not allow you to have a single database, as each of your OS deployment servers can function more or less independently of one another and does not require constant connection to a distant database. For example, if one of the databases is down, the OS deployment servers linked to the other databases are not hindered at all and can perform their usual tasks. The drawbacks of a multiple database infrastructure is the need of a database server in each location you have an OS deployment server, the need to use a configuration file to setup the multiserver infrastructure, and the limited visibility of the OS deployment servers as you can only see the children in the hierarchy through the web interface of a given OS deployment server.

## Installing a multiserver infrastructure with a centralized database

You can set up a multiserver infrastructure with a single, centralized database supporting concurrent access (such as MySQL or SQL server) shared by all OS deployment servers, with server replication. Note that the default database does not support concurrent access, so you must therefore set up your own database before installing your OS deployment servers.

To install replicated OS deployment servers, perform the following steps:

1. Create an empty database. The OS deployment server automatically populates the database with the necessary tables.
2. For each server, follow the appropriate process
   - On Windows, follow the installation process for installing with an alternative database from step 2 on page 51 onwards.
   - On UNIX, follow the installation process, making sure that you provide the correct path to the centralized database.

     The installation process includes the following database examples:
     - "MySQL 4.1 example" on page 22
     - "DB2 on AIX example" on page 24
     - "Oracle 11i on Linux example" on page 24

     **Note:** Ensure that the Tivoli Provisioning Manager for OS Deployment administrator passwords are identical on all servers.
3. After all OS deployment servers have been installed and started, select **Server replication** from the **Server parameters** menu. Your new OS deployment servers appear in the **Standalone servers** folder of the server tree.
4. For each server that you want to include in the infrastructure, move your standalone server into the **Replicated OS deployment servers** list by making it either a parent server or a child server.

   **Note:** Before assigning a child role to a server, you must have at least one parent server to which the new child server can belong.
5. To make a parent server, click **Make this OS deployment server a parent OS deployment server**
6. To make a child server:
   a. Click **Make this OS deployment server a child OS deployment server**
   b. Indicate which parent the server will be child to by selecting an appropriate parent with its specific network connection.
7. Restart all the OS deployment servers.
8. Click **Replicate this OS deployment server only** to replicate your child server with its parent

   **Note:** A child server is never up-to-date until it has been replicated with its parent.
9. Optionally, click **edit** under **Schedule** to set up a replication schedule to perform regular replication of your child servers. If you have a hierarchy of more than two parent and child servers, the schedule must match the hierarchy. Top servers must be replicated first, and child servers after.

## Example: Installing a Windows multiserver infrastructure with a DB2 centralized database

You can set up a multiserver with a single, centralized DB2 database that is shared by all Windows OS deployment servers, with server replication, by performing the following steps:

1. Create an empty DB2 database on a DB2 server machine (UNIX or Windows). Tivoli Provisioning Manager for OS Deployment has no restrictions for the database name that you choose, but DB2 allows a maximum of 8 characters. The OS deployment server automatically populates the database with the

necessary tables. Proceed with steps 2 to 5 for each Windows server on which you want to install the OS deployment server.

2. Install the IBM DB2 Client and catalog the remote DB2 database. Test the client connection to the database, for example, locally on the database client by running the following commands on the DB2 CLP window:

```
db2 catalog TCPIP NODE <node-name> REMOTE <db2-server-hostname> SERVER <server-name/port>
db2 catalog DATABASE <db-name> AT NODE <node-name>
db2 connect to <db-name> user <dbuser> using <dbpasswd>
```

   You can also perform these steps using the DB2 Configuration Assistant.

3. Create an ODBC system DSN pointing to the database cataloged on the machine in step 2. You must name the DSN **AutoDeploy**; this is case-sensitive.

   **Note:** If you installTivoli Provisioning Manager for OS Deployment before creating the ODBC DSN, or if you fail to name the DSN AutoDeploy, the installer creates a local Microsoft Access database for this OS deployment server.

4. Run the Tivoli Provisioning Manager for OS Deployment installation wizard.
5. At the end of the installation process, you are prompted for an account and a password to use for the database table creation and for the connection. You must have sufficient privileges to create tables, columns, and so on (database owner).

When you have installed your OS deployment servers, set up the server hierarchy from the Web UI and the replication. Refer to the Tivoli Provisioning Manager for OS Deployment documentation for details about the various replication methods that can apply to this scenario.

## DB2 on Linux example

**Note:** Some paths and parameter values depend on your DB2 installation. Paths and values provided here are only examples.

1. Install DB2 9.1 ESE at the required fix pack level on the parent server machine.
2. Create an instance where you can create the database, choosing, for example, rembodb as its name.
3. Ensure that you have Sun Java 1.4.2 in your server $PATH for both the parent and child. If there are multiple JVMs installed, ensure that 1.4.2 is the first JVM listed in the $PATH.
4. Create two links in `/usr/tpmfos` for the files `db2jcc.jar` and `db2jcc_license_cu.jar`:

   ```
   cd /usr/tpmfos
   ln -s /opt/IBM/DB2/V9.1/java/db2jcc.jar db2jcc.jar
   ln -s /opt/IBM/DB2/V9.1/java/db2jcc_license_cu.jar db2jcc_license_cu.jar
   ```

5. From the directory where you have extracted the build of Tivoli Provisioning Manager for OS Deployment, start the Java database gateway issuing the following command from a shell prompt:

   ```
   java -cp dbgw.jar:db2jcc.jar:db2jcc_license_cu.jar \
   -Djdbc.drivers=com.ibm.db2.jcc.DB2Driver com.rembo.dbgw.Dbgw -d
   ```

6. On another terminal, verify the database connectivity using:

   ```
   telnet localhost 2020
   use db2://127.0.0.1:50000/tpmfosd,db2inst1,<password>
   ```

7. Stop the `dbgw` process. If you do not, you will encounter errors when running setup.

8. Go to the /usr/tpmfos directory.
9. To install Tivoli Provisioning Manager for OS Deployment, start ./setup in the current directory and follow the setup program instructions. Mandatory parameters specific to using the product with DB2 are CLASSPATH, JDBCDRIVER, and JDBCURL. Default values for these parameters may not fit your particular installation of DB2. Make sure to provide appropriate values to the installer to have a working OS deployment server. In this example, you should use the following values:

   ```
   CLASSPATH = /usr/tpmfos/db2jcc.jar:/usr/tpmfos/db2jcc_license_cu.jar
   JDBCDRIVER = com.ibm.db2.jcc.DB2Driver
   JDBCURL = db2://127.0.0.1:50000/tpmfosd
   ```
10. On the child server, install the DB2 9.1 client, matching the Fix Pack level running on the parent. Allow setup to create a new instance.
11. Install the child server, specifying the parent database when asked.
12. After the parent and child have completed the Tivoli Provisioning Manager for OS Deployment installation, stop the following on both the parent and child: rembo server, rembo agent, and dbgw
13. The parent's (IP 192.168.128.10) radb.ini file is: cat ./files/global/rad/radb.ini with settings: ODBC_Source=db2://192.168.128.10:50000/rembodb, ODBC_Username=db2inst1, ODBC_Password=A740034ED1FDAF767F1177B230031971
14. The child's (IP 192.168,128.11), radb.ini file is: cat /opt/tpmfos/files/global/rad/radb.ini with settings: ODBC_Source=db2://192.168.128.10:50000/rembodb , ODBC_Username=db2inst1, ODBC_Password=A740034ED1FDAF767F1177B230031971
15. The child now points to the parent's database as its own database and the field MasterDbName is left empty, as are MasterDbUser and MasterDbPass.
16. When you finish the configuration, restart dbgw, rembo server, and rembo agent on both servers.
17. Connect to the parent Web GUI and select Server Parameters -> Replicate
18. You can now see both servers under "Standalone OS Deployment Servers"
19. Select the parent and follow the link "Make this OS deployment server a parent OS deployement server"
20. Select the child and follow the link "Make this OS deployment server a child OS deployement server"
21. Point the child server to the parent server.

## Installing a multiserver infrastructure with multiple databases

If you need to use multiple databases for your multiserver architecture

1. Plan your architecture carefully, knowing where the OS deployment servers are located and their IP addresses, to which type of database they are connected, to whom each server is parent and/or child, and so on.

   **Note:**

   - Each database must support concurrent access.
   - All the databases must use the same collation, for example SQL_Latin1_General_CP850_BIN, because the collation controls how records are sorted.
2. Transfer your plan in the configuration file config.csv.

For more information, see “Configuring Tivoli Provisioning Manager for OS Deployment with a text file” in the information center.

3. Install your databases and your OS deployment servers as if they were independent servers.
4. Stop your OS deployment server.
5. Copy `config.csv` in the `DataDir/global/rad` directory of each OS deployment server, usually `c:\TPMfOS Files\global\rad` on Windows.
6. Restart the OS deployment server.

You might get an information message in the logs looking like

```
2008/03/15 15:50:29.828] <INF> ODBC ERROR: [ERROR 958 (IM002): [Microsoft]
[ODBC Driver Manager] Data source name not found and no default driver specified]
<br>Configuration of the server has been changed, restarting server in 2 sec...
[2008/03/15 15:50:32.000] <INF> Thread shutdown
```

You can disregard this message because, when restarting, the server finds the database from the `config.csv` file. If the server does not restart on its own, which happens only in rare cases, you must restart it manually.

## Example: Synchronizing Tivoli Provisioning Manager for OS Deployment with DB2

In this scenario we have two OS deployment servers, a parent and a child, on Windows XP systems. Each server manages its own database. The DB2 software will be installed on both servers.

1. On the parent server, log in as Administrator.
2. Install the IBM DB2 Server .The db2admin user is created by default.
3. From the IBM DB2 Control Center, create a database, for example, rembodb .
4. Publish this database, rembodb, as an ODBC with name AutoDeploy . Tthe ODBC name must be AutoDeploy.
5. Save the config.csv file in the location where you want to have your DATADIR located, for example, "`c:\TPMfOS Files`".See “Sample config.csv” on page 61 for a sample file and its contents. Place the file in a correct structure ("`c:\TPMfOS Files\global\rad`").
6. Install IBM Tivoli Provisioning Manager for OS Deployment with any required fix packs. During the installation, the IBM DB2 ODBC source AutoDeploy will be discovered by the installer. Insert the db2admin credentials. Verify if the following string is contained in the logs: `VM > A multi-server configuration file is present and contains an entry for this server.` When running in debug level 4, you will see all the parameters of your config.csv.
7. On the child server, log in as Administrator.
8. Install the IBM DB2 Client.
9. Install IBM Tivoli Provisioning Manager for OS Deployment with any required fix packs, (use the embedded AutoDeploy MS Access database for the child server).
10. From the IBM DB2 Control Center, create a connection to the parent database, rembodb, created on the IBM DB2 Server in step 3 as follows:
    a. Click **Add system** > **Discover** and then select the IBM Tivoli Provisioning Manager for OS Deployment parent computer.
    b. Click**Add instance** > **Discover** and then select the IBM Tivoli Provisioning Manager for OS Deployment parent instance.

c. Click **Add database** > **Discover** and then select the IBM Tivoli Provisioning Manager for OS Deployment parent database ( rembodb ).

To test theIBM Tivoli Provisioning Manager for OS Deployment parent database connection, type the db2admin credentials and try to view some tables.

11. Publish this database pointing to the parent database, rembodb, as an ODBC. For example, ODBC name: parentdb.
12. Put the config.csv file and restart the child server with the command: **net stop/start remboserver**. Verify if the following string is contained in the logs:

    ```
    VM > A multi-server configuration file is present and contains an entry for this server.
    ```

13. Perform a final check.
    a. Open the Tivoli Provisioning Manager for OS Deployment web interface from the parent server. You should see something like, "You have 2 servers in the database, 2 active, and 2 replicated servers "..
    b. From the web interface, create a software package. After the synchronization completes, the child is automatically updated with the new software package.

### Sample config.csv

```
"HostName";"Interfaces";"DbName";"DbUser";"DbPass";"ParentIP";"ParentDbName";
"ParentDbUser";"ParentDbPass";"AutoSync";"PollInterval"
"ibm-6f3e2932072";"192.168.119.1";"AutoDeploy";"db2admin";"db2admin";
"SELF";"";"";"";""
"pc-000c2999b1fa";"192.168.119.128";"AutoDeploy";"";"";"192.168.119.1";
"ParentDB";"db2admin";"db2admin";"fcsd";"5"
```

Replicate all the deployment objects (configurations, software, and so on), but not the hosts. When you want to have multiple servers for the same hosts, the servers should share the same database (rather than using the "h" flag).

## Example: Installing multiserver infrastructure with a parent Microsoft SQL database and child servers with synchronized databases

This example demonstrates how you can use Microsoft Access to maintain a deployment server, the "parent" server that uses a Microsoft SQL database and some "child" servers that manage their own copy of the parent database. The authentication is based on the Microsoft SQL server and therefore not all the computers are required to be in the same domain. This hierarchy of parent and child servers is achieved using a text file.

In this scenario we have a "parent" OS deployment server using a Microsoft SQL database and some "child" servers that manage a synchronized copy of the parent database.

1. Install Microsoft SQL 2000 on the parent.
   a. Insert the MSSQL 2000 CD-ROM and run the following .bat file: \ENGLISH\ENT\SETUP.BAT. Click **Continue** when prompted with an SQL 2000 warning screen.
   b. Click **Next** at the Welcome Screen.
   c. Leave **Local Computer** selected and click **Next**.
   d. Leave **Create New Instance of MSSQL Server** selected and click **Next**.
   e. Click **Next** at the User Information screen.
   f. Click **Yes** to accept the license agreement.

g. Select to install the Server and Client Tools and click **Next**.
h. Leave **Default** selected and click **Next**.
i. Leave **Typical** selected and either confirm or change the installation paths as necessary.
j. Leave the selections **Use Same Account for each Service** and **Select Domain User Account**, and enter a valid user ID and password that has been enabled in Windows to run services. Click **Next**.
k. Select **Mixed Mode** (Windows Authentication and SQL Server Authentication) and enter a password for the "sa" account and click **Next**.
l. Click **Next** to begin the installation.
m. Select the appropriate licensing mode for your environment and click **Continue**.

2. Install service pack 4 for Microsoft SQL 2000 on the parent.
   a. Insert the SP4 CD-ROM and run **SETUP.BAT**.
   b. Click **Next** at the Welcome Screen.
   c. Click **Yes** to accept the license agreement.
   d. Click **Next** at the Instance Name screen.
   e. Select the SQL server system administration login and enter the "sa" user password. Click **Next**.
   f. Select **Upgrade Microsoft Search** and **Apply SQL Server 2000 SP4**.
   g. At the Error Reporting screen, do not select to send reports to Microsoft. Click **OK**.
   h. Click **Next** on the Start Copying files screen.
   i. Restart the server.
3. Configure Microsoft SQL server on the parent.
   a. Select **Start** > **Programs** > **Microsoft SQL Server** > **Enterprise Manager**.
   b. Open **Microsoft SQL Servers** > **SQL Server Group** > **(Local)(Windows NT)** > **Databases**.
   c. Right-click **Databases** and create a new database. Enter a name, for example, `TPMDB` and click **OK**.
   d. Expand **Security** and select **Logins**
   e. Right-click **Logins** and create a new login ID.
   f. Enter a user name, for example, `tivoli`.
   g. Select **SQL Server Authentication** and enter a password.
   h. Select the new database name as the default database for this user.
   i. Select the **Server Roles** tab and select **System Administrators**.
   j. Select the **Database Access** tab and select the new database.
   k. In the Permit in Database Role, ensure all check boxes are selected.
   l. Click **OK** to complete.
   m. Re-type the user password to confirm and click **OK**
   n. Confirm that the new user appears in the list and that the new database is listed as the default for this user.
4. Create ODBC source on the parent.
   a. Select **Start** > **Programs** > **Administrative Tools** > **Data Sources (ODBC)**
   b. Select the **System DSN** tab.
   c. Click **Add**.
   d. Scroll to the bottom of the list and select **SQL Server** and click **Finish**.

e. Enter the name as `AutoDeploy` (case sensitive).
f. Enter the name as `AutoDeploy` (case sensitive).
g. Select **SQL Server Authentication** and enter the ID and password of the new user ID you created, for example, tivoli. Click **Next**.
h. Change the default database to the new database defined in the previous step, for example, TPMDB, and then click **Next**.
i. Click **Finish** and test the datasource connection at the next screen. If there are no errors, click **OK** to exit.

5. Install the OS deployment server on the parent using the standard installation method.
   a. In the database parameters window, ensure the ODBC DSN displays the following parameters:
      - ODBC DSN: `AutoDeploy`
      - ODBC Driver: `SQL Server untrusted`
   b. The account fields must be set with the following values:
      - ODBC Username: `tivoli`
      - ODBC Password:<*your chosen password*>
   c. Confirm the password and finish the installation.
   d. In the web interface of the OS deployment server, on the page **Server** > **Server Parameters** > **Server replication**, the server must appear in the tree as a "standalone" server. Ensure the following values are specified:
      - Server role: `Head server`
      - Database: <*server_name*>:`TPMDB`
   e. Check the server name in the path of the web console, for example, <*child_name*>)
6. Install the OS deployment server on the child using the standard installation method. You then have a server working with a default Microsoft Access database installed locally.
   a. In the web interface of the OS deployment server, on the page **Server** > **Server Parameters** > **Server replication**, the server must appear in the tree as a "standalone" server. Ensure the following values are specified:
      - Server role: `Head server`
      - Database: `ACCESS:c:\Program Files\IBM\TPMfOSd\AutoDeploy`
   b. Check the server name in the path of the web console, for example, <*child_name*>.
   c. Open a DOS window and at the prompt type "`net stop remboserver`" .
7. Create the ODBC Source and config.csv file on both the child and the parent.
   a. Select **Start** > **Programs** > **Administrative Tools** > **Data Sources(ODBC)**.
   b. Select the **System DSN** tab.
   c. Click **Add**.
   d. Scroll to the bottom of the list and select **SQL Server** and then click **Finish**.
   e. Enter the name for the parent database to synchronize, for example, `ParentDB`.
   f. Select your SQL server name from the drop-down list and click **Next**.
   g. Select **SQL Server Authentication**, and then enter the ID and password of the new user ID you created, for example, tivoli. Click **Next**.

h. Click **Finish** and test the datasource connection at the next screen. If there are no errors then click **OK** to exit.

i. Create a "config.csv" file in the folder "`c:\TPMfOS Files\global\rad`" with the following values:

- HostName;DbName,DbUser,DbPass;ParentIP;
- ParentDbName;ParentDbUser;ParentDbPass;AutoSync
- "<*child_name*>";AutoDeploy;"";"";
- "<*IP_of_the_Parent*>";"ParentDB";"tivoli";
- "<*password_new_user*>";"fcsdh"
- "<ParentHostname>";"AutoDeploy";"tivoli";
- "<*password_new_user*>";"SELF";"";"";"";""

**Note:** The keyword "SELF" is important because it states that the parent is at the top of the hierarchy and that no other server can manage the database.

j. On the parent, open a DOS window and at the prompt type "`net stop remboserver`"

k. On the parent, copy the file config.csv in the folder "`c:\TPMfOS Files\global\rad`"

8. Complete the installation of the OS deployment server on the child. You have now prepared all the parameters for the synchronization. You can the restart the server. The restart can take a little longer than usual because after it restarts, it reads the file config.csv, changes some of the internal parameters, and then restarts automatically.

a. Open a DOS window and at the prompt type "`net start remboserver`"

b. After logging on to the web interface, a warning message is displayed to inform you that you are now working on a child server and must not create or modify the objects of the server.

c. In the web interface of the OS deployment server, on the page **Server** > **Server Parameters** > **Server replication**, the server must appear in the tree as a "synchronized" server. Ensure the following values are specified:

- Server role: Child (with cache)
- Database: `ACCESS:c:\Program Files\IBM\TPMfOSd\AutoDeploy`
- Parent database: <*server_name*>:TPMDB
- Replicated items: deployment objects and hosts

9. Repeat the steps 6 on page 63, 7 on page 63, and 8 for each child that you want to install in the hierarchy.

10. Check the hierarchy of your servers on the parent.

a. Open a DOS window and at the prompt type "**`net start remboserver`**" .

b. From the web console, select **Server** > **Server Parameters** > **Server replication** and then select the parent server which must appear in the tree as a "standalone" server. Ensure the following values are specified:

- Server role: Host server
- Database: <*server_name*>:TPMDB

c. From the web interface of the OS deployment server, on the page **Server** > **Server Parameters** > **Server replication**, the server must appear in the tree as a "synchronized" server. Ensure the following values are specified:

- Server role: Child (with cache)
- Database: `ACCESS:c:\Program Files\IBM\TPMfOSd\AutoDeploy`

- Parent database: <*server_name*>:TPMDB

11. Start the synchronization operation on the master.

## Working with Tivoli Provisioning Manager for OS Deployment locally

In some circumstances, it is useful to work with Tivoli Provisioning Manager for OS Deployment locally, without the intent of deploying remote-boot computers in the current environment, to work on Tivoli Provisioning Manager for OS Deployment objects. For example, you might want to prepare a web interface extension or generate a CD set without starting a remote-boot target.

In this case, no PXE server is needed, but the web interface still needs to be able to contact the OS deployment server to work properly, because the OS deployment server is the main repository for all objects. Install all components on the computer in order to be able to work with web interface, even if you do not want to use the PXE boot capability. If you do not intend to let unknown targets boot remotely, you can disable network boot for unknown targets in the **Idle Layout** of the task templates.

# Appendix B. Installing and uninstalling the web interface extension

The web interface extension is installed automatically when the full Tivoli Provisioning Manager for OS Deployment product is installed on a computer. It is however possible to install only the web interface extension on a computer.

When the web interface extension is running on a target, it can be used by the OS deployment server to perform actions on the target and to gather information from it.

When browsing the web interface from a computer which is not the OS deployment server, the web interface extension allows the computer on which the web interface is running to exchange informations with the OS deployment server. Without the web interface extension, several features of the web interface are disabled.

## Status of the web interface extension

To learn the status of the status the web interface extension on a computer, perform the following steps:

1. Open a web interface
   a. Open a browser
   b. Go to `http://<serverIPaddress>:8080` where *<serverIPaddress>* is the IP address of your OS deployment server.
2. Go to **> Server > Server Status > Web interface extension**

   Icons and messages indicate whether the web interface extension is installed or not.

**Note:** The web interface extension is not designed for multitasking. If the web interface extension is busy with another task, such as uploading files to the OS deployment server, it cannot respond to the status enquiry and it is not detected. To refresh the status,

1. Wait until all tasks of the web interface extension are closed.
2. Click the web interface extension icon at the lower right corner of the web interface.

- If the web interface extension is not installed, you might want to install it.
- If the web interface extension is already installed, but of a different version than the OS deployment server, you might want to upgrade by uninstalling and reinstalling it.
- If the web interface extension is already installed with a version identical to the one on the OS deployment server, you do not need to do anything.

## Installing the web interface extension on Windows operating systems

**Note:** On Windows Vista/2008/7 64-bit, User Access Control must be disabled during installation.

To install the web interface extension (`rbagent.exe`):

- For interactive installation
  1. On **Server** > **Server Status** > **Web interface extension for Windows**, follow the link **Click here to download the Web interface extension installer for Windows**
  2. Run `rbagent.msi`
  3. Follow the instructions of the installer

  **Note:** The password required must match the super user password of the OS deployment server to which you link the web interface extension.
- For a silent installation, you can use the parameters `LANG_ID`, `LANG_PROMPT`, `NET_PASSWORD`, `SERVER_IP`, `RBAGENT_ACCOUNT`, `RBAGENT_PASSWORD`, and `FORCE_FRESH` described in Table 7 on page 12.

In case of an invalid IP address is provided in web interface extension installer, a long timeout can occur.

## Uninstalling the web interface extension on Windows operating systems

To uninstall the web interface extension on Windows operating systems:

1. Open the **Control Panel**
2. Open **Add or Remove Programs**
3. Select **IBM Tivoli Web interface extension**
4. Click **Remove**
5. Follow the instructions of the installer

**Note:** The `rbagent.conf` file is not removed from the `Program Files/Common Files/IBM Tivoli` directory. This allows you to keep the same configuration in case of upgrade. If you want to remove the web interface extension completely, you must delete the `rbagent.conf` file manually.

## Installing the web interface extension on UNIX operating systems

The extension of the executable file for the web interface extension depends on the operating system. The operating systems and their corresponding names for the executable files are given in Table 14.

*Table 14. Executable file names for the web interface extension on UNIX*

| Operating system | Executable file name |
|---|---|
| AIX | `rbagent.aix` |
| Linux x86-32 | `rbagent.linux` |
| Linux x86-x64 | `rbagent.lnx64` |
| Linux on Power | `rbagent.linuxppc64` |

*Table 14. Executable file names for the web interface extension on UNIX (continued)*

| Operating system | Executable file name |
|---|---|
| Linux on zSeries | `rbagent.zlnx` |
| OS X | `rbagent.osx` |
| Solaris | `rbagent.solaris` |

The installation steps for the web interface extension are similar for all UNIX operating systems. They are exemplified here for a Linux operating system.

1. On **> Server > Server Status > Web interface extension**, follow the link **Click here to download the Web interface extension installer for Linux**
2. Download the `rbagent.linux` file
3. Open a shell, login as `root`, and go to the directory where `rbagent.linux` was downloaded
4. To transform the downloaded file into an executable, type:
   ```
   chmod +x rbagent.linux
   ```
5. To run the web interface extension, type:
   ```
   ./rbagent.linux -s <IP>:<password>
   ```
   where *IP* is the IP address of the OS deployment server to which you want to connect, and *password* is the password of the Web user administrator.

In case of an invalid IP address is provided in web interface extension installer, a long timeout can occur.

## Uninstalling the web interface extension on UNIX operating systems

To uninstall the web interface extension on UNIX operating systems, simply delete the executable file.

# Appendix C. Federal Information Processing Standards Compliance

Federal Information Processing Standards (FIPS) are standards and guidelines issued by the National Institute of Standards and Technology (NIST) for federal government computer systems. FIPS are developed when there are compelling federal government requirements for standards, such as for security and interoperability, but acceptable industry standards or solutions do not exist. Government agencies and financial institutions use these standards to ensure that the products conform to specified security requirements.

## Procedure to render the product FIPS-compliant

To make Tivoli Provisioning Manager for OS Deployment FIPS 140-2 compliant, after installing the product, install IBM Global Security Kit version 7.0.4.20 or higher, and then configure it.

You can perform key database management with the GSKit tools:

**gsk7capicmd**
Native command line

**ikeyman**
Optional graphic interface.

After installing Tivoli Provisioning Manager for OS Deployment, proceed as follows:

1. Install GSKit by following the instructions in the GSKit documentation and following these guidelines:
   - On Windows, set it up with the short product name **tpmfosd**.
   - On UNIX, ensure that the *LIBPATH* environment variable contains the path to the GSKit libraries.
2. Create a key database (keyring) in FIPS mode with a stash file.
3. Get a certificate from a Certificate Authority and receive it into your key database. Alternatively, you can create a self-assigned certificate in FIPS mode in the key database for test purposes. You can choose to give your certificate a label.
4. To enable Tivoli Provisioning Manager for OS Deployment to use GSKit, set the environment variables *GSK_KEYRING_FILE* and *GSK_KEYRING_STASH_FILE*. These variables must specify an absolute path, not including any environment variable. The following example is incorrect: `GSK_KEYRING_FILE=%basedir%\key.kdb`. If you specified a label for your certificate, also set the environment variable *GSK_KEYRING_LABEL*, otherwise the default certificate is used. Set the environment variables as follows:
   - On Windows, set the system environment variable (not the user variable) and then restart the computer, because the service environment is read at system startup.
   - On UNIX, add the variables to the `tpmfosdvars` file, which is located in `/etc/sysconfig` or `/etc/` depending on the platform, prepending **export** to the variable name as in the following example:

     ```
     export GSK_KEYRING_FILE=/root/key.kdb.
     ```

Restart the OS deployment server, because the `tpmfosdvars` file is read at server startup.

5. The next time you go to the Tivoli Provisioning Manager for OS Deployment web interface you will see the string **"(FIPS)"** after the server name.

These steps ensure that the SSL library used is FIPS-compliant.

## Components that are FIPS-compliant

Once you have installed and configured the IBM Global Security Kit, the following components of Tivoli Provisioning Manager for OS Deployment are FIPS-compliant:

- the SSL library used
- the Web server
- encryption on the OS deployment server
- connection and access to all pages of the web interface

## FIPS compliance and versions older that 7.1.1

If you have a non-FIPS-compliant version of Tivoli Provisioning Manager for OS Deployment installed that is older than version 7.1.1, and you want to make it FIPS-compliant, you must first upgrade to version 7.1.1 and then follow the installation and configuration steps above.

## Returning to a non-FIPS-compliant version

If you want to return to a non-FIPS-compliant system, you must remove the GSKit environment variables *GSK_KEYRING_FILE GSK_KEYRING_STASH_FILE*, and *GSK_KEYRING_LABEL* as follows:

- On Windows, remove the variables from the system environment variables and then restart your computer.
- On UNIX, remove the variables from the `tpmfosdvars` file and then restart the OS deployment server.

# Notices

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 may 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 may not display.

## 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, Intel logo, 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