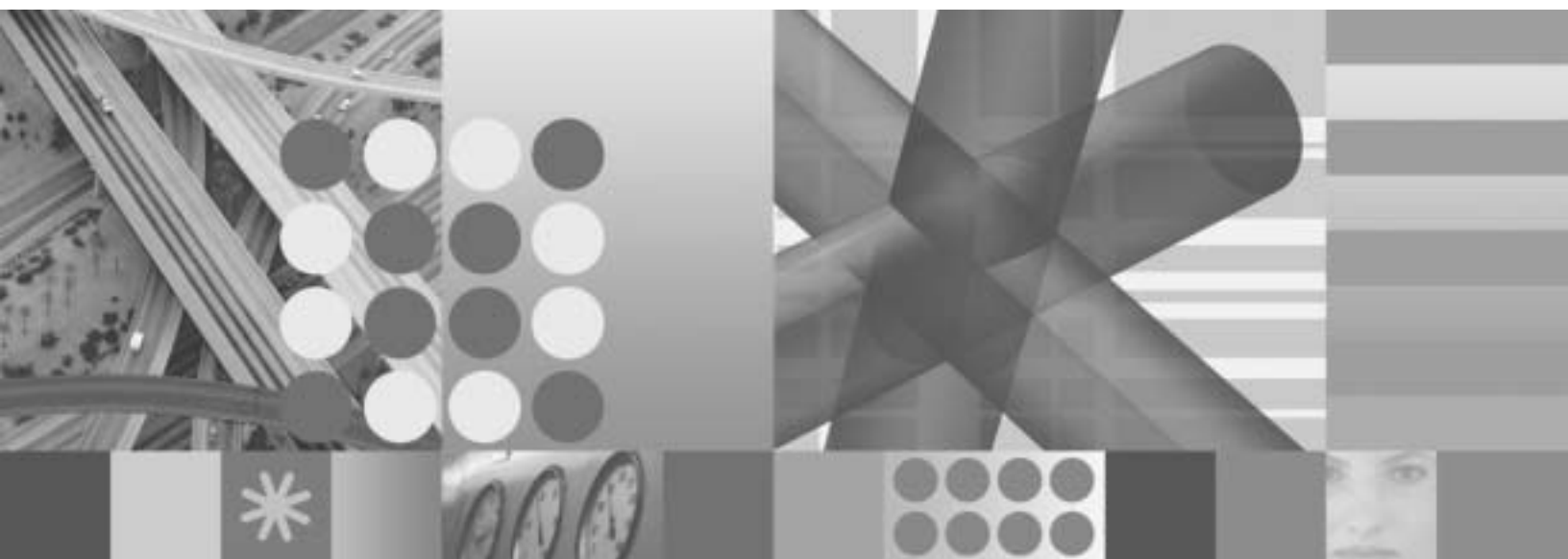




Patch Management Guide



Patch Management Guide

Note

Before using this information and the product it supports, read the information in “Notices” on page 115.

This edition applies to version 4 release 3 modification level 1 of IBM Tivoli Configuration Manager (program number 5724-C06) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-5263-03.

© **Copyright International Business Machines Corporation 2005, 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
--------------------------	----------

Tables	vii
-------------------------	------------

About this guide	ix
-----------------------------------	-----------

Who should read this guide	ix
What this guide contains	ix
Publications	x
IBM Tivoli Configuration Manager library	x
Related publications	xi
Accessing terminology online	xi
Accessing publications online	xii
Ordering publications	xii
Tivoli technical training	xii
Support information	xii
Conventions used in this guide	xiii
Typeface conventions	xiii
Operating system-dependent variables and paths	xiii

Chapter 1. Introduction	1
--	----------

Why patch management is important	1
An automated patch management solution	2
Operating systems and applications managed with this solution	3
Planning your environment	3
A road map to implement the automated patch management solution	6

Chapter 2. Installing the automated patch management solution	7
--	----------

Installing the Tivoli Configuration Manager Automation Server component	7
Hardware and operating environment requirements	7
Installing and configuring Cygwin	7
Changing locales	9
Installing the software prerequisites and Tivoli Configuration Manager Automation Server	9
Installing the Patch Management component	16
Prerequisites	16
Before you begin	16
InstallShield Server installation	17
Installing using Tivoli Software Installation Service	18
Tivoli Desktop installation	19
Tivoli command line	20
Objects created by the installation	21
Installing and upgrading the Patch Management Automation Server driver	22
Installing the Patch Management Automation Server driver	22
Upgrading the Patch Management Automation Server driver	24

Chapter 3. Configuration and administrative tasks	27
--	-----------

Installing and Configuring the Microsoft WSUS Server	27
Configuring SSH communications between Automation Server and WSUS	28
Configuring the SSH server	28
Configuring the SSH client	30
Enabling client server SSH communications	30
Checking SSH communications	30
Enabling TCP/IP communications	30
Deploying WUA and Qchain on endpoints	30
Upgrading WUA and .cab file	31
Configuring automated patch management settings	31
Configuring Automation Server to download the .cab file	34
Considerations for an interconnected environment	35
Hub and Spoke configuration examples	35
Administering Tivoli Configuration Manager Automation Server	37
Starting the Tivoli Configuration Manager Automation Server	37
Logging on to the Tivoli Configuration Manager Automation Server console	38
Stopping the Tivoli Configuration Manager Automation Server	38
Changing default passwords	38
Changing WSUS values	39
Administering the workflow	40
Running the workflow	41
Stopping the workflow	41
Checking workflow status	41
Scheduling the workflow	42

Chapter 4. Scanning for patches	43
--	-----------

Optimizing the wsusscan.cab download	43
Submitting the patch scan	44
Discovering missing and installed patches	44
Patch management tables and views	44
Information stored in the PM_PATCH_INFO table	45
Information stored in the H_PM_PATCH_INFO table	46
Information stored in the PM_PRODUCT_INFO table	47
Information stored in the PM_PATCH_PKG table	48
Information stored in the INV_GROUP table	49
Information stored in the INV_GROUP_EP table	49
Filtering View	50

Chapter 5. Patch installation	51
--	-----------

Deployment Paradigms	51
Performing the initial patch scan	52
An end-to-end scenario	53
Managing emergency patches	55

TCM_Emergency_Patches Workflow	56
Deploying patches to groups of endpoints	57
Viewing the software packages	58
Viewing the activity plan	59
Submitting and monitoring the plan	60
Checking the results	60
Managing the patch lifecycle	61
Customizing the software package and plan templates	61
Customizing the software package templates	63
Customizing the activity plan templates	63
Customizing the target filter template	65

Chapter 6. Automated patch management command line. 67

wseccfg.	68
wseccgenplan	77
wseccgensp.	80
wseccgrp	83
wseccrpt	85
wtransfer	90

Chapter 7. Troubleshooting 93

Automation Server logs	93
Log locations	93
Automation Server workflow logging levels	94
Tivoli Configuration Manager Automation Server silent installer log	95
Tivoli Configuration Manager Automation Server start and stop logs	96
Tivoli Configuration Manager Automation Server prerequisites uninstall logs	96
Patch management component logs and traces	96
Patch Management environment installation logs and traces	96

Patch Management command line traces	96
Patch Management environment uninstall logs and traces	96
Common problems and troubleshooting scenarios	96
Problems with the Tivoli Configuration Manager Automation Server silent installation	97
Automation Server workflow	97
Other common problems	104

Appendix A. Uninstalling the automated patch management solution. 109

Uninstall the product from the Tivoli server	109
Uninstalling the Tivoli Configuration Manager Automation Server	109

Appendix B. Support information . . . 111

Searching knowledge bases	111
Searching the information center	111
Searching the Internet	111
Obtaining fixes	111
Receiving weekly support updates	112
Contacting IBM Software Support	113
Determining the business impact	113
Describing problems and gathering information	114
Submitting problems	114

Notices 115

Trademarks	117
----------------------	-----

Index 119

Figures

1. Patch management network topology 4

Tables

1.	Automated patch management environment	4	11.	Patch Management component on Hub only	35
2.	Patch tools	5	12.	Default User ID	38
3.	Roadmap	6	13.	Workflow schedule values	42
4.	Cygwin packages	8	14.	Patch management tables	44
5.	List of CDs	9	15.	Patch management views	45
6.	Response file values.	12	16.	Activity plan naming convention	59
7.	Files to download	16	17.	Software package and plan templates	62
8.	Objects created after installation.	21	18.	Tivoli Configuration Manager Automation Server logs.	93
9.	tpm_update.req file values	23	19.	Tasks performed by the Automation Server workflow	98
10.	Patch Management component on Hub and Spoke	35			

About this guide

The purpose of this guide is to describe how you can implement an automated patch management solution in a Windows® environment, taking advantage of the functions provided by IBM® Tivoli® Configuration Manager version 4.3.1.

This guide explains how to install and configure the environment and how to put in place a solution that collects new patches, scans endpoints to provide inventory of installed and missing patches, and distributes the missing patches.

Who should read this guide

This guide is intended for IT specialists and administrators who want to implement an automated patch management solution using IBM Tivoli Configuration Manager, version 4.3.1. This solution covers the distribution and management of Microsoft patches on Windows endpoints in a Tivoli environment.

Readers should be familiar with the following topics:

- Windows operating systems
- Tivoli environment
- IBM Tivoli Configuration Manager environment
- Supported database architectures and concepts
- Microsoft® Windows Server Update Services (WSUS)

What this guide contains

This guide contains the following sections:

- Chapter 1, “Introduction”
Provides an introduction to the automated patch management solution, including prerequisites, platforms and applications managed with this solution. It also discusses basic planning considerations before installing an automated patch management solution.
- Chapter 2, “Installing the automated patch management solution”
Describes how to install the automated patch management solution and which steps and operations to perform to start using it.
- Chapter 3, “Configuration and administrative tasks”
Describes configuration and administrative tasks to be performed after installation.
- Chapter 4, “Scanning for patches”
Describes how the inventory component of IBM Tivoli Configuration Manager collects information about which patches are installed and missing in the environment.
- Chapter 5, “Patch installation”
Describes how software packages and activity plans are automatically generated to install security and software patches.
- Chapter 6, “Automated patch management command line”
Describes the syntax and arguments for the automated patch management command line, as well as examples.

About this guide

- Chapter 7, “Troubleshooting”
Describes how to manage logs and traces in the automated patch management solution and how to perform problem determination.
- Appendix A, “Uninstalling the automated patch management solution”
Describes how to uninstall the automated patch management solution.
- Appendix B, “Support information”
Describes options for obtaining support for IBM products.

Publications

This section lists publications in the IBM Tivoli Configuration Manager library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

IBM Tivoli Configuration Manager library

The following documents are available in the IBM Tivoli Configuration Manager library:

- *IBM Tivoli Configuration Manager: Introducing IBM Tivoli Configuration Manager*, GC23-4703
Provides an introduction to the product.
- *IBM Tivoli Configuration Manager: Planning and Installation*, GC23-4702
Describes how to plan for installing Tivoli Configuration Manager components and how to perform the installation.
- *IBM Tivoli Configuration Manager: User's Guide for Software Distribution*, SC23-4711
Provides user information about how to use the Software Distribution component of Tivoli Configuration Manager.
- *IBM Tivoli Configuration Manager: Reference Manual for Software Distribution*, SC23-4712
Provides advanced information about how to use and customize the Software Distribution component of Tivoli Configuration Manager.
- *IBM Tivoli Configuration Manager: User's Guide for Deployment Services*, SC23-4710
Provides information about the different services provided as part of Tivoli Configuration Manager.
- *IBM Tivoli Configuration Manager: Database Schema Reference*, SC23-4783
Provides information about the configuration repository of Tivoli Configuration Manager.
- *IBM Tivoli Configuration Manager: User's Guide for Inventory*, SC23-4713
Describes the Inventory component and the management tasks that you can perform.
- *IBM Tivoli Configuration Manager: Messages and Codes*, SC23-4706
Describes the messages issued by Tivoli Configuration Manager, its components, and its services.
- *IBM Tivoli Configuration Manager: Guide for Microsoft Active Directory Integration*, SC32-2285
Describes how you can integrate the Microsoft Active Directory environment with the Tivoli environment.
- *IBM Tivoli Configuration Manager: User's Guide for Operating System Deployment Solution*, SC32-2578
Describes how you can implement an operating system imaging solution.

- *IBM Tivoli Configuration Manager: License Management Extension*, SC32-2260-01
Describes how you can implement the license management facilities in your IBM Tivoli Configuration Manager environment.
- *IBM Tivoli Configuration Manager: Release Notes*, GI11-0926
Provides late-breaking information about Tivoli Configuration Manager, its components, and its services.

Related publications

The following documents also provide useful information:

- *Tivoli Management Framework: Planning for Deployment Guide*, GC32-0803
Explains how to plan for deploying your Tivoli environment.
- *Tivoli Management Framework: User's Guide*, GC32-0805.
Describes the concepts and procedures for using Tivoli Management Framework services.
- *Tivoli Management Framework: Reference Manual* GC32-0806
Provides in-depth information about Tivoli Management Framework commands.

The following documents are useful for information about Tivoli Provisioning Manager, on which the Automation Server technology is based:

- *Tivoli Intelligent ThinkDynamic Orchestrator Installation Guide*
- *Tivoli Intelligent ThinkDynamic Orchestrator Migration Guide*
- *Tivoli Intelligent ThinkDynamic Orchestrator Release Notes*
- *Tivoli Intelligent ThinkDynamic Orchestrator Problem Determination Guide*

The Tivoli Intelligent ThinkDynamic Orchestrator library can be found at <http://publib.boulder.ibm.com/tividd/td/IBMTivoliProvisioningManager2.1.html>. The product documentation is also available in the online help, which can be launched from the Automation Server administrative console Web interface.

The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available at the following Tivoli software library Web site:

<http://www.ibm.com/software/tivoli/library/>

Access the glossary by clicking the **Glossary** link on the left pane of the Tivoli software library window.

Accessing terminology online

The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available at the following Tivoli software library Web site:

<http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm>

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

<http://www.ibm.com/software/globalization/terminology>

Accessing publications online

The product CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both. To access the publications using a Web browser, open the `infocenter.html` file. The file is in the appropriate publications directory on the product CD.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:

<http://publib.boulder.ibm.com/tividd/td/link/tdprodlist.html>

Click **Tivoli product manuals**. In the Tivoli Technical Product Documents Alphabetical Listing window, click **IBM Tivoli Configuration Manager** to access your product library at the Tivoli software information center.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File** » **Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.link.ibm.com/public/applications/publications/cgibin/pbi.cgi>

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

<http://www.ibm.com/software/tivoli/education>

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see Appendix B, “Support information,” on page 111.

Conventions used in this guide

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Words defined in text
- Emphasis of words (words as words)
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This guide uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace *\$variable* with *%variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. The names of environment variables are not always the same in Windows and UNIX®. For example, %TEMP% in Windows is equivalent to \$tmp in UNIX.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

About this guide

Chapter 1. Introduction

This chapter provides an introduction to the automated patch management solution and concepts, detailing the following:

- “Why patch management is important”
- “An automated patch management solution” on page 2
- “Operating systems and applications managed with this solution” on page 3
- “Planning your environment” on page 3
- “A road map to implement the automated patch management solution” on page 6

Why patch management is important

Today more and more companies rely on their information technology infrastructure to enhance their business, save them money, and increase their profitability. The threat of malicious virus and worm attacks has been increasing thus forcing businesses to reinvestigate their security needs to better protect their environment.

Combatting these attacks on the Microsoft platform has become a high priority. In response, Microsoft produces security patches for their system vulnerabilities and makes them available to users. Research has shown that the most efficient way to be protected against attacks is to ensure that every machine in the environment has the latest patches installed. If one computer in the environment is not patched, it can threaten the stability of the entire environment and possibly inhibit normal functionality.

Today’s environments demand that the solution to managing patches for Microsoft operating systems and applications be an automated yet stringently controlled process. The IBM Tivoli Configuration Manager automated patch management solution is a solution designed to address this need. An automated patch management solution involves the following processes:

Assess the vulnerability

Audit software in your production environment, evaluate potential security threats, vulnerabilities and non-compliances. This requires accurate inventory of IT assets to assess exposures.

Automated patch management relies on the Inventory component of Configuration Manager to perform scans that use the Windows Update Agent (WUA) to scan endpoints and report information about found and missing patches.

Patch identification and download

Determine a reliable, timely source of information on software updates and a documented and secure download process.

Automated patch management uses Windows Server Update Services (WSUS) for downloading fixes for Windows operating systems and applications.

Patch testing

Validate a given patch in a test environment, provide the assurance that all

Why patch management is important

necessary packages, pre-requisites, co-requisites, conflicts have been identified before deploying to production.

Patches can be deployed in a test environment to troubleshoot problems before patches are deployed in the enterprise.

Patch approval

Maintain strict control over what is being changed, which vulnerability the fix addresses, what services and applications are being impacted, and priority. Requires an approval process.

You use the WSUS interface to approve patches so that the automated patch management solution automatically creates software packages only for patches that have been approved.

Patch deployment

Prioritize the urgency of the patch deployment, schedule the deployment, build the installable unit, and deploy the patch.

An automated process generates software packages and activity plans, and then notifies the Administrator when they are ready to be submitted. The process relies on IBM Tivoli Configuration Manager components and services, such as, Software Distribution and Activity Planner.

Patch verification

Validate that the patch was successfully applied on all eligible endpoints.

The automated patch management command line can be used to retrieve patch status information. Patch installations can also be monitored from the Activity Plan Monitor graphical user interface where activity plans are submitted.

Compliance management

Update the configuration baseline definitions to include the new patches, regularly analyse to assure that all endpoints remain in compliance, identify improvements and customize the patch management process accordingly.

Automated patch management is a dynamic process designed to identify any missing patches in your environment and to automatically create patches to cover the current vulnerabilities.

An automated patch management solution

IBM Tivoli Configuration Manager version 4.3.1 offers an automated patch management solution using Tivoli Provisioning Manager technology. Configuration Manager, with the addition of this technology, automates a series of otherwise manual steps to maintain your network at a desired level of installed patches. The automation technology is managed by the Tivoli Configuration Manager Automation Server.

The Automation Server automates a sequence of Configuration Manager operations defined in the Automation Server workflow. A workflow is a sequenced set of commands that run in a synchronous manner. The workflow is designed to automate some of the processes involved in implementing an automated patch management solution in your network environment. The Automation Server automates these operations by running the workflow from the Automation Server administrative console Web interface.

You can run the Automation Server workflow from either the Automation Server administrative console Web interface or by modifying a configuration file, TEDWScheduler.ini. See “Administering the workflow” on page 40.

Note: This guide might refer you to the Tivoli Provisioning Manager library for reference information and troubleshooting purposes. Refer to <http://publib.boulder.ibm.com/tividd/td/IBMTivoliProvisioningManager2.1.html> to view the library. The Tivoli Provisioning Manager library is part of the IBM Tivoli Intelligent ThinkDynamic Orchestrator library. The documentation is also available in the online help, which can be launched from the Automation Server administrative console Web interface.

Operating systems and applications managed with this solution

The solution manages Microsoft patches, service packs, and update rollups for the following operating systems and applications:

- Windows 2000 Professional SP™ 3 or later
- Windows 2000 Server SP 3 or later
- Windows 2000 Advanced Server SP 3 or later
- Windows XP Professional
- Windows XP Home Edition
- Windows 2003 Server Standard Edition
- Windows 2003 Server Enterprise Edition
- Windows 2003 Server Web Edition
- Windows Vista (valid with WSUS server 2.1)
- Windows Server 2008 Standard (ix86 only)
- Windows Server 2008 Enterprise (ix86 only)
- Internet Explorer 5.01 or later
- Media Player 6.4 or later
- Exchange 2000 Server
- Exchange Server 2003
- Microsoft Office 2000
- Microsoft Office XP
- Microsoft Office 2003

Note: No operating system patches are supported for Windows Server 2008, only patches related to software applications.

Planning your environment

Before starting the installation and configuration of an automated patch management solution, some basic considerations of your environment need to be made. With the addition of the patch management component to the IBM Tivoli Configuration Manager environment, the network topology is extended as follows:

Planning your environment

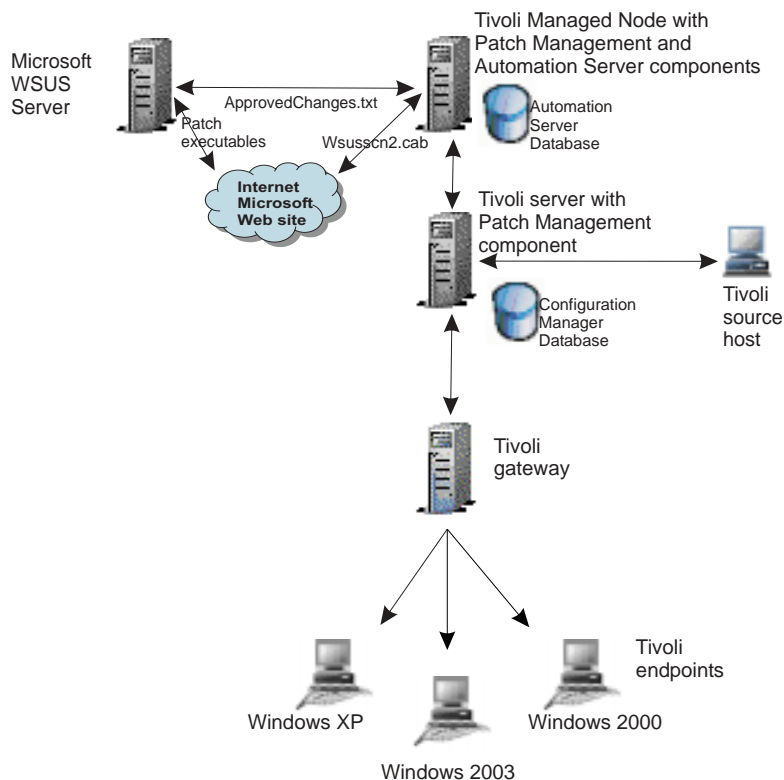


Figure 1. Patch management network topology

Table 1 lists the resources in the automated patch management environment:

Table 1. Automated patch management environment

Resource	Component	Description
Tivoli management region server	Patch Management	A Tivoli server component to perform patch management. It is installed on the Tivoli server.
Tivoli managed node	Automation Server Patch Management	A Windows 2003 Standard Edition machine on which to install the Automation Server component. This component provides automation in downloading files required by the solution, downloading patches, and preparing all the elements needed to perform the patch installation.
Windows Server Update Services (WSUS)		A server machine used to implement the approval mechanism of patches and to download up-to-date patches from the Microsoft Web site.

Table 2 on page 5 lists the tools used in the automated patch management environment:

Table 2. Patch tools

Resource	Component	Description
Patch tools	Patch Management	<p>The solution uses the following Microsoft tools:</p> <p>Windows Update (WU) offline scan file It is the security policy catalog in a Microsoft environment where missing patches are discovered running WUA. This file is downloaded and updated regularly using an automated process defined in the Automation Server workflow. For details on the .cab file changes, see http://support.microsoft.com/kb/926464.</p> <p>Depending on the installed WUA, Patch Management solution supports one of the following .cab files:</p> <p>wsusscn2.cab This new offline scan file is available in addition to the existing WU offline scan file, Wsusscan.cab. The Wsusscn2.cab offline scan file has a new format. To download the updated version of the .cab file see http://go.microsoft.com/fwlink/?LinkId=74689.</p> <p>To manage mixed environments, where different levels of WUA are installed, deploy the new .cab file only after having updated WUA on all endpoints. Note: The wsusscn2.cab file is renamed into wsuscan.cab during the download process. After updating the catalog with the new wsusscn2.cab file, verify if the date of the wsuscan.cab file matches the size and date of the new wsusscn2.cab file.</p> <p>wsusscan.cab You can download this file from http://go.microsoft.com/fwlink/?LinkId=39043. By March 2007, Microsoft will no longer support the existing WU offline scan file. At that time, you will have to use the new Wsusscn2.cab file.</p> <p>QChain.exe Enables the installation of multiple patches without restarting the computer between each installation.</p>

A road map to implement the automated patch management solution

Table 3 outlines the sequence of steps necessary to get the automated patch management solution up and running.

Table 3. Roadmap

Step	Task	Refer to...
1	Install IBM Tivoli Configuration Manager, version 4.3.1 and ensure that the Inventory Integration with Software Distribution is enabled using the <code>wswdmgr -s is_cmstatus_enabled=true</code> command.	<i>Planning and Installation Guide, Reference Manual for Software Distribution</i>
2	Install Tivoli Configuration Manager Automation Server component	"Installing the Tivoli Configuration Manager Automation Server component" on page 7
3	Install IBM Tivoli Configuration Manager, version 4.3.1	<i>Planning and Installation Guide</i>
5	Download WUA and qchain.exe	"Before you begin" on page 16
6	Configure the automated patch management settings using <code>wseccfg</code> command. Before deploying the patches, you can also decide to organize your endpoints in different groups.	"Configuring automated patch management settings" on page 31 "Deploying patches to groups of endpoints" on page 57
7	Run the Automation Server workflow to download the latest .cab file	"Running the workflow" on page 41
8	Move WUA and qchain.exe files to the endpoints	"Deploying WUA and Qchain on endpoints" on page 30
9	Run the Windows_Initial_Patch_Scan	"Submitting the patch scan" on page 44
10	Approve patches on WSUS Web site	Refer to Windows Server Update Services documentation.
11	Run the Automation Server workflow	"Running the workflow" on page 41
12	Submit activity plans	"Submitting and monitoring the plan" on page 60
13	Run the <code>wsecrpt</code> command to retrieve report of installed patches	"Checking the results" on page 60

Chapter 2. Installing the automated patch management solution

This chapter describes how to install the components required for the automated patch management solution. To install the solution, perform the following steps:

1. Install IBM Tivoli Configuration Manager, version 4.3.1. Refer to *Planning and Installation Guide*.
2. Install the Automation Server component, using the silent install method, on a Windows 2003 managed node. Refer to “Installing the Tivoli Configuration Manager Automation Server component”
3. Install the Patch Management component on the Tivoli server and on the managed node where the Automation Server component is installed. Optionally install this component on a managed node from where you want to run patch management commands. Refer to “Installing the Patch Management component” on page 16.

Installing the Tivoli Configuration Manager Automation Server component

The automated patch management solution relies on automated processes managed by the Tivoli Configuration Manager Automation Server. The automated processes are defined in the Automation Server workflow, Group_Status_Updater. The Automation Server requires a TCP/IP connection to access the Microsoft WSUS server. Avoid installing this component on the Tivoli server. Install it on a dedicated managed node machine. The following sections describe how to fully implement the Tivoli Configuration Manager Automation Server.

Hardware and operating environment requirements

This section describes the hardware and operating environment required to install the Tivoli Configuration Manager Automation Server. Ensure that:

1. You have an IBM Compatible PC with the hardware requirements specified below:
 - 2.8 GHz Intel® Pentium® 4 processor or equivalent
 - Minimum 4 GB of memory
 - Minimum 20 GB of free disk space
 - CD drive
2. Your machine meets the operating environment requirement specified below:
 - Windows Server 2003 Standard Edition

Installing and configuring Cygwin

This section provides details about installing and configuring Cygwin on the Tivoli Configuration Manager Automation Server before running the silent installer. Cygwin is required for both the silent installer and workflows within Tivoli Configuration Manager Automation Server. Cygwin Version 1.5.12.0 or 1.5.18.0 must be installed. For information on obtaining and installing Cygwin, refer to the following Web site: <http://www.cygwin.com>.

To install Cygwin:

Installing and configuring Cygwin

1. Ensure you have Windows Server 2003 Standard Edition.
2. Log on using a user account with administrator privileges.
3. Run Cygwin Setup and select **All Users** in the **Install for** option and **Unix** in the **Default Text File Type** option.
4. During the installation of the Cygwin tool, on the **Choose Installation Directory** panel, ensure that you select **UNIX** in the **Default Text File Type** box.

Note: If you installed Cygwin selecting DOS as the default text file type, you can modify this setting by entering the following command in a command prompt:

```
eval "`mount -m | grep -i cygwin | sed 's/-t/-b/'`"
```

5. In the **Select Package** panel, select the packages listed in Table 4:

Table 4. Cygwin packages

Category	Package
Admin	Default, cron, cygrunsrv, shutdown
Archive	Default, cabextract, sharutils, unzip, zip
Base	All default packages
Database	All default packages
Devel	All default packages, cvs
Doc	All default packages, perl-manpages
Editors	All default packages, ed, vim
Gnome	All default packages
Graphics	All default packages
Interpreters	Expect, gawk, perl
Libs	All default packages
Mail	All default packages
Math	All default packages
Misc	All default packages
Net	All default packages, inetutils, openssh, ping, rsync
Publishing	All default packages
Shells	All default packages (ensure that ash and bash are selected)
System	All default packages
Text	All default packages, more
Utils	All default packages, ccrypt, cpio, clear, cygutils, time, file, keychain
Web	All default packages, wget
+X11	All default packages
ZZZRemovedPackages	All default packages
+_PostInstallLast	All default packages

Note: The location of specific packages can vary depending on the version of Cygwin you are installing.

Changing locales

This section describes the steps to perform prior to the Tivoli Configuration Manager Automation Server installation if you are using a locale other than **English**.

For all other languages except **Korean**, follow these instructions:

1. Click **Start-> Control Panel-> Regional and Language Options**.
2. Under the **Regional Options** tab, select **English (United States)** from the language drop down.
3. Click **Advanced** and select **English (United States)** from the list of languages.
4. Click **OK**.
5. Reboot your system for the changes to take place.
6. Once the Tivoli Configuration Manager Automation Server installation is completed, you can change the locale back to the original value.

For **Korean**, follow these instructions:

1. Click **Start-> Control Panel-> Regional and Language Options**.
2. Under the **Regional Options** tab, select **English (United Kingdom)** from the list of languages.
3. Click **Advanced** and select **English (United Kingdom)** from the list of languages.
4. Click **OK**.
5. Reboot your system for the changes to take place.
6. Once the Tivoli Configuration Manager Automation Server installation is completed, you can change the locale back to the original value.

Installing the software prerequisites and Tivoli Configuration Manager Automation Server

This section describes how to install the prerequisite software and Tivoli Configuration Manager Automation Server using the silent installation method. With this method, you customize a template response file which contains all your selected installation options. You can then reuse this template file if you need to reinstall at any time.

Before you begin

Ensure that you meet these requirements before you complete the procedures in this chapter:

1. You have the hardware and operating environment requirements as specified in “Hardware and operating environment requirements” on page 7.
2. You have completed the steps in “Installing and configuring Cygwin” on page 7.
3. If applicable, you have completed the steps in “Changing locales” on page 9.
4. You have all the CDs as specified in Table 5 on page 9. The CDs are at 4.2.3 level.

Table 5. List of CDs

CD	CD label	Function
1	IBM Tivoli Configuration Manager Automation Server, version 4.2.3	Installs the Tivoli Configuration Manager Automation Server for Windows application.

Installing software prerequisites and Configuration Manager Automation Server

Table 5. List of CDs (continued)

CD	CD label	Function
2	IBM Tivoli Configuration Manager Pre-requisite Software Installer for Automation Server, version 4.2.3	Installs the prerequisite software and workflows required.
3	IBM Tivoli Configuration Manager Generic Fixes, version 4.2.3	Installs the Tivoli Directory Server fix, the WebSphere® Application Server 5.1 cumulative fix 3, WebSphere Enabled Messaging Interim fixes for WebSphere Application Server 5.1.
4	IBM Tivoli Configuration Manager WebSphere Application Server Version 5.1 for Windows, version 4.2.3	Installs the WebSphere Application Server 5.1 for Windows.
5	IBM Tivoli Configuration Manager DB2® Workgroup Server Version 8.1 for Windows, version 4.2.3	Installs the DB2® Universal Database™ Workgroup Server Workgroup Server 8.1 for Windows.
6	IBM Tivoli Configuration Manager DB2 Workgroup Server Version 8.1 fix pack 3 for Windows, version 4.2.3	Installs the fix pack 3 for the DB2 Universal Database Workgroup Server 8.1 for Windows.
7	IBM Tivoli Configuration Manager Tivoli Directory Server Version 5.2 for Windows, version 4.2.3	Installs the Tivoli Directory Server 5.2 for Windows.
8	IBM Tivoli Configuration Manager Automation Server fix pack 1, version 4.2.3	Installs the fix pack 1 for the Tivoli Configuration Manager Automation Server for Windows application.

About the installation

A response file contains all the information necessary to install Tivoli Configuration Manager Automation Server. It is a plain text file consisting of sections containing data entries. InstallShield reads the necessary input from the response file at run time while performing a silent installation.

Note: It is mandatory that you do not have any of the software prerequisites (for example, DB2 Universal Database Workgroup Server, Tivoli Directory Server or WebSphere Application Server) already installed on your machine as specific versions of these prerequisites are required.

Guidelines for silent installation: Follow these guidelines for a successful silent installation:

- Do not create an original response file, use the template provided on the CD.
- Save a copy of the response file before making any edits.
- Copy all the install images to a hard disk. The path information will be required for the response file.
- Do not modify parameters, except to edit their values.
- Do not remove a parameter, even if it does not have a value.
- Do not add a parameter.
- Use these guidelines when editing the values:
 - Replace any value that you delete. If the parameter is required, installation or configuration could fail.
 - Retain the case of the original value.

Silent installation requirements

To run the Tivoli Configuration Manager Automation Server installer using silent installation, you must first complete the tasks below:

1. Create an installation image depot that contains the install images of all the middleware products that are to be installed. See the “Creating an installation image depot for the Tivoli Configuration Manager Automation Server CDs” on page 11 for details about creating the installation image depot.
2. Edit the response file template. Refer to the section “Installing using the response file template” on page 11 for more information.

Creating an installation image depot for the Tivoli Configuration Manager Automation Server CDs

You cannot install using multiple installation CDs. You must copy all installation media to your hard drive or other file system. The installation program does not prompt for media when running silently. Given below are guidelines to create directories for the installation media on your hard disk.

1. Log on as the local Administrator on the local machine console.

Note: Tivoli does not support installation from a terminal server or cross-domain installations.

2. Create a main directory for Tivoli Configuration Manager Automation Server, such as:

`\tcmas_install`

Note: Do not name this directory Disk or any other name starting with Disk.

3. Create a subdirectory for each of the CDs provided with the Tivoli Configuration Manager Automation Server package, such as:

Tivoli Configuration Manager Automation Server

`\tcmas_install\Automation Server`

Tivoli Configuration Manager Generic Fixes

`\tcmas_install\Generic Fixes`

4. Place each CD in the CD-ROM drive, and copy its contents into the subdirectory you created.
5. Repeat for all the CDs.

Important

IBM Tivoli Configuration Manager includes an enhanced version of Automation Server installation, that runs several additional checks.

To run this new installation, perform the following steps:

1. Remove the contents of the `tpm_install` directory, you copied from the **Pre-requisite Software Installer for Automation Server** CD, version 4.2.3.
2. Copy the new contents of the `tpm_install` directory from the `tpm_install` directory of IBM Tivoli Configuration Manager CD 5.

For more information on the new installation, see “Problems with the Tivoli Configuration Manager Automation Server silent installation” on page 97.

Installing using the response file template

To install the Tivoli Configuration Manager Automation Server, using the response file template that was provided in the CD, do the following:

Installing software prerequisites and Configuration Manager Automation Server

1. Ensure you are still logged on as the local Administrator and run the following command:

```
change user /install
```

Note: InstallShield determines if the prerequisite software is installed on the system by reading the vpd.properties file. InstallShield sometimes writes to the directory C:\Documents and Settings\Administrator\Windows\vpd.properties and sometimes to C:\Windows\vpd.properties. This causes problems for the installer if it reads from the wrong vpd.properties file to determine if all the prerequisite software is installed on the system. To avoid this problem, run the change user /install command to change windows to installation mode and force reading or writing to the C:\Windows\vpd.properties path.

2. Locate the response file template tpm_install.req within the directory where the contents of the **Pre-requisite Software Installer for Automation Server** CD were copied. The file is located in the tpm_install subdirectory.
3. Rename and save a copy of the original template. Use the original copy as your working copy.
4. Open the working copy in Cygwin and change the parameter values in the file as appropriate for your choice of products and configuration. Refer to the section “Specifying the response file values” below for the values for each variable.
5. After editing and saving the response file, open a Cygwin bash window and start the installer from the tpm_install subdirectory by typing the following command:

```
./tpm_install.sh
```
6. The installer begins to install all the options that were selected in the response file. The Cygwin window displays a bar indicating progress.
7. The system automatically reboots twice during the installation process. Following each reboot, you must log on as the local Administrator for the installation to continue.
8. After installation, refer to the log file to determine if the silent installation was successful. The tpm_install.log file is located in the *BASE_DIR/TPM_SRC/tpm_install/* directory.

Specifying the response file values: To edit the response file template:

1. Specify a value for a setting by replacing the *<value>* variable with the appropriate values. Refer to Table 6 on page 12 for a description of each of the variables.
2. Save the changes to the file.

Note: Installation on a drive other than C:\ is supported. The name you choose for the installation directory and all subdirectories must be specified in the response file template.

Table 6. Response file values

User Input Field	Description
HOSTNAME=" <i><value></i> "	Replace <i><value></i> with the fully qualified name of the Tivoli Configuration Manager Automation Server.

Table 6. Response file values (continued)

User Input Field	Description
<code>BASE_DIR="<i><value></i>"</code>	<p>Replace <i><value></i> with the full path of the Tivoli Configuration Manager Automation Server main directory. For example:</p> <pre>BASE_DIR="C:\tcmas_install"</pre> <p>This directory will contain subdirectories for each of the installation CDs.</p>
<code>PICS_SRC="<i><value></i>"</code>	<p>Replace <i><value></i> with the path where the contents of the Pre-requisite Software Installer for Automation Server CD are copied. Note that it is a relative path to the BASE_DIR directory. For example:</p> <pre>PICS_SRC="Prerequisite_Software"</pre>
<code>DB2_SRC="<i><value></i>"</code>	<p>Replace <i><value></i> with the path where the contents of the DB2 Universal Database Workgroup Server Version 9.5 for Windows CD are copied. Note that it is a relative path to the BASE_DIR directory.</p>
<code>DB2_FIXPACK_SRC="<i><value></i>"</code>	<p>Replace <i><value></i> with the path where the contents of the DB2 Universal Database Workgroup Server Version 9.5 for Windows fix pack 3 CD are copied. Note that it is a relative path to the BASE_DIR directory.</p>
<code>DB2_HOME="<i><value></i>"</code>	<p>Replace <i><value></i> with the target directory for the DB2 Universal Database Workgroup Server Version 9.5 for Windows installation. You can use directory names with spaces only if you specify the abbreviated form that appears if you enter <code>dir /x</code>. For example, <code>C:\program files</code> must be specified using the abbreviated form containing the tilde character as follows: <code>C:\progra~1</code>. If the directory does not exist, it will be automatically created.</p>
<code>DB2_ADMIN_USER="<i><value></i>"</code>	<p>Replace <i><value></i> with the user ID used to manage the DB2 Universal Database Workgroup Server instance. The following characters are not supported:</p> <ul style="list-style-type: none"> • \ • / • [] • : • • < • > • + • = • ; • , • ? • * • @
<code>DB2_ADMIN_PWD="<i><value></i>"</code>	<p>Replace <i><value></i> with the password for the user ID used to manage the DB2 Universal Database Workgroup Server instance.</p>

Installing software prerequisites and Configuration Manager Automation Server

Table 6. Response file values (continued)

User Input Field	Description
<code>ITDS_SRC=<value></code>	Replace <code><value></code> with the path where the contents of the Tivoli Directory Server Version 5.2 for Windows CD are copied. Note that it is a relative path to the <code>BASE_DIR</code> directory.
<code>ITDS_HOME=<value></code>	Replace <code><value></code> with the target directory for the Tivoli Directory Server Version 5.2 for Windows installation. You can use directory names with spaces only if you specify the abbreviated form that appears if you enter <code>dir /x</code> . For example, <code>C:\program files</code> must be specified using the abbreviated form containing the tilde character as follows: <code>C:\progra~1</code> . If the directory does not exist, it will be automatically created.
<code>ITDS_DB_HOME=<value></code>	Replace <code><value></code> with a local drive letter to be used for the creation of the LDAP database. For example: <code>ITDS_DB_HOME="C"</code> . You can only enter one character.
<code>ITDS_ADMIN_DN=<value></code>	Replace <code><value></code> with the DN value for the Tivoli Directory Server administrator. For example: <code>ITDS_ADMIN_DN="cn=root"</code> The following characters are not supported: <ul style="list-style-type: none"> • \ • / • [] • ; • • < • > • + • = • ; • , • ? • * • @ The syntax for this key is as follows: <code>cn=string</code> .
<code>ITDS_ADMIN_PWD=<value></code>	Replace <code><value></code> with the password for the Tivoli Directory Server administrator.
<code>ITDS_DB_ADMIN_USER=<value></code>	Replace <code><value></code> with the user ID used by Tivoli Configuration Manager Automation Server to connect to the LDAP server.
<code>ITDS_DB_ADMIN_PWD=<value></code>	Replace <code><value></code> with the password for the user ID used by Tivoli Configuration Manager Automation Server to connect to the LDAP server.

Table 6. Response file values (continued)

User Input Field	Description
<code>ITDS_DB_NAME=<value></code>	<p>Replace <value> with the name of the LDAP database. The following characters are not supported:</p> <ul style="list-style-type: none"> • \ • / • : • * • ? • " • < • > •
<code>WAS_SRC=<value></code>	Replace <value> with the path where the contents of the WebSphere Application Server Version 5.1 for Windows CD are copied. Note that it is a relative path to the BASE_DIR directory.
<code>WAS_HOME=<value></code>	Replace <value> with the target directory for the WebSphere Application Server Version 5.1 for Windows installation. You can use directory names with spaces only if you specify the abbreviated form that appears if you enter <code>dir /x</code> . For example, <code>C:\program files</code> must be specified using the abbreviated form containing the tilde character as follows: <code>C:\progra~1</code> . If the directory does not exist, it will be automatically created.
<code>FIX_DIR=<value></code>	Replace <value> with the path where the contents of the Generic Fixes CD are copied. Note that it is a relative path to the BASE_DIR directory.
<code>TPM_SRC=<value></code>	Replace <value> with the path where the contents of the Tivoli Configuration Manager Automation Server CD are copied. Note that it is a relative path to the BASE_DIR directory.
<code>TPM_HOME=<value></code>	Replace <value> with the target directory for the Tivoli Configuration Manager Automation Server installation. You can use directory names with spaces only if you specify the abbreviated form that appears if you enter <code>dir /x</code> . For example, <code>C:\program files</code> must be specified using the abbreviated form containing the tilde character as follows: <code>C:\progra~1</code> . If the directory does not exist, it will be automatically created.
<code>TIO_ADMIN_PWD=<value></code>	Replace <value> with the password for the Tivoli Configuration Manager Automation Server administrator. Ensure that the password complies with the password policy on the Win2k3 server.
<code>TIO_DB_NAME=<value></code>	Replace <value> with the name of the Tivoli Configuration Manager Automation Server database.
<code>BASE_DN=<value></code>	<p>Replace <value> with the BaseDN value of the LDAP server. This value should match the real domain of the Tivoli Configuration Manager Automation Server. For example:</p> <p><code>BASE_DN="dc=ibm,dc=com"</code></p> <p>The syntax for this key is as follows: <code>cn=string</code>.</p>

Table 6. Response file values (continued)

User Input Field	Description
<code>TPM_FIXPACK1=<value></code>	Replace <value> with the path where the contents of the Tivoli Configuration Manager Automation Server fix pack 1 CD are copied. Note that it is a relative path to the BASE_DIR directory.
<code>SUS_SERVER_NAME=<value></code>	Specify any value because this key is no longer used.
<code>SUS_IP_ADDRESS=<value></code>	Specify any value because this key is no longer used. Ensure that you specify a value with the correct syntax.
<code>SUS_SERVER_USER=<value></code>	Specify any value because this key is no longer used.
<code>SUS_SERVER_PASSWORD=<value></code>	Specify any value because this key is no longer used.

Installing the Patch Management component

You can use the InstallShield wizard server installation, SIS, Tivoli Desktop or Tivoli command line to install the Patch Management component on the Tivoli management region server (Tivoli server) or any managed node in your environment. By installing the component on the managed nodes you can use the command line from such managed nodes. The component must be installed on the Tivoli server and on the managed node hosting the Automation Server component.

Prerequisites

The solution is designed to work on IBM Tivoli Configuration Manager version 4.3.1. Therefore, all the software and hardware prerequisites required by IBM Tivoli Configuration Manager are required by this solution as well.

For information on the software and hardware prerequisites required by IBM Tivoli Configuration Manager, refer to *IBM Tivoli Configuration Manager: Planning and Installation*.

Before you begin

The automated patch management solution requires both WUA and qchain.exe files. The WUA code is provided as a Microsoft Software Installer (MSI) installation. Manually install WUA on a Windows machine and accept the End User Licence Agreement (EULA). After the EULA license has been accepted, WUA can be used in the network without having to accept the license again. The qchain.exe file enables the installation of multiple updates without restarting the computer between each installation. Both WUA and qchain.exe files can be downloaded from the following Microsoft Web sites.

Table 7. Files to download

File to download	Microsoft Web site
WUA	For x86-based computers: WindowsUpdateAgent20-x86.exe. See go.microsoft.com/fwlink/?linkid=43264
qchain.exe	http://www.microsoft.com/downloads/details.aspx?FamilyID=A85C9CFA-E84C-4723-9C28-F66859060F5D&displaylang=en

Run the q296861_W2k_spl_x86_en.exe executable file obtained from the link provided in Table 7 to extract the qchain.exe file.

Distribute the two files to the endpoints in your environment, as described in “Deploying WUA and Qchain on endpoints” on page 30.

InstallShield Server installation

The Patch Management component can be installed using the ISMP server custom installation wizard. The installation creates software packages that are required by the automated patch management solution. The complete InstallShield wizard server installation is documented in the *Planning and Installation Guide*.

If you select the Patch Management component from the list of components you must also select the Software Distribution, Inventory, and Activity Planner components.

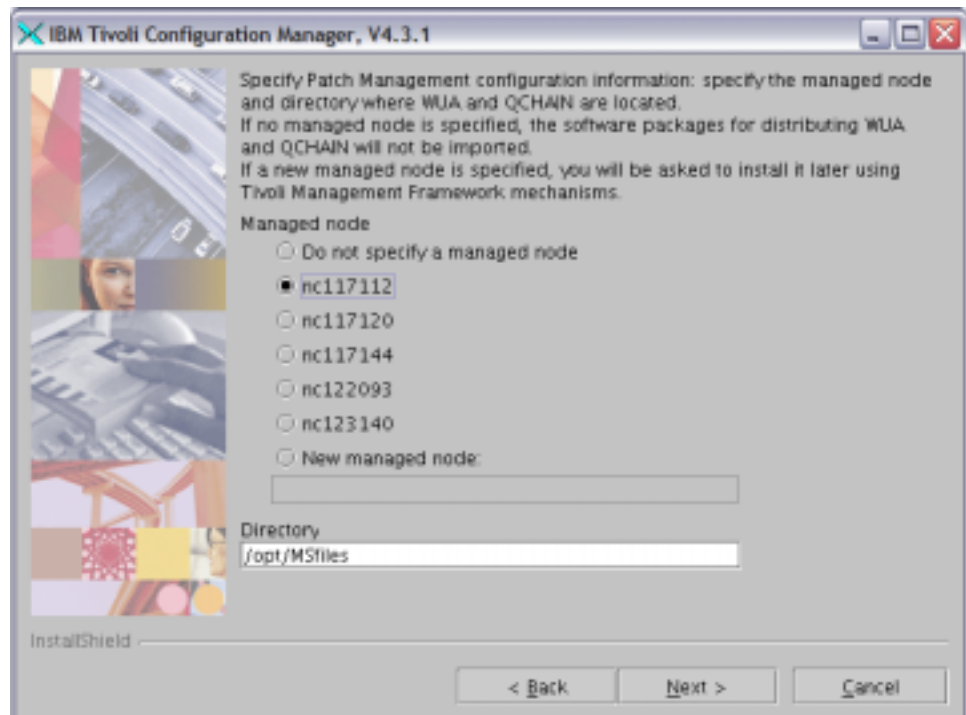
The Patch Management component installation prompts you to specify the following information:

1. The source host where the WUA and qchain.exe files are located or will be located. These files are used when creating packages. You must specify a valid source host (for example the Tivoli server).

If the source host is not defined the installation completes, the WUA^1.0 software package profile is created empty, and a message is logged in the Tivoli Administrator Notices about the error encountered during the import operation.

Select one of the following:

- **Do not specify a managed node** if you do not want to import the packages.
 - One of the existing managed nodes that are listed in the window.
 - **New managed node**, then type in the name of the managed node.
2. In the **Directory** text box, type the name of the directory in the source host where the WUA and qchain.exe files are located or will be located. Click **Next**.



3. Specify on which managed node the IBM Tivoli Configuration Manager Automation Server is, or will be, installed. If you have not yet identified the

Installing the Patch Management component

machine which will assume the role of the Automation Server, you can install the Automation Server component at a later time and launch the InstallShield wizard a second time to finish the installation of the Patch Management component. You have three options:

- Select **Do not specify a managed node** if you do not want the IBM Tivoli Configuration Manager Automation Server to be configured with IBM Tivoli Configuration Manager components.
- Select one of the existing Windows 2003 managed nodes that are listed in the window.
- Select **New managed node**, then type in the name of the managed node when the server is to be installed.



You can use the InstallShield wizard installation to put individual installation steps in Held status and continue with the installation of objects not dependent on the existence of the required managed nodes and then put them in Ready status when all requirements are in place.

Note: The InstallShield wizard does not install the IBM Tivoli Configuration Manager Automation Server. For more information about installing the Automation Server, see “Installing the Tivoli Configuration Manager Automation Server component” on page 7.

Installing using Tivoli Software Installation Service

Tivoli Software Installation Service can install multiple products on multiple machines in parallel. It can install more products on more computer systems in less time than using the installation mechanisms provided by Tivoli Management Framework.

Tivoli Software Installation Service does not distinguish between products and patches. Independent of whether the installation image is used for an installation or upgrade, Tivoli Software Installation Service refers to all installation images as products.

Note: If an installation is started with Tivoli Software Installation Service, it cannot be continued with the InstallShield wizard.

The basic procedure for using Tivoli Software Installation Service to install products is as follows:

1. Import the product images into the Tivoli Software Installation Service depot.
2. Select the components to be installed.
3. Select the machines where each component is to be installed.
4. Click **Install**.

During the installation procedure, you are asked to provide installation options, when applicable.

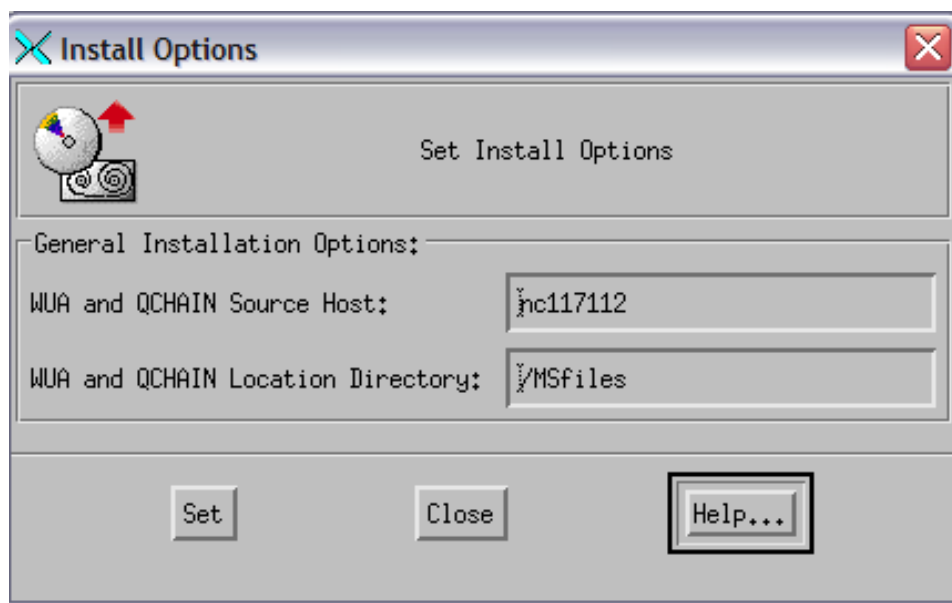
Note: The Tivoli Software Installation Service does not support blank spaces in any of the installation attributes.

For detailed information about using Tivoli Software Installation Service, see *Tivoli Enterprise Installation Guide*.

Tivoli Desktop installation

The Tivoli Desktop can install the same product on multiple machines sequentially. The basic procedure for using the Tivoli Desktop to install the Patch Management component is as follows:

1. From the Tivoli Desktop, select **Install -> Install Product** from the **Desktop** menu.
2. From the Install Product window, click **Select Media**, and the File Browser window opens.
3. In the File Browser window, browse to the Patch Management Images directory and click **Set Path**.
4. Click **Set Media & Close**.
5. From the Install Product window, select **Patch Management** and the managed node you are installing on.
6. The Set Install Options dialog opens.



- a. In **WUA and QCHAIN Source Host**, type the name of a valid source host.
- b. In **WUA and QCHAIN Location Directory**, type the name of the directory in the source host where these files are located or will be located.

Click **Set** and then **Close**.

7. Click **Install & Close**.

Tivoli command line

To install the Patch Management component from the Tivoli command line, launch the following command:

```
winstall -c source-dir -i PATCHMGT.IND location TMR_hostname
```

where:

-c source-dir

Specifies the complete path to the directory containing the installation image.

-i PATCHMGT.IND

Specifies the product index file from which the product is installed. Index files have an IND extension.

TMR_hostname

The name of the Tivoli management region server.

Objects created by the installation

The installation creates the following objects in your Tivoli environment:

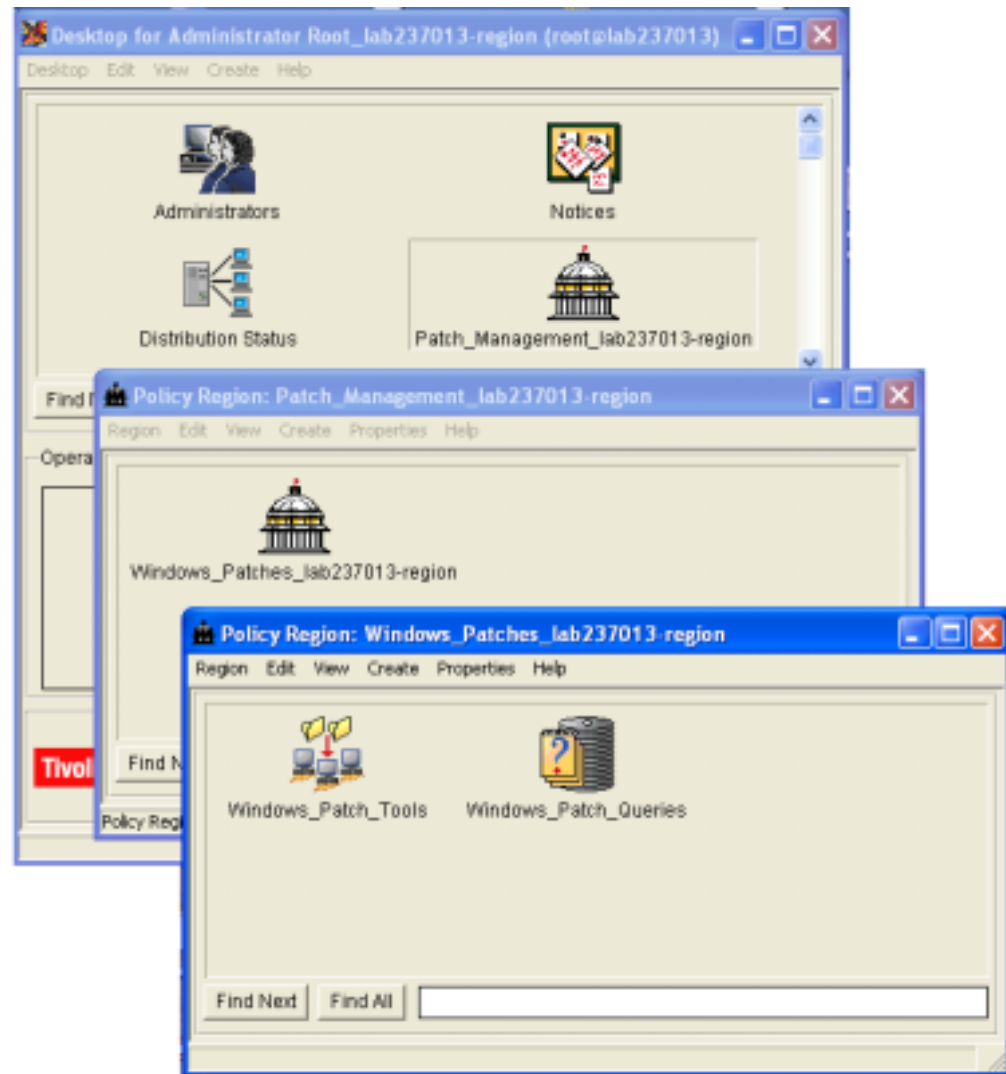


Table 8. Objects created after installation

Object	Description
Command lines are added to the directory \$(BINDIR)/bin directory	Patch management commands for specifying patch installation settings and for reporting patch status.
Uninstall commands are copied to the \$(BINDIR)/TME/PATCH_MGMT/SCRIPTS directory.	Commands used when the Patch Management component is uninstalled.
Patch_Management and Windows Patches policy regions	Contains a set of profile managers (dataless) that contain the packages to install patches to the related operating system and the Windows_Patch_Queries query library.
Windows_Patch_Tools profile manager	The profile manager will contain the Inventory scan profiles properly customized to run scans for patch management. This also includes the reboot package.

Objects created by the installation

Table 8. Objects created after installation (continued)

Object	Description
Windows_Patch_Queries query library	Query library is created empty and queries are created as software packages are automatically generated.
Windows_Initial_Patch_Scan and Windows_Patch_Scan inventory scans	Inventory scan profiles to discover patch information.
WUA^1.0, WSUSSCANCAB^1.0, RebootMSG^1.0 software package objects	<ul style="list-style-type: none">• The WUA^1.0 software package contains the Windows Update Agent.• The WSUSSCANCAB^1.0 software package contains the scan file downloaded from the Microsoft Web site.• The RebootMSG^1.0 package is used to force a reboot on the endpoint for those patches that require a reboot after installation.

Installing and upgrading the Patch Management Automation Server driver

To install or upgrade the Patch Management Automation Server driver, follow the steps described in “Installing the Patch Management Automation Server driver” and “Upgrading the Patch Management Automation Server driver” on page 24.

Installing the Patch Management Automation Server driver

If you installed the Patch Management solution, version 4.2.3, you need to uninstall the previous Patch Management Automation Server driver and install the new driver.

To install the new driver, perform the following steps:

1. Stop the Tivoli Configuration Manager Automation Server:
 - a. Log on as user `tioadmin`.
 - b. Open a Cygwin bash window and switch to the `$TIO_HOME/tools` directory.
 - c. To stop the application, run the command: `./tio_stop.cmd`.
 - d. At the **User name** prompt, type `wasadmin` and press **Enter**.
 - e. At the **Password** prompt, if you have not changed the password for WebSphere Application Server, type the default password `wasadmin` and press **Enter**.
 - f. Optionally, check the log file `$TIO_HOME/logs/tio_stop.log` for errors.
2. Uninstall the previous Patch Management Automation Server driver:
 - a. In the `$TIO_HOME/tools` directory, run the `./tc-driver-manager.cmd uninstallDriver tcm-ms-patches` command
3. Copy the new version of the Patch Management Automation Server driver:
 - a. Switch to the `$BINDIR/TME/PATCH_MGMT/TPM_TCM_DRIVER` directory.
 - b. Copy the new `tcm-ms-patches.tcdriver` file to the `$TIO_HOME/drivers` directory.
 - c. Remove the `$TIO_HOME/drivers/lib/tcm-ms-patches.jar` file. Do not rename it, because this might cause unpredictable results.
4. Install the new version of the Patch Management component:

Installing and upgrading the Patch Management Automation Server driver

- a. Switch to the `$TIO_HOME/tools` directory.
 - b. Run the `./tc-driver-manager.cmd installDriver tcm-ms-patches` command. The following files are created in the `$TIO_HOME/xml` directory:
 - `xml/xml_update.sh`
 - `xml/tpm_update.req`
5. Modify the **tpm_update.req** file using Cygwin by providing the appropriate values for the following keywords:

Table 9. *tpm_update.req* file values

User Input Field	Description
<code>TPM_HOST_NAME=<value></code>	Replace <code><value></code> with only the relative hostname of the workstation where the Automation Server component is installed.
<code>TPM_HOME=<value></code>	Replace <code><value></code> with the full path of the Tivoli Configuration Manager Automation Server main directory. For example: <code>TPM_HOME="C:\IBM\tivoli\thinkcontrol"</code>
<code>WSUS_SERVER_NAME=<value></code>	Replace <code><value></code> with the fully qualified name of the WSUS server. For example: <code>WSUS_SERVER_NAME="server.ibm.com"</code> Note: The server name must be registered in DNS or in the local hosts file.
<code>WSUS_IP_ADDRESS=<value></code>	Replace <code><value></code> with the IP address of the WSUS server.
<code>PROXY_HOST=<value></code>	Replace <code><value></code> with the host name of the proxy server.
<code>PROXY_PORT=<value></code>	Replace <code><value></code> with the port of the proxy server.
<code>PROXY_USER=<value></code>	Replace <code><value></code> with the user ID used to manage the proxy.
<code>PROXY_PASSWD=<value></code>	Replace <code><value></code> with the password for the user ID used to manage the proxy.
<code>TPM_SERVER_INTERP=<value></code>	Do not modify the value of this key.
<code>WSUS_MANAGED_NODE=<value></code>	Do not modify the value of this key.

6. Create the `tcm-dcm_xx.xml` file by running the `./xml_update.sh` command from the `$TIO_HOME/xml` directory. The `tcm-dcm_xx.xml` file is created, where the `xx` value depends on the level of fix pack you are installing.

Note: To determine which XML file to use, run the following command from a command prompt:

```
ls -la "$TIO_HOME"/xml
```

7. Reinitialize the Patch Management environment:
- a. Switch to the `$TIO_HOME/tools` directory.
 - b. Run the `./reinit.cmd $TIO_HOME/xml/tcm-dcm_xx.xml` command.
 - c. Optionally, check the log file `$TIO_HOME/logs/reinit.log` for any errors, if unable to start the application.

Note: After running the `reinit.cmd` procedure, the Automation Server points to the URL of the new .cab file. If you are in a mixed environment where the latest WUA has not been deployed on all workstations and you need to use the old .cab, modify the **mscab_url** value in the Tivoli Configuration Manager Automation Server console by performing the following steps:

Installing and upgrading the Patch Management Automation Server driver

- a. On the User defined variables page, locate the **mscab_url** key and click the pointer associated with that key.
- b. Enter the URL of the Microsoft Web site where the .cab file is located:
`/fwlink/?LinkId=39043`

This path must start with a forward slash (/) and is obtained by removing the `http://new_server_name` section from the complete URL.

- c. Click **Save**.
8. Start the Tivoli Configuration Manager Automation Server:
 - a. Switch to the `$TIO_HOME/tools` directory.
 - b. To start the application, run the `./tio_start.cmd` command.
 - c. At the **User name** prompt, type `wasadmin` and press **Enter**.
 - d. At the **Password** prompt, if you have not changed the password for WebSphere Application Server, type the default password `wasadmin` and press **Enter**.
 - e. The window displays a message that Tivoli Configuration Manager Automation Server is ready to run.

Important

Do not close the window that informs you that the application is running. If you close the window, the Tivoli Configuration Manager Automation Server does not start.

- f. Check the log file `$TIO_HOME/logs/tpm_start.log` for any errors, if unable to start the application.

Upgrading the Patch Management Automation Server driver

If you installed the Patch Management solution, version 4.2.3, you need to uninstall the previous Patch Management Automation Server driver and install the new driver provided with the current version of the product. To install the new driver, perform the following steps:

1. Stop the Tivoli Configuration Manager Automation Server:
 - a. Log on as user `tioadmin`.
 - b. Open a Cygwin bash window and switch to the `$TIO_HOME/tools` directory.
 - c. Run the `./tio_stop.cmd` command
 - d. At the **User name** prompt, type `wasadmin` and press **Enter**.
 - e. At the **Password** prompt, if you have not changed the password for WebSphere Application Server, type the default password `wasadmin` and press **Enter**.
 - f. Check the `$TIO_HOME/logs/tio_stop.log` log file for errors.
2. Uninstall the previous Patch Management Automation Server driver:
 - a. In the `$TIO_HOME/tools` directory, run the `./tc-driver-manager.cmd UninstallDriver tcm-ms-patches` command.
 - b. Remove the `$TIO_HOME/drivers/lib/tcm-ms-patches.jar` file. Do not rename it, because this might cause unpredictable results.
3. Install the new version of the Patch Management component:
 - a. Switch to the `$BINDIR/TME/PATCH_MGMT/TPM_TCM_DRIVER` directory.

Installing and upgrading the Patch Management Automation Server driver

- b. Copy the `tcm-ms-patches.tcdriver` file, which represents the new version of the Patch Management Automation Server driver, to the `$TIO_HOME/drivers` directory.
- c. Switch to the `$TIO_HOME/tools` directory.
- d. Run the `./tc-driver-manager.cmd installDriver tcm-ms-patches` command.

Note: You do not need to reinitialize the Patch Management environment.

4. Start the Tivoli Configuration Manager Automation Server:
 - a. Switch to the `$TIO_HOME/tools` directory.
 - b. Start the application by running the `./tio_start.cmd` command.
 - c. At the **User name** prompt, type `wasadmin` and press **Enter**.
 - d. At the **Password** prompt, if you have not changed the password for WebSphere Application Server, type the default password `wasadmin` and press **Enter**. A window displays a message that Tivoli Configuration Manager Automation Server is ready to run.

Important: Do not close the window informing you that the application is running. If you do, the Tivoli Configuration Manager Automation Server does not start.

- e. Check the `$TIO_HOME/logs/tpm_start.log` log file for any errors.
5. If you want to retrieve the new version of the `.cab` file, `wsusscn2.cab`, you must modify the `mscab_url` value in the Tivoli Configuration Manager Automation Server console by performing the following steps:
 - a. Identify the `MSSecure_Server` name as follows:
 - 1) Select the **System configuration and workflow management** tab and click **Configuration**. This displays the Data Center Configuration page.
 - 2) From the Data Center Configuration page, click the **Variables** tab. This displays the User defined variables page. On the User defined variables page, you find the value of the `MSSecure_Server` key.
 - b. Modify the value of the `mscab_url` key as follows:
 - 1) From the **Data center assets and resources** tab, expand **Inventory > Servers > MS-WSUS**.
 - 2) Select the `MSSecure_Server` name you previously identified and click **Variables**.
 - 3) Edit the `mscab_url` key by entering the URL of the Microsoft Web site where the `.cab` file is located:
`/fwlink/?LinkId=74689`

This path must start with a forward slash (/) and is obtained by removing the `http://new_server_name` section from the complete URL.

- 4) Click **Save**.

Chapter 3. Configuration and administrative tasks

This chapter describes the following administrative tasks you can perform after installing the automated patch management components:

- “Installing and Configuring the Microsoft WSUS Server”
- “Configuring SSH communications between Automation Server and WSUS” on page 28
- “Deploying WUA and Qchain on endpoints” on page 30
- “Configuring automated patch management settings” on page 31
- “Configuring Automation Server to download the .cab file” on page 34
- “Considerations for an interconnected environment” on page 35
- “Administering Tivoli Configuration Manager Automation Server” on page 37
 - “Starting the Tivoli Configuration Manager Automation Server” on page 37
 - “Logging on to the Tivoli Configuration Manager Automation Server console” on page 38
 - “Stopping the Tivoli Configuration Manager Automation Server” on page 38
 - “Changing default passwords” on page 38
 - “Changing WSUS values” on page 39
- “Administering the workflow” on page 40
 - “Running the workflow” on page 41
 - “Stopping the workflow” on page 41
 - “Checking workflow status” on page 41
 - “Scheduling the workflow” on page 42

Installing and Configuring the Microsoft WSUS Server

The Microsoft WSUS server is used to implement the approval mechanism of patches and to download up-to-date patches from the Microsoft Web site. Approved patches are listed in the ApprovedChanges.txt file that is downloaded to a Tivoli management region server and moved to the endpoints in your Tivoli environment with an Inventory scan profile. An internet connection is required to communicate with the Microsoft Web site so that the public Windows Update service can synchronize with the server running Windows Server Update Services. It is recommended to avoid installing this service on the Automation Server managed node.

The WSUS version available at the time this manual is released is version 3.0.

To obtain, install and configure the latest version of WSUS, perform the following steps:

1. Install WSUS from the following Web site: <http://support.microsoft.com/kb/935524>.
2. Install the Microsoft SQL Server Native Client from the following Web site: <http://www.microsoft.com/downloads/details.aspx?familyid=d09c1d60-a13c-4479-9b91-9e8b9d835cdc&displaylang=en>.
3. Install the Microsoft SQL Server 2005 Command Line Query Utility from the following Web site: <http://www.microsoft.com/downloads/details.aspx?familyid=d09c1d60-a13c-4479-9b91-9e8b9d835cdc&displaylang=en>.

Configuring Microsoft WSUS Server

4. Set the `wsus_db_host` and `wsus_version` parameters in the `wseccefg` command as follows:
`wsus_db_host=\\.\pipe\MSSQL$MICROSOFT##SSEE\sql\query`
`wsus_version=3`
5. Ensure that the `sqlcmd.exe` file is located in the `$progFiles/Microsoft SQL Server/90/Tools/binn` directory or that the related path is included in the `PATH` variable.

To use the automated patch management solution, ensure that the Windows Server Update Services are configured to store files locally:

1. From the WSUS homepage, select **Options**.
2. From the Options dialog, select **Synchronization Options**.
3. From the Synchronization Options dialog, click the **Advanced** push button.
4. In the Update Files box, ensure that the **Store update files locally on this server** check box is selected.

For detailed information about deploying Microsoft WSUS, refer to *Deploying Microsoft Windows Server Update Services* at the following Web site:
<http://technet2.microsoft.com/windowsserver/en/technologies/featured/wsus/default.mspx>

To allow Tivoli Configuration Manager Automation Server to communicate with the Microsoft WSUS server, you must install and configure Cygwin on the WSUS server, as described in “Installing and configuring Cygwin” on page 7, and configure the SSH protocol as described in “Configuring SSH communications between Automation Server and WSUS” on page 28.

To improve network security, you can install a local WSUS server located behind the firewall which connects to another WSUS server located outside the firewall. The server outside the firewall can connect to the Internet and download the information from the Microsoft Web site. This configuration allows only one port to be opened in the firewall for the connection between the two WSUS servers.

Configuring SSH communications between Automation Server and WSUS

To configure SSH communications between Automation Server and WSUS you need to configure WSUS as the SSH server and Automation Server as the SSH client.

Configuring the SSH server

To configure WSUS as the SSH server, perform the following tasks:

1. Verify that all servers in your configuration are set up correctly in either DNS and or `/etc/hosts`.
2. Create the `tioadmin` user on WSUS server and add this user to the Administrators group.
3. Log on as `tioadmin` and install Cygwin.
4. Invoke a Cygwin window. Ensure that the home directory is `/home/tioadmin`.
5. To generate host keys, open a Cygwin bash shell window and run the following command:
`/usr/bin/ssh-host-config -y`

This command generates three different keys-DSS, RSA, RSA1, each corresponding to a different encryption algorithm. This allows a system to establish SSH sessions with systems requiring any one of these encryption algorithms.

On Windows 2003, you will be prompted to create a new account with special privileges for enabling the **passwordless logon** functionality. A new account is created and you will then be prompted for a password. Enter a password for the new user and ensure that this password matches the password rules given on your system.

Accept the default value for the environment variable CYGWIN and press Enter.

On Windows 2003, ensure that the user account created by Cygwin belongs to the **Administrators** group, otherwise add it to this group by clicking **My Computers->Manage**.

6. Start the sshd service by running the following command (This service is automatically start after a reboot):
`cygrunsrv -S sshd`
7. Verify that tioadmin has a password in /etc/passwd. If no password is found, generate it by running the command:
`mkpasswd -l > /etc/passwd`
8. Configure SSH for tioadmin, by typing the following command:
`ssh-user-config`

When prompted for creating the identity files, choose the option for the ssh version to be used. A configuration example for using SSH2 follows. Press Enter when prompted for a passphrase. Output similar to the following one is displayed:

```
Shall I create an SSH1 RSA identity file for you? (yes/no) no
Shall I create an SSH2 RSA identity file for you? (yes/no) (yes/no) yes
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Do you want to use this identity to login to this machine? (yes/no) yes
Shall I create an SSH2 DSA identity file for you? (yes/no) (yes/no) no
```

Configuration finished. Have fun!

9. Switch to the .ssh directory by typing:
`cd .ssh`

and press Enter.
10. Insert the user key into the authorized_keys file of the user account on the server. To perform this task, run the command:
`cat id_rsa.pub >>authorized_keys`
11. To configure SSH to accept connections from new hosts without prompting for confirmation, create a file in /home/tioadmin/.ssh called config and run the command:
`echo "StrictHostKeyChecking no" > config`

The file contains the value of StrictHostKeyChecking no

12. Verify that SSH is configured properly by typing
`ssh tioadmin@localhost`

If SSH is properly configured, the following message is displayed:

Configuring SSH server

```
Fanfare!!!  
You are successfully logged in to this server!!!
```

Configuring the SSH client

To configure the Automation Server as the SSH client, perform the following tasks:

1. On the Automation Server log on as tioadmin.
2. Invoke a Cygwin window.
3. Ensure that the ssh directory has write permission.
4. Generate client keys

```
$ cd  
$ pwd  
/home/tioadmin  
  
$  
$ ssh-keygen -t rsa -N ""  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/tioadmin/.ssh/id_rsa):  
Created directory '/home/tioadmin/.ssh'.  
Your identification has been saved in /home/tioadmin/.ssh/id_rsa.  
Your public key has been saved in /home/tioadmin/.ssh/id_rsa.pub.  
The key fingerprint is: fd:ca:21:d3:3f:db:fd:d9:56:b2:30:68:16:43:1c:11  
tioadmin@tio12
```

5. Export the client public key to the SSH server:

```
scp /home/tioadmin/.ssh/id_rsa.pub ssh_server_ipaddress  
:/home/tioadmin/.ssh/ssh_client_hostname.txt
```

Enabling client server SSH communications

1. On WSUS server log on as tioadmin.
2. Invoke a Cygwin window.
3. Run the following commands:

```
cd /home/tioadmin/.ssh  
cat client_hostname.txt >> authorized_keys2
```

Checking SSH communications

1. On the Automation Server, log on as tioadmin.
2. Invoke a Cygwin window.
3. Run the following command:

```
ssh tioadmin@wsus_ip_address
```
4. Check that a successful login is performed without a password request.

Enabling TCP/IP communications

Before using WSUS, ensure you ran the SQL Server Network Utility (svrnetcn.exe) to enable TCP/IP communications.

Deploying WUA and Qchain on endpoints

Before deploying WUA on endpoints, ensure that its version is correct for the .cab file you want to use.

After having downloaded WUA and Qchain from the Microsoft site as described in Table 2 on page 5, distribute them to the endpoints by performing the following steps:

1. Open the WUA.spd file created during the installation.

2. Customize the WUACLIROOT key in the WUA.spd file, specifying the directory on the source host where both the WindowsUpdateAgent20-x86.exe and qchain.exe are located.
3. Import WUA.spd in the WUA^1.0 software package created during the installation.
4. Run the available query to define the endpoints on which to install the software package.
5. Install WUA^1.0 software package on the endpoints.

To discover Windows product patches, ensure you installed the latest level of Windows Installer on the endpoints.

Upgrading WUA and .cab file

Every time you update the scan package format (.cab file), you must ensure you upgrade WUA to a version that supports that .cab file.

To manage mixed environments, where different levels of WUA are installed, deploy the new .cab file only after having updated WUA on all endpoints.

If you install a version of WUA that does not meet the minimum version requirement of the corresponding .cab file, the installed WUA version will not work with the scan package.

For more information, see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wua_sdk/wua/portal_client.asp.

Configuring automated patch management settings

This section describes configuration settings you can manually set to define your automated patch management solution.

wseccfg command

Using this command, you can configure the following settings. See “wseccfg” on page 68 for the command syntax and arguments.

- Set the **TMR_server_list** key in a single Tivoli management region environment, to specify the Tivoli server on which to download the .cab and ApprovedChanges.txt files. See “Considerations for an interconnected environment” on page 35.
- Set the **source_host_list** key to list the source hosts where patches are prepared. Specify one source host for each Tivoli management region.
- Set the **plan_grouping_mode** key to define how plans are generated: one plan containing all patches or one plan for each patch.
- Set the **plan_creation_mode** key to define how plans are generated based on topology: one plan addressing all the targets in the network (configuration for small environments) or one plan for each region. See “Considerations for an interconnected environment” on page 35.
- Set the **provider_spb_dir** key to specify the directory to use on the source host to save software package blocks. Specify a directory with a large amount of space because many software package blocks are copied to this directory. Ensure that the specified directory exists on the source host.
- Set the **email_notification_address** key to specify the e-mail of the Administrator for notification once the Automation Server workflow is

Configuring automated patch management settings

complete and the activity plan is ready to be submitted. If this key is not set, no notification is sent when the workflow is complete.

- Set the **trace_size** key to specify the size of the trace file.
- Set the **trace_level** key to specify the trace level.
- Set the **max_apm_bootable_threshold** key to specify how many activities that require a reboot are to be grouped together.
- Set the **wsus_inst_path** key to specify the WSUS Update Source directory (for example, c:\WSUS). This key is mandatory. For details, see *Deploying Microsoft Windows Server Update Services*.
- Set the **wsus_db_name** key to specify the name of the database used by WSUS to store data. This key is case-sensitive and optional. The default value is SUSDB.
- Set the **wsus_db_host** key to specify the hostname of the SQL Server used by WSUS to store data. You can retrieve this name from the SQL Server Service Manager (sqlmangr.exe) on the WSUS server. If you specify a remote database for WSUS, you must modify the WSUS_info_retriever.sh script, as described in “Osql failure with a remote WSUS database” on page 103. If you are using WSUS , version 3.0, set this key to

```
\\.\pipe\MSSQL$MICROSOFT##SSEE\sql\query
```

. This key is mandatory.

- Set the **wsus_version** key to specify the version of WSUS being used.
- Set the **cab_gateways_list** key to specify the list of gateways to which the .cab and ApprovedChanges.txt files must be uploaded if changed since the last download from the Microsoft Web site. These files are downloaded by the Automation Server workflow to the lcf_bundle directory on the specified gateways. The keyword all_gw can be used to indicate all the gateways in the network. If you do not specify this key, the .cab and ApprovedChanges.txt files are not copied onto any gateways.
- Set the **cab_gateways_file** key to specify the name of the file that contains the list of the gateways to which the file has to be uploaded (see cab_gateways_list option).
- Set the **prepare_patches_requiring_connectivity** key to yes or no. Some Microsoft patches require connectivity to Internet to be installed. These patches are prepared only if the value of this key is set to yes. The parameter is optional and its default value is no.
- Set the **prepare_patches_requiring_user_input** key to yes or no. Some Microsoft patches require user input (for example acceptance of EULA license) when installed. These patches are prepared only if the value of the prepare_patches_requiring_user_input is set to yes. This key is optional and the default value is no.
- Set the **catalog_proxy_enabled** key to yes or no. This key enables or disables proxy support. You can use an HTTP proxy to access the Microsoft Web site, or your local HTTP server where the .cab file has been downloaded. Proxy parameters are defined at installation time in the tpm_update.req file. This key is optional and the default value is no.
- Set the **workflow_activities** key to one of the following values:

sync Performs the following operations:

1. Synchronizes WSUS approved patches with the Automation Server database.

Configuring automated patch management settings

2. Download the .cab file from the Microsoft Web site or the HTTP server.
3. Creates ApprovedChanges.txt.
4. Copies the .cab and ApprovedChanges.txt file on the workstations defined in the TMR_server_list, cab_gateways_list, and cab_gateways_file.

preparation

Creates software packages, queries, and APM plans.

all Performs all sync and preparation operations. This is the default value.

This key is optional and the default value is all.

- Set the **target_filtering_mode** to specify the filter to be used for grouping targets.
- Set the **submit_plans** to yes or no. This key specifies the submission of activity plan, which is generated at the end of the patch management workflow.
- Set the **skip_plans_creation** to yes or no. This key specify to avoid the creation of the activity plan at the end of the patch management workflow.

Note: If your environment contains interconnected regions, some of the parameters set with the **wseccfg** command have more than one value. See “Considerations for an interconnected environment” on page 35

Add tioadmin login

Before running the Automation Server workflow, associate the tioadmin user to a Tivoli administrator. The **tioadmin** user is the user of the Automation Server used to run all Patch Management operations. From the Tivoli desktop, add the **tioadmin** user to the list of logins for a Tivoli administrator with the following characteristics:

- User Login Name set to the user login map "\$root_user", "\$root_group"
- TMR roles: senior, policy, RIM_view, APM_Edit, APM_View

Note: Ensure that the tioadmin user is allowed to write to the following product directories:

- product_dir
- provider_patch_dir
- provider_spb_dir

This is valid only if these directories are local.

wmailhost command

Run the Tivoli Framework command, **wmailhost** on Windows operating systems to generate and send e-mail notifications. The command specifies the mail server used by Tivoli Management Framework on Windows operating systems. Refer to the *Tivoli Management Framework Reference Manual* for information about the **wmailhost** command.

Note: See also how to specify the **email_notification_address** key using “wseccfg” on page 68.

winvmgr command

Specify if you want to use the Microsoft WSUS Server for your patch management operations by running the following Inventory command:

```
winvmgr -c PM_WSUS_enabled=y
```

Configuring Automation Server to download the .cab file

To download the .cab file from the Microsoft Web site you must connect to the Internet. You can use a proxy server to ensure a higher security when accessing the Internet or your local network. At installation time you can customize the HTTP proxy server name (if enabled) and the related user ID and password (if required by the proxy settings) in the `tpm_update.req` file.

If the access to the Microsoft Windows Web site through Internet does not work, use the following workaround:

1. Manually download the .cab file and move it to a workstation where an HTTP server is running. You can download one of following files:
 - `wsusscn2.cab` from <http://go.microsoft.com/fwlink/?LinkId=74689>
 - `wsusscan.cab` from <http://go.microsoft.com/fwlink/?LinkId=39043>
2. Save the .cab file granting unrestricted HTTP get access.
3. Modify the `MSSecure_Server` and `mscab_url` values in the Tivoli Configuration Manager Automation Server console to retrieve the .cab file. To modify the `MSSecure_Server` and `mscab_url` values, perform the following steps:
 - a. Log on the Tivoli Configuration Manager Automation Server console.
 - b. From the **Data center assets and resources** tab, click **Inventory » Servers » MS-WSUS** and select the existing MSSecure server name. This displays the MSSecure server page.
 - c. On the MSSecure server page, click the **Management** menu on the top right and select **To Maintenance**. This brings the server offline and turns the maintenance mode on. The MSSecure server status changes from **available** to **in maintenance**.
 - d. On the same page, click the **Edit** menu on the top right and select **Properties**. This allows you to enter the new MSSecure server name.
 - e. Enter the new MSSecure server name and click **Save**.
 - f. Click the **Management** menu again and select **Out of Maintenance**. This brings the server online again and turns the maintenance mode off.
 - g. Select the **System configuration and workflow management** tab and click **Configuration**. This displays the Data Center Configuration page.
 - h. From the Data Center Configuration page, click the **Variables** tab. This displays the User defined variables page.
 - i. On the User defined variables page, find the `MSSecure_Server` key and click the pointer associated with that key.
 - j. Enter the new MSSecure server name in the value field and click **Save**.
 - k. On the MSSecure server page, click the **Variables** tab. This displays the User defined variables page.
 - l. On the User defined variables page, locate the `mscab_url` key and click the pointer associated with that key.
 - m. Enter the virtual path defined on the HTTP server where the .cab file is located. This path must start with a forward slash (/) and is obtained by removing the `http://new_server_name` section from the complete URL.
 - n. Click **Save**.

Considerations for an interconnected environment

If your environment is composed of interconnected Tivoli management region servers, you can install and configure your environment differently, depending on how you configure specific options using the **wseccfg** command.

For more information on Hub and Spoke regions, refer to the *IBM Tivoli Configuration Manager Planning and Installation Guide* and the *Tivoli Management Framework Planning for Deployment Guide*.

Hub and Spoke configuration examples

An environment where both the Hub and Spoke environments are running IBM Tivoli Configuration Manager, version 4.3.1 can include two different configurations.

First configuration example

Table 10. Patch Management component on Hub and Spoke

Tivoli region	Role	Component
Hub	Server	Inventory
		Software Distribution
		Activity Planner
		Patch Management
	Managed node/Automation Server	Software Distribution
		Activity Planner
		Patch Management
	Managed node/Source host	Software Distribution
Spoke	Tivoli server	Inventory
		Software Distribution
		Activity Planner
		Patch Management
	Managed node/Source host	Software Distribution

In this type of configuration, you can define patch management settings using the **wseccfg** command without any particular limitations. See “wseccfg” on page 68.

Second configuration example

Table 11. Patch Management component on Hub only

Tivoli region	Role	Component
Hub	Server	Inventory
		Software Distribution
		Activity Planner
		Patch Management
	Managed node/Automation Server	Software Distribution
		Activity Planner
		Patch Management
	Managed node/Source host	Software Distribution

Considerations for interconnected environment

Table 11. Patch Management component on Hub only (continued)

Tivoli region	Role	Component
Spoke	Tivoli server	Inventory
		Software Distribution
	Managed node/Source host	Software Distribution

In this type of configuration, the following keys set using the **wseccfg** command must assume the following values:

```
wseccfg -s tme_object_scope_for_plan=hub
wseccfg -s TMR_server_list=<HUB-server>
wseccfg -s source_host_list=<HUB-source_host>
wseccfg -s plan_creation_mode=per_enterprise
```

These configuration settings imply that the Automation Server workflow generates plans whose activities refer to software packages and inventory scans belonging to the Hub region. A suffix of "#hub-region-name" is appended to the activity name.

To avoid excessive bandwidth allocation caused from sending software packages and Inventory scans from the Hub region to the endpoint, you can use a source host in each Spoke that is closer to the endpoint that you want to manage using the following settings:

```
wseccfg -s tme_object_scope_for_plan=spoke
wseccfg -s source_host_list=<HUB-source-host>,<SPOKE-source_host>
wseccfg -s plan_creation_mode=per_tmr_region
wseccfg -s TMR_server_list=<HUB-server>,<SPOKE-server>
```

To use these settings however, you must perform one of the following tasks:

- Install the Patch Management component on each Spoke, then the keys managed by **wseccfg** can be used in any combination.

OR

- Create on each Spoke, an Inventory profile scan with the name defined in the plan that is automatically generated by the workflow as follows:

```
wcrtprf @ProfileManager:<SPOKE-ProfileManager-Name> InventoryConfig
Windows_Patch_Scan
```

Set the options for the patch scan as follows:

```
wsetinvpcsw -m YES @InventoryConfig:Windows_Patch_Scan
```

Create the reboot package on the Spokes using the following command:

```
wimpspo -c <SPOKE-ProfileManager> -h <SPOKE-source-host> -f
@<HUB-TMR-server>:<${BINDIR}/TME/PATCH_MGMT/TEMPLATES/RebootMSG.spd
```

Note: An example of the warning messages generated by the import operation and related to variables defined on the endpoint, follows:

```
DISSE0224W Variable 'REBOOT' near line '30' cannot be resolved.
DISSE0224W Variable 'REBOOT' near line '38' cannot be resolved.
DISSE0224W Variable 'INTERP' near line '41' cannot be resolved.
DISSE0224W Variable 'LCF_DATDIR' near line '41' cannot be resolved
```

The variables are resolved at installation and therefore you can ignore these messages.

Administering Tivoli Configuration Manager Automation Server

This section describes how to start and stop the Tivoli Configuration Manager Automation Server, as well as how to access the main administrative console for configuration purposes.

Starting the Tivoli Configuration Manager Automation Server

Before you start Tivoli Configuration Manager Automation Server, ensure that you have:

- Started the DB2 services including DB2 Universal Database Workgroup Server - TIOLDAP.
- Started the Tivoli Directory Server V5.2 services.
- Stopped the WebSphere Application Server service.

There are two ways to start Tivoli Configuration Manager Automation Server:

- “Starting Tivoli Configuration Manager Automation Server using the command line”
- “Starting Tivoli Configuration Manager Automation Server using services”

Starting Tivoli Configuration Manager Automation Server using the command line

To start Tivoli Configuration Manager Automation Server using the command line:

1. Log in as `tioadmin`.
2. Open a Cygwin bash window and switch to the `$TIO_HOME\tools` directory.
3. To start the application, run the command: `./tio_start.cmd`
4. At the **User name** prompt, type `wasadmin` and press **Enter**.
5. At the **Password** prompt, if you have not changed the password for WebSphere Application Server, type the default password `wasadmin` and press **Enter**.
6. The window will display a message that Tivoli Configuration Manager Automation Server is ready to run.

Important

Do not close the window that informs you that the application is running. If you do, Tivoli Configuration Manager Automation Server will fail.

7. Check the log file `$TIO_HOME\logs\tpm_start.log` for any errors.

Starting Tivoli Configuration Manager Automation Server using services

To start Tivoli Configuration Manager Automation Server using services:

1. Click **Start-> Settings-> Control Panel-> Service Manager**.
2. Select the IBM WebSphere Application Server V5 IBM_TPM from the list of services, right-click and then click **Start**. The status column will indicate if the service has started.
3. Select the IBM Tivoli Configuration Manager Automation Server V5.2 from the list of services, right-click and then click **Start**. The status column will indicate if the service has started.

Logging on to the Tivoli Configuration Manager Automation Server console

Before logging on to the Tivoli Configuration Manager Automation Server, ensure that you have:

- Started the Tivoli Configuration Manager Automation Server.
- Started the DB2 services including DB2 Universal Database Workgroup Server - TIOLDAP.
- Started the Tivoli Directory Server V5.2 services.
- Internet Explorer full version 6.0 (also known as Internet Explorer 6.0 Service Pack 1 and Internet Tools) or later with the latest security updates from Microsoft.
- The fully-qualified domain name (for example, `hostname.domain.com`) and port number for the Tivoli Configuration Manager Automation Server. The default port number is 9080.

To log on to the main Tivoli Configuration Manager Automation Server console:

1. Open a Web browser and enter the following URL:
`http://host_name:9080/tcWebUI`
where *host_name* is the fully-qualified domain name of the server. The Log On window opens.
2. Enter your user name and password. The default user name is `tioappadmin` and if you have not already changed the password, the default password is `tioappadmin`.

Note: To log off from the system, click Log off. You are automatically logged off the system after thirty minutes of session inactivity. To stop Tivoli Configuration Manager Automation Server, refer to the section, “Stopping the Tivoli Configuration Manager Automation Server” on page 38.

Stopping the Tivoli Configuration Manager Automation Server

To stop the Tivoli Configuration Manager Automation Server:

1. Log on as user `tioadmin`.
2. Open a Cygwin bash window and switch to the `$TIO_HOME\tools` directory.
3. To stop the application, run the command: `./tio_stop.cmd`
4. At the **User name** prompt, type `wasadmin` and press **Enter**.
5. At the **Password** prompt, if you have not changed the password for WebSphere Application Server, type the default password `wasadmin` and press **Enter**.
6. Review the log file `$TIO_HOME\logs\tio_stop.log` for errors.

Changing default passwords

Tivoli Configuration Manager Automation Server requires that a default set of user IDs and default passwords be created and used during installation and configuration. A command line tool is provided to change the passwords after Tivoli Configuration Manager Automation Server has been installed. Using the command line tool, you can change the passwords for the following user IDs:

Table 12. Default User ID

User ID	Description
<i>tioldap</i>	User ID used by Tivoli Configuration Manager Automation Server to connect to the LDAP server

Table 12. Default User ID (continued)

User ID	Description
<i>wasadmin</i>	User ID used to administer WebSphere Application Server and start and stop Tivoli Configuration Manager Automation Server
<i>tioappadmin</i>	User ID used to administer Tivoli Configuration Manager Automation Server through its Web interface

To change the password for one of the default user IDs:

1. Log on as *tioadmin*.
2. Ensure that the WebSphere Application Server is started.
3. Open a Cygwin bash window.
4. Ensure that the following variables are currently defined: *WAS_HOME*, *JAVA_HOME*, *TIO_HOME*. These environment variables should be defined after the Tivoli Configuration Manager Automation Server installation process is complete.
5. Switch to the *\$TIO_HOME/tools* directory, where *\$TIO_HOME* is the directory where Tivoli Configuration Manager Automation Server is installed and run the following command:

- `./changePassword.cmd user_ID new_password current_was_password`

where:

user_ID The user ID that has the password you want to change.

new_password The new password you want to use for the user ID.

current_was_password

The current password for the *wasadmin* user ID. If you have not yet changed the *wasadmin* password, use the default value.

You can only change the password for one user ID at a time.

6. Except when changing the password for user ID *tioappadmin*, you must restart the WebSphere Application Server and the Tivoli Configuration Manager Automation Server after each password change for the change to take effect. For more information, refer to “Starting the Tivoli Configuration Manager Automation Server” on page 37 and “Stopping the Tivoli Configuration Manager Automation Server” on page 38.

Notes:

- a. Stopping and starting the Tivoli Configuration Manager Automation Server requires you to enter the *wasadmin* user name and password. After using the **changePassword** command to change the *wasadmin* password, stop the server and enter the *old* *wasadmin* password. When starting the server, with the *./tio_start.cmd* command, use the *new* *wasadmin* password.
- b. Passwords used for the Automated Patch Management solution must be changed while the Tivoli Configuration Manager Automation Server is running. If, however, the password for user *tioldap* is changed in error while the Automation Server is down, you must reset the password to the previous one, start the Automation Server, and then issue the **changePassword** command to implement the new password change.

Changing WSUS values

You can modify the WSUS values specified in the installation response file using the Tivoli Configuration Manager Automation Server console.


Changing WSUS values

To modify any of the WSUS values, you must first log on to the main Tivoli Configuration Manager Automation Server console:

1. Open a Web browser and enter the following URL:
`http://host_name:9080/tcWebUI`
where *host_name* is the fully-qualified domain name of the server. The Log On window opens.
2. Enter your user name and password. The default user name is `tioappadmin` and if you have not already changed the password, the default password is `tioappadmin`.


Changing the WSUS_SERVER_NAME value

To change the `WSUS_SERVER_NAME` value, do the following:

1. From the **Data center assets and resources** tab, expand **Inventory > Servers > MS-WSUS** and click the existing WSUS server name. This displays the WSUS general server page.
2. On the WSUS general server page, click the **Management** menu on the top right and select **To Maintenance**. This brings the server offline and turns the maintenance mode on.
3. The WSUS server status changes from **available** to **in maintenance**.
4. On the same page, click the **Edit** menu on the top right and select **Properties**.
5. Enter the new WSUS server name and click **Save**.
6. Click the **Management** menu again and select **Out of Maintenance**. This brings the server online again and turns the maintenance mode off.
7. Select the **System configuration and workflow management** tab and click **Configuration**. This displays the Data Center Configuration page.
8. From the Data Center Configuration page, click the **Variables** tab. This displays the User defined variables page.
9. On the User defined variables page, find the `WSUS_Server` key and click 
associated with that key.
10. Enter the new WSUS server name in the value field and click **Save**.

Changing the WSUS_IP_ADDRESS value

To change the `WSUS_IP_ADDRESS` value, do the following:

1. From the **Data center assets and resources** tab, expand **Inventory > Servers > MS-WSUS** and click the existing WSUS server name. This displays the WSUS server page.
2. On the WSUS server page, click 
and select **Edit Interface**.
3. Enter the new IP address in the IP and subnetwork fields and click **Save**.

Administering the workflow

This section describes how to run, stop, and schedule the workflow within the Tivoli Configuration Manager Automation Server. For a description of the operations performed by the workflow, see “Workflow Description” on page 98.

Running the workflow

The workflow used within Tivoli Configuration Manager Automation Server is named `Group_Status_Updater` and is scheduled to run automatically at 11:00 p.m. every day. If you want to run the workflow outside the scheduled time, you can do so by following the instructions below. For information on modifying the workflow schedule, refer to “Scheduling the workflow” on page 42

To run the workflow:

1. Open a Web browser and enter the following URL:
`http://host_name:9080/tcWebUI`
where *host_name* is the fully-qualified domain name of the server. The Log On window opens.
2. In the Log On window, enter your user name and password. The default user name is `tioappadmin` and if you have not already changed the password, the default password is `tioappadmin`, and click **Log On**.
3. From the System Configuration tab, click **Configuration > Workflows**. The All Workflows page displays all of the workflows currently defined in the system.
4. On the All Workflows page, click the `Group_Status_Updater` workflow.
5. Click **Execute > Run > Run**.

Stopping the workflow

To stop a workflow that is running:

1. After you have run a workflow, the Deployment Requests page is displayed. On that page, find the deployment request ID that you want to work with. Each deployment request is identified by **Request ID**, **Start Time**, and **Status**.
2. Select the **Request ID** to be stopped. The Execution Logs page is displayed.
3. On the Execution Logs page, click



Stop Execution.

Note: Selecting **Stop Execution** displays a cancelled status but the underlying process might still be creating the software package and activity plan, or it might still be moving patch files from the WSUS server workstation to the Automation Server workstation depending on when the process was stopped.

Checking workflow status

To display the status for the `Group_Status_Updater` workflows:

1. Click **System configuration and workflow management > Workflow Executions**.
2. In the **Workflow Name** list, select the `Group_Status_Updater` workflow or select **All** to view all workflows with a run history.
3. Select a status of **all**, **success**, **failed**, **in-progress** or **created** from the Status list.
4. In the **From** and **To** lists, specify the time interval that you are interested in.
5. Click **Search**. The table displays the history for all of the workflows available in the system, according to your criteria.

For workflows with a **failed** status, click **X** in the Status column to display the related error message and other details.

Scheduling the workflow

In Tivoli Configuration Manager Automation Server, scheduling is set by default at installation time to run the workflow daily. You can change the scheduled workflow to run at a specific date and time, or at regular intervals by editing the `TEDWScheduler.ini` file. This option is useful when you need to run the workflow for testing during non-peak hours, or when you want to run the workflow on a regular basis.

To change the workflow schedule, do the following:

1. Locate the `TEDWScheduler.ini` file in the `$TIO_HOME/config` directory.
2. Open the `TEDWScheduler.ini` file in an editor and change the parameter values in the file to schedule the workflow. Refer to Table 13 on page 42 for the values for each variable.

Note: To implement your changes immediately, you must stop and restart the Automation Server to run the workflow according to the new schedule. Otherwise, the next workflow runs according to the original schedule and only the subsequent workflows will run according to the new schedule. See “Administering Tivoli Configuration Manager Automation Server” on page 37 for information about starting and stopping the Automation Server.

Table 13. Workflow schedule values

User Input Field	Description
<code>START_TIME1=<value></code>	Replace <code><value></code> with the time and date the workflow should run. The time and date specified should be in the <code>H:mm yyyy.MM.dd</code> format where H = Hour in a day (0-23) mm = Minutes in an hour yyyy = Year MM = Month in a year dd = Day in a month
<code>REPEAT_TIME1=<value></code>	Replace <code><value></code> with the time interval in which the workflow will be repeated. The <code><value></code> specified should be in minutes.

Chapter 4. Scanning for patches

This chapter describes how the patch scan works, how to scan for patches, and how to use the Inventory component of IBM Tivoli Configuration Manager to collect information about which patches are installed or missing in the environment. It also contains information about the patch management tables and views in the Inventory database. The tables and views are updated when you run `inv_dbvendor_schema_423_FP01.sql`, and `h_inv_dbvendor_patch_mgmt_423_FP01.sql` scripts. Refer to the *Planning and Installation Guide* for more information about running database scripts.

The automated patch management solution uses the Windows Update Agent (WUA), which is a tool used to scan Windows computers for security updates.

It can scan both the base operating system and other add-on applications for missing security updates. The scanning mechanism is effective only if performed with an up-to-date .cab file. The .cab file contains the latest information about security updates that are available for Microsoft products. The Windows Update Agent then queries the system to discover all patches that are currently installed and compares this list with the information contained in the .cab file.

After the comparison, the Windows Update Agent produces a list of all patches with their related status. This data is stored in the Inventory database. To discover all Windows products, ensure you installed the latest level of Windows Installer on the endpoints.

The .cab file is frequently updated by Microsoft and is regularly downloaded by the Automation Server workflow to the `$DBDIR/inventory` directory on the Tivoli server. If the .cab file is not present on the endpoint, the Windows Update Agent cannot run on that endpoint and the inventory scan fails with an error level of 1.

The approval mechanism supplied with Microsoft Windows Server Update Services (WSUS) can be used by WUA at scan time to limit the discovery process only to approved patches. Patches approved by the Administrator are flagged in the `ApprovedChanges.txt` file that contains a list of all available Microsoft patches. The `ApprovedChanges.txt` file is automatically downloaded to the same location as the .cab file. An Inventory profile distribution then copies the files to the `$(LCF_DATDIR)/../../INV/SCAN` path on all endpoints addressed by the distribution.

Optimizing the wsusscan.cab download

To avoid excessive bandwidth allocation caused by downloading `wsusscan.cab`, you can decide to download this file to the endpoints only when it is newer than the version already present at the endpoints and not at every scan.

To download this file only if needed, perform the following tasks:

1. Copy the `wsusscan.cab` file to the Tivoli gateways using the `cab_gateway_list` or `cab_gateway_file` keys in the `wseccfg` command.
2. Prevent the Inventory Profile from downloading the .cab file during the distribution of the profile to avoid downloading it twice:

```
wsetinvpcsw -d Y -n wsusscan.cab @InventoryConfig:Windows_Patch_Scan
```

Optimizing the wsusscan.cab download

To enable the download of the wsusscan.cab file again, run the following command:

```
wsetinvpcsw -d N -n wsusscan.cab @InventoryConfig:Windows_Patch_Scan
```

Submitting the patch scan

During the installation of the Patch Management component, the Windows_Patch_Tools profile manager is created containing two predefined Inventory scans. These scans rely on WUA, so ensure that you have previously downloaded the WUA to the endpoints, as described in “Deploying WUA and Qchain on endpoints” on page 30. The two predefined scans are:

- **Windows_Initial_Patch_Scan:** Run this scan manually the first time to populate the database with missing and found patch information about the endpoints in your environment. It replaces the current data with up-to-date data. This scan ignores the ApprovedChanges.txt file.
- **Windows_Patch_Scan:** Run this scan to return any differences found between the information on the endpoint and that currently stored in the database. The ApprovedChanges.txt file filters the missing patches list discovered by the scan by eliminating those patches not flagged as being approved. If the ApprovedChanges.txt file is missing, the scan fails.

On disconnected endpoints, before running the isolated patch scan as described in section “Scanning disconnected systems” of the *User’s Guide for Inventory*, you must:

1. Ensure you installed WUA.
2. Copy the .cab, ApprovedChanges.txt and qchain.exe files into the directory where you copied the other files needed for scanning.

To submit the scan, distribute the related inventory profile from the Tivoli desktop or use the **wdistinv** command from the command line. Refer to the *IBM Tivoli Configuration Manager: User’s Guide for Inventory* for more detailed information about distributing an inventory profile from the Tivoli desktop or CLI.

Discovering missing and installed patches

There are two types of information retrieved from the endpoints using the two pre-defined scans:

- Information about discovered vulnerabilities. This information is stored in the PM_PATCH_INFO table of the database.
- Information about the supported products and related operating systems. This information is stored in the PM_PRODUCT_INFO table of the database.

Patch management tables and views

The following tables are created in the database to store patch-related information:

Table 14. Patch management tables

Table name	Description
PM_PATCH_INFO	Stores patch information discovered by the scan engine for each endpoint.
H_PM_PATCH_INFO	Maintains historical data on information discovered by the scan engine.

Table 14. Patch management tables (continued)

Table name	Description
PM_PRODUCT_INFO	Maintains information on supported products, related operating systems, and installation languages of applications supported by the Patch Management component.
PM_PATCH_PKG	Maintains a link between the name of the software package and the patch for which the software package is created.
INV_GROUP	Stores group information.
INV_GROUP_EP	Store endpoint group information.

New patch management views are created:

Table 15. Patch management views

Views	Description
PM_PATCHES_VIEW	Returns information on patches and products.
SP_PATCHES_VIEW	Returns information on patches and related software packages, if any.
EP_PATCHES_VIEW	Returns information on patches, software packages status on the endpoints
H_PM_PATCHES_VIEW	Returns historical information on patches and products. Contains the RECORD_TIME field, which indicates the time the data was updated at the database.
H_SP_PATCHES_VIEW	Returns historical information on patches and related software packages, if any. Contains the RECORD_TIME field, which indicates the time the data was updated at the database.
H_EP_PATCHES_VIEW	Returns historical information on patches, software packages status on the endpoints. Contains the EXEC_TIME field, which indicates the time, on the endpoint, the last successful operation was performed on the software package.
INV_GRP_EP_VIEW	Returns the endpoints belonging to a group.

Information stored in the PM_PATCH_INFO table

The following is an excerpt of the `inv_vendor_schema.sql` SQL script that creates the PM_PATCH_INFO table for DB2 databases.

```
create table PM_PATCH_INFO (
  COMPUTER_SYS_ID  varchar(64) not null,
  ID               varchar(128),
  PRODUCT          varchar(255) not null,
  BULLETIN         varchar(32) not null,
  QNUM             varchar(32) not null,
  GUID             varchar(64) not null,
  UPDATEID         varchar(64) not null,
  REBOOTBEHAVIOR   varchar(20) not null,
  REASON           varchar(255) not null,
```

Patch management tables and views

```
STATUS          varchar(64) not null,  
PRODUCT_CODE    varchar(4)  not null,  
RECORD_TIME     TIMESTAMP   DEFAULT CURRENT_TIMESTAMP,  
  
constraint PMPATCH_PK primary key(COMPUTER_SYS_ID, QNUM, GUID),  
constraint PMPATCH_FK foreign key(COMPUTER_SYS_ID) references  
COMPUTER(COMPUTER_SYS_ID)  
);
```

where:

COMPUTER_SYS_ID

Is the computer system ID.

ID Is a checksum of bulletin and Q number values.

PRODUCT

Identifies the product that needs the patch.

BULLETIN

Identifies the Microsoft bulletin that released the patch.

QNUM

Indicates the Hotfix number.

GUID Defines the internal unique identifier WSUS uses to identify the components of a patch.

UPDATEID

Defines the unique identifier with which WSUS identifies the patches.

REBOOTBEHAVIOR

Defines if the patch requires a reboot. It is not used.

REASON

Is a comment to the patch, when provided.

STATUS

Identifies the status of the patch. It can be: Found or NOT Found.

PRODUCT_CODE

Identifies the code of the product. It can be OS (Operating System).

RECORD_TIME

A timestamp which indicates the last update performed on the row.

Information stored in the H_PM_PATCH_INFO table

The following is an excerpt of the `h_inv_vendor_patch_mgmt_423_FP01.sql` script that creates the H_PM_PATCH_INFO table for DB2.

```
create table H_PM_PATCH_INFO (  
  COMPUTER_SYS_ID varchar(64) not null,  
  ID               varchar(128),  
  PRODUCT          varchar(255) not null,  
  BULLETIN         varchar(32)  not null,  
  QNUM             varchar(32)  not null,  
  GUID             varchar(64)  not null,  
  UPDATEID         varchar(64)  not null,  
  REBOOTBEHAVIOR   varchar(20)  not null,  
  REASON           varchar(255) not null,  
  STATUS           varchar(64)  not null,  
  PRODUCT_CODE     varchar(4)   not null,  
  RECORD_ACTION    char(6),  
  PRFL_ACTION      varchar(20),  
  RECORD_TIME      timestamp not null,
```

```

constraint H_HPMPATCH_PK primary key(COMPUTER_SYS_ID, GUID, RECORD_TIME),
constraint H_HPMPATCH_FK foreign key(COMPUTER_SYS_ID) references
    COMPUTER(COMPUTER_SYS_ID)
);

```

where:

COMPUTER_SYS_ID

Is the computer system ID.

ID Is a checksum of bulletin and Q number values.

PRODUCT

Identifies the product that needs the patch.

BULLETIN

Identifies the Microsoft bulletin that released the patch.

QNUM

Indicates the Hotfix number.

GUID Defines the internal unique identifier WSUS uses to identify the components of a patch.

UPDATEID

Defines the unique identifier with which WSUS identifies the patches.

REBOOTBEHAVIOR

Defines if the patch requires a reboot. It is not used.

REASON

Is a comment to the patch, when provided.

STATUS

Identifies the status of the patch. It can be: Found or NOT Found.

PRODUCT_CODE

Identifies the code of the product. It can be OS (Operating System).

RECORD_ACTION

Specifies whether the record is an INSERT (new information is being added to the operational data table), an UPDATE (some part of a record in the operational data table is being modified), or a DELETE (the record no longer exists in the operational data table).

PRFL_ACTION

Specifies whether the profile configuration option was REPLACE (Replace with Current® Results) or REPLACE_WITH_DIFF (Update with Differences).

RECORD_TIME

A timestamp which indicates when the last update was performed on the row.

Information stored in the PM_PRODUCT_INFO table

The following is an excerpt of the SQL script that creates the PM_PRODUCT_INFO table for DB2.

```

create table PM_PRODUCT_INFO (
  COMPUTER_SYS_ID  varchar(64)  not null,
  PRODUCT_CODE     varchar(4)   not null,
  PROD_MAJOR_VER   varchar(4)   not null,
  PROD_MINOR_VER   varchar(4)   not null,
  PROD_LANG        varchar(16)  not null,
  OS_BASE_NAME     varchar(16)  not null,

```

Patch management tables and views

```
OS_ARCHITECTURE  varchar(16) not null,  
OS_TYPE          varchar(16),  
OS_SUBTYPE       varchar(16),  
OS_SP_MAJOR_VER  varchar(4),  
OS_SP_MINOR_VER  varchar(4),  
RECORD_TIME      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
constraint PMPROD_PK primary key(COMPUTER_SYS_ID, PRODUCT_CODE, PROD_LANG),  
constraint PMPROD_FK foreign key(COMPUTER_SYS_ID) references  
    COMPUTER(COMPUTER_SYS_ID)  
);
```

where:

COMPUTER_SYS_ID

Is the computer system ID.

PRODUCT_CODE

Identifies the code of the product. It can be OS (Operating System).

PROD_MAJOR_VER

Identifies the major version of the product.

PROD_MINOR_VER

Identifies the minor version of the product.

PROD_LANG

Identifies the language of the product, for example, ENUS.

OS_BASE_NAME

Identifies the name of the operating system.

OS_ARCHITECTURE

Identifies the architecture of the operating system, for example, x86.

OS_TYPE

Identifies the type of the operating system, for example, srv.

OS_SUBTYPE

Identifies the subtype of the operating system, for example, ent.

OS_SP_MAJOR_VER

Identifies the major version of the service pack.

OS_SP_MINOR_VER

Identifies the minor version of the service pack.

RECORD_TIME

A timestamp which indicates when the last update was performed on the row.

Information stored in the PM_PATCH_PKG table

The following is an excerpt of the SQL script that creates the PM_PATCH_PKG table for DB2.

```
create table PM_PATCH_PKG (  
    SWARE_NAME      varchar(128) not null,  
    SWARE_VERS      varchar(64)  not null,  
    REGION_ID       varchar(20)  not null,  
    PRODUCT_CODE    varchar(4)   not null,  
    QNUM            varchar(32)  not null,  
    GUID            varchar(64)  not null,  
    constraint PMPATCHPKG_PK primary key(GUID, SWARE_NAME, REGION_ID)  
);
```

where:

SWARE_NAME

Indicates the name of the file associated with the software application.

SWARE_VERS

Indicates the version of the software application.

REGION_ID

Indicates the Tivoli region ID for the software package.

PRODUCT_CODE

Identifies the code of the product. It can be OS (Operating System) or IE (Internet Explorer).

QNUM

Indicates the Hotfix number.

GUID Defines the internal unique identifier WSUS uses to identify the components of a patch.

Information stored in the INV_GROUP table

The following is an excerpt of the SQL script that creates the INV_GROUP table for DB2.

```
create table INV_GROUP (
    GROUP_LABEL      varchar(64) not null,
    GROUP_LABEL_ID   varchar(128) not null,
    GROUP_DESCR      varchar(255) not null,

    constraint GROUP_PK primary key(GROUP_LABEL_ID)
);
```

where:

GROUP_LABEL

Indicates the label of the group.

GROUP_LABEL_ID

Indicates the label ID of the group.

GROUP_DESCR

Indicates the description of the group

Information stored in the INV_GROUP_EP table

The following is an excerpt of the SQL script that creates the INV_GROUP_EP table for DB2.

```
create table INV_GROUP_EP (
    GROUP_LABEL_ID   varchar(128) not null,
    COMPUTER_SYS_ID  varchar(64) not null,
    Constraint GROUP_EP_PK primary key(GROUP_LABEL_ID,COMPUTER_SYS_ID),
    constraint GROUP_EP_CP_FK foreign key(COMPUTER_SYS_ID)
    references COMPUTER(COMPUTER_SYS_ID),
    constraint GROUP_EP_FK foreign key(GROUP_LABEL_ID)
    references INV_GROUP(GROUP_LABEL_ID)
);
```

where:

GROUP_LABEL_ID

Indicates the label ID of the group.

COMPUTER_SYS_ID

Indicates the computer sys ID of the endpoint.

Filtering View

The following is an excerpt of the SQL script that creates the INV_GRP_EP_VIEW view for DB2.

```
-- drop view INV_GRP_EP_VIEW;
-- delete from QUERY_VIEWS where VIEW_NAME = 'INV_GRP_EP_VIEW';
create view INV_GRP_EP_VIEW
as
select
    INV_GROUP.GROUP_LABEL,
    INV_GROUP.GROUP_LABEL_ID,
    INV_GROUP_EP.COMPUTER_SYS_ID
from
    INV_GROUP
JOIN
    INV_GROUP_EP
ON
    INV_GROUP.GROUP_LABEL_ID=
    INV_GROUP_EP.GROUP_LABEL_ID
;
```

Chapter 5. Patch installation

This chapter describes the tasks you must perform to make an initial assessment of the missing and found patches and vulnerabilities in your environment.

This chapter also describes the tasks you can perform once the patches are ready to be installed in the context of a use-case scenario. The scenario demonstrates the following tasks:

- “Deployment Paradigms”
- “Performing the initial patch scan” on page 52
- “An end-to-end scenario” on page 53
- “Managing emergency patches” on page 55
- “Deploying patches to groups of endpoints” on page 57
- “Viewing the software packages” on page 58
- “Viewing the activity plan” on page 59
- “Submitting and monitoring the plan” on page 60
- “Checking the results” on page 60
- “Managing the patch lifecycle” on page 61
- “Customizing the software package and plan templates” on page 61

Before performing a scan, ensure you installed the WUA at the same level of the .cab file you have on your endpoints. You can then group the endpoints according to the .cab file they are using.

Deployment Paradigms

You can extend the Software Distribution capability of retrieving software packages to be installed from a depot or a file server to Patch Management by using the `deployment_paradigm` key in the `wseccfg` command as follows:

```
wseccfg -s deployment_paradigm=[standard | from_depot | from_fileserver]
```

where:

- | | |
|-------------------|---|
| standard | Specifies that the software package to be installed resides on the source host. This is the default value. |
| from_depot | Specifies that the software package to be installed resides on the repeater depot, rather than on the source host. With the <code>from_depot</code> key you must specify one of the following keys in the <code>wseccfg</code> command: <ul style="list-style-type: none">-s <code>sp_patches_depots=depot1, depot2,...</code>
Use this key to specify in <code>depot1, depot2,...</code> the name of the gateways or stand-alone repeaters where to load the packages. These names must be separated by a comma.-s <code>sp_patches_depots_file=depotfile</code>
Use this key to specify in the <code>depotfile</code> file the list of gateways or stand-alone repeaters where to load the packages. |

You can also specify the following key:

-s depots_unload=[yes(default) | no]

Use this key to specify if the unload operation must be performed or not.

from_fileserver

Specifies that the images referenced in the software package are to be retrieved from a file server. After the software package, query, and plan creation through the workflow and before the plan submission, perform the following configuration steps:

1. Create an installable image of the software package block using the following command:

```
wldsp -l depot_image_dir=provider_spb_dir  
@SoftwarePackage:spobj_name [@subscribers ...]
```

where:

provider_spb_dir

Is the directory on the source host where the .spb files are copied after the software package for a given patch has been created. This value is specified in the **wseccfg** command.

@subscribers

Specifies the source hosts defined using the **wseccfg** command.

Refer to the *Reference Manual for Software Distribution* for the syntax and usage of the **wldsp** command.

These commands create two files with the extensions .itc or .toc and .dat. These two files contain the data to be used during the distribution operation.

2. Load or copy the .itc or .toc and .dat files to the file server in a directory shared by all endpoints.
3. Create and configure the remote.dir file on the endpoints to access the file server.
4. Copy the remote.dir file under \$LCF_DATDIR on each endpoint. This file contains a list of available file servers, one per line. You can use Software Distribution to distribute the remote.dir file to the endpoints in a software package. When the endpoint lcfld receives the distribution, it looks for the content of the distribution in the specified shared directory of the first file server listed in the remote.dir file. If it does not find the file, it looks for it on the next file server listed in the remote.dir file.
5. Mount the shared directory of the file server on each endpoint using, for example, the **wlcfmap** command.

Performing the initial patch scan

Before the patch management automation is up and running, you need to determine which patches are already present in your environment, and which patches need to be installed on which endpoints. This chapter describes the tasks required to populate the database with the data on missing and found patches:

1. Make sure WUA and qchain.exe files have been distributed to all endpoints in your environment. For more information on distributing these files, see “Deploying WUA and Qchain on endpoints” on page 30.

2. Set the **workflow_activities** key to sync using the **wseccfg** command.
3. Run the Automation Server workflow manually or wait for the next scheduled workflow to start. In this phase of the process, the workflow is in charge of downloading the latest version of the .cab file from the Microsoft Web site. This file is required to perform any patch scan. For more information, see “Running the workflow” on page 41.
4. If the **notification_email_address** key was configured using the **wseccfg** command, the Administrator receives an e-mail when the workflow has completed. If this option is not configured, the Administrator needs to open the Automation Server console, as described in “Checking workflow status” on page 41. The following is the text of the e-mail notification:

```
*
* THIS MESSAGE HAS BEEN SENT BY AN AUTOMATIC SERVICE MACHINE. PLEASE DO NOT REPLY TO THIS MAIL.
*
```

The Automation Server has completed running the automated patch management workflow.

General status (RequestId - 10000): SUCCESS

*** Execution Report:

```
>> Download of patch executable files to the Automation Server Workstation:
- Synchronization process for Windows All succeeded.
```

```
>> Download of wsusscn2.cab:
- Operation successful.
```

```
>> Download of ApprovedChanges.txt:
- Operation successful.
```

```
>> Distribution of wsusscn2.cab and ApprovedChanges.txt to the Tivoli Servers:
- Operation successful.
```

(For additional details see the Automation Server Execution Log)

5. The Administrator submits the **Windows_Initial_Patch_Scan** which populates the database with the status of the patches listed in the .cab. This task needs to be performed only at this time.
6. After the results from the scan are returned to the Inventory database, the Administrator runs the **wsecrpt** command to obtain the list of missing patches as described in the example:

```
wsecrpt -sM -f header
```

The automated patch management solution can now automatically download approved patches from WSUS and manage the environment with minimal user intervention.

An end-to-end scenario

This scenario integrates some of the steps described in the previous chapter and integrates them in a detailed scenario. In this scenario, Microsoft releases an important security bulletin, MS05-013 (KB891781), for Windows systems. The Administrator responsible for approval determines that this security update needs to be implemented in the enterprise. It applies to Windows XP family systems only. The following process outlines the sequence of steps involved in deploying the patch using the automated patch management solution. See “A road map to implement the automated patch management solution” on page 6 for the sequence of steps to follow for first-time implementation.

End-to-end scenario

1. The Administrator responsible for patch approval connects to the Microsoft WSUS Web site and synchronizes the WSUS server with the Microsoft Windows Update server.
2. The Administrator approves the Security Update for Windows XP from the WSUS Web site.
3. Set the **workflow_activities** key to sync using the **wseccfg** command.
4. Run the Automation Server workflow manually, or wait for the next scheduled workflow to complete if it is set to run within a short time. For more information, see “Running the workflow” on page 41. The purpose of submitting the workflow at this point is to download the latest .cab and ApprovedChanges.txt files to the Tivoli server.
5. If the **notification_email_address** key was configured with the **wseccfg** command, the Administrator receives an e-mail when the workflow has completed. If this option is not configured, the Administrator needs to open the Automation Server console, as described in “Checking workflow status” on page 41.
6. The Administrator responsible for running Inventory scans submits the Windows_Patch_Scan. This assumes that the Windows_Initial_Patch_Scan was run at an earlier time.
7. As the results of the scan are returned to the Inventory database, the Administrator runs the **wsecrpt** command to generate a list of endpoints requiring the important patch as follows:

```
wsecrpt -sM -ge -f header -p 891781
```

The command produces the following output.

```
** lab133050
```

QNUMBER	PACKAGE NAME	REGION ID	BULLETIN	PRODUCT LANGUAGE	PRODUCT CODE
891781			MS05-013	ENUS	OS

```
** lab134162
```

891781			MS05-013	ITIT	OS
--------	--	--	----------	------	----

See Chapter 6, “Automated patch management command line,” on page 67 for the command syntax and arguments for the **wsecrpt** command.

8. Optionally, you can modify the way in which software packages and activity plans are automatically prepared by modifying the corresponding .spd template (software package) and .xml template (activity plan). See “Customizing the software package and plan templates” on page 61.
9. Set the **workflow_activities** key to preparation with the **wseccfg** command.
10. Submit the Automation Server workflow a second time, or wait for the next scheduled workflow to run if it is set to run within a short time. See “Running the workflow” on page 41 for information.
11. The Administrator receives an e-mail indicating that the workflow has completed and that activity plans have been generated and are ready to be submitted. The notification is sent if you have previously configured the **email_notification_address** key with the **wseccfg** command. If this option is not configured, the Administrator needs to open the Automation Server console, as described in “Checking workflow status” on page 41.
12. The Administrator runs the **wsecrpt** command again to check that the software packages related to the important patch have been generated for each endpoint, but have not yet been installed:

```
wsecrprt -cR -ge -f header -p 891781
```

The command produces the following output. The "PACKAGE NAME" column contains the name of the software package containing the patch executable.

```
** lab133050
```

QNUMBER	PACKAGE NAME	REGION ID	BULLETIN	PRODUCT LANGUAGE	PRODUCT CODE
891781	patch.891781.46 5CE113-6F63-45 AC-BFD7-ABF 1E93686FA.b	1429578020	MS05-013	ENUS	OS

```
** lab134162
```

891781	patch.891781.835 FD4E2-AB77-4F FA-834B-77FB8 B88BC66.b	1429578020	MS05-013	ITIT	OS
--------	---	------------	----------	------	----

13. Launch the Tivoli desktop to view the software packages prepared by the Automation Server that will be distributed to the endpoints when the activity plan containing the software packages is submitted. See "Viewing the software packages" on page 58.
14. Open the plan or plans in the Activity Plan Editor GUI to view or modify the plan, if necessary. See "Viewing the activity plan" on page 59. Refer to the *User's Guide for Deployment Services* for more information about using the GUI and modifying plans.
15. Submit the plan to run from the Activity Plan Monitor GUI, and monitor the patch installation progress. Refer to the *User's Guide for Deployment Services* for more information about submitting and monitoring activity plans.
16. A final activity is defined in each plan that updates the database with the status of the software packages on the endpoints.
17. Run the **wsecrprt** command to view the results of the patch installation and ensure that endpoints that previously reported the patch "NOT Found" now have a status of "Found", as described in the example:

```
wsecrprt -sM -sF -ge -f header -p 891781
```

For more information, see "Checking the results" on page 60.

Managing emergency patches

If you are the Administrator responsible for approval and you determine that an update, released in an important Microsoft security bulletin, needs to be implemented immediately, you can use the emergency patch feature to defer the preventive inventory scan and install the update as soon as possible.

You can specify the list of patches you want to manage as emergency patches using the `emergency_patches` configuration key in the **wseccfg** command as follows:

```
wseccfg -s emergency_patches=patchInfo1, patchInfo2,..., patchInfoN
```

The *patchInfoN* values must be separated by a comma.

Managing emergency patches

Before defining the *patchInfoN* values, you must identify the patch on WSUS and approve it. You can then collect the *updateID* and the platform on which to install the patch and then define the *patchInfoN* value as follows:

```
updateID.os_base_name[.os_architecture[.os_type[.os_subtype]]]
```

where:

updateID Is the unique identifier for the patch (it is not the Qnumber) and is defined in WSUS.

os_base_name Identifies the name of the operating system (winxp, win2k, win2k3, vista).

os_architecture Identifies the x86 architecture.

os_type Identifies the type of the operating system: srv, wks.

os_subtype Identifies the subtype of the operating system: dtc, ent, pro, bld, hom, ts, std.

The *updateID* and *os_base_name* parameters are mandatory. They are used to determine the targets to which the patch has to be deployed. If other non-required parameters (such as *os_architecture*, *os_type*, *os_subtype*) are not specified, a very complex APM plan might be created which addresses a high number of endpoints causing network overload problems.

For example, if you want to distribute a patch for Windows 2000 Professional and the *patchInfoN* key that you specify is *updateID.win2k*, the patch is sent to all Windows 2000 endpoints.

Instead, if the *patchInfoN* key that you specify is *updateID.win2k.x86.wks.pro*, the patch is sent only to Windows 2000 Professional endpoints.

TCM_Emergency_Patches Workflow

You can manage emergency patches using the new *TCM_Emergency_Patches.wkf* workflow. Its execution depends on the *emergency_patches* variable settings. If the variable is not set, an error is reported.

Before running the *TCM_Emergency_Patches.wkf* workflow, ensure you run the *Windows_Initial_Patch_Scan* on the endpoints.

Software packages, queries, and APM plans for the emergency patches are created with the following naming convention and are considered separately from the standard patches:

Emergency software packages	<code>hot_patch.qnumber.guid.os_base_name.[os_architecture].[os_type].[os_subtype].[b]^1.0</code>
Emergency queries	<code>hot_query.qnumber.guid.os_base_name.[os_architecture].[os_type].[os_subtype].[b]</code>
Emergency plan	<code>hot_patch.plan_name</code>

The workflow performs the following steps:

1. Downloads emergency patches and *ApprovedChanges.txt* from WSUS.
2. Downloads the Microsoft Security Policy Catalog from the Microsoft Web site.
3. Uploads the catalogs to the Tivoli servers and Tivoli gateways.
4. Invokes the **wsecgensp** command to create all software packages and queries related to the specified emergency patches.

Note: To delete software packages, queries, and the APM plan related to previous emergency patches, set the `emergency_delete_packages_plan` configuration key to yes.

5. Invokes the **wsecgenplan** command to generate APM plans with activities that install emergency patches. These activities are conditioned by the installation of the WSUSSCANCAB^1.0 software package, used to:
 - Refresh the .cab catalog on the endpoints. The .cab catalog is taken from the \$DBDIR/inventory directory of the Tivoli server specified in the TMR_server_list key of the **wseccfg** command.
 - Deploy the check_patch.cmd file on the endpoints. This batch file is run to determine if a patch has to be installed on an endpoint.
6. If the email_notification_address key was configured using the **wseccfg** command, sends an e-mail to the Administrator when the workflow has completed.

When you submit the APM plans with activities that install emergency patches, you can check if the emergency patch has been installed by looking at the Software Distribution log file. This log lists the status of the emergency patch installation on the endpoints:

```
DISSE0442I Execution of user program 'during_install - ..\..\inv\SCAN\check_path.cmd
(-status)' completed with result: 'success'
DISSE0198I User program exit code: n
```

Where *n* can have the following values:

- | | |
|----------|--|
| 0 | The emergency patch was not installed. The emergency patch installation is run. |
| 1 | The emergency patch was installed. The emergency patch installation is not run. |
| 2 | No information about the emergency patch installation is available. The emergency patch installation is not run. |

Deploying patches to groups of endpoints

You can define filters on endpoints to group them and then deploy the security patches to these groups of endpoints.

The filters must be defined as filter types in the `filtering_template.xml` file, as described in “Customizing the target filter template” on page 65.

After you identify which of the defined filter types to use, you can associate it to the plan that distributes patches in one of the following ways, depending on how you create the plan:

- If you are using workflows:

Assign the filter type to the `target_filtering_mode` key of the **wseccfg** command. An example is:

```
wseccfg target_filtering_mode=inventory
```

where `inventory` is a filter type defined in the `filtering_template.xml` file.

- If you are using the **wsecgenplan** command:

Specify the filter type, in the `-g` parameter. An example is:

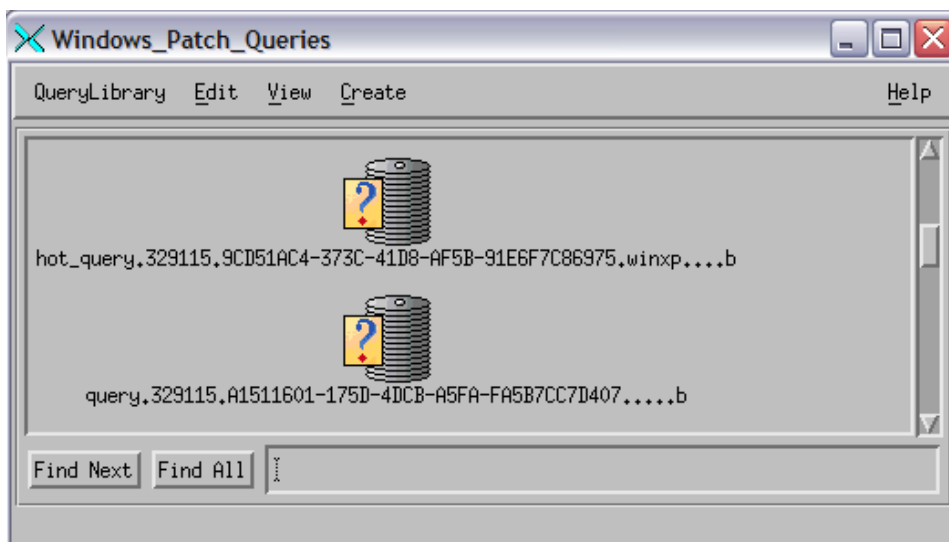
```
wsecgenplan -p patch.842526.DF8B7CBC-80DB-427D-BACD-42F7F8DD2C0A.b^1.0
-g inventory
```

where `inventory` is a filter type defined in the `filtering_template.xml` file.

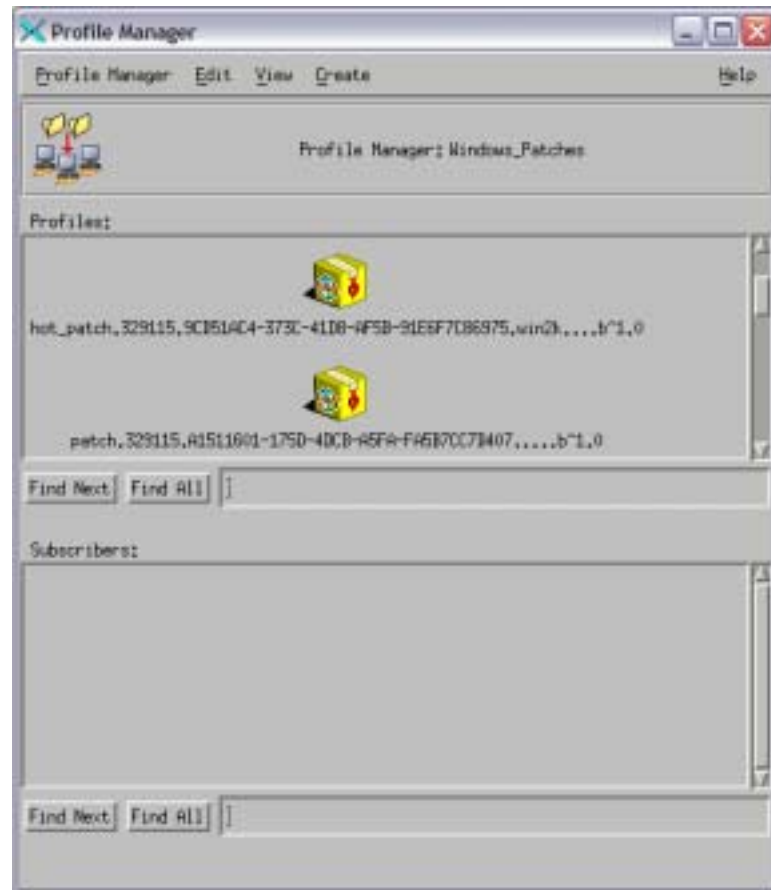
Viewing the software packages

You can view the software packages automatically generated by the Automation Server workflow from the Tivoli desktop. For the automated patch management solution, a profile manager is created in the Windows_Patches_<region_name>.

To view the software packages created for the MS05-013 (KB891781) security update, double-click the Windows_Patches profile manager from the Windows_Patches_<region_name> sub-region. The following are the software packages created to install the MS05-013 (KB891781) security update.



In addition to software packages, you will also find queries generated to retrieve the targets of the patch. The following are the queries generated for the MS05-013 (KB891781) security update.



Viewing the activity plan

Launch the Activity Plan Editor to view a graphical representation of the activity plan created by the workflow. This representation enables you to view the generated activities, conditioning, if any, as well as the Final Activity that is contained in every plan that is responsible for reporting the status of the software package installations to the Inventory database. The presence of the Final Activity depends on the template you used, as described in “Customizing the software package and plan templates” on page 61. The plan names are defined based on the following naming convention:

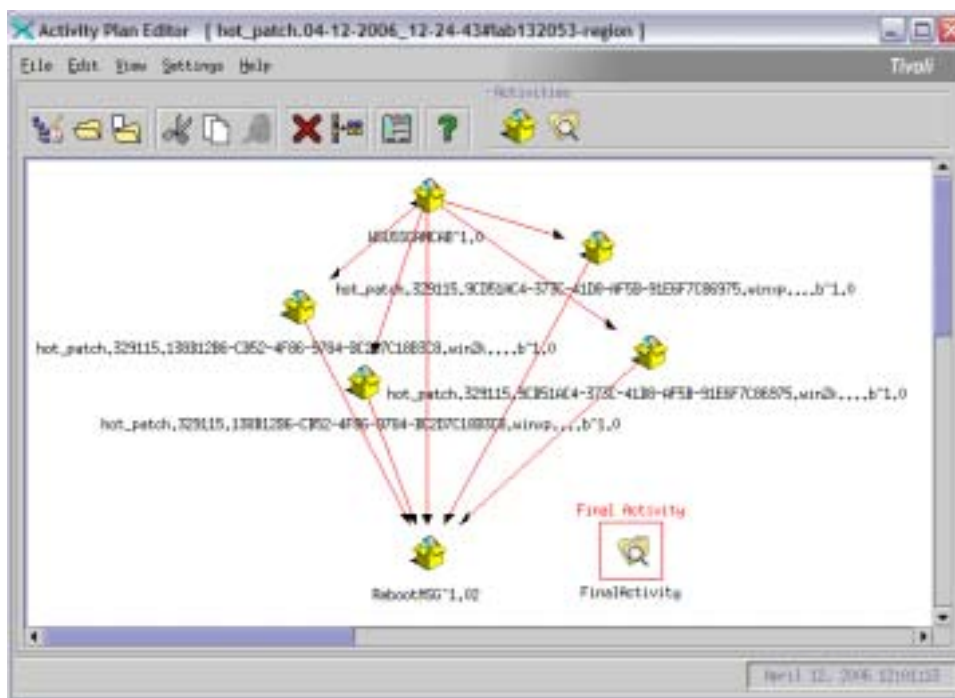
Table 16. Activity plan naming convention

Type of plan	Naming convention
Standard plan	patch.mm-dd-yyyy_hh_mm_ss_patch_id#region_id
Emergency plan	hot_patch.mm-dd-yyyy_hh_mm_ss_patch_id#region_id
Filtered standard plan	filtered.mm-dd-yyyy_hh_mm_ss_patch_id#region_id
Filtered emergency plan	hot_filtered.mm-dd-yyyy_hh_mm_ss_patch_id#region_id

The patch_id and region_id suffixes might not be present, depending on Patch Management configuration.

Viewing the activity plan

In the MS05-013 (KB891781) security update scenario, a reboot is necessary on the endpoints after the installation of the patch. The Automation Server workflow recognizes the need for the reboot and includes the RebootMSG software package, created during the installation of the Patch Management component, in the activity plan to perform the reboot. The reboot activity is conditioned by Completion-Target to the other installation activities. Refer to *IBM Tivoli Configuration Manager: User's Guide for Deployment Services* for more information about conditions on activities.



You may find other software packages created by the workflow that are not related to the approved patch. For example, if a new machine has been added to the environment that is not compliant with the list of approved patches, the Automation Server workflow will create packages and plans to address the vulnerabilities and bring the new machine up-to-date with security updates.

Submitting and monitoring the plan

Launch the Activity Plan Monitor to submit the plan. To view the activities contained within the activity plan, click the expand icon to the left of the plan name. To view the targets assigned to the activity and the status of the activity on that particular target, click the expand icon to the left of the activity. Refer to the *IBM Tivoli Configuration Manager: User's Guide for Deployment Services* for more information about submitting and monitoring the plan.

Checking the results

The **wsecrpt** command returns information about found and missing patches in your environment. You can perform advanced queries by combining the available options. For more information on this command, refer to “wsecrpt” on page 85.

You can also create specific queries to retrieve the same information retrieved by the **wsecrpt** command. For more information on creating queries, refer to *IBM Tivoli Configuration Manager: Database Schema Reference*.

Activity plans automatically generated by the Automation Server contain a final scan activity responsible for updating the Inventory repository with the state of the software packages contained in the plan. You can use the **wsecrprt** command to retrieve the information reported by the final scan activity to ascertain whether previously reported missing patches are no longer missing.

To check whether endpoints previously reported as having MS05-013 (KB891781) security update NOT Found are now reported as Found, run the command as follows:

```
wsecrprt -sF -sM -ge -f header -p 891781
```

The following is the output of the command:

```
** lab133050
QNUMBER      PACKAGE      REGION ID    BULLETIN     PRODUCT      PRODUCT      STATUS
              NAME
891781        patch.891781.  1429578020   MS05-013     ENUS         OS           Found
              465CE113-6F
              63-45AC-BF
              D7-ABF1E93
              686FA.b

** lab134162
891781        patch.891781.  1429578020   MS05-013     ITIT         OS           Found
              835FD4E2-A
              B77-4FFA-83
              4B-77FB8B8
              8BC66.b
```

Managing the patch lifecycle

The Automation Server workflow contains a mechanism to delete obsolete patches and related queries. If the Administrator revokes the approval from patches that are no longer needed, the Automation Server workflow deletes the related software package objects from the Tivoli region. To enable this behavior, set the **delete_packages** key to **yes** with the **wseccfg** command. This mechanism does not delete the software package blocks from the source host.

You can also set the **delete_plans** key to **yes** with the **wseccfg** command to delete obsolete plans. The Automation Server generates one or more plans after each workflow has completed running which addresses the latest vulnerabilities and distributes the latest patches, therefore previous plans are no longer necessary. For more information on the **wseccfg** command, see “wseccfg” on page 68.

Note: If you set the **delete_plans** parameter to **yes**, then the Automation Server workflow deletes all activity plans, not only the plans generated using the automated patch management solution.

Customizing the software package and plan templates

The software package and activity plan templates used by the Automation Server are located in \$(BINDIR)\TME\PATCH_MGMT\TEMPLATES. The following are the custom templates installed during the installation of the automated patch management solution which you can customize, if necessary:

Customizing the software package and plan templates

Table 17. Software package and plan templates

Template	Description
exclusive_template.spd	A template for a software package containing exclusive patches.
standard_template.spd	A template for a software package containing standard patches.
reboot_template.spd	A template for a software package containing patches that require a reboot
emergency_exclusive_template.spd	A template for a software package containing exclusive emergency patches.
emergency_template.spd	A template for a software package containing standard emergency patches.
emergency_reboot_template.spd	A template for a software package containing emergency patches that require a reboot
APM_REBOOT_Activity_Template.xml	A template for an activity to install a software package that performs a reboot.
APM_EXCL_Activity_Template.xml	A template for an activity to install an exclusive patch.
APM_STD_Activity_Template.xml	A template for an activity to install a standard patch.
APM_DEPOT_Activity_Template.xml	A template for an activity to install a patch from a depot.
APM_CAB_Activity_Template.xml	A template for an activity to deploy the .cab and check_patch.cmd files on the endpoints.
APM_LAST_Activity_Template.xml	A template for a final activity. By default the final activity is a Inventory scan.
APM_PLAN_Template.xml	A template for a standard activity plan.
filtering_template.xml	A template for defining target filters.

The templates provided address the three different patch types provided by Microsoft:

reboot patches

Require a reboot. The reboot operation is not necessarily performed immediately after the patch installation. In the activity plan, several activities which install reboot patches can be conditioned to a single reboot activity, so that a single reboot is performed after several patches are installed. You can configure how many activities requiring a reboot can be conditioned to a reboot activity using the **max_apm_bootable_threshold** option with the **wseccfg** command. For more information on this command, see “wseccfg” on page 68. Software packages containing reboot patches contain the “b” character in their name.

exclusive patches

Require an immediate reboot. Software packages containing exclusive patches contain the “x” character in their name.

standard patches

Patches which do not require a reboot.

Use the `filtering_template` file to distribute security patches to groups of endpoints as described in “Customizing the target filter template” on page 65.

Customizing the software package templates

Three different template files are provided to manage the different kinds of packages that can be created.

You can optionally create new templates to meet your environment requirements, although this operation is not recommended and should not be necessary. Do not modify the templates provided.

If you created new templates, you can use the **wseccfg -s** command and one of the following keys to specify which templates should be used:

- `reboot_template_file`
- `standard_template_file`
- `exclusive_template_file`
- `emergency_reboot_template_file`
- `emergency_standard_template_file`
- `emergency_exclusive_template_file`

For more information on the **wseccfg** command, see “wseccfg” on page 68.

Customizing the activity plan templates

You can customize the activity plan templates to adjust the default settings to your environment or to the targets to which you are distributing the plan. For example, if you are distributing the plan to server systems, you might want to modify the timeout settings to ensure that the plan is applied successfully.

You can modify the XML template manually or using the Activity Plan Editor. The XML file is made up of components called elements. Tags are used to describe elements. A start tag marks the beginning of an element and an end tag marks the end of the element. For more information on activity plans, refer to *IBM Tivoli Configuration Manager: User's Guide for Deployment Services*.

The following is the `APM_REBOOT_Activity_Template.xml` followed by the explanation of its most meaningful default tags:

```
<activity>
  <application>SoftwareDistribution</application>
  <name>__[ACT_NAME]__</name>
  <target_resource_type>endpoint</target_resource_type>
  __[TARGETS_QUERY]__
  <targets_computation>p</targets_computation>
  <operation>Install</operation>
  __[CONDITION]__
  .....
  <parameter>
    <name>SoftwarePackage</name>
    <value>__[SP_NAME]__</value>
  </parameter>
  .....
  <parameter>
    <name>DefaultTimeout</name>
    <value>120</value>
  </parameter>
  <parameter>
    <name>UserNote</name>
    <value>__[USER_NOTE]__</value>
  </parameter>
  .....
  <parameter>
    <name>RelativeDeadline</name>
```

Customizing the activity plan templates

```
        <value>8:0</value>
      </parameter>
      <parameter>
        <name>EnableNotification</name>
        <value>T</value>
      </parameter>
      <parameter>
        <name>DefaultAction</name>
        <value>accept</value>
      </parameter>
      .....
      <parameter>
        <name>AllowDefer</name>
        <value>T</value>
      </parameter>
      .....
    </activity>
```

__[ACT_NAME]__

This is a placeholder for the activity name. This data is retrieved from the software package the plan must install and is automatically added when the plan is generated. Do not modify this tag.

__[TARGETS_QUERY]__

This is a placeholder for the target query. This data is retrieved from the software package the plan must install and is automatically added when the plan is generated. Do not modify this tag.

__[CONDITION]__

This is a placeholder for the conditions set on the activities in the plan, if any. This data is retrieved from the software package the plan must install and is automatically added when the plan is generated. Do not modify this tag.

__[SP_NAME]__

This is a placeholder for the software package name. This data is retrieved from the software package the plan must install and is automatically added when the plan is generated. Do not modify this tag.

__[USER_NOTE]__

This is a placeholder for the note to be displayed to the user. This data is retrieved from the software package the plan must install and is automatically added when the plan is generated. Do not modify this tag.

DefaultTimeout

Specifies the interval of time the notification dialog is displayed. The default is 120 seconds. When the timeout period elapses, the default action is launched. Depending on whether you are distributing the plan on server systems or end-user workstations, you might want to specify different timeouts.

RelativeDeadline

Specifies the time after which the distribution expires.

EnableNotification

Specifies whether the end user is notified that a distribution is taking place on the workstation.

DefaultAction

Specifies which action is performed on the endpoint in the case that the user is not logged on the machine or is not physically present.

AllowDefer

Specifies whether the user can defer the distribution.

Customizing the target filter template

You can customize the target filter template to define groups of endpoints to which to distribute the plan.

Use this template to define groups of endpoints satisfying specific filtering conditions. You can combine the conditions by using the `<and>` or `<or>` logical operators. In particular you can combine a set of `<or>` operators using the `<and>` condition or viceversa you can combine a set of `<and>` operators using the `<or>` condition.

For each condition ensure you do not use different logical operators at the same hierarchical level. The following is an example of incorrect syntax:

```
<target_filtering_mode type="environ" >
  <table>INV_GRP_EP_VIEW</table>
  <or>
    <condition default="test1">group_label='${ENV1}'</condition>
  </or>
  <and>
    <condition default="test2">group_label='${ENV2}'</condition>
  </and>
</target_filtering_mode>
```

The syntax of the each condition tag is the following:

```
<condition default="APM_var_default_value">column_label='${APM_var}'</condition>
```

where:

column_label

Represents the name of the column of the table or view specified in the `<table>` tag, for which you define this filtering condition.

APM_var

Represents the name of a variable defined in the plan created using this filtering condition.

APM_var_default_value

Represents the default value assigned to the *APM_variable* defined in the plan created using this filtering condition.

The following is a customized `$BINDIR/../../TME/PATCH_MGMT/TEMPLATES/filtering_template.xml` file:

```
<!-- FILTERING TEMPLATE -->
<grouping>
  <target_filtering_mode type="mygroup">
    <table>INV_GRP_EP_VIEW</table>
    <or>
      <condition default="testgroup">group_label='${GRP1}'</condition>
    </or>
  </target_filtering_mode>

  <target_filtering_mode type="inventory">
    <table>INVENTORYDATA</table>
    <and>
      <condition default="Windows 2003">OS_TYPE='${OSTYPE}'</condition>
    </and>
    <and>
      <condition default="658483G">COMPUTER_MODEL='${MODEL}'</condition>
    </and>
  </target_filtering_mode>
</grouping>
```

Customizing the target filter template

In this example, two filter types are defined:

- The mygroup filter type is created on the basis of a query on the INV_GRP_EP_VIEW table, filled in using the **wsecgrp** command. For details see “wsecgrp” on page 83. This command creates static groups of endpoints.

By default, the plan generated using this filter addresses the group called “testgroup”. To address a different group, as for example the productionGroup created with the **wsecgrp** command, when you submit the plan you can use the following command:

```
wsubpln -r plan_name -VGRP1=productionGroup
```

- The inventory filter type is created on the basis of a query on the INVENTORYDATA table. In this case it groups by default all Windows 2003 systems, model 658483G. To address computers having an operating system different from Windows 2003 and a computer model different from 658483G, when you submit the plan you can use the following command:

```
wsubpln -r plan_name -VOSTYPE=Linux -VCOMPUTER_MODEL=1234
```

The following example shows how to combine a set of <or> and <and> operators to create a composite condition:

```
<target_filtering_mode type="computers" >
  <table>COMPUTER</table>
  <and>
    <or>
      <condition default="Windows Vista">OS_TYPE='${OS1}'</condition>
    </or>
    <or>
      <condition default="Tivoli">REGISTERED_OWNER='${OWN1}'</condition>
    </or>
  </and>
  <and>
    <or>
      <condition default="-[621942G]-">COMPUTER_MODEL='${MOD1}'</condition>
    </or>
    <or>
      <condition default="6579NAG">COMPUTER_MODEL='${MOD2}'</condition>
    </or>
  </and>
</target_filtering_mode>
```

The computers filter type selects all the COMPUTER_SYS_ID of the COMPUTER table that have OS_TYPE="Windows Vista" or REGISTERED_OWNER="Tivoli" and COMPUTER_MODEL="-[621942G]-" or COMPUTER_MODEL="6579NAG". As for the previous examples, to address a different set of computers, when submitting the related plan with the **wsubpln** command, use the **-V** option.

In addition to this kind of composite condition, you can also create a composite condition having nested <and> conditions inside <or> conditions.

To distribute the patches to the endpoints defined in one of these filters, specify the name of the filter type either in the **-s target_filtering_mode** of **wseccfg** or in the **-g target_filtering_mode** of **wsecgenplan** command line, depending on how you create the plan for distributing patches. For details see “Deploying patches to groups of endpoints” on page 57.

Chapter 6. Automated patch management command line

This chapter describes commands to configure automated patch management settings and retrieve automated patch management reporting information. You can use the **wseccfg**, **wsecgrp**, and **wsecrpt** commands for configuring and reporting purposes. The **wsecgensp**, **wtransfer**, and **wsecgenplan** commands are used by the Automation Server workflow and can be used for troubleshooting purposes.

The commands are described in alphabetical order.

wseccfg

Modifies or retrieves patch management settings for the managed node.

Syntax

wseccfg -s [*key* [=*value*]]

wseccfg -d *key*

wseccfg [-x *separator*] {-a *key value* | -c *key value*}

Description

You can use the **wseccfg** command to configure, retrieve, and change the patch management settings for the managed node. Some of the keys listed below are not displayed by default, but are available for configuration.

Options

-s *key=value*

Sets a custom key and its value, or allows you to define existing variables and their values. Specifying the **wseccfg -s** command without the *key* argument, displays all keys with the corresponding settings currently used. Specifying the **wseccfg -s** *key* command without a value, displays the value set for the specified key. Specifying the *key* argument with a value, sets the key to the specified value.

product_dir

Identifies the directory where Patch Management data, such as logs and traces, is stored. The default directory is `$(BINDIR)/../patch_mgmt`.

trace_size

Specifies the size of the trace file. The default value is 1 000 000 bytes. When the maximum size is exceeded, a new trace file is created.

trace_level

Specifies the trace level. Supported values are as follows:

0	none
1	fatal
2	error
3	warning
4	information
5	verbose

The default value is **0**.

source_host_list

Identifies a list of source hosts, separated by commas, used by Automation Server to send patch files. Specify only one source per region. At installation time, this option is completed with the name of the Tivoli server. This option is used only by the Automation Server.

TMR_server_list

Identifies a list of Tivoli servers, separated by commas, that Automation Server will use to send the .cab file and **ApprovedChanges.txt** files. At

installation time, this option is completed with the name of the Tivoli server. This option is used only by Automation Server.

delete_packages

Specifies whether software packages objects containing patches with revoked approval must be removed from Tivoli Framework. Supported values are **yes** and **no**. The default value is **no**.

delete_plans

Specifies whether existing activity plans are to be deleted before a new plan is created. Supported values are **yes** and **no**. The default value is **no**.

Note: If you set the **delete_plans** parameter to **yes**, then the Automation Server workflow deletes all activity plans, not only the plans generated using the automated patch management solution.

plan_creation_mode

Specifies whether only one plan or whether a plan for each Tivoli region in the Enterprise should be created. Specify **per_enterprise** to create one plan, or specify **per_tmr_region** to create a plan for each Tivoli region in the Enterprise. The default value is **per_enterprise**. This option applies to the creation of plans in the workflow.

plan_grouping_mode

Specifies how the activity plan must be created. Specify **by_patch_id** to create one plan per patch ID, or specify **none** to create one plan for all the available patch IDs. The default value is **none**. This option applies to the creation of plans in the workflow.

tme_object_scope_for_plan

Specifies whether resources such as software packages or Inventory scans used in activity plans must be located on the hub or spoke region. Specify **hub** to indicate that activities in the plan must refer to resources located on the hub region, specify **spoke** to indicate that activities in the plan must refer to resources located on the spoke regions for which the plan was created. If you specify **spoke**, a source host located in the spoke region must be specified with the **source_host_list** key. The default value is **hub**. This option applies to interconnected regions.

email_notification_address

Specifies the e-mail address used to send notifications about the workflow completion. This option is used only by Automation Server.

remove_patch_files_if_built

Specifies whether the patch executable files must be removed from the *\$BINDIR/./patch_repos* directory after creating the software package. Specify **yes** to remove the patch files, or specify **no** not to remove the patch files. The default value is **yes**. For more information on the **wsecgensp** command, see “wsecgensp” on page 80.

reboot_template_file

Specifies a relative path to the .spd template used to create the software package for patches requiring a reboot. The relative path must already exist. It is appended to the *\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES* default path as follows: *\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/reboot_template_file*. If you do not specify this key, the following default template is used:

\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/reboot_template.spd

standard_template_file

Specifies a relative path to the .spd template used to create the software package for standard patches. The relative path must already exist. It is appended to the $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES$ default path as follows: $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/standard_template_file$. If you do not specify this key, the following default template is used:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/standard_template.spd$

exclusive_template_file

Specifies a relative path to the .spd template used to create the software package for exclusive patches. The relative path must already exist. It is appended to the $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES$ default path as follows: $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/exclusive_template_file$. If you do not specify this key, the following default template is used:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/exclusive_template.spd$

emergency_reboot_template_file

Specifies a relative path to the .spd template used to create the software package for emergency patches requiring a reboot. The relative path must already exist. It is appended to the $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES$ default path as follows:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/emergency_reboot_template_file$

If you do not specify this key, the following default template is used:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/emergency_reboot_template.spd$

emergency_template_file

Specifies a relative path to the .spd template used to create the software package for emergency standard patches. The relative path must already exist. It is appended to the $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES$ default path as follows: $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/emergency_template_file$. If you do not specify this key, the following default template is used:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/emergency_template.spd$

emergency_exclusive_template_file

Specifies a relative path to the .spd template used to create the software package for exclusive emergency patches. The relative path must already exist. It is appended to the $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES$ default path as follows:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/emergency_exclusive_template_file$

If you do not specify this key, the following default template is used:

$\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/emergency_exclusive_template.spd$

emergency_delete_packages_plan

Specifies whether existing emergency packages and plans are to be deleted before a new iteration. Supported values are **yes** and **no**.

If you specify **no**, none of the previous packages and plans are deleted. If you specify **yes**, the previous emergency plans are deleted and only the packages of the last iteration are included in the plan. The default value is **no**.

emergency_patches

Specify this key to identify the list of patches to be installed immediately. Supported values have the following format: *patchInfo1,patchInfo2,...,patchInfoN* separated by a comma. Do not insert spaces between the items.

Before defining the *patchInfoN* values, you must identify the patch on WSUS and approve it. You can then collect the *updateID* and the platform on which to install the patch, and define the *patchInfoN* value as follows:

```
updateID.os_base_name[.os_architecture[.os_type[.os_subtype]]]
```

where:

<i>updateID</i>	Is the unique identifier for the patch (it is not the Qnumber) and is defined in WSUS.
<i>os_base_name</i>	Identifies the name of the operating system (winxp, win2k, win2k3, win2k8, vista).
<i>os_architecture</i>	Identifies the x86 architecture.
<i>os_type</i>	Identifies the type of the operating system: srv, wks.
<i>os_subtype</i>	Identifies the subtype of the operating system: dtc, ent, pro, bld, hom, ts, std.

The *updateID* and *os_base_name* parameters are mandatory. They are used to determine the targets to which the patch must be deployed. If you specify a non-required parameter, you must specify all the previous parameters in the *patchInfoN* key. This key is optional.

scp_timeout

Specifies the time, in seconds, allowed for copying the patch from the WSUS server to the Application Server. The default is 600 seconds.

provider_spb_dir

Specifies the directory on the source host where the .spb files are copied after the software package for the given patch has been created. If no value is specified, the .spb files are created in the $\$(BINDIR)/../patch_mgmt$ directory.

provider_patch_dir

Specifies the directory on the source host containing the patch executables used to build the software package. If this option is not specified, the $\$(BINDIR)/../patch_mgmt$ directory is assumed. Automation Server uses this path to store traces and spd. files.

provider_patch_host

Specifies the source host used to build the software package. If this option is not specified, the local workstation is assumed.

custom_plan_template

Specifies a relative path to the template file used to customize the activity plan. The relative path must already exist. It is appended to the $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES$ default path as follows: $\$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/custom_plan_template$. If you do not specify this key, the following default template is used:

```
 $\$(BINDIR)/TME/PATCH\_MGMT/TEMPLATES/APM\_PLAN\_Template.xml$ 
```

std_activity_template

Specifies a relative path to the template file used to customize an activity that installs a standard patch. The relative path must already

exist. It is appended to the `$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/` default path as follows: `$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/std_activity_template`. If you do not specify this key, the following default template is used:

```
$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/APM_STD_Activity_Template.xml
```

reboot_activity_template

Specifies a relative path to a template file used to customize an activity that performs a reboot. The relative path must already exist. It is appended to the `$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/` default path as follows: `$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/reboot_activity_template`. If you do not specify this key, the following default template is used:

```
$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/APM_REBOOT_Activity_Template.xml
```

excl_activity_template

Specifies a relative path to a template file used to customize an activity that installs an exclusive patch. The relative path must already exist. It is appended to the default path as follows: `$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/excl_activity_template`. If you do not specify this key, the following default template is used:

```
$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/APM_EXCL_Activity_Template.xml
```

last_activity_template

Specifies a relative path to a template file used to customize the last activity of the plan. The relative path must already exist. It is appended to the default path as follows: `$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/last_activity_template`. If you do not specify this key, the following default template is used:

```
$(BINDIR)/TME/PATCH_MGMT/TEMPLATES/APM_LAST_Activity_Template.xml
```

If you modify the name of the last activity in this template, you must use the same name in the `APM_PLAN_TEMPLATE.xml` or in the template you specify with the **std_activity_template** key.

max_apm_bootable_threshold

Specifies how many activities requiring a reboot are to be grouped together. For each group of activities requiring a reboot, a specific reboot activity is generated and conditioned to all the above activities. The default value is **10**.

wsus_inst_path

Specifies the WSUS Update Source directory (for example `c:\WSUS`). You can use directory names containing spaces only if you specify the abbreviated form that appears if you enter `dir /x`. For example, `C:\program files` must be specified using the abbreviated form containing the tilde character as follows: `C:\progra~1`.

This key is mandatory. For details, see *Deploying Microsoft Windows Server Update Services*.

wsus_db_name

Specifies the name of the database used by WSUS to store data. This key is not mandatory and its default value is `SUSDB`.

wsus_db_host

Specifies the hostname of the SQL Server used by WSUS to store data. You can retrieve this name from the SQL Server Service Manager (`sqlmangr.exe`) on the WSUS server. If you specify a remote database for WSUS, you must modify the `WSUS_info_retriever.sh` script, as

described in “Osql failure with a remote WSUS database” on page 103.
 If you are using WSUS , version 3.0, set this key to
 \\.\pipe\MSSQL\$MICROSOFT##SSEE\sql\query

. This key is mandatory.

wsus_version

Specifies the version of WSUS being used. Supported values are 2 and 3. This key is not mandatory.

cab_gateways_list

Specifies the list of gateways to which the .cab and ApprovedChanges.txt files must be uploaded if changed since the last download from the Microsoft site. These files are downloaded by the Automation Server workflow to the lcf_bundle directory on the specified gateways.

The keyword all_gw can be used to indicate all the gateways in the network. If you do not specify this key, the .cab and ApprovedChanges.txt files are not copied.

cab_gateways_file

Specifies the full path to a file that contains the list of the gateways to which the file must be uploaded (see cab_gateways_list option). This file must reside on the Automation Server.

prepare_patches_requiring_connectivity

Some Microsoft patches require connectivity to Internet to be installed. Specifies if these patches are to be prepared. You can set the key to yes or no. The parameter is optional and its default value is no.

prepare_patches_requiring_user_input

Some Microsoft patches require user input (for example acceptance of EULA license) when installed. These patches are prepared only if the value of the prepare_patches_requiring_user_input is set to yes. You can set the key to yes or no. This key is optional and the default value is no.

catalog_proxy_enabled

Enables or disables proxy support to download the .cab file. You can use an HTTP proxy to access the Microsoft Web site, or your local HTTP server where the .cab file has been downloaded. Proxy parameters are defined at installation time in the tpm_update.req file. This key is optional and the default value is no.

workflow_activities

Specifies whether workflows are completed in one step or are separated into two steps. Supported values are as follows:

- | | |
|--------------------|--|
| sync | Performs the following operations: <ol style="list-style-type: none"> 1. Synchronizes WSUS approved patches with the Automation Server database. 2. Downloads wsuscan.cab from the Microsoft Web site. 3. Creates ApprovedChanges.txt. 4. Copies wsuscan.cab and ApprovedChanges.txt on the workstations defined in the TMR_server_list, cab_gateways_list, and cab_gateways_file. |
| preparation | Creates software packages, queries, and APM plans. |

all Performs all sync and preparation operations. This is the default value.

This key is optional and the default value is **all**.

deployment_paradigm

Specifies how to retrieve software packages. Supported values are as follows:

standard

Specifies that the software package to be installed resides on the source host. This is the default value.

from_depot

Specifies that the software package to be installed resides on the repeater depot, rather than on the source host. With the **from_depot** key you must specify one of the following keys in the **wseccfg** command:

-s sp_patches_depots=depot1, depot2,...

Use this key to specify in *depot1, depot2,...* the name of the gateways or stand-alone repeaters where to load the packages. These names must be separated by a comma.

-s sp_patches_depots_file=depotfile

Use this key to specify in the *depotfile* file the list of gateways or stand-alone repeaters where to load the packages.

You can also specify the following key:

-s depots_unload=[yes(default) | no]

Use this key to specify if the unload operation must be performed or not.

from_fileserver

Specifies that the images referenced in the software package are to be retrieved from a file server. File servers must be configured if this value is used. For more details, see “Deployment Paradigms” on page 51.

This key is optional.

skip_plans_creation

Avoids the creation of the activity plan at the end of the patch management workflow. Supported values are **yes** and **no**. The default value is **no**.

submit_plans

Submits the activity plan, which is generated at the end of the patch management workflow, automatically. Supported values are **yes** and **no**. The default value is **no**. If the **skip_plans_creation** key is set to yes, the **submit_plans** key is ignored.

If you set **deployment_paradigm** to **from_fileserver**, the plan is not automatically submitted because you would need to perform some additional manual operations before submitting the plan, as described in “Deployment Paradigms” on page 51.

target_filtering_mode

Specifies one of the filter types you defined in the

target_filtering_mode type key of the filtering_template.xml file. The filter type filters the endpoints on which the activity plan to be created must install the patches. This key is used only by the Automation Server and is optional.

-d key Deletes the specified key.

-a key value

Appends the specified value to the given key. If the **-x** option is not specified, the comma is used as separator.

-c key value

Removes the specified value from the given key. If the **-x** option is not specified, the comma is used as separator. The specified value is removed only if you specify the correct separator. Before running this command, run **wseccfg -s** to find out the exact separator used for the key.

-x separator

Specifies the separator symbol used to process the arguments in the string when the option **-a** or **-c** is specified. If this option is not specified, the default separator is a comma (,).

Note: If you are using a semicolon as separator on UNIX systems, include the semicolon between single inverted quotes.

Authorization

user For viewing configuration information

senior For modifying configuration information

Return Values

The **wseccfg** command returns one of the following:

0 The **wseccfg** command completed successfully.

other than 0 The **wseccfg** command failed due to an error.

Examples

1. To set the trace level, enter the following command:
wseccfg -s trace_level=5
2. To set the trace size, enter the following command:
wseccfg -s trace_size=1000000
3. To set the directory on the source host where the software package blocks are stored, enter the following command:
wseccfg -s provider_spb_dir=/Tivoli/bin/patch_mgmt/spb
4. To set the directory on the source host containing the patch executables used to build the software package blocks, enter the following command:
wseccfg -s provider_patch_dir=/Tivoli/Patches
5. To set the source host where the software package blocks are built, enter the following command:
wseccfg -s provider_patch_host=lab133109
6. To specify the source host list used by Automation Server to send patch files, enter the following command:
wseccfg -s provider_patch_host=lab133109,lab13486,lab13879

7. To specify the Tivoli server list for Automation Server to use to send the .cab and ApprovedChanges.txt files, enter the following command:

```
wseccfg -s TMR_server_list=linux111,lab13875,lab13542
```

8. To add a key and its value to the Patch Management configuration, enter the following command:

```
wseccfg -x , -a test 80
```

9. To verify that the specified key was correctly added, enter the following command:

```
wseccfg -s
```

10. To remove the specified value, enter the following command:

```
wseccfg -x , -c test 80
```

This command removes the value, while the test key is still present. To remove the test key, enter the following command:

```
wseccfg -d test
```

11. To remove values from the colors key, enter the following commands:

- a. To discover the separators used in the colors key, enter the following command:

```
wseccfg -s
```

The following is an abstract of the output returned by the command:

```
colors=green,red;blue;black,white,purple
```

- b. To remove the white value from the colors key, enter the following command:

```
wseccfg -x , -c colors white
```

- c. To remove the red;blue;black values from the colors key, enter the following command:

```
wseccfg -x , -c colors red;blue;black
```

See Also

“wsecgensp” on page 80, “wsecgenplan” on page 77.

wsecgenplan

The **wsecgenplan** command creates the activity plan to be submitted for installing one or more patches. If you set **submit_plans** to **yes** using the **wseccfg** command, the plan is also submitted.

Syntax

```
wsecgenplan {-p software_package_list | -P packages_file} {-e
emergency_software_package_list | -E emergency_software_package_file} [-f
subscriber_file | -g target_filtering_mode] [-t] [-n name] [-d deployment_paradigm] [-D
depot_file] [-u]]
```

Description

The activity plans are generated based on templates stored in the `$BINDIR/TME/PATCH_MGMT/TEMPLATES` directory. To specify a different template, use the following options with the **wseccfg** command:

- **custom_plan_template**
- **std_activity_template**
- **reboot_activity_template**
- **excl_activity_template**
- **last_activity_template**
- **emergency_template_file**
- **emergency_reboot_template_file**
- **emergency_exclusive_template_file**

For more information on this command, see “wseccfg” on page 68.

Options

-p *software_package_list*

Specifies a list of patch IDs, separated by commas, that the activity plan to be created must install.

-P *packages_file*

Specifies a file containing the list of patches that the activity plan to be created must install. Specify one package per line. Blank or empty lines are skipped. Lines starting with # are considered as comments and are ignored.

-e *emergency_software_package_list*

Specifies a list of emergency patches, separated by commas, that the activity plan to be created must install.

-E *emergency_software_package_file*

Specifies a file containing the list of emergency patches that the activity plan to be created must install. Specify one package per line. Blank or empty lines are skipped. Lines starting with # are considered as comments and are ignored.

-f *subscriber_file*

Specifies the fully-qualified path to the file containing the target list for each activity in the plan. In the file, specify one subscriber per line. If this option is not specified, the query associated to the patch package is used to evaluate the targets for each activity in the plan.

-g target_filtering_mode

Specifies one of the filter types you defined in the `target_filtering_mode` type key of the `filtering_template.xml` file. The filter type filters the endpoints on which the activity plan to be created, must install the patches. For details see “Customizing the target filter template” on page 65.

-t

Specifies whether a single plan must be created for each Tivoli region. The targets of each plan are filtered by the region number. Therefore, each plan addresses only endpoints belonging to the same region. If this option is not specified, only one plan is generated to address all the endpoints in the enterprise.

-n name

Specifies the name to be assigned to the plan to be created. If this option is not specified, a default name is automatically generated. The default naming convention is as follows: `patch.mm-dd-yy_hh_mm_ss_patchID#regionID`, where

mm-dd-yy The date the plan was created.

hh_mm_ss The time the plan was created.

patchID The ID of the patch contained in the plan. This item might not be present depending on the plan configuration.

regionID The ID of the region for which the plan is created. This item might not be present depending on the plan configuration.

-d deployment_paradigm

Specifies how to retrieve software packages. If you do not specify this key, the software package is retrieved from the source host. Supported values are as follows:

standard Specifies that the software package to be installed resides on the source host. This is the default value.

from_depot Specifies that the software package to be installed resides on the repeater depot, rather than on the source host.

from_fileserver Specifies that the images referenced in the software package are to be retrieved from a file server.

-D depot_file

Specifies the file containing the list of gateways or stand-alone repeaters on which to load the packages. This option is valid only with the `-d from_depot` parameter. This file must reside on the workstation where the **wsecgenplan** command is run.

-u

Specifies if the unload operation must be performed or not. This option is valid only with the `-D depot_file` parameter. The following APM templates manage the activities of WSUSSCANCAB^1.0 installation and of the load and unload of depots:

- `APM_CAB_Activity_Template.xml`
- `APM_DEPOT_Activity_Template.xml`

Authorization

user For viewing configuration information

APM_Edit

For editing configuration information

Return Values

The **wsecgenplan** command returns one of the following:

- 0** The **wsecgenplan** command completed successfully.
- other than 0** The **wsecgenplan** command failed due to an error.

Examples

1. To generate an activity plan for packages

```
patch.842526.DF8B7CBC-80DB-427D-BACD-42F7F8DD2C0A.b^1.0,
patch.888113.3754C547-772F-4C0F-9A17-1A7754CB4A6C.b^1.0,
patch.891781.3128B719-219E-4273-A706-F417D8872C39.b^1.0
```

using the `tgt_list.txt` file to specify subscribers, enter the following command:

```
wsecgenplan -p patch.842526.DF8B7CBC-80DB-427D-BACD-42F7F8DD2C0A.b^1.0,
patch.888113.3754C547-772F-4C0F-9A17-1A7754CB4A6C.b^1.0,
patch.891781.3128B719-219E-4273-A706-F417D8872C39.b^1.0
-f /Tivoli/act_plans/tgt_list.txt
```

2. Creates a plan by including all the software packages specified in the `package.txt` file.

```
wsecgenplan -P packages.txt
```

3. Creates a plan for each Tivoli region with source hosts specified with the **source_host_list** key in the **wseccfg** command that includes the specified package.

```
wsecgenplan -p patch.891781.3128B719-219E-4273-A706-F417D8872C39.b^1.0 -t
```

4. Specifies one of the filter types you defined in the `filtering_template.xml` file.

```
wsecgenplan -p patch.842526.DF8B7CBC-80DB-427D-BACD-42F7F8DD2C0A.b^1.0
-g inventory
```

where `inventory` is the filter type defined in the `filtering_template.xml` file.

See Also

“wseccfg” on page 68.

wsecgensp

The **wsecgensp** command generates the software packages required for installing the specified patch or patches on target workstations and creates the queries for determining the target workstations on which the patches need to be installed.

Syntax

```
wsecgensp -p patch_id -g GUID [-t patchInfo] [-q { 0 | 1 }] [{ -b | -x }] [-c file_type] -f patch_file -a args [-h source_host -d patch_dir] [-v] [-o] [-H 9]
```

Description

The **wsecgensp** command generates the software package and the associated query for the specified patch ID. Packages are prepared using templates stored in the *\$BINDIR/TME/PATCH_MGMT/TEMPLATES* directory. To modify the default template, use the following options with the **wseccfg** command:

- **reboot_template_file**
- **standard_template_file**
- **exclusive_template_file**
- **emergency_template_file**
- **emergency_reboot_template_file**
- **emergency_exclusive_template_file**

Options

-p *patch_id*

Specifies the Q number of the patch.

-g *GUID*

Specifies the globally unique identifier.

-t *patchInfo*

Specifies that the patch is an emergency patch. The *patchInfo* syntax is the following:

```
os_base_name[.os_architecture[.os_type[.os_subtype]]]
```

where:

os_base_name Identifies the name of the operating system (winxp, win2k, win2k3, win2k8, vista).

os_architecture Identifies the x86 architecture.

os_type Identifies the type of the operating system: srv, wks.

os_subtype Identifies the subtype of the operating system: dtc, ent, pro, bld, hom, ts, std.

The *os_base_name* parameter is mandatory. It is used to determine the targets to which the patch has to be deployed. If other non-required parameters (such as *os_architecture*, *os_type*, *os_subtype*) are not specified, a very complex APM plan might be created that addresses a high number of endpoints causing network overload problems.

-q **0** | **1**

Specifies whether the query should be created. Specify **1** to create the query, specify **0** to skip the creation of the query. The default value is **1**.

- b** Indicates that the patch requires a reboot. Uses the `reboot_template_file` template.
- x** Indicates that the patch is exclusive. Uses the `exclusive_template_file` template.
- c *file_type***
Specifies the type of file that is included in the package, either EXE or CAB. The default is EXE.
- f *patch_file***
Specify the name of the file that installs the patch. This file must be located in the directory specified by the **provider_patch_dir** option.
- a *args*** Specifies the arguments for the patch installation. The arguments vary depending on the patch to be installed. For more information on supported arguments for a patch, search the patch ID on the Microsoft WSUS server.
- h *source_host***
Specifies the source host node where the patch files are stored. If this option is not specified, the value defined in the **provider_patch_host** option in the **wseccfg** command is assumed. For more information on this command, see “wseccfg” on page 68.
- d *directory***
Specifies the directory on the source host where the patch executable files are stored. If this option is not specified, the value defined in the **provider_patch_dir** option in the **wseccfg** command is assumed. For more information on this command, see “wseccfg” on page 68.
- v** Displays in preview the software package name without generating the software package. The software package name contains information about the type of patch to be installed and must not be modified.
- o** Overwrites any existing packages with the same name, if present.
- H 9** Specifies whether software packages are generated using the installer for Windows 2008 and Windows Vista. If this option is not specified or if you enter a value different from **9**, the packages are created using the default installer.

Authorization

senior For modifying configuration information

Return Values

The **wsecgensp** command returns one of the following:

- 0** The **wsecgensp** command completed successfully.
- other than 0** The **wsecgensp** command failed due to an error.

Examples

To create a software package for the patch 873339 to be applied on English Windows 2000 workstations with the specified installation arguments, enter the following command:

```
wsecgensp -p 873339 -g 47159B36-A90E-4253-B1F7-1629DA5D0328 -b
-f 500E4656B4F0CA3431565631989090BBEEB74BCC.exe -a "-q /Z -ER"
```

The patch.873339.47159B36-A90E-4253-B1F7-1629DA5D0328.b software package is created.

wsecgensp

See Also

“wseccfg” on page 68.

wsecgrp

The **wsecgrp** command generates groups of targets on which to install a specified patch.

Syntax

```
wsecgrp -l [-t | {-n group_label }]
```

```
wsecgrp -g -n group_label {-c [-D descr] | -d}
```

```
wsecgrp -m -n group_label -D descr
```

```
wsecgrp -e -n group_label {-s endpoint_sys_id | -S endpoint_sys_id_file} {-c | -d}
```

Description

The **wsecgrp** command generates groups of targets on which to install a specified patch.

Options

- l** Lists all the groups.
- t** Lists all the computers belonging to each group.
- n group_name**
Lists the system ID of the endpoints belonging to the specified group.
- g** Creates or deletes a group.
 - n group_name**
Specifies the name of the group to be created or deleted.
 - c** Creates the specified group. You can also add a description of the new group by specifying the following parameter:
 - D descr**
Gives a description of the group.
 - d** Deletes the specified group.
- m** Modifies the existing description of a group by specifying the new description using the following parameter:
 - D descr**
Gives a description of the group.
- e** Adds or deletes a specified computer system ID to a specified group.
 - n group_name**
Specifies the name of the group.
 - s endpoint_sys_id**
Specifies the system ID of the endpoint.
 - S endpoint_sys_id_file**
File containing the computer system IDs, one ID per line.
 - c** Adds computer system IDs to a group.
 - d** Deletes computer system IDs from a group.

wsecgrp

Authorization

senior For creating or deleting groups.

Return Values

The **wsecgrp** command returns one of the following:

- 0** The **wsecgrp** command completed successfully.
- other than 0** The **wsecgrp** command failed due to an error.

Examples

To create a group, enter the following command:

```
wsecgrp -n Windows_Endpoints -c
```

The Windows_Endpoints group is created.

See Also

“wsecgenplan” on page 77.

wsecrprt

Retrieves patch management information from the IBM Tivoli Configuration Manager database.

Syntax

```
wsecrprt [ -s M] [-s F] [-k C] [-k N] [-c IC] [-c IP] [-c IBC] [-c E] [-c R]
[ -g { e | p } ] [ -p patch_id ]... [ -P patch_id_file ]... [ -e ep_label ]... [ -E ep_label_file ]...
[ -t begin_date ] [ -T end_date ]
[ -v1 | -v2 ] [ -f {flat | header | csv | html}]
[ -n ] [ -o file_name ]
```

Description

The **wsecrprt** command returns information about patches. You can perform advanced queries by combining the available options.

Options

[-s M] [-s F]

Returns information about patches discovered in the specified state by the latest patch scan

M Returns information about patches discovered in missing state by the latest patch scan.

F Returns information about patches discovered in found state by the latest patch scan.

[-k C] [-k N]

Returns information about software packages in relation to patches.

C Returns information about all patches with an associated software package. Software packages in **created** state are listed in the SD_PACKAGES table in the Inventory database.

N Returns information about all patches without an associated software package. Software packages in **not created** state are not listed in the SD_PACKAGES table in the Inventory database.

[-c IC] [-c IP] [-c IBC] [-c E] [-c R]

Returns information about software packages used to install a patch. Software Distribution states are considered.

IC Returns information about software packages in Installed Committed state. The state of the package in the SD_INST table is **IC*--**.

IP Returns information about software packages in **prepared** state. The state of the package in the SD_INST table is **IP*--**.

IBC Returns information about software packages in **need reboot** state. The state of the package in the SD_INST table is **IC-BC**.

R Returns information about patches that have not been packaged or whose software package is not yet installed.

- E** Returns information about software packages in **error** state. The state of the package in the SD_INST table is **I***E**.
- g {e | p}**
Lists results by endpoint or by patch ID.
- e ep_label**
Displays results for the specified endpoint. If you specify an endpoint, the results are returned for the specified endpoint.
- E ep_label_file**
Displays results for the endpoints specified in the file. Specify the full path to a file containing a list of endpoint labels, separated by commas.
- p patch_id**
Displays results for the specified patch IDs. If you specify a patch ID, the results are returned for the specified patch ID.
- P patch_id_file**
Displays results for the patch IDs specified in the file. Specify the full path to a file containing a list of patch IDs, separated by commas.
- t begin_date**
Specifies the time after which to perform the search on history tables for the specified patch. Ensure that the time on the endpoint is synchronized with the time on the Tivoli server, because the date and time set on the endpoint are used when returning information. For this option to work correctly, allow a large amount of disk space for the Inventory **inv_temp_ts** temporary tablespace.
- T end_date**
Specifies the time before which to perform the search on history tables for the specified patch. Ensure that the time on the endpoint is synchronized with the time on the Tivoli server, because the date and time set on the endpoint are used when returning information. For this option to work correctly, allow a large amount of disk space for the Inventory **inv_temp_ts** temporary tablespace.
- v1** Returns additional information for the specified query. The information returned varies depending on the query.
- v2** Returns additional information for the specified query. The information returned varies depending on the query. The information returned by the **-v1** option is included.
- n** Omits the table headers and column names from the output.
- f {flat | header | csv | html}**
Specifies the format options for the results. If you do not specify the **-o file_name** option, the results are returned to standard output.
- flat** The output is returned in flat format. This is the default value. The following is an example output.

QNUMBER	BULLETIN	PRODUCT	PRODUCT LANGUAGE	PRODUCT CODE	OS BASE NAME
---------	----------	---------	---------------------	-----------------	--------------

885492	MS05-009	Security Update for Windows Media Player 9 Series (KB885492)	ENUS	OS	win2k
896358	MS05-026	Security Update for Windows 2000 (KB896358)	ENUS	OS	win2k
896422	MS05-027	Security Update for Windows 2000 (KB896422)	ENUS	OS	win2k

header

The output is returned in header format. The following is an example output.

** lab133011-w2ks

QNUMBER	PACKAGE NAME	REGION ID	BULLETIN	PRODUCT LANGUAGE	PRODUCT CODE
835732	patch.835732.C DE4D5D0-A9 83-4A98-B B9 B-628ADFC96 F09.b	1489730723	MS04-011	ENUS	OS

** lab133061-w2k

896422	patch.896422.23 A6CA60-1AB4- 4AF3-9D66-C97 54FBF989B.b	1489730723	MS05-027	ENUS	OS
--------	---	------------	----------	------	----

csv

The output is returned in csv format.

QNUMBER,	BULLETIN,	PRODUCT,	PRODUCT LANGUAGE,	PRODUCT CODE,	OS BASE NAME
835732,	MS04-011,	Security Update for Windows XP (KB835732),	ENUS,	OS,	winxp
896358,	MS05-026,	Security Update for Windows 2000 (KB896358),	ENUS,	OS,	win2k
896422,	MS05-027,	Security Update for Windows 2000 (KB896422),	ENUS,	OS,	win2k

html

The output is returned in html format.

-o filepath

Specifies the path to a file where the results are to be stored.

Authorization

user For viewing configuration information

RIM_view

For viewing RIM information

Return Values

The **wsecrprt** command returns one of the following:

- 0** The **wsecrprt** command completed successfully.
- other than 0** The **wsecrprt** command failed due to an error.

Examples

1. To list the endpoints on which patches are missing, enter the following command:
`wsecrprt -s M -g e`
2. To list the endpoints on which the specified patches are missing, enter the following command:
`wsecrprt -s M -g p -p 279328 -p 279328`
3. To discover available patches with an associated software package which have not been distributed yet, enter the following command:
`wsecrprt -k C -c R`
4. To discover the patched endpoints grouping the results by endpoint, enter the following command:
`wsecrprt -s F -g e`
5. To discover the patched endpoints grouping the results by patch, enter the following command:
`wsecrprt -s F -g p`
6. To list all patches for which a distribution failed, enter the following command:
`wsecrprt -c E -g p`
7. To list all endpoints for which a reboot is required, enter the following command:
`wsecrprt -c IBC -g e`
8. To list all patches missing because of an error, enter the following command:
`wsecrprt -s M -c E -g p`
9. To list all patches missing on the specified endpoint, enter the following command:
`wsecrprt -s M -e ept1`
10. To list all endpoints on which the specified patch is present, enter the following command:
`wsecrprt -s F -p 279328 -g e`
11. To list all endpoints on which the specified patch is missing, enter the following command:
`wsecrprt -s M -p 279328 -g e`
12. To list all patches that were installed without using Software Distribution, do not have an associated software package, and were discovered by the scan, ordering the results by patch ID, enter the following command:
`wsecrprt -s F -k N -g p`
13. To list all patches that were discovered by the scan and have an associated software package, ordering the results by patch ID, enter the following command:
`wsecrprt -s F -k C -g p`
14. To list the patches installed in a specified time interval, enter the following command:
`wsecrprt -c IC -t 03/10/2005 -T 04/10/2005`
15. To list patches created in a specified time interval, enter the following command:


```
wsecrprt -k C -t 03/10/2005 -T 04/10/2005
```

16. To verify whether the specified patch is missing in your environment, enter the following command:

```
wsecrprt -s M -p 822679
```

17. To list all patches missing on the specified endpoint, enter the following command:

```
wsecrprt -s M -e lab14586
```

See Also

None.

wtransfer

Copies files and directories from one managed node to another.

Syntax

```
wtransfer [-h source_host] -s source_file {-t target_label... | -t all_gw | -t all_mn | -T targets_list_file} [-d destination_dir] [-f]
```

Description

The **wtransfer** command copies files and directories from one managed node to another. It can be used only on the directories managed by Tivoli.

Options

- h *source_host*
The name of the managed node where the file or directories to be copied are stored.
- s *source_file*
The absolute path of the files or directories on the source host. You can use wild cards and the Tivoli variables \$(DBDIR), \$(BINDIR), \$(TEMP), and \$(TMP). The Tivoli variables are solved on the source host machine.
- t *target_label*
The name of the target managed node or gateway.
- t **all_gw** | -t **all_mn**
Specifies either all the gateways (**all_gw**) of the Tivoli management region as the targets of the operation, or all the managed nodes (**all_mn**) of the Tivoli management region as the targets of the operation.
- T *targets_list_file*
The absolute name of the file containing the list of target managed node or gateway names.
- d *destination_dir*
Is the path relative to the \$(BINDIR)/../ directory and identifies the target destination path. If the specified destination directory does not exist, it is not created unless you specify the -f option to force the creation of the destination directory. You can use the \$(DBDIR), \$(BINDIR), \$(TMP), and \$(TEMP) Framework variables to specify this option. If you use these variables, the path is considered absolute. On UNIX, enclose this path in quotes.
- f
Forces the creation of the destination directory specified by the -d *destination_dir* option if the directory does not exist.

Authorization

senior

Return Values

The **wtransfer** command returns one of the following:

- 0** The **wtransfer** command completed successfully.
- other than 0** The **wtransfer** command failed due to an error.

Examples

1. To move the to_transfer.txt file to all gateways in the Tivoli Management region forcing the destination path to be created if it does not exist, enter the following command:

```
wtransfer -h lab145864 -s /test/to_transfer.txt -t all_gw -d /statistics -f
```

2. To move all files and directories matching the c:\te*p pattern to the \$BINDIR/../statistics directory, enter the following command:

```
wtransfer -h lab145864 -s c:\te*p -t all_mn -d /statistics -f
```

See Also

None.

wtransfer

Chapter 7. Troubleshooting

This chapter describes how to manage logs and traces in the automated patch management solution. In particular, it specifies the location of log and trace files and how to enable a particular trace level.

It also describes topics to aid you in determining the cause of problems you encounter while implementing the automated patch management solution in your environment.

Automation Server logs

Log locations

The location of the message logs for Tivoli Configuration Manager Automation Server follows the Tivoli Common Directory standard. The log locations are shown in the following table.

Table 18. Tivoli Configuration Manager Automation Server logs

Directory name	Description	Path
deploymentengine	Logs Automation Server workflow information	<i>\$TIO_LOGS</i> /deploymentengine
install	Logs information on the installation of Automation Server prerequisites	<i>\$TIO_LOGS</i> /install
uninstall	Logs information on the uninstallation of Automation Server prerequisites	<i>\$TIO_LOGS</i> /uninstall
tpm_install	Logs information on Automation Server silent installation	<i>BASE_DIR</i> / <i>TPM_SRC</i> /tpm_install/
logs	Logs information on start, stop, and reinitialize events for the Automation Server	<i>\$TIO_HOME</i> /logs

where:

\$TIO_LOGS Represents the path Program Files/ibm/tivoli/common/COP/logs

BASE_DIR Represents the path where the subdirectories for each of the installation CDs are created.

TPM_SRC Represents the path where the contents of the Tivoli Configuration Manager Automation Server CD are copied. Note that it is a relative path to the *BASE_DIR* directory.

\$TIO_HOME The directory where Tivoli Configuration Manager Automation Server is installed.

Automation Server workflow logging levels

Log data in Tivoli Configuration Manager Automation Server is managed by **log4j**, an established open source logging tool. This section details the default log4j settings, the customized Tivoli Configuration Manager Automation Server settings, and how you can modify settings dynamically. For complete log4j documentation, go to <http://logging.apache.org/log4j/docs/documentation.html>.

Data is logged in the **msg.log**, **trace.log**, and **console.log** files located in the \$TIO_LOGS/deploymentengine directory. To modify logging levels and configuration parameters, set the related values in the **log4j.prop** file, located in \$TIO_HOME\config directory.

The following is an example of the log4j.prop file:

```
# output directory. Can be overwritten with -Dkanaha.logs=<directory>
#
kanaha.logs=logs

# message formats
# normal used to write to console.log
# error used to format error messages (prints location of a problem)
# module is meant for messages written module specific files
#
output.normal=%d{ISO8601} %-5p [%t] (%13F:%L) %c{2}: %m%n
output.error=%d{ISO8601} %-5p [%t] (%13F:%L): %m%n
output.module=%d{ISO8601} %-5p [%t] (%13F:%L): %m%n

#
# configure root category
# note that this configuration is inherited by all other categories (see
# example below if you want to suppress this behaviour)
#
log4j.rootCategory=DEBUG, consolefile, errorfile, messagefile

# everything goes to console.log
# rolling by log size. For other rolling options, see see http://logging.
# apache.org/log4j/docs/api/index.html
#
log4j.appender.consolefile=org.apache.log4j.RollingFileAppender
log4j.appender.consolefile.MaxFileSize=100MB
log4j.appender.consolefile.MaxBackupIndex=10
log4j.appender.consolefile.File=${kanaha.logs}/console.log
log4j.appender.consolefile.layout=org.apache.log4j.PatternLayout
log4j.appender.consolefile.layout.ConversionPattern=${output.normal}
log4j.appender.consolefile.threshold=info
log4j.appender.consolefile.append=true

# errors to trace log file, for FFDC
# rolling by log size
#
log4j.appender.errorfile=org.apache.log4j.RollingFileAppender
log4j.appender.errorfile.MaxFileSize=10MB
log4j.appender.errorfile.MaxBackupIndex=10
log4j.appender.errorfile.File=${kanaha.logs}/trace.log
log4j.appender.errorfile.layout=org.apache.log4j.PatternLayout
log4j.appender.errorfile.layout.ConversionPattern=${output.error}
log4j.appender.errorfile.threshold=error
log4j.appender.errorfile.append=true

# globalized message log to msg.log (user log)
# rolling by log size
#
log4j.appender.messagefile=org.apache.log4j.RollingFileAppender
log4j.appender.messagefile.MaxFileSize=10MB
log4j.appender.messagefile.MaxBackupIndex=10
```

```
log4j.appender.messagefile.File=${kanaha.logs}/msg.log
log4j.appender.messagefile.layout=org.apache.log4j.PatternLayout
log4j.appender.messagefile.layout.ConversionPattern=${output.normal}
log4j.appender.messagefile.threshold=MSG_INFO#com.thinkdynamics.
    kanaha.util.logging.MessageLevel
log4j.appender.messagefile.append=true
```

The three sections in the file refer to the three available log files, as described below:

log4j.appender.consolefile

Sets the maximum file size and maximum number for the **console.log** file.

log4j.appender.errorfile

Sets the maximum file size and maximum number for the **trace.log** file.

log4j.appender.messagefile

Sets the maximum file size and maximum number for the **msg.log** file.

Similar keys are used to define the same settings for each component. You can use the log4j.prop file to define the following keys:

MaxFileSize Specifies the maximum file size.

MaxBackupIndex

Specifies the maximum file number.

threshold Specifies the level of detail. Supported values are as follows:

- error
- warn
- info
- debug. This is the default value.

append Specifies whether new information is to be appended to the log file. Supported values are **true** or **false**.

If the three sections listed above are missing, the setting defined in the log4j.rootCategory key is applied to the three log files.

To modify the log4j.prop file, perform the following steps:

1. Open log4j.prop in a text editor.
2. Modify settings as required.
3. Save the file.

The Tivoli Configuration Manager Automation Server automatically reloads the log4j configuration and applies the new settings 60 seconds after you save the log4j.prop file.

Tivoli Configuration Manager Automation Server silent installer log

After you complete a silent installation, refer to the log file to determine if the silent installation was successful. The tpm_install.log file is located in the following directory:

```
BASE_DIR/TPM_SRC/tpm_install/
```

Tivoli Configuration Manager Automation Server start and stop logs

When you start Tivoli Configuration Manager Automation Server, the application creates a log file, `tpm_start.log`. When you stop Tivoli Configuration Manager Automation Server, the application creates a log file, `tpm_stop.log`. Both files are located in the following directory:

`$TIO_HOME\logs`

Tivoli Configuration Manager Automation Server prerequisites uninstall logs

When you uninstall Tivoli Configuration Manager Automation Server prerequisites, the log files for the uninstall process follow the Tivoli Common Directory standard, and are located in the following directory:

`$TIO_LOGS\uninstall\`

Patch management component logs and traces

The following sections describe the logs and traces related to the Patch Management component.

Patch Management environment installation logs and traces

The log files for the Patch Management environment are named **`inv_install.log`** and **`patchmgmt_install.log`** and are located in the following directory: `$DBDIR/tmp/`.

The trace files for the Patch Management environment are named **`inv_install.trc`** and **`patchmgmt_install.trc`** and are located in the following directory: `$DBDIR/tmp/`.

Patch Management command line traces

The trace files store information on the commands performed by the Patch Management command line. By default, the tracing function is not enabled. To enable the function and define the trace file size and number, use the **`wseccfg`** command.

Traces are saved in the `$(BINDIR)/../patch_mgmt` default directory. To modify the default directory, use the **`wseccfg`** command. For more information on this command, see “`wseccfg`” on page 68.

Patch Management environment uninstall logs and traces

The log files for the Patch Management environment are named **`inv_uninst.log`** and **`patchmgmt_uninst.log`** and are located in the following directory: `$DBDIR/tmp/`.

The trace files for the Patch Management environment are named **`inv_uninst.trc`** and **`patchmgmt_uninst.trc`** and are located in the following directory: `$DBDIR/tmp/`.

Common problems and troubleshooting scenarios

This section describes how to recover from the following types of patch management problems:

- “Problems with the Tivoli Configuration Manager Automation Server silent installation” on page 97
- “Automation Server workflow” on page 97

- “Other common problems” on page 104

Problems with the Tivoli Configuration Manager Automation Server silent installation

This section covers the following areas of Tivoli Configuration Manager Automation Server silent installation:

- “Checks run during installation” on page 97
- “Steps for debugging installation failure”

Checks run during installation

The Tivoli Configuration Manager Automation Server installation, available with IBM Tivoli Configuration Manager performs the following additional actions:

1. Checks if the correct version of Cygwin is installed
2. Checks if Cygwin has been installed with the correct UNIX file type.
3. Removes the ^M characters present in the files of the tpm_install directory.
4. Before starting the Tivoli Configuration Manager Automation Server installation, checks the correctness of the directories for the configuration files and that Windows Instrumentation Service is not running to prevent the failure of the MQ Series component .
5. Validates the response file parameters.
6. Adds a timestamp to the tpm_install.log installation log file.
7. Verifies that the Windows Firewall/Internet Connection Sharing (ICS) service is started and the Windows Firewall is disabled.

Steps for debugging installation failure

1. If the silent installation is unsuccessful, review the log file. After addressing any issues found, you can run the installer again using the following command:
`./tpm_install.cmd continue`

This will continue the installation from the failed task onwards.

2. If the silent installation is still unsuccessful after running the command above, collect the following information to send to IBM Tivoli Software Support to help diagnose the problem:
 - Any Windows event log relevant to the failed installation.
 - tpm_install.log file. For more information on this file, see “Tivoli Configuration Manager Automation Server silent installer log” on page 95.
 - Operating system level.
 - Service pack information.
 - Hardware description.
 - Installation package (CD or electronic download) and level.
 - Windows services that were active during the unsuccessful installation (for example, antivirus software).
 - Whether you are logged on to the local machine console (not through a terminal server).

Automation Server workflow

This section covers the following areas of workflows:

- “Workflow Description” on page 98
- “Troubleshooting Workflows” on page 99

Common problems and troubleshooting scenarios

- “Problems with Workflows” on page 100

Workflow Description

This section briefly describes the steps the workflow performs as part of the Tivoli Configuration Manager Automation Server. In general, the workflow is responsible for downloading patches from Microsoft Software Update Services, creating software packages and generating activity plans. The workflow is scheduled by default to begin running at 11 p.m. every day. For information on modifying the workflow schedule, refer to “Scheduling the workflow” on page 42.

Table 19. Tasks performed by the Automation Server workflow

Step	Description
1	Downloads the .cab file from the Microsoft Web site to the Automation Server workstation and stores it in the \$TIO_HOME/mscab directory.
2	Determines the list of approved patches based on Microsoft WSUS and downloads the ApprovedChanges.txt file to the Automation Server workstation storing it in the \$TIO_HOME/mscab directory.
3	Identifies the list of Tivoli servers to which the .cab and ApprovedChanges.txt file must be distributed. You can view and modify the Tivoli server list using the wseccfg command. The ApprovedChanges.txt file is distributed using the wtransfer command and stored in the \$DBDIR/Inventory directory.
4	Downloads the patch executable files to the Automation Server workstation and stores them in the WSUS directory.
5	Determines the list of missing patches in the Tivoli Configuration Manager Automation Server environment using the wseccrpt command. The Inventory database must have been previously populated by manually running the Windows_Initial_Patch_Scan and Windows_Patch_Scan, as described in “Submitting the patch scan” on page 44.
6	Prepares the list of patches that are required based on the list of missing and approved patches. You can obtain information on the status of patches and endpoints in your environment with the wseccrpt command.
7	Identifies the list of source hosts to which the patch executables must be distributed. You can view and modify the source host list using the wseccfg command. The patch executables are distributed using the wtransfer command and stored in the directory specified using the provider_patch_dir key with the wseccfg command.
8	For each source host, it determines the identifier of the Tivoli management region to which it is connected.
9	For each Tivoli management region, performs the following operations: <ol style="list-style-type: none">1. Removes packages that are no longer required from the Tivoli management regions. For a package to be removed, the Administrator must first revoke the related patch approval. The removal operation is performed if you set the delete_packages key to yes with the wseccfg command.2. Removes obsolete plans if you set the delete_plans key to yes with the wseccfg command.3. Determines the list of packages to be created in the specific Tivoli management region based on the list of required patches and packages already present in the region.4. For each software package, performs the following operations:<ul style="list-style-type: none">• Transfers the patch executables to the source host associated to the Tivoli management region using the wtransfer command.• Generates the software package and related query using the wsecgensp command. The package generation is responsible for deleting the patch executable files after preparing the package, if you set the remove_patch_files_if_built key to yes with the wseccfg command.

Table 19. Tasks performed by the Automation Server workflow (continued)

Step	Description
10	Generates the IBM Tivoli Configuration Manager activity plan for patch distribution using the wsecgenplan command.
11	Notifies the Administrator of the plan creation with an e-mail if you defined the email_notification_address key with the wseccfg command.

Troubleshooting Workflows

All deployment engine run-time results are logged to the \$TIO_LOGS/deploymentengine/console.log file. If you are looking for additional details to help determine why a particular transition within a workflow has failed, review this log file first.

The standard functionality of Tivoli Configuration Manager Automation Server is to suppress stack trace Java™ exception error messages within the user interface, and the actual commands that have been run and issued by the deployment engine at each transition. It is important that you understand exactly which commands are being issued by the deployment engine at run time (for debugging purposes), and to achieve this, you must enable debug mode as described in “Automation Server workflow logging levels” on page 94

- By only configuring the log4j settings, debug mode (default) still suppresses stack trace error messages from the error window within the Tivoli Configuration Manager Automation Server user interface when you run workflows, but does not log actual commands issued by the deployment engine. To enable this level of logging, you must define a global variable. To do this:
 1. Click **Configuration** and click the **Variables** tab.
 2. Assign the name **debug** to the new variable, select **Deployment Engine** as the component, and then assign the value of **true** to the new variable.

This enables full debug mode when you run workflows for both the Tivoli Configuration Manager Automation Server user interface and the log file.

Note: When you add this variable, the change takes effect immediately for each instance that a workflow is run.

For all errors that occur when you run your workflow, check the run history.

Displaying execution details for the workflow

To display the details of the workflow execution:

1. Click **System configuration and workflow management > Deployment Requests**.
2. In the list, identify the workflow execution whose details you want to display and click on the related **RequestId**.
3. The Execution Logs tab is displayed which contains all the transition executions for the workflow, including their date and time. Further levels of detail are displayed by clicking the Execution log detail icon



for each executed transition.

Exporting workflow run history logs

You can export the log files of your workflow history to help troubleshoot any workflow problems. The workflow run logs are exported into an XML file with a filename that you specify by running the following command:

```
workflowLogExport.cmd {-n<workflow_name>} {-f<export_filename>}
```

For example:

```
workflowLogExport.cmd -nGroup_Status_Updater -f"c:/myDirectory/myWorkflowExport.xml"
```

Problems with Workflows

This section includes troubleshooting scenarios for the following problems:

- “Microsoft WSUS Server not communicating with the Automation Server machine”
- “Automation Server machine cannot access the WSUS database”
- “Automation Server workflow fails”
- “DB2 Universal Database Workgroup Server deadlocks occur when the system runs a logical operation” on page 101
- “Shell command error: Resource temporarily unavailable” on page 101
- “WSUS Authentication Failure” on page 101
- “Group_Status_Updater not starting for data center model lock” on page 102
- “TCM_Update_Patches failure due to WSUS synchronization” on page 102
- “Gateway not considered when running workflows” on page 103
- “Osql failure with a remote WSUS database” on page 103
- “Duplicated entries in the wsecrpt output” on page 107

For any additional problems with workflows, refer to the *Tivoli Provisioning Manager 2.1 Problem Determination Guide*.

Microsoft WSUS Server not communicating with the Automation Server machine:

Cause: If you receive an error related to access denial and invalid authentication scheme when trying to connect to the WSUS Server then you have not properly configured the SSH protocol.

Solution: Follow the steps outlined in “Installing and Configuring the Microsoft WSUS Server” on page 27.

Automation Server machine cannot access the WSUS database:

Cause: If you receive an error when trying to access the WSUS database then you have not properly defined the `wsus_db_name` and `wsus_db_host` keys or you have not properly configured TCP/IP.

Solution: Perform the following steps:

1. See the WSUS keys descriptions at page 72, to check the definition correctness.
2. Run the SQL Server Network Utility (`svrnetcn.exe`) to verify that the TCP/IP is an enabled library among the SQL Server network libraries. To enable this library, restart the SQL Server before using WSUS.

Automation Server workflow fails:

Cause: Automation Server workflow fails with an error message related to Microsoft WSUS access. The patch executable files are not downloaded to the path \$drive:\WSUS.

Solution:

- Check internet connection on Automation server machine.
- Verify Microsoft WSUS server configuration, see “Installing and Configuring the Microsoft WSUS Server” on page 27.
- Verify WSUS configuration parameters of Automation server, see “Changing WSUS values” on page 39.

Cause: Automation Server workflow fails with the following error message:

```
COPDEX123E The workflow threw a Group_Status_Updater_Errors exception.  
The message is: Update of patches into TMR failed. Error message:  
COPDEX040E An unexpected deployment engine exception COPCOM123E  
This shell command error occurred: Exit value=2, Error stream="AMN0005E  
An unrecoverable error occurred during an attempt to connect to the  
RIM object. Activity Planner engine cannot start. Check whether the  
RIM connection works properly. ", Result stream="". occurred.
```

Solution: Check APM RIM connection. Issue command `wrimtest -l planner`.

Cause: Automation Server workflow loops. User `tioadmin` has not been added to the list of Tivoli administrators logins.

Solution: Add `tioadmin` user to the list of the Administrators logins, see 33.

DB2 Universal Database Workgroup Server deadlocks occur when the system runs a logical operation:

Cause: The DB2 Universal Database Workgroup Server database configuration parameter (locklist) value is not large enough.

Solution: Adjust the size of the lock list value from 50 to 2000. Depending on the volume of traffic on the servers and the size of your Data Center Model, you might want to select a more appropriate value.

Shell command error: Resource temporarily unavailable:

Cause: If you get this error: `bash: fork: Resource temporarily unavailable`, it might be accompanied by this Tivoli Configuration Manager Automation Server message:

```
COPCOM123E This shell command error occurred: Exit value=128, Error stream=  
"/usr/bin/bash: fork: Resource temporarily unavailable ", Result stream="".
```

Solution: Increase the maximum number of processes allowed per user on your system.

For instructions on setting the maximum number of processes allowed per user, refer to the documentation for your operating system.

WSUS Authentication Failure:

Cause: If you log on to the Automation Server console, change the user name associated to the credential of the Service Access Point "SSH_Server2" of the WSUS

Common problems and troubleshooting scenarios

machine with a wrong value and then run the TCM_MS_Discover_Patches and TCM_MS_Get_ApprovedChanges workflows, both the workflows fail with the following misleading message:

```
COPDEX044E An error occurred in the embedded logical operation:
"COPDEX044E An error occurred in the embedded logical operation:
"COPCOM116E The operation timed out."."
```

The problem is an authentication failure generated by the Automation Server when trying to use SSH to establish a connection with a remote machine.

Solution: Ensure you entered a valid user name.

Group_Status_Updater not starting for data center model lock:

Cause: If the Automation Server is disconnected from the network when Group_Status_Updater is working, when the network connection is re-established the workflow does not continue and you can only stop its execution. However when a new instance of Group_Status_Updater workflow starts, it stops immediately on a data center model lock.

Solution: A workaround for this situation is to run the following command to reinitialize the Automation Server database:

```
cd $TIO_HOME/tools
./reinit.cmd ../xml/tcm-dcm_22.xml
```

If you do not want to run the reinit operation, you can perform the following steps:

1. Stop the Automation Server with the following command:

```
cd $TIO_HOME/tools
./tio_stop.cmd
```

2. Open a DB2 command line processor window and run the following commands:

```
connect to TC user <DB2_ADMIN_USER> using <DB2_ADMIN_PWD>
update dcm_object set locked_until = null
```

3. Check with the following command that there are no other not null records
select locked_until from dcm_object where name='WSUS server name'

If you get a not null record repeat the previous command.

4. When there are no other not null records, run the following commands:

```
delete from local_variable
delete from stack_frame
delete from workflow_execution_thread
commit
```

5. Restart the Automation Server. A new instance of Group_Status_Updater workflow should run correctly.

TCM_Update_Patches failure due to WSUS synchronization:

Cause: After a WSUS synchronization the number of patches managed by Group_Status_Updater is increased. It could occur that the inner workflow named TCM_Update_Patches fails in the patches, queries, and plans preparation phase, and displays the following message:

```
COPDEX032E The system cannot evaluate the expression:
com.ibm.tivoli.orchestrator.TCMHelper#addMissingPatchesToTCM.
```

The Automation Server console.log file will also report messages similar to the following:

```
Caused by: java.lang.OutOfMemoryError
at java.lang.String.(String.java:Compiled Code))
at java.io.BufferedReader.readLine(BufferedReader.java:Compiled Code))
at java.io.BufferedReader .readLine(BufferedReader.java:Inlined Compiled Code))
at com.ibm.tivoli.orchestrator.TCMHelper.getPatchesFromItemsFiles(TCMHelper.
java:Compiled Code))
at com.ibm.tivoli.orchestrator.TCMHelper.addMissingPatchesToTCM(TCMHelper.java:550)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.
java:Compiled Code))
at java.lang.reflect.Method.invoke(Method.java:Compiled Code))
at com.ibm.tivoli.orchestrator.de.util.ReflectionHelper.evaluate(ReflectionHelper.
java:221)
at com.ibm.tivoli.orchestrator.de.instruction.impl.InvokeJavaHelper.evaluate
(InvokeJavaHelper.java:74)
```

This is because the Java maximum heap size is set by default to 128 MB in the \$TIO_HOME/tools/run-deploymentengine.cmd script.

Solution: To solve the problem increase the maximum heap size for instance to 512 Mb with the following commands:

```
cd $TIO_HOME/tools
./tio_stop.cmd
```

Edit the file *run-deploymentengine.cmd*, modifying the line:

```
"%JAVA_HOME%\bin\java" -Xms32m -Xmx128m ^
```

to:

```
"%JAVA_HOME%\bin\java" -Xms32m -Xmx512m ^
```

Run the following command:

```
./tio_start.cmd
```

Gateway not considered when running workflows:

Cause: If you set the configuration key *cab_gateways_file* to a file which has been created from an MS-DOS command prompt window by using either the Notepad or WordPad editor without adding a new line after the last gateway name specified in the file, the last gateway name is not considered when running the TCM_Upload_Catalog or Group_Status_Updater workflows.

If there is not a "new line" command after the last element in the file, the element is ignored.

Solution: Edit the file with the vi editor because the new line command is inserted automatically.

Osql failure with a remote WSUS database:

Cause: If you have a remote WSUS database, the workflow might fail.

Solution: Edit the "\$osqlprog" line of the \$TIO_HOME/misc/WSUS_info_retriever.sh script as follows:

Common problems and troubleshooting scenarios

- Add the user name (-U) and the password (-P) options specifying the values of the SQL server account the database administrator created. To avoid the password definition you can set an environment variable.
- Remove the -E option.

The "\$osqlprog" line becomes:

```
"$osqlprog" -U user_name -P password -S "$DBSERVER" -d "$DBNAME"  
-h-1 -b -n -w $DBROWLENGHT-i "$sqlInputFile"
```

Other common problems

This section includes troubleshooting scenarios for the following problems:

- “wsecrpt command does not find any patches”
- “Patch scan fails”
- “Software package distribution fails” on page 105
- “Automation Server cannot start” on page 105
- “Patch Management policy regions available only from the top level policy region” on page 105
- “Workflow failure due to wtransfer error” on page 106
- “The activity plan fails on targets deleted from the Tivoli database” on page 106
- “Error downloading files if patch is a cabinet file” on page 106
- “Duplicated entries related to the same QNUM” on page 106
- “Incorrect endpoint status report” on page 106
- “Office installation failure” on page 107
- “Duplicated entries in the wsecrpt output” on page 107
- “Windows Update Agent (WUA) does not work properly” on page 107
- “Wrong code set” on page 107
- “2007 Microsoft Office Suite Service Pack 1 installation fails” on page 107

For any additional common problems, refer to the Tivoli Provisioning Manager 2.1 Problem Determination Guide.

wsecrpt command does not find any patches

Cause: Command **wsecrpt** returns message: "CMYSE0113I No data available in the database for the specified query". An initial patch scan is required to populate the Patch Management database. The initial patch scan has not been performed.

Solution: Perform the initial patch scan procedure by distributing the Windows_Initial_Patch_Scan profile. See “Performing the initial patch scan” on page 52.

Patch scan fails

Cause: The distribution of the Windows_Initial_Patch_Scan or the Windows_Patch_Scan profile fails with error message

```
inv_config_msgs:0031 INVCFO0031E Failed to get the file /Tivoli/db/  
Tab237013.db/inventory/wsusscan.cab
```

The wsusscan.cab file is required to perform scanning of Windows patches. The file is automatically downloaded from the Microsoft Web site by the Automation Server workflow to the path \$BDDIR/inventory on the Tivoli server.

Solution: Check that the above directory contains the wsusscan.cab file. If the file is not present, then submit the Automation Server workflow again to download the wsusscan.cab file from Microsoft Web site.

Cause: The distribution of the Windows_Initial_Patch_Scan, or Windows_Patch_Scan profile is successfully submitted, but the distribution fails on one or more endpoints. WUA is not installed on the endpoints reporting the failure.

Solution: Install WUA to all Windows endpoints in your environment. See “Deploying WUA and Qchain on endpoints” on page 30.

Cause: If, before installing WUA on a Windows Server with Office XP, you distribute the Windows_Initial_Patch_Scan profile, the resulting tivpatchscan.mif is empty. This file remains empty even if you install WUA and re-distribute the Windows_Initial_Patch_Scan.

Cause: The distribution of the Windows_Initial_Patch_Scan, or Windows_Patch_Scan profile is successfully run but the patchscan.mif is empty.

Solution: Check if:

- You are using an updated version of the .cab file together with an old version of WUA.
- The .cab is corrupt.

Software package distribution fails

Cause: The plan fails on one or more endpoints due to a software package distribution failure. The file qchain.exe has not been distributed on the endpoints reporting the failure. The Software Distribution log file shows the following error message:

```
ISSE0123E Unable to execute or complete execution of program 'during_install  
- \..\inv\SCAN\qchain.exe ()'.
```

Solution: Distribute the qchain.exe file to all endpoints reporting the failure. See “Deploying WUA and Qchain on endpoints” on page 30.

Automation Server cannot start

Cause: The Automation Server cannot successfully start if it has been previously started or if you attempt to start it with a user different from tioadmin.

Solution: To restore a working environment:

1. Verify that the Automation Server is not running. Stop the Automation Server with the same user you used to start it.
2. Delete tpm.lck, tpm_start.log, and tpm_stop.log files under \$TIO_HOME\logs.
3. Logoff the current user.
4. Start the Automation Server as described in “Starting the Tivoli Configuration Manager Automation Server” on page 37.

Patch Management policy regions available only from the top level policy region

In an environment with interconnected regions, Patch Management regions issuing the **wsecgensp** command from a Tivoli region with Patch Management installed on a source host belonging to a Tivoli server without Patch Management installed, are

Common problems and troubleshooting scenarios

available only from the top level policy region. On the contrary, on the Tivoli server with Patch Management installed, Patch Management regions are available directly from the Tivoli desktop.

To workaround this problem, enter the following command to move the policy region to the Administrators collection:

```
wmv @PolicyRegion:policy_region_name /Administrators
```

then drag and drop the policy region from the Administrators collection to the Administrator's Desktop. For more information on the **wmv** command, refer to *Tivoli Management Framework: Reference Manual*.

Workflow failure due to wtransfer error

Cause: The **wtransfer** command fails on the Automation Server and **odstat** displays an e=128 error code.

Solution: Verify that the Windows Firewall/Internet Connection Sharing (ICS) service is started and the Windows Firewall is disabled.

The activity plan fails on targets deleted from the Tivoli database

Deleting endpoints from the Tivoli database does not delete those endpoints from the configuration repository. This might cause the activity plan to fail on the deleted endpoints because targets for the workflow are defined based on the information in the configuration repository.

To prevent this problem, after deleting the endpoints using the **wdelep** command, run the **winvrnode** command to remove hardware and software scan information from the configuration repository. For more information on these commands refer to *Tivoli Management Framework: Reference Manual* and *IBM Tivoli Configuration Manager: User's Guide for Inventory*.

Error downloading files if patch is a cabinet file

Cause: The Patch Management solution does not manage those hot fixes that have a cabinet file (.cab file) instead of an executable file on the WSUS server. If there is a cabinet file, the patches are not downloaded from the WSUS server to the Automation Server machine.

Duplicated entries related to the same QNUM

Cause: The report generated after the Inventory scan contains some duplicated entries. It is only a report problem, all the patches have been handled correctly. The problem is how Windows Update Service (WUA) returns the results of a patch scan.

Incorrect endpoint status report

Cause: When a patch installation requires the endpoints to be rebooted, after the reboot, the status can be different for the affected endpoints. This is due to a timing condition on the endpoints themselves.

Solution: The status is synchronized when a new distribution or **wsyncsp** command is performed on the endpoint for which the patch status is still the status before the reboot.

Office installation failure

Cause: During the installation of Office XP Service Pack 3 (KB832671) on a computer running Windows Server 2003 or Windows 2000 Server, with Microsoft IIS and Microsoft FrontPage, you might receive error 1603.

Solution: This problem is described in "Known issues when you install Office XP SP3" page on the Microsoft support Web site: <http://support.microsoft.com>. Check this site to see if the problem has been solved.

Duplicated entries in the wsecrpt output

Cause: If you distribute a patch normally and subsequently distribute it a second time indicating the emergency configuration key, then running the wsecrpt command might report the patch twice for the same Qnumber.

Windows Update Agent (WUA) does not work properly

For the Microsoft Windows Update Agent (WUA) to work properly, ensure that the following Windows services are enabled and set to Automatic:

- Automatic Updates
- Background Intelligent Transfer Service (BITS)

In addition, for the WUA to receive necessary updates from other Microsoft products, ensure that Windows Installer 3.1 is installed.

Wrong code set

Cause: If you are defining a name for a group, a patch, or any other object, containing a character code set that is not defined in the Tivoli environment, the character is not displayed correctly in the name.

Solution: Set the TIS_CODESET variable to the appropriate code set as follows:

1. Copy the Tivoli environment settings to a temporary file:

```
odadmin environ get >env.out
```

2. Add the following line to the temporary file:

```
TIS_CODESET = TIS_CODESET
```

where *TIS_CODESET* is the new value of the code set.

3. Import the new TIS_CODESET setting value in the Tivoli environment:

```
odadmin environ set < env.out
```

4. Stop and start the Tivoli server:

```
reexec oserv all
```

2007 Microsoft Office Suite Service Pack 1 installation fails

Cause: The "2007 Microsoft Office Suite Service Pack 1" installation fails. If you perform a query in Patch Management using the **wsecrpt** command, the patch is missing.

Solution: Despite this error, in most cases the patch has been installed correctly. Verify the patch installation from the Add or Remove Programs list of the Windows operating system. A ticket has been opened to Microsoft for this software limitation.

Appendix A. Uninstalling the automated patch management solution

This appendix describes the steps required to remove patch management components from your environment.

Uninstall the product from the Tivoli server

To uninstall the product from the Tivoli server, enter the following command:

```
sh wuninst patch_mgmt node [-rmfiles]
```

This command deletes all the Patch Management objects in the Tivoli Desktop, except for custom software packages located in the Windows_Patch_Tools policy region.

Specifying the **rmfiles** option in the command, you delete recursively the directory \$BINDIR/../../generic/TME/PATCH_MGMT.

It does not delete the activity plans generated during the use of the automated patch management solution.

Note: If the uninstall operation fails and not all the objects are removed from the Tivoli Desktop, you need to manually delete the remaining objects.

Uninstalling the Tivoli Configuration Manager Automation Server

This section describes how to uninstall the Tivoli Configuration Manager Automation Server.

Note: The uninstaller does not check if the Tivoli Configuration Manager Automation Server is running before uninstalling. However, there will be a warning message on a separate panel at the beginning of the uninstall informing you that Tivoli Configuration Manager Automation Server must be shut down before you uninstall it.

To uninstall using the graphical uninstaller:

1. Open a Cygwin bash window, switch to the \$TIO_HOME directory and run the command: `chown -R Administrator .`
2. From the Cygwin bash window, switch to the \$WAS_HOME directory and run the command: `chown -R Administrator .`
3. Stop all services related to DB2 and ITDS.
4. Stop the Tivoli Configuration Manager Automation Server. To do this, refer to the section, "Stopping the Tivoli Configuration Manager Automation Server" on page 38.
5. Click **Start-> Control Panel-> Add/Remove Programs-> Tivoli Provisioning Manager 2.1**, and then click **Remove**.
6. In the Tivoli Configuration Manager Automation Server box, select the language you want the graphical uninstaller to use, and click **OK**.
7. The Welcome panel opens. Click **Next**.

Uninstalling automated patch management

8. On the next panel, click the check box to remove the Tivoli Configuration Manager Automation Server database. This will drop the Tivoli Configuration Manager Automation Server database. It will not uninstall your database server. Click **Next**.
9. The Database Configuration panel appears. Enter the instance owner and password. This is required to drop the database. Click **Next**.
10. On the WebSphere Application Server configuration panel, the WebSphere Application Server installation directory is detected automatically, along with the user ID and password.

Note: The WebSphere Application Server security will be turned off after the uninstall and runAsUserproperty of the WebSphere Application Server will be rolled back to root. The WebSphere Application Server unconfiguration removes the Tivoli Configuration Manager Automation Server configuration.

11. On the Uninstallation Preview panel, review the selections you have made. To correct any of the options you have selected, click **Back** and then make any required changes. When the selections are correct, click **Next**. Tivoli Configuration Manager Automation Server will be uninstalled.
12. When the uninstallation and unconfiguration is complete, the Uninstallation summary panel opens and indicates whether the uninstall completed successfully. Click **Finish**.
13. From **Add/Remove Programs**, click WebSphere Application Server 5.1, then click **Remove** and follow the removal prompts provided.
14. From **Add/Remove Programs**, click WebSphere EMPS, then click **Remove** and follow the removal prompts provided.
15. From **Add/Remove Programs**, click WebSphere MQ, then click **Remove** and follow the removal prompts provided.
16. From **Add/Remove Programs**, click Tivoli Directory Server 5.2, then click **Remove** and follow the removal prompts provided.
17. From **Add/Remove Programs**, click TivGUID, then click **Remove** and follow the removal prompts provided.
18. From **Add/Remove Programs**, click DB2 Universal Database Workgroup Server Workgroup Server Edition, then click **Change** followed by **Remove** and follow the removal prompts provided.
19. Delete the following system users and groups:
 - Click **Start-> Control Panel-> Administrative tools-> Computer Management** and navigate to the System Tools-> Local Users and Groups-> Users directory.
 - From the Users directory, delete the following system users: tioadmin, db2admin, tioldap.
 - Click **Start-> Control Panel-> Administrative tools-> Computer Management** and navigate to the System Tools-> Local Users and Groups-> Groups directory.
 - From the Groups directory, delete the following system groups: tivoli, mqm.
20. Delete the folders for \$ITDS_HOME, \$WAS_HOME, \$TIO_HOME, and Tivoli Directory Server database.
21. To reinstall, you must run the installer again using the following command:
`./tpm_install.sh reinstall`

This starts the installation from the beginning.

Appendix B. Support information

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- “Searching knowledge bases”
- “Obtaining fixes”
- “Receiving weekly support updates” on page 112
- “Contacting IBM Software Support” on page 113

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching the information center

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

Searching the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the **Web search** topic in your information center. In the navigation frame, click **Troubleshooting and support ► Searching knowledge bases** and select **Web search**. From this topic, you can search a variety of resources, including the following:

- IBM technotes
- IBM downloads
- IBM Redbooks®
- IBM developerWorks®
- Forums and newsgroups
- Google

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

1. Go to the IBM Software & download Web page at <http://www.ibm.com/support/us>.
2. Click **Downloads and drivers** in the **Support topics** section.
3. Select the **Software** category.
4. Select a product in the **Sub-category** list.
5. In the **Find downloads and drivers by product** section, select one software category from the **Category** list.

Support information

6. Select one product from the **Sub-category** list.
7. Type more search terms in the **Search within results** if you want to refine your search.
8. Click **Search**.
9. From the list of downloads returned by your search, click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at <http://techsupport.services.ibm.com/guides/handbook.html>.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support Web site at <http://www.ibm.com/software/support>.
2. Click **My support** in the upper right corner of the page.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. Click **Edit profile**.
5. In the **Products** list, select **Software**. A second list is displayed.
6. In the second list, select a product segment, for example, **Application servers**. A third list is displayed.
7. In the third list, select a product sub-segment, for example, **Distributed Application & Web Servers**. A list of applicable products is displayed.
8. Select the products for which you want to receive updates, for example, **IBM HTTP Server** and **WebSphere Application Server**.
9. Click **Add products**.
10. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
11. Select **Please send these documents by weekly email**.
12. Update your e-mail address as needed.
13. In the **Documents** list, select **Software**.
14. Select the types of documents that you want to receive information about.
15. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

- | | |
|-----------------|---|
| Online | Send an e-mail message to erchelp@ca.ibm.com , describing your problem. |
| By phone | Call 1-800-IBM-4You (1-800-426-4968). |

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

Online

Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home and click **How to Enroll**.

By phone

For the phone number to call in your country, go to the IBM Software Support Web site at <http://techsupport.services.ibm.com/guides/contacts.html> and click the name of your geographic region.

- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at <https://techsupport.services.ibm.com/ssr/login>.
- For customers with IBMLink™, CATIA, Linux®, S/390®, iSeries®, pSeries®, zSeries®, and other support agreements, go to the IBM Support Line Web site at <http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006>.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at <http://www.ibm.com/servers/eserver/techsupport.html>.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at <http://techsupport.services.ibm.com/guides/contacts.html> and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. “Determining the business impact”
2. “Describing problems and gathering information” on page 114
3. “Submitting problems” on page 114

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1

The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2

The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3

The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4

The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Submit and track problems** on the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>. Type your information into the appropriate problem submission form.

By phone

For the phone number to call in your country, go to the contacts page of the *IBM Software Support Handbook* at <http://techsupport.services.ibm.com/guides/contacts.html> and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

Trademarks

AS/400, DB2, DB2 Universal Database, developerWorks, eServer, IBM, the IBM logo, IBMLink, IMS, iSeries, Lotus, Passport Advantage, pSeries, Rational, Redbooks, S/390, Tivoli, the Tivoli logo, Tivoli Enterprise Console, WebSphere, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, Celeron, Intel Centrino, Intel Xeon, Itanium, Pentium and Pentium III Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows NT are registered trademarks of Microsoft Corporation 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

- activity plan
 - submit 60
- activity plans
 - customizing 61
 - viewing 59
- administrative console
 - change password 38
 - log on 38
- approval
 - revoking 61
- automated patch management
 - installing 7
 - uninstalling 109
- Automation Server
 - logs 93
 - stopping 38
 - troubleshooting installation 97
- Automation server component 7
 - hardware requirements 7
 - software prerequisites 9

B

- books
 - see publications x, xii

C

- commands
 - winvmgr 34
 - wmailhost 33
 - wseccfg 68
 - wsecgenplan 77
 - wsecrpt 85
 - wtransfer 90
- Configuring
 - WSUS settings 27
- Configuring the environment 31
- conventions
 - typeface xiii
- Conventions used xiii
- customer support
 - See Software Support
- customizing
 - activity plans 61
 - software packages 61
- cygwin
 - install 7
 - package 7

D

- directory names, notation xiii
- discovery association
 - add 40
 - remove 40
- Downloading
 - qchain.exe 16

- Downloading (*continued*)
 - WUA 16

E

- e-mail notification 33
- education
 - see Tivoli technical training xii
- environment variables, notation xiii
- exclusive
 - patches 62

F

- files
 - qchain.exe 5
 - TEDWScheduler.ini 42
 - wsusscan.cab 5
 - wsusscn2.cab 5
- filtering
 - endpoints 62
- fixes, obtaining 111

G

- grouping
 - endpoints 62

I

- IIS
 - settings 28
- information centers, searching for
 - problem resolution 111
- initial scan 44, 52
- install
 - cygwin 7
- installation 7
 - using Tivoli Software Installation Service 18
- installing
 - Automation Server component 7
 - Patch Management component 16
- interconnected regions 35
- Internet
 - searching for problem resolution 111
- Internet Information Services
 - settings 28

K

- knowledge bases, searching for problem
 - resolution 111

L

- locale 9

- log
 - silent install 95
- logging levels, workflow 94
- logging on
 - administrative console 38
- logs
 - Automation Server 93
 - Automation Server prerequisites 96
 - Automation Server start and stop 96
 - Patch Management 96

M

- manuals
 - see publications x, xii

N

- notation
 - environment variables xiii
 - path names xiii
 - typeface xiii

O

- obsolete patches
 - delete_packages 61
 - delete_plans 61
 - revoking approval 61
- online publications
 - accessing xii
- ordering publications xii

P

- passwords
 - changing default 38
- patch installation results 60
- patch management
 - scenario 53
 - tables 44
 - views 45
- Patch management 1
- Patch Management
 - logs 96
 - traces 96
- Patch Management component
 - installing 16
 - ISMP install 17
 - SIS install 18
 - Tivoli CLI install 20
 - Tivoli desktop install 19
- patch scan 44
 - initial 52
- patch scan results 44
- patches
 - exclusive 62
 - reboot 60, 62
 - standard 62

- patches (*continued*)
 - supported 3
- path names, notation xiii
- Prerequisites 16
- problem determination
 - describing problems 114
 - determining business impact 113
 - submitting problems 114
- publications x
 - accessing online xii
 - ordering xii
 - related xi

Q

- qchain.exe
 - description 5
 - downloading 16

R

- reboot 60
 - patches 62
- response file
 - installing 9
 - template 9
 - values 9, 12
- road map 6
- run
 - workflow 40

S

- scanning 44
- scenario 53
- schedule
 - workflow 42
- server
 - start 37
 - stop 37
- silent install 9
 - log 95
- software packages
 - customizing 61
 - viewing 58
- Software Support
 - contacting 113
 - describing problems 114
 - determining business impact 113
 - receiving weekly updates 112
 - submitting problems 114
- standard
 - patches 62
- Start log
 - Automation Server 96
- status
 - checking workflow 41
- Stop log
 - Automation Server 96
- stopping
 - Automation Server 38
 - workflow 41
- submitting activity plan 60
- Support information xii
- supported patches 3

T

- tables
 - patch management 44
- TEDWScheduler.ini file 42
- templates
 - activity plans 61
 - software packages 61
- tioadmin
 - adding login 33
- Tivoli software information center xii
- Tivoli Software Installation Service,
 - installing from 18
- Tivoli technical training xii
- traces
 - Patch Management 96
- training, Tivoli technical xii
- troubleshooting
 - Automation Server installation 97
 - workflow 99
- Troubleshooting 93
- typeface conventions xiii

U

- uninstalling 109
- Uninstalling the product from the Tivoli
 - server 109
- uninstalling Tivoli Configuration
 - Manager Automation Server 109

V

- variables, notation for xiii
- views
 - patch management 45

W

- winvmgr command 34
- wmailhost command 33
- workflow
 - checking status 41
 - detailed description 98
 - logging levels 94
 - run 40
 - schedule 42
 - stopping 41
 - troubleshooting 99
- wseccfg 68
 - configuring the environment 31
- wsecgenplan 77
- wsecrprt 85
- WSUS
 - change values 39
 - configure 27
- wsusscan.cab
 - description 5
- wsusscn2.cab
 - description 5
- wtransfer 90
- WUA
 - downloading 16



Program Number: 5724-C06

Printed in USA

SC23-5263-04

