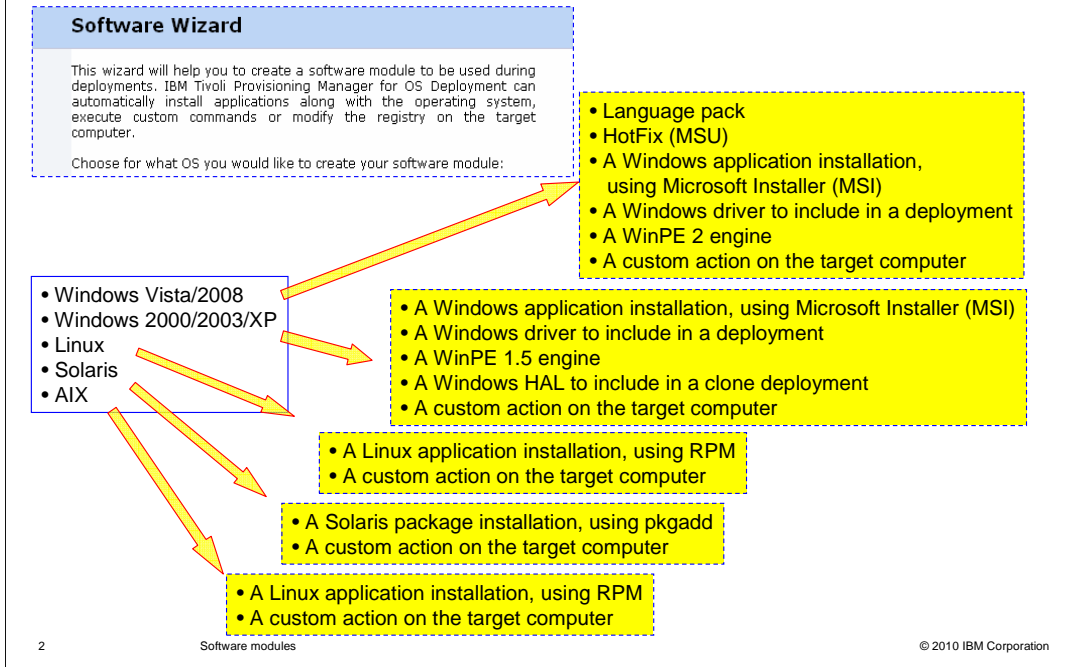

Tivoli Provisioning Manager for OS Deployment 7.1.1

Software modules



In this module, you learn how to use the software modules feature that is provided by Tivoli® Provisioning Manager for OS Deployment. You can use this feature to install software and drivers during a system deployment and to solve some common problems encountered during an operating system installation.

Creating a software module



You can use Tivoli Provisioning Manager for OS Deployment 7.1.1 to create and install software modules, covering most of the software deployment scenarios on the supported target platforms (Windows®, Linux®, Solaris, and AIX®). On Windows, you can install language packs and hot fixes for Windows Vista/2008. You can also install applications that are packaged in MSI format. You can copy a file or set of files to the target computer, run an additional command or script, or modify the Windows registries. You can also install drivers, inject a specific Hardware Abstraction Layer during the deployment of a clone profile, or create a WinPE software module. On Linux and AIX, you can install applications using RPM or copy a set of files and execute a command or script. On Solaris, you can install applications using pkgadd or copy a set of files and execute a command or script. A software module can be created using the Tivoli Provisioning Manager for OS Deployment WEBUI from the **OS Deployment > Software modules** panel. You can click the **New software** button or select **Add a new software** from the Contextual actions menu.

Language packs and hot fixes

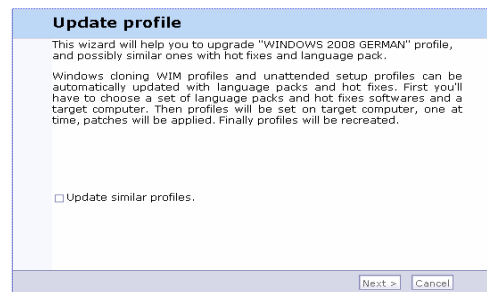
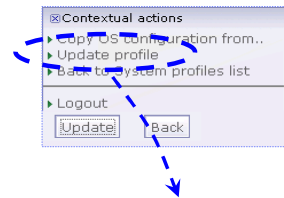
- Add a language pack or hot fix on Windows Vista or Windows 2008
- Clone
- WIM Clone (cloning from a reference file)
- Unattended setup

Example:

Inject a LP (language pack) or Hot fix into a WIM cloning image

Create a standard software package with the LP/Hot fix

Edit the WIM cloning system profile and click the **Update profile** link or the **Update** button. In the wizard, select the LP or Hot Fix software package that you want to include



3

Software modules

© 2010 IBM Corporation

In Tivoli Provisioning Manager for OS Deployment you can install language packs and hot fixes when deploying Windows Vista/2008 system profiles. You can install them for clone profiles (clone from a reference machine or clone from a reference WIM file) and unattended profiles. The steps for installing the language packs and hot fixes are: 1) Create the language pack or hot fix software module; 2) Update the Windows Vista/2008 profile to inject the specific software module previously created; 3) From the software module page, you click the **New software** button. Alternatively, you can select the **Add a new software** option from the Contextual actions menu. Then, select Windows Vista/2008 and the language pack or hotfix menu option. Follow the wizard by providing the location of the needed files; 4) After creating the software module, go to the System profiles page and select the specific Windows Vista/2008 profile to be updated; 5) Double-click the profile and click the **Update** button. The wizard guides you in the selection of the language pack or hot fix software module. **Note:** To update an unattended setup profile or a WIM cloning profile, you need a Windows 32-bit machine with Windows Automated Installation Kit (WAIC) 1.1 32-bit installed. You must also have the Web interface extension running.

A Windows application installation using Microsoft Installer

- Available for Windows 2000, 2003, XP, Vista, and 2008
- You must copy your .msi file and the related files (cab files and so on) into an empty directory locally on the boot server or on another machine with the Web Interface Extension running
- If you do not have an .msi file but have a self-extracting archive (.exe), you first execute the archive to extract the .msi file

Software Wizard

Here are the parameters that IBM Tivoli Provisioning Manager for Images will use to create the software module:

When to apply this software module:
 When the OS is installed

Package filename on the OS deployment server:
 install9AB8963F3C.pkg

Destination path on the target:
 \\install9bmsi

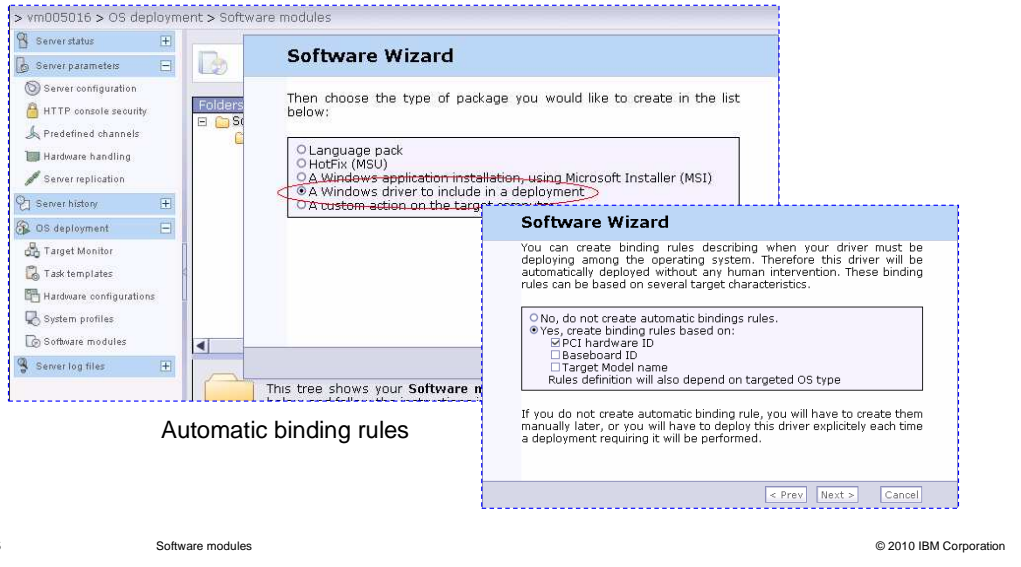
Command line to run on the target:
 MSIEXEC /qb /I "\\install9bmsi\\yb7.1.00-081.13.msi"

< Prev Next > Cancel

You can use this software module option to install an MSI package on the deployed machine for Windows 2000, 2003, XP, Vista, and 2008. To build a software module, you copy the msi and any related file into a directory on a machine with the Web Interface Extension running. The machine can be the Tivoli Provisioning Manager for OS Deployment server where the Web Interface Extension is installed by default. If you have an .exe archive, execute it to have the .msi file available in that folder. Then, from the new software module wizard, select the specific Windows operating system and select **A Windows application installation, using Microsoft® Installer (MSI)**. Follow the wizard and specify the machine that is running the Web interface extension and hosting the folder containing the .msi package. Specify the deployment stage when the software module will be installed. You can install it after additional reboots to avoid installation conflicts with other packages' installations. Also, specify the msixec options to run the .msi package. The options might be necessary to run it in unattended mode.

Windows driver (1 of 2)

Save driver files (.inf .cat .sys files) in a folder and provide the path to the wizard



You can inject drivers to Windows 2000, 2003, XP, Vista, and 2008 through the software module option named **A Windows driver to include in a deployment**. This option provides the appropriate driver files for unattended and clone profiles installation. It is also useful for injecting drivers for the disk controller (mass storage driver injection) of the target system to avoid deployment failure reported by the appearance of code 0x0000007b on a blue screen. To create these software modules, save your driver files, the .inf, .cat, and .sys files, into a directory on the Tivoli Provisioning Manager for OS Deployment server. You can also save your files on a machine with the Web Interface Extension running and provide that path to the Software Module wizard. The folder might have multiple subdirectories containing different drivers. During the software module creation, you can decide whether to create automatic binding rules that are based on PCI hardware ID, Baseboard ID or Target Model name. The drivers are then automatically bound to the target machine by matching the specific rules.

Windows driver (2 of 2)

The sequence of screenshots shows the following steps:

- Software Wizard (Step 1):** Prompting the user to select a folder for driver files. The folder path is `E:\images\1000drv`.
- Software Wizard (Step 2):** Displaying information about the driver: Provider: Intel, Driver class: Net, Version: 08/20/2008 - 12/11/2008.
- Software Wizard (Step 3):** Allowing the user to create binding rules. The user selects "Yes, create binding rules based on:" and "PCI hardware ID".
- Software Wizard (Step 4):** Configuring deployment parameters:
 - When to apply this software module: When the OS is installed
 - Package filename on the OS deployment server: `edrvCF09963F3C.pkg`
 - Destination path on the target: `\drivers\Net-93B2F`
 - Note: Drivers must be stored in the \drivers folder

This sequence of screen captures is displayed when you create a software module with the option, **A Windows driver to include in a deployment**. As mentioned previously, you must specify the machine that is running the Web interface extension where the folder containing the drivers files is located. The wizard parses the files and recognizes the included driver or drivers. In the next panel, you can choose to create binding rules, which automate the driver deployment to targets. More precisely, you can choose to create binding rules (that is, links) between the drivers that you are including in the software module and target system attributes. One of the target system attributes is the PCI hardware ID, which is a key value that identifies a specified driver and is based on values discovered on the machine. These values are discovered by the PCI/DMI scan that is performed by Tivoli Provisioning Manager for OS Deployment. You can see these values in the Target Inventory tab when browsing the target details information through the product WEBUI. Other target system attributes are the target system Baseboard ID and the target model name. You can select one or multiple choices. If you decide not to create new automatic binding rules, you can create them later by editing the software module. If you choose one option, you must provide information in the next panels. For example, if you select the automatic binding rules based on the PCI hardware ID characteristic, you have to specify if the rule applies to the exact same device or similar devices. A device driver is an exact match if the discovered PCI vendor ID, device ID, and subdevice ID values for the specific device match the ones discovered on the machine. The device is similar only if the PCI vendor ID and device ID match. After you specify the information about the binding types that you selected, you decide when during the deployment the driver package will be installed. For drivers, you leave the stage set to when the OS is installed. You can also specify the destination directory for the drivers. The destination directory must be a subdirectory of the `\drivers` folder.

Windows universal image

Create a Windows profile with various disks drivers and network drivers to fit different hardware

- ➔ WinPE2 manages the installation so that any NIC and disk controller driver must be also injected into a WinPE2 software package
 1. Install the Microsoft tool WAIK (Windows Automated Installation Kit) 32-bit and the Web Interface extension on a Windows 32-bit machine (can be the Tivoli Provisioning Manager for OS Deployment server)
 2. Create a standard software module with the drivers
 3. Edit the WinPE2 package
 4. Use the Update drivers link
 5. Specify the machine where the WAIK and rbagent (Web Interface Extension) is installed
 6. Select the software modules created in step 2


Software modules that include drivers are important if you want to create a Windows universal image. This universal image is a Windows system profile, including all disk controller drivers and NIC drivers fitting different hardware. To create the universal image, you package the needed drivers for all the target system hardware in software modules. You use the automatic binding rules option so that any requested driver is automatically installed during the deployment. WinPE manages the Windows systems installation so that any NIC or disk controller driver is injected into the WinPE software module. To create the universal image, you use the following procedure: 1. Ensure that you have a Windows 32-bit system with WAIK 1.1 32-bit installed on a running Web Interface Extension. 2. Create the software module or modules, including any needed NIC or disk driver. 3. Edit the WinPE2 software module in the Software Modules page in the product WEBUI. 4. Click Update drivers and follow the wizard by specifying the machine with WAIK and the running Web Interface Extension. Select all the software modules needed to inject into WinPE. 5. Specify the machine where the WAIK and rbagent (Web Interface Extension) is installed. 6. Select the software modules created in step 2.

Mass storage and HAL driver injection

Bind and install mass storage (disk) and Hardware Abstraction Layer (HAL) drivers with the deployment

BSOD 7b

- Avoids error BSOD 7b
- For disk drivers, you create and deploy software modules, including drivers



```
A problem has been detected and windows has been shut down to prevent damage to your computer.
If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:
Check for viruses on your computer. Remove any newly installed hard drives or hard drive controllers. Check your hard drive to make sure it is properly configured and terminated. Run CHKDSK /F to check for hard drive corruption, and then restart your computer.
Technical information:
*** STOP: 0x0000007B (0xF78A2640, 0xC0000034, 0x00000000, 0x00000000)
```

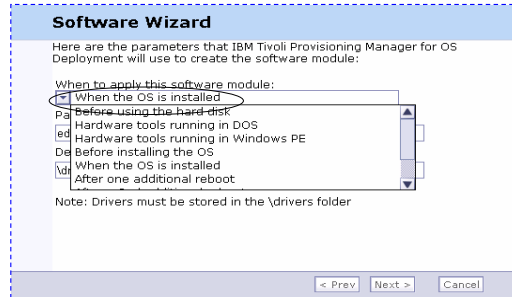
A Windows deployment fails and reports the stop code 7b if you do not inject the correct disk driver or do not inject the correct Hardware Abstraction Layer drivers. For mass storage drivers, you create and deploy software modules, including the needed disk driver. This process is valid for both unattended and clone profiles and can use the automatic binding rules feature. For HAL drivers, Tivoli Provisioning Manager for OS Deployment provides a specific software module type for injecting those drivers during the deployment and avoiding the problem. The HAL applies specifically to clone profiles.

Injecting mass storage drivers



Software modules including Windows drivers can be used to inject **Mass Storage drivers**

- 1) Identify Disk Controller type (IDE Controller, SCSI controller) and the needed driver
- 2) Download the specific disk driver
- 3) Create a software module.
- 4) Choose the PCI binding rule based on the PCI hardware ID and customize the driver to be applied when the OS is installed



Note: Remember to update the WinPE software module with the create software module

To inject the correct disk drivers during a Windows deployment and avoid the 7b BSOD, you identify the disk controller characteristics for the target system. You can access the target details for the specific system from the target monitor page (product WEBUI), change to the inventory tab, and expand the PCI devices information. Depending on the Disk controller type, for example, IDE or SCSI, check the driver identifiers and, based on them, download the specific driver from the hardware vendor Web site. The driver is typically identified by the key VEN_<Vendor_id>&DEV_<device_id>&SUBSYS_YYYYZZZZ, where YYYY is the SubDeviceID and ZZZZ is the SubVendorID. After you have the driver files (composed of at least one .sys and one .inf file), you can create the software module that includes them. You can choose the PCI binding rule based on the PCI Hardware ID. The installation state must be when the OS is installed. Remember to update the WinPE software module with the created software module too.

How to inject a Windows HAL (1 of 2)

1

Software Wizard

Then choose the type of package you would like to create in the list below:

- A Windows application installation, using Microsoft Installer (MSI)
- A Windows driver to include in a deployment
- A Windows HAL to include in a clone deployment
- A custom action on the target computer

2

Software Wizard

Please select the folder under which the HAL files can be found. It can be either the root directory of a standard Windows installation CD-ROM, or another folder in which you have copied the I386 (or AMD64) directory.

Folder: Browse local disk.

3

Software Wizard

Multiple HAL types have been found. Select the hardware type for which you want a package to be created, and click on Next to go to the next step. You can click on each name to get more details on the type of hardware.

OS detected: Windows XP Professional

OS architecture: i386

Hardware type:

- 486c_up
- acpiapic_up
- mps_up
- mps_mp
- syspro_mp
- acpiapic_up
- acpiapic_mp

4

Software Wizard

Here is some information that IBM Tivoli Provisioning Manager for OS Deployment has extracted from the folder you have specified. Please review this information and click on Next to go to the next step.

OS detected: Windows XP Professional

OS architecture: i386

Hardware type: acpiapic_mp

11 Software modules © 2010 IBM Corporation

A specific software module type injects HAL during the deployment of a clone profile. When creating the software module you select the option, **A Windows HAL to include in a clone deployment**. On this slide and in the next one, you can see the sequence of screens displayed during the specific software module creation. You must provide the folder where the HAL files can be found, typically on the specific operating system CD/DVD image. The creation process might have different HAL types. You must choose the hardware types that match the ones on your target machine.

How to inject a Windows HAL (2 of 2)

Software Wizard

You can create binding rules describing when your new HAL must be deployed among the operating system. Therefore this HAL will be automatically deployed without any human intervention. These binding rules can be based on several target characteristics.

No, do not create automatic bindings rules.
 Yes, create binding rules based on:

- OS Deployed
- HAL Type
- Target Architecture
- Target Model name

If you do not create automatic binding rule, you will have to create manually later, or you will have to deploy this HAL explicitly each deployment requiring it will be performed.

Software Wizard

If you do not choose the right parameters, you may get a blue screen of death at start up time on target computer.

OS architecture:

HAL Type:

OS targeted:
 Windows XP Professional build 5.1.2600

Software Wizard

Please enter a description to identify your software module in the IBM Tivoli Provisioning Manager for OS Deployment database.

Description:

Comment:

Software Wizard

Your software module has been successfully created.

It will automatically be applied when matching devices are found on a target computer.

12 Software modules © 2010 IBM Corporation

For the software modules including HAL, you can create automatic binding rules that are based on several factors: the operating system you are deploying, the Hardware Abstraction Layer type, the target system architecture, and its model name. You must provide more details about the OS architecture, Hardware Abstraction Layer type, and deployed OS that this software module will apply to.

Custom action on the target computer

- You can choose one of these items:
 - A WinPE 1.5 Ramdisk image
 - A WinPE 2 engine for OS deployment server ramdisk image
 - A configuration change to perform on the target computer, for example, a command to execute or a registry patch
 - A set of files to copy on the target computer with an optional command to execute

Various software module types are available when you select the custom action option. A WinPE 1.5 Ramdisk image option is available for Windows 2000, 2003, and XP. You can use this option to create a WinPE 1.5 Ramdisk image from your Windows 2000, 2003, or XP image CD/DVD. A WinPE 2.0 Ramdisk image option is available only for Windows Vista and 2008. You can create a WinPE2 software module from the Vista/2008 installation files. The WinPE2 software package creation requires that the computer where the source images for Vista/2008 are located has a Windows 32-bit operating system, the Web Interface Extension started with local administrator privileges, and the Windows Automated Installation Kit (WAIK) 1.1. 32-bit installed. A configuration change to perform on the target computer (a command to execute, a registry patch) is available for you to copy and execute a single file. You can apply a Windows registry change, apply a Windows .ini file change, copy a single text file, execute a single command file, and boot a virtual floppy disk. A set of files to copy on the target computer (with an optional command to execute) is available for you to copy the content of a directory to the target computer and to execute a command.

Software modules for Linux and AIX

- Software module options available for Linux and AIX target computers
 - A Linux application installation, using RPM
 - A custom action on the target computer
 - A configuration change to perform on the target computer (a command to execute)
 - A set of files to copy on the target computer (with an optional command to execute)

The software module options available for Linux and AIX target computers are: A Linux application installation, using RPM; a custom action on the target computer such as a configuration change to perform on the target computer (a command to execute) or a set of files to copy on the target computer (with an optional command to execute). Basically, you can provide an RPM to be installed. You can also specify a command to be executed or a set of files to be copied onto the target computer with a command to be run. The wizard guides you through the specific software module creation. This option is like the related option that is available for MSI packages for Windows.

Software modules for Solaris

- Software module options available for Solaris target computers
 - A Solaris package installation, using pkgadd
 - A custom action on the target computer
 - A configuration change to perform on the target computer (a command to execute)
 - A set of files to copy on the target computer (with an optional command to execute)

The software module options available for Solaris target computers are: a Solaris package installation, using pkgadd; and a custom action on the target computer, such as a configuration change to perform on the target computer (a command to execute) or a set of files to copy on the target computer (with an optional command to execute). Basically, you can provide a Solaris package to be installed. You can also specify a command to be executed or a set of files to be copied on the target computer with a command to be run. The wizard guides you through the specific software module creation.

Software bindings

- Two binding types:
- Automatic binding rules displayed with the notation "by rule"
- Manual bindings displayed with the notation "explicit"

Es: binding "by rule"

software bindings

The list below shows software modules bound to the selected target. Note that extra packages actual configuration deployed.
Click on a configuration in the list above to update this list accordingly..

Software	Order	binding
Echo	stage 3	by generic rule
Intel Net driver (ver. 08/20/2008 - 12/11/2008)	stage 3	by generic rule
OSd 71 for Windows	stage 4	by generic rule

```
[Software]
nsofts="3"
nsoftsdesc="3 (by rule only)"
soft0_SoftSeqID="3"
soft0_SoftItemID="2222996"
soft0_Description="Echo"
soft0_Type="pkg"
soft0_Flags="1"
soft0_Source="echo95C16059C2.pkg"
soft0_Dest="\app600002"
soft0_CmdLine="\app600002\echo.k
soft1_SoftSeqID="3"
soft1_SoftItemID="9316750"
soft1_Description="Intel Net driver (ver. 08/20/2008 - 12/11/2008)"
soft1_Type="pkg"
soft1_Content="win-drv"
soft1_Flags="0"
soft1_Source="win73AD25C30C.pkg"
soft1_Dest="\drivers\Net-A1A369"
soft2_SoftSeqID="4"
soft2_SoftItemID="3110055"
```

During deployment, the .ini file for the deployment reports the specific binding rule and the bound software module

16 Software modules © 2010 IBM Corporation

There are two binding types: automatic binding rules and manual bindings. **Automatic binding rules** are used to link software modules/OS configurations to targets without having to specifically bind them on each target. Rules are created in each software module/OS Configuration. They determine which targets are automatically bound to the software module/OS configuration. Rules are created from criteria and values. If a target has a matching value for all criteria in the rule, the software module/OS configuration is bound to that target. The bindings are displayed with the notation "by rule". You can add automatic binding rules for software modules or an OS configuration by editing them. **Manual bindings** explicitly bind software modules/OS configurations to targets to enable their automatic deployment. You can set up manual bindings on the target monitor by accessing the specific target details page and accessing the Bindings tab. You can either edit the OS Configuration bindings or the Software bindings and select the OS configuration/software module to associate with the computer. In this case, the binding is reported as "explicit." When you create a software module including drivers, you can create automatic binding rules based, for example, on the PCI hardware ID. This binding is reported in the software bindings tab as "by rule" if a specific target matches the condition. When you deploy an operating system onto a target machine, the deployment .ini file reports the software modules that match the automatic binding rules (if any) for the specific target machine.

PCI binding rules

- Data from Inventory PCI devices is used
- VendorID and SubdeviceID (VEN_<VendorID>&DEV_<DeviceID>) are matched to the hex values reported in the driver .inf file

The screenshot displays the 'Inventory' tab of the Tivoli Provisioning Manager. On the left, a tree view shows 'PCI devices' expanded. The main pane shows a table of properties for a selected device:

Property	Value
BomID	28
Bus	0
Dev	18
Fun	0
Slot	4
VendorID	8086
SubVendorID	15ad
DeviceID	100f
SubdeviceID	0750
Revision	1
Class	02
Subclass	00
ProgIf	0
IRQ	10
ClassName	Ethernet controller
Address	00C:29:EF:C9:B7
DeviceName	Intel 82545EM Gigabit Ethernet Controller (Copper)
name	Ethernet controller

Below the inventory table, a Notepad window shows the contents of 'e1000325.inf'. The following lines are highlighted, showing the mapping of PCI IDs to driver files:

```

%E101E.dev1cesck = E101E.ndf, PCI\VEN_8086&DEV_101E&SUBSYS_00011179
%E100E1BM.dev1cesck = E100Ecopper.ndf, PCI\VEN_8086&DEV_100E&SUBSYS_027E1014
%E100F.dev1cesck = E100Fcopper.ndf, PCI\VEN_8086&DEV_100F&SUBSYS_10018086
%E100F1BM.dev1cesck = E100Fcopper.ndf, PCI\VEN_8086&DEV_100F&SUBSYS_02691014
%E100F5EM.dev1cesck = E100F.ndf, PCI\VEN_8086&DEV_100F
%FERN_LGM
%E100FLOM.dev1cesck = E100Fcopper.ndf, PCI\VEN_8086&DEV_100F&SUBSYS_028E1014
%E100FLOM.dev1cesck = E100Ecopper.ndf, PCI\VEN_8086&DEV_100E&SUBSYS_027E1014
  
```

A yellow box labeled 'Driver .inf file' has a dashed arrow pointing to the VendorID (8086) and SubdeviceID (0750) in the inventory table, and another dashed arrow pointing to the corresponding entries in the .inf file.

17

Software modules

© 2010 IBM Corporation

If you specify the binding rules that are based on the PCI hardware ID, Tivoli Provisioning Manager for OS Deployment matches the PCI device IDs. The IDs that are discovered on the target through the PCI/DMI scan are matched to the values reported in the drivers .inf files included in the specific software module. If they match, the software module is bound by rule to the target.

Summary

In this module, you learned:

- The software module types available with Tivoli Provisioning Manager for OS Deployment 7.1.1
- The common deployment problems that can be avoided by using specific software modules
- Automatic and manual bindings and their uses

In this module you learned about: the various software module types that can be created with Tivoli Provisioning Manager for OS Deployment; the deployment issues that can be addressed using the specific software modules; and what are bindings, how to setup automatic or manual bindings for a software module or OS configuration and how the automatic bindings work.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_software_modules.ppt

This module is also available in PDF format at: [./software_modules.pdf](http://software_modules.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, AIX, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.