Tivoli Enterprise

Installation Guide

Version 4.3.1

GC32-0804-02

Tivoli Enterprise

Installation Guide

Version 4.3.1

GC32-0804-02

Note

Before using this information and the product it supports, read the information in Appendix D, “Notices,” on page 399.

Tivoli Enterprise Installation Guide

Contents

Chapter 13. Using response files . . . 143

Chapter 14. Using Tivoli Software Installation Service log files 171

Chapter 15. Checking prerequisites 179

Part 4. Resource and product installation 195

Chapter 16. Installing resources managed by Tivoli Management Framework 197

Chapter 17. Installing Tivoli products and patches 221

Chapter 18. Using RIM objects 243

Chapter 19. Creating InstallShield response files 253

Chapter 20. Uninstalling a Tivoli environment 255

Part 5. Scenario and additional considerations 269

Chapter 21. Installation scenario . . . 271

Preface

The *Tivoli Enterprise Installation Guide* explains how to install and upgrade Tivoli Enterprise software within your Tivoli® region using the available installation mechanisms provided by Tivoli Software Installation Service and Tivoli Management Framework. Tivoli software includes the Tivoli server, managed nodes, gateways, endpoints, and RDBMS Interface Module (RIM) objects.

This guide uses a task-oriented approach to explain how to manage and install Tivoli product images. You can install product images using any of the following installation mechanisms:

- Tivoli Software Installation Service
- Tivoli desktop
- Command line
- Platform-specific methods

Tivoli Software Installation Service is an enhancement to the Tivoli Management Framework installation mechanisms (desktop and command line).

Besides providing information and procedures for installing and upgrading Tivoli Management Framework resources (Tivoli server, managed nodes, gateways, and endpoints) using each of the installation mechanisms, this guide provides the procedures for installing and upgrading Tivoli products.

Who should read this guide

The target audience for this guide is system administrators responsible for the installation of Tivoli products and patches. Users of this guide should have knowledge of the following:

- PC and UNIX® operating systems
- Database architecture and concepts
- Shell programming
- Graphical user interfaces

Related documents

Tivoli provides the following related documentation:

- *Tivoli Management Framework Planning for Deployment Guide*

  Explains how to plan for deploying your Tivoli environment. It also describes Tivoli Management Framework and its services.
- *Tivoli Management Framework User's Guide*

  Describes the concepts and procedures for using Tivoli Management Framework services. It provides instructions for performing tasks from the Tivoli desktop and from the command line.
- *Tivoli Management Framework Maintenance and Troubleshooting Guide*

  Explains how to maintain a Tivoli environment and troubleshoot problems that can arise during normal operations.
- *Tivoli Management Framework Reference Manual*

Provides in-depth information about Tivoli Management Framework commands. This manual is helpful when writing scripts that are later run as Tivoli tasks. This manual also documents default and validation policy scripts used by Tivoli Management Framework.

- *Tivoli Management Framework Planning for Deployment Guide*

  Explains how to plan for deploying your Tivoli environment. It also describes Tivoli Management Framework and its services.
- *Tivoli Management Framework Maintenance and Troubleshooting Guide*

  Explains how to maintain a Tivoli environment and troubleshoot problems that can arise during normal operations.
- *Tivoli Management Framework Release Notes*

  Describes the latest installation information, including supported platforms, defects, and limitations.

In addition to being familiar with information about Tivoli Management Framework, you should be familiar with the product documentation for each Tivoli product that you plan to install after you install Tivoli Management Framework.

What this guide contains

The *Tivoli Enterprise Installation Guide* contains the following sections:

- Part 1—Introduction to Installation
  - Chapter 1, “Overview of installation,” on page 3

    Provides an overview of installing Tivoli Management Framework and Tivoli Enterprise software using the available installation mechanisms.
  - Chapter 2, “Internationalization,” on page 9

    Provides information about creating and running a Tivoli region in languages other than English.
- Part 2—Tivoli Server Installation
  - Chapter 3, “Installing a Tivoli server,” on page 23

    Provides detailed instructions for installing a Tivoli server on UNIX and Microsoft® Windows® operating systems
  - Chapter 4, “Configuring SSH,” on page 35

    Provides instructions for configuring secure shell (SSH), which enables you to encrypt data while performing remote installations of managed nodes and endpoints.
  - Chapter 5, “Tivoli Management Framework in a Microsoft Cluster Server,” on page 39

    Provides detailed information about and instructions for implementing a Tivoli server on a Microsoft Cluster Server (MSCS).
  - Chapter 6, “Enabling Tivoli Web Interfaces,” on page 43

    Provides detailed information about installing Web access for Web servers and for redirecting HTTP requests.
  - Chapter 7, “Configuring for SMTP e-mail,” on page 61

    Provides detailed information about and instructions for configuring the Simple Mail Transfer Protocol (SMTP) e-mail service in your Tivoli environment.
  - Chapter 8, “Tivoli Desktop for Windows,” on page 65

    Provides detailed instructions for installing and starting Tivoli Desktop for Windows on supported Windows and OS/2® operating systems.

- Part 3—Tivoli Software Installation Service
  - Chapter 9, “Introduction to Tivoli Software Installation Service,” on page 75

    Provides an introduction to Tivoli Software Installation Service and information about planing its deployment.
  - Chapter 10, “Installing Tivoli Software Installation Service,” on page 83

    Provides detailed information about how to install, upgrade, and uninstall the components of Tivoli Software Installation Service.
  - Chapter 11, “Configuring Tivoli Software Installation Service,” on page 99

    Provides detailed information about how to configure Tivoli Software Installation Service.
  - Chapter 12, “Using Tivoli Software Installation Service,” on page 111

    Provides step-by-step instructions for using Tivoli Software Installation Service to install Tivoli resources and products.
  - Chapter 13, “Using response files,” on page 143

    Provides instructions for how to use Tivoli Software Installation Service response files to install multiple products on multiple machines at the same time.
  - Chapter 14, “Using Tivoli Software Installation Service log files,” on page 171

    Provides detailed information about using the log files created by Tivoli Software Installation Service.
  - Chapter 15, “Checking prerequisites,” on page 179

    Provides detailed information about creating and modifying prerequisites, and describes each prerequisite check provided by Tivoli Software Installation Service.
- Part 4—Resource and Product Installation
  - Chapter 16, “Installing resources managed by Tivoli Management Framework,” on page 197

    Provides detailed instructions for creating managed nodes, gateways, and endpoints in a Tivoli region using each of the installation mechanisms.
  - Chapter 17, “Installing Tivoli products and patches,” on page 221

    Provides detailed instructions for installing and upgrading Tivoli products in a Tivoli region using each of the installation mechanisms.
  - Chapter 18, “Using RIM objects,” on page 243

    Provides detailed instructions for creating and configuring an RDBMS Interface Module (RIM) object, and discusses RIM host and RDBMS server installation and configuration considerations.
  - Chapter 19, “Creating InstallShield response files,” on page 253

    Provides information about recording and playing back response files that are used for custom resource installations.
  - Chapter 20, “Uninstalling a Tivoli environment,” on page 255

    Provides detailed instructions for uninstalling Tivoli products and managed resources from a Tivoli region.
- Part 5—Scenario and Additional Considerations
  - Chapter 21, “Installation scenario,” on page 271

    Demonstrates installing a small Tivoli region using each of the installation mechanisms described in this guide: the Tivoli Software Installation Service console and commands, the Tivoli desktop, and Tivoli Management Framework commands.
  - Chapter 23, “Troubleshooting,” on page 323

  Provides troubleshooting information about problems commonly encountered while installing Tivoli Enterprise software and while using the available installation mechanisms.
  - Chapter 24, "Reading index files," on page 331

    Describes how to the read the product index (.IND) files used to install Tivoli Enterprise software.
  - Chapter 25, "Operating system considerations," on page 339

    Provides operating system-specific information about running Tivoli Enterprise software on Windows, OS/2, NetWare, and OS/400® operating systems.
- Part 6—Appendixes
  - Appendix A, "Installation Commands," on page 361

    Lists the Tivoli Management Framework installation commands and provides reference information for the Tivoli Software Installation Service commands.
  - Appendix B, "Directory structure and system variables," on page 383

    Provides detailed information about the directory structure and system environment variables created by Tivoli Management Framework.
  - Appendix C, "X Window Resources," on page 397

    Provides information about using X Window System resources with Tivoli Management Framework.

Accessing publications online

The documentation CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both.

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/tdprodlist.html

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.ibmlink.ibm.com

From this Web page, select **Publications** and follow the instructions.

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, see the following Web site for a list of telephone numbers:

http://www.ibm.com/software/tivoli/order-lit

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

Contacting software support

If you have a problem with any Tivoli product, refer to the following IBM Software Support Web site:

http://www.ibm.com/software/sysmgmt/products/support/

If you want to contact software support, see the *IBM Software Support Guide* at the following Web site:

http://techsupport.services.ibm.com/guides/handbook.html

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers, depending on the country in which you are located
- Information you must have before contacting IBM Software Support

Conventions used in this guide

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
- 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
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

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.

- When using the bash shell on Windows operating systems, use the UNIX conventions.

Part 1. Introduction to installation

Chapter 1. Overview of installation

After you have created your deployment plan as described in the *Tivoli Management Framework Planning for Deployment Guide*, you are ready to install Tivoli Enterprise software on the machines in your distributed environment. The Tivoli Enterprise software that you install on these machines is known as your *Tivoli environment*. If you already have Tivoli Enterprise software installed, you should follow your updated deployment plan to perform the following:

- Upgrade existing Tivoli managed resources
- Upgrade Tivoli Enterprise software already installed
- Add new Tivoli managed resources
- Add additional Tivoli Enterprise software

You can install most Tivoli managed resources and Tivoli Enterprise software by using the installation mechanisms provided by either Tivoli Software Installation Service or Tivoli Management Framework. Some Tivoli managed resources, such as the Tivoli server, cannot be installed using these installation mechanisms. In these cases, Tivoli Enterprise products provide alternative installation mechanisms that are documented in this guide.

Where both Tivoli Software Installation Service and Tivoli Management Framework installation mechanisms are available, it is recommended that you use Tivoli Software Installation Service.

This chapter contains the following sections:

- "Installation overview"
- "Installing with Tivoli Software Installation Service" on page 4
- "Installing with Tivoli Management Framework" on page 5
- "Comparing installation mechanisms" on page 5
- "Packaging of installation images" on page 7
- "Disk space requirements" on page 7

Installation overview

Based on the information in your deployment plan, you will either create a new Tivoli environment or upgrade an existing one. Remember that upgrading an existing Tivoli environment can include adding new machines as well as upgrading existing Tivoli managed resources and software that was previously installed.

If you are creating a new Tivoli environment, the general installation sequence is as follows:

1. Install the Tivoli server.

   Tivoli Management Framework provides a special installation mechanism for the Tivoli server. Refer to Chapter 3, "Installing a Tivoli server," on page 23 for instructions. A Tivoli server cannot be installed using Tivoli Software Installation Service.

2. If the Tivoli server is a supported Windows operating system, install the Tivoli desktop. If your Tivoli server is a UNIX machine, the Tivoli desktop is installed when you install the Tivoli server.

   Refer to Chapter 8, “Tivoli Desktop for Windows,” on page 65 for instructions.
3. Create the appropriate administrators and policy regions as described in your deployment plan.

   For details on creating administrators with the required authorization roles as well as for details on creating policy regions, refer to the *Tivoli Management Framework User’s Guide*.
4. If you do not want to use the Tivoli server to create managed resources and install Tivoli Enterprise software, create at least one managed node.

   You can use either the Tivoli desktop or the **wclient** command. These procedures are described in “Creating managed nodes” on page 197.

   Note: If the Tivoli server is not a supported Windows operating system, you must manually install the Tivoli Remote Execution Service on at least one Windows operating system in your environment before installing more managed nodes on Windows operating systems. These instructions are in “Using Tivoli Remote Execution Service” on page 341.

   If the managed node is running on a Windows operating system, install the Tivoli desktop. Refer to Chapter 8, “Tivoli Desktop for Windows,” on page 65 for instructions.
5. If you plan to use Tivoli Software Installation Service to deploy managed resources and Tivoli Enterprise software, install it on either the Tivoli server or a managed node.

   Refer to Chapter 9, “Introduction to Tivoli Software Installation Service,” on page 75 for information about where to install the components of Tivoli Software Installation Service. Refer to Chapter 10, “Installing Tivoli Software Installation Service,” on page 83 for the details about installing Tivoli Software Installation Service.
6. Create managed resources such as endpoints, managed nodes, gateways, and so forth, and install Tivoli Enterprise software.

   You can use Tivoli Software Installation Service, Tivoli Management Framework, or platform-specific installation mechanisms to add new managed resources and to install or upgrade Tivoli Enterprise software in your Tivoli environment.

Note: Before and after each major change to your Tivoli environment, you should back up your Tivoli object database to be able to return your Tivoli environment to a known working state. These instructions are in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*. Additionally, system backups of management servers should be performed, following your operating procedures.

Installing with Tivoli Software Installation Service

Tivoli Software Installation Service lets you plan and perform the installation of multiple products and patches on multiple machines and checks the success of each installation. It installs the software in the order required by their dependencies. The product stores installation images in a reusable, customizable, and easily maintained install repository.

Tivoli Software Installation Service provides both a command line and a console, a graphical user interface (GUI) known as the Tivoli Software Installation Service console. The commands enable you to automate installations using a response file, a text file that defines the installation to be performed. The console has an installation worksheet on which you plan an installation, and helps you visualize the dependencies between software components.

To use Tivoli Software Installation Service to install or upgrade Tivoli managed resources or software, perform the following high-level tasks:

1. Select the products to be installed from the install repository. If the products are not already in the install repository, import them.

   Note: Tivoli Software Installation Service considers most Tivoli managed resources to be product installation images. These images can be for managed resources, products, or patches.
2. Set the default installation options for each product.
3. Select the machines where the products are to be installed. If the machines are not defined, add them.
4. If necessary, modify the installation options of a product for a specific machine.
5. Start the installation.

Whether you are using the console or commands, the overall procedure remains the same. However, the individual procedures differ. For detailed information about using Tivoli Software Installation Service to install and upgrade Tivoli managed resources and products, refer to Chapter 12, "Using Tivoli Software Installation Service," on page 111 and Chapter 13, "Using response files," on page 143.

Installing with Tivoli Management Framework

Tivoli Management Framework uses a different procedure to create each Tivoli managed resource. Additionally, there are different procedures for installing and for upgrading Tivoli managed resources and products.

For example, the following list shows some of the procedures used to create Tivoli managed resources or to install Tivoli Enterprise products or patches using the installation mechanisms provided by Tivoli Management Framework:

- Create a managed node by using the **Create → Managed Node** option within a policy region from the Tivoli desktop or by using the **wclient** command.
- Install a Tivoli Enterprise product by using the **Install → Product** option of the Tivoli desktop or by using the **winstall** command.
- Install a Tivoli Enterprise patch by using the **Install → Patch** option of the Tivoli desktop or by using the **wpatch** command.

This list is not complete. Refer to Chapter 16, "Installing resources managed by Tivoli Management Framework," on page 197 and Chapter 17, "Installing Tivoli products and patches," on page 221 for detailed instructions.

Comparing installation mechanisms

You can install most Tivoli Enterprise software using either Tivoli Software Installation Service or Tivoli Management Framework. Some Tivoli Enterprise software can also be installed using platform-specific methods. This section helps you understand the differences between these mechanisms.

Tivoli Software Installation Service and Tivoli Management Framework both use the same images and communicate with the machines in the same way. However, Tivoli Software Installation Service checks additional dependencies and prerequisites before attempting the installation. These checks prevent problems that can occur when installing with Tivoli Management Framework.

Using Tivoli Management Framework, you can install only one component at a time, although it can be installed to multiple machines at the same time. For example, you can create one or more managed nodes, or you can install a specific product to one or more machines, or you can apply a specific patch to one or more machines. However, you cannot combine these tasks.

Using Tivoli Software Installation Service, you can install a different set of products on each machine and you can install multiple products, patches, and managed resources on a machine during a single installation. For example, you can use Tivoli Software Installation Service to create a fully functional gateway that uses the TCP/IP protocol on a new machine in your Tivoli region in a single installation step. However, Tivoli Management Framework installation process requires the following separate actions:

1. Install Tivoli Management Framework to create a managed node.
2. Define a gateway on the managed node.
3. Install one or more products, each of which might require a base component and a gateway component. Each of these components requires a separate installation.
4. If necessary, apply upgrades, each requiring a separate installation.
5. Optionally, create an endpoint.

The following sections describe the differences between using Tivoli Software Installation Service and Tivoli Management Framework to install the following:

- Managed nodes
- Endpoints
- Tivoli Enterprise products and patches

Installing managed nodes

Both Tivoli Management Framework and Tivoli Software Installation Service can install multiple managed nodes at one time. With Tivoli Management Framework, all the managed nodes in a single installation are created in the same policy region. With Tivoli Software Installation Service you can specify a different policy region for each managed node in an installation.

Installing endpoints

You can install an endpoint using Tivoli Software Installation Service, Tivoli Management Framework, or platform-specific installation tools.

Important differences include the following:

- Tivoli Software Installation Service checks additional dependencies and prerequisites before installing, which ensures that the endpoint can communicate with the gateway.
- After installing, Tivoli Software Installation Service checks that the endpoint and gateway connected properly.
- Tivoli Software Installation Service and Tivoli Management Framework both create an endpoint using an installation image that is maintained on the managed node that performs the installation. Upgrading Tivoli Management Framework on a managed node upgrades the endpoint installation image. In

contrast, platform-specific methods such as InstallShield use prepackaged images, which require you to keep track of the most current installation media.

- When installing on Windows operating systems, the default destination directory is %SystemRoot%\Program Files\Tivoli, while the default location when installing with the **winstlcf** command or Tivoli Software Installation Service is c:\Tivoli\lcf on Windows operating systems and /opt/Tivoli/lcf on UNIX operating systems.

Installing products and patches

You can install a Tivoli product or patch using Tivoli Software Installation Service or Tivoli Management Framework. In addition, some products have platform-specific installation methods such as InstallShield images, tar files, or zip files.

Using Tivoli Software Installation Service, you can install multiple products to a machine at the same time, and install different products to different machines in parallel. Using Tivoli Management Framework, you can install only one product to multiple machines at the same time.

A few products can be installed directly on an endpoint. For these products, Tivoli Software Installation Service and platform-specific installation tools are the only methods available. Tivoli Management Framework cannot install products on endpoints. For more information, refer to "Installing Tivoli Enterprise software on an endpoint" on page 78.

Packaging of installation images

The CD for each Tivoli Enterprise product has the following files, among others:

CONTENTS.LST
An ASCII file that contains a list of installation components for that Tivoli Enterprise product. Each component in this list has an associated index (.IND) file. You select these components for the initial installation of that Tivoli product.

PATCHES.LST
An ASCII file that contains a list of upgrade components for that Tivoli Enterprise product. Each component in this list has an associated index (.IND) file. You select these components to upgrade an existing product.

Index (.IND) Files
An ASCII file that contains the component-specific instructions for each installation image. These files specify the registered product tag, dependency statements, and the information required to install this component on each of its supported operating systems. For information about how to read index files, refer to Chapter 24, "Reading index files," on page 331.

Disk space requirements

The Tivoli binaries and libraries can require a large amount of disk space. You should ensure that there is enough disk space in the file systems where you are going to install the Tivoli Enterprise files. The Tivoli server or managed node installation allows you to customize the location of Tivoli Enterprise binaries, libraries, and other files.

The locations you specify during the installation of the Tivoli server or managed node are the locations where files are installed for all Tivoli Enterprise software. As each Tivoli Enterprise product is added to your Tivoli environment, additional disk space is required. Ensure that there is enough available disk space to allow for future installation of Tivoli Enterprise software.

Note: Tivoli strongly recommends that you do not share binaries or libraries across Tivoli region boundaries. Upgrading to future releases and installing service packs cannot be completed on a region-by-region basis when these files are shared.

Refer to the *Tivoli Management Framework Release Notes* for a complete listing of disk space requirements for Tivoli Management Framework and Tivoli Software Installation Service. Refer to the release notes for each Tivoli Enterprise product for its disk space requirements.

Chapter 2. Internationalization

This chapter describes the internationalization features for a Tivoli server and how to use them. It covers the following topics:

- Enabling language support
- Text encoding or code set support
- Installing and configuring an non-English Tivoli server
- Product notes for internationalization

Enabling language support

Tivoli Enterprise software is translated into the following languages:

- Brazilian Portuguese
- Chinese (Simplified)
- Chinese (Traditional)
- French
- German
- Italian
- Japanese
- Korean
- Spanish

The translations for these languages are provided as language packages on the language support CD for each product. After you install or upgrade a Tivoli Enterprise product in your Tivoli region, you optionally install the language support for that product. If you do not install the language support after installing, the associated product displays all text in English. Each language is a separately installable product installation image.

If language support for a product is installed and you upgrade the product, you must install the corresponding language support product, if one exists. Refer to the upgrade documentation for the specific product to determine if language support is required. If you do not install the language support after upgrading, the associated product can display some fields and messages in English.

Language packages are installed in the same manner as Tivoli Enterprise products. For detailed instructions on installing a language support product, refer to Chapter 17, “Installing Tivoli products and patches,” on page 221.

Note: For OS/400 endpoints, you enable language support during installation. For details, refer to “Installing internationalization support” on page 356.

Locale environment variables

As with most current operating systems, localized behavior is obtained by specifying the desired locale. For Tivoli Enterprise software, you set the LANG environment variable to the desired locale name as specified by POSIX, X/Open, or other open systems standards.

Note: If you are in a Microsoft Windows environment, you can alternatively modify the language setting in the Regional Settings of the Control Panel.

If you specify the LANG environment variable and modify the regional settings, the LANG environment variable overrides this regional setting.

As specified by open systems standards, other environment variables will override LANG for some or all locale categories. These variables include the following:
- LC_CTYPE
- LC_TIME
- LC_NUMERIC
- LC_MONETARY
- LC_COLLATE
- LC_MESSAGES
- LC_ALL

If any of the previous variables are set, you must remove their setting for the LANG variable to have full effect.

Locale names

Locale names are generally based on the ISO 639 two-letter language codes and the ISO 3166 two-letter territory codes. The syntax for the value of LANG is as follows:

```
ll[_TT][.CodeSet][@Modifier]
```

where:

ll Specifies the two-letter language code.

TT Specifies the optional two-letter territory or country code.

CodeSet
Specifies the optional name of text encoding.

Modifier
Specifies an optional modifier.

For example:
- `fr` is the locale name for French.
- `fr_FR` is the locale name for French in France.
- `fr_FR.ISO8859-1` is the locale name for French in France using ISO8859-1 text encoding.

All implementations support a `C` locale.

LANG variable and UNIX operating systems

Most UNIX systems use the LANG variable to specify the desired locale. Different UNIX operating systems, however, require different locale names to specify the same language. Be sure to use a value for LANG that is supported by the UNIX operating system that you are using.

You can obtain the locale names for your UNIX operating system by running the following command:

```
locale -a
```

The following table shows examples of the locale names that are supported by Solaris, HP-UX, and IBM® AIX® operating systems.

Language	**Solaris**	**HP-UX**	**AIX**
Brazilian Portuguese	pt_BR	pt_PT.iso88591	pt_BR

Language	Solaris	HP-UX	AIX
Chinese, Simplified	zh	zh_CN.eucCN	zh_CN
Chinese, Traditional	zh_TW	zh_TW.eucTW	zh_TW
English in C locale	C	C	C
French	fr	fr_FR.iso88591	fr_FR
German	de	de_DE.iso88591	de_DE
Italian	it	it_IT.iso88591	it_IT
Japanese	ja	ja_JP.eucJP	ja_JP
Korean	ko	ko_KR.eucKR	ko_KR
Brazilian Portuguese	pt_BR	pt_PT.iso88591	pt_BR

LANG variable and Windows operating systems

Most PC operating systems do not use the LANG variable. Tivoli Enterprise software, however, can use LANG to determine the desired language even on PC systems. On PC systems, LANG should be set to the canonical locale name based on the ISO language or territory codes without a code set suffix. For example:

- `fr` is the locale for standard French
- `ja` is the locale for Japanese
- `pt_BR` is the locale for Brazilian Portuguese
- `C` is the locale for English in C locale

On Windows operating systems, if LANG is not set, Tivoli Management Framework uses the current selection in the Regional Settings object of the Windows Control Panel.

Using locale variants

Although Tivoli Enterprise software currently provides only one translated version for each language, you can use a preferred locale variant, and Tivoli Management Framework finds the corresponding language translation. For example, Tivoli Management Framework provides one translation for French, but each of the following locale settings finds the appropriate translation:

- `fr` is the locale name for standard French
- `fr_FR` is the locale name for French in France
- `fr_CA` is the locale name for French in Canada
- `fr_CH` is the locale name for French in Switzerland

Message catalogs

Tivoli Enterprise message catalogs are installed in a location specified at installation time. Message catalogs are typically installed under a top-level \msg_cat directory, and each of these message catalogs is installed under a language-specific subdirectory. For example:

.../msg_cat/de
German message catalogs

.../msg_cat/ko
Korean message catalogs

.../msg_cat/C
English message catalogs

Tivoli Management Framework recognizes variations in UNIX locale names and is usually able to map the specified value to the appropriate message catalog.

The NLSPATH variable is used to find the appropriate message catalog directory, as specified by open systems standards. For example, if the message catalogs are in /usr/local/Tivoli/msg_cat directory, the NLSPATH variable should begin with the following:

```
/usr/local/Tivoli/msg_cat/%L/%N.cat:
```

The `%L` directive is expanded to the message catalog directory that most closely matches the current user language selection, and `%N.cat` expands to the desired message catalog.

If a message catalog is not found for the desired language, the English C message catalogs are used.

For example, suppose you specify the AIX locale for German in Switzerland as follows:

```
LANG=De_CH.IBM-850
```

The `%L` directive is expanded in the following order to locate the specified locale:

1. `de_CH`
2. `de`
3. `C`

Because Tivoli Management Framework does not provide a German in Switzerland language package, `de_CH` will not be found. If the Tivoli Management Framework German language package is installed, `de` is used. Otherwise, the default locale `C` is used, causing text to be displayed in English.

Text encoding (code set) support

Different operating systems often encode text in different ways. For example, PC operating systems use SJIS (code page 932) for Japanese text, but UNIX operating systems often use eucJP.

In addition, multiple locales can be provided for the same language so that different code sets can be used for the same language on the same machine. This can cause problems when text is moved from system to system or between different locale environments.

Tivoli Management Framework addresses these problems by using Unicode and UTF-8 (the multi-byte form of Unicode) as the internal canonical representation for text.

Message catalogs are encoded using UTF-8, and the text is converted to the locale encoding before being presented to the user. In this way, the same French message catalog files can be used to support a variety of Latin 1 code sets, such as ISO8859-1, Microsoft 1252, IBM PC 850, and IBM MVS™ 1047.

UTF-8 is also used to achieve text interoperability across the Tivoli environment. For example, Common Object Request Broker Architecture (CORBA) strings are transmitted as UTF-8 within the Tivoli environment. This enables remote management within a heterogeneous network in which local text encoding can vary. For example, Japanese file names can be manipulated on Japanese PC endpoints from a desktop executing in the UNIX Japanese EUC locale.

Text interoperability across the Tivoli environment is also achieved by storing strings as UTF-8 within the Tivoli object database. Strings are converted to the local encoding for viewing and manipulation by applications that are executing on different operating system code sets.

Location of code set files

Interoperability across your Tivoli environment depends on code set files, which are used to perform UTF-8 conversion and other types of encoding-specific text processing. These files are installed in the generic/codeset subdirectory under the directory specified for binary files during installation.

For example, if binaries are installed in /usr/local/Tivoli/bin, the code set files are in /usr/local/Tivoli/bin/generic/codeset directory. The TISDIR variable, which points to the directory that contains the code set directory, is used to find the files. The value for the TISDIR variable in this example is /usr/local/Tivoli/bin/generic.

Code set files provided

Tivoli Management Framework provides the following code set files for Solaris, HP-UX, and AIX systems.

Language Code Set	Solaris	HP-UX	AIX
Latin 1 (Western European)	ISO98591	ISO88591 ROMAN8	ISO88591 850
Latin 2 (Eastern European)	ISO88592	ISO88592	ISO88592
Turkish	ISO88599	ISO88599	ISO88599
Latin 4 (Baltic)	ISO88594		921
Estonian	ISO88591		922
Greek	ISO88597	ISO88597	ISO88597
Cyrillic	ISO88595	ISO88595	ISO88595
Arabic†	ISO88596	ISO88596 ARABIC8‡	ISO88596 1046
Hebrew†	ISO88598	ISO88598	ISO88598 856
Simplified Chinese	EUCCN GB2312‡	EUCCN HP15CN‡	EUCCN 936 GBK‡
Traditional Chinese	EUCTW CNS11643‡ 950 BIG5‡	EUCTW 950 BIG5‡	EUCTW 950 BIG5‡
Japanese	EUCJP 932 SJIS‡ PCK‡	EUCJP 932 SJIS‡	EUCJP 932
Korean	EUCKR	EUCKR 5601‡	EUCKR

Language Code Set	Solaris	HP-UX	AIX
Thai†	874 TIS620‡	874 TIS620‡	874 TIS620‡
† The code set is available, but the product does not fully support it. ‡ Denotes an alias for real table.			

Tivoli Management Framework provides the following code set files for Windows, OS/2, NetWare, and EBCDIC systems.

Language Code Set	Windows	OS/2, NetWare	EBCDIC
Latin 1 (Western European)	1252	ISO88591 437 819† 850 860 861 863 865	37 273 274 277 278 280 282 284 285 297 500 871 1047
Latin 2 (Eastern European)	1250	ISO88592 852 912†	870
Turkish	1254	ISO88599 857 920†	1026
Latin 4 (Baltic)	1257	ISO88594 775 914† 921 1117	1112 1122
Estonian		922 1116	
Greek	1253	ISO88597 855 866 878 915†	875
Cyrillic	1251	ISO88595 855 866 878 915†	1025
Arabic‡	1256	ISO88596 864 1046 1089†	
Hebrew‡	1255	ISO88598 856 862 916†	424

Language Code Set	Windows	OS/2, NetWare	EBCDIC
Chinese, Simplified	936	936 1381[†] 1386[†]	935 1388
Chinese, Traditional	950	950	937
Japanese	932	932 942[†] 943[†]	939
Korean	949	949 1363[†]	933
Thai[‡]	874	838	
[†] The code set is available, but the product does not fully support it. [‡] Denotes an alias for real table.			

Endpoints and code set tables

When installed, endpoints do not contain code set tables. However, when the endpoint logs in to the gateway, the appropriate code set is determined by the locale in which the endpoint is running, and the appropriate code set file is downloaded.

At login, the endpoint checks to determine if it has its code set file; if it does not, the gateway sends it. The file is placed in the TISDIR/codeset subdirectory in the endpoint tree. TISDIR has the same value as LCF_DATDIR. For example, if LCF_DATDIR is /usr/local/Tivoli/dat/1, the code set files is in /usr/local/Tivoli/dat/1/codeset directory.

Installing a non-English Tivoli region

This section contains information about how to install and configure your Tivoli environment for languages other than English.

Language support for this release is delivered as products. If you do not install the language support after installing a product, all window text and error messages are in English. If you do not install the language support after upgrading, a previous release of the language support can be in use, which might cause problems.

Software that processes non-English text must use a code set that contains all of the characters in that text. The code set that a process uses is determined by its language. Therefore, it is critical that all Tivoli Enterprise components run in a language that supports all the data they need to process.

Before you begin

A Tivoli region does not fully support operating in more than one language. Although Tivoli Management Framework does provide support for generating informational, warning, or error messages in the supported language of the locale, converting program or user-generated messages from one language to another is not supported. However, data from clients running in multiple languages can be handled by the Tivoli server without data loss by running the Tivoli server in a UTF-8 locale (including an English locale).

Non-English characters in host names and host labels are not supported in this release. For some locales, using non-English characters for the following can cause problems:

- User and group names
- Passwords
- File, directory, and object names

As a workaround, avoid using non-English characters if you have a problem in one of these areas.

Setting the LANG variable

Where possible, set your entire operating system to run in the desired language. For example, in a Tivoli region that must process Japanese text, all Tivoli processes on Solaris should be initiated with a LANG value of `ja`, and Windows nodes should have Japanese selected as the system language.

Configuring a system so that all processes are started in the correct locale varies by operating system. Consult your operating system documentation for information about how to set the LANG variable for your system. For example, on Solaris, you can add the desired LANG setting to the /etc/default/init file (for example, `LANG=ja`). On Windows operating systems, you can select the appropriate locale in the Regional Settings object of the Control Panel. This applies to the Tivoli server, managed nodes, and endpoints.

On OS/2 operating systems, set LANG by modifying the **autoexec.bat** file to include the following line:

```
SET LANG=locale_name
```

where *locale_name* is the canonical locale name. Restart the operating system after you make this change.

Installing on Windows operating systems

When installing on Windows operating systems, you must install kbdus.dll before installing any Tivoli Enterprise software. This library file is normally installed on English Windows operating systems. It is available through Microsoft or on the Windows CD.

Install kbdus.dll on the system drive in the %SystemRoot%\system32 directory.

The requirement to install kbdus.dll stems from a Microsoft limitation. The Microsoft Incident number is SRZ980727000321.

Mixed-language text during installation

If you are using a language other than English, be aware that some text is in English during the installation process; after installation, you will not see such mixed-language text.

Setting locale on Tivoli server and managed nodes

Immediately after installing either a Tivoli server or managed node, you must set the LC_ALL environment variable to a value supported by that operating system. You must do this for the Tivoli server and for each managed node.

Use the **wsetlang** command to set the LC_ALL environment variable. For more information, refer to the **wsetlang** command in the *Tivoli Management Framework Reference Manual*.

The **wsetlang** command maps standard locale names, such as `fr`, `ja`, `pt_BR`, and `en`, to a value appropriate for the local system. The **wsetlang** command can be included in a task that runs on all managed nodes. This updates the environment inherited by methods.

For example, to set a managed node to run in Brazilian Portuguese, enter the following command:

```
wsetlang -o -l pt_BR
```

To set a managed node to run in Japanese, enter the following command:

```
wsetlang -o -l ja
```

Shut down and restart the Tivoli server or managed node with the following commands:

odadmin shutdown [*object_dispatch* | **clients** | **all**]
odadmin start [*object_dispatch* | **clients** | **all**]

For more information about the **wsetlang** and **odadmin** commands, refer to the *Tivoli Management Framework Reference Manual*.

Setting locale on endpoints

On UNIX endpoints, you can set the LC_ALL variable when you install the endpoint with the **–C** option of the **winstlcf** command:

```
winstlcf -C ja other_options
```

If LC_ALL was not specified when the endpoint was installed, it can be set as follows:

1. Set up the endpoint command line environment:

   ```
   . /etc/Tivoli/lcf/N/lcf_env.sh
   ```

   where *N* is the number of this endpoint instance.
2. Stop the endpoint:

   ```
   $LCF_DATDIR/lcfd.sh stop
   ```
3. Add the following lines to lcfd.sh, immediately after the other environment settings:

   ```
   LC_ALL=value
   export LC_ALL
   ```
4. Restart the endpoint:

   ```
   $LCF_DATDIR/lcfd.sh
   ```

Setting environment variables for methods

The NLSPATH and TISDIR variables must be present in the environment inherited by methods for all operating systems. They should be automatically set when managed nodes are installed.

On managed nodes, you can verify these settings by entering the following command:

```
odadmin environ get > file_name
```

You can modify them by editing *file_name* and then entering the following command:

```
odadmin environ set < file_name
```

For additional information about the **odadmin environ** command, refer to the *Tivoli Management Framework Reference Manual*.

Using non-English text on Solaris Open Windows

Use one of the following methods to display non-English text on Solaris Open Windows:

- Specify a font list on the Tivoli desktop command line, as shown in the following example:

  ```
  tivoli -xrm '*fontList:-dt-interface \
  user-medium-r-normal-s*-*-*-*:'
  ```

- Add the font list to the X Window System resource file by performing the following steps:
  1. Edit the app-defaults/Tivoli file and add the following line at the end of the file:

     ```
     *fontList:-dt-interface user-medium-r-normal-s*-*-*-*:
     ```

  2. Create a subdirectory under the app-defaults directory with the same name as the current LANG variable value.
  3. Copy the file into this subdirectory.
  4. Start the Tivoli desktop by entering the following command:

     ```
     tivoli
     ```

Product notes for internationalization

Consider the following important information before using an internationalized version of Tivoli Management Framework.

AIX 4.3 libraries

In the AIX 4.3 release, there is a problem with the X Window System libraries that only occurs in non-English environments. To start the Tivoli desktop, you must perform the following steps:

1. Change to the /etc/Tivoli directory.
2. Run the setup_env.sh script.
3. From the command line, enter the following command:

   ```
   export LIBPATH=/usr/lpp/X11/lib/R5:$LIBPATH
   ```

4. Start the Tivoli desktop by entering the following command:

   ```
   tivoli
   ```

You can make this change permanent by adding the LIBPATH variable to the following setup files:

- /etc/Tivoli/setup_env.sh
- /etc/Tivoli/setup_env.csh
- /etc/Tivoli/oserv.rc

Command line output on Windows operating systems

In Western European languages, the translated output of Tivoli commands on Windows operating systems is encoded using the Windows 1252 code page. To view this text correctly in a DOS command window, perform the following steps:

1. Enter the following command:
   ```
   chcp 1252
   ```
2. Set the font for that window to Lucida Console.

Web browser support

To view the translated versions of most Web pages used by Tivoli Enterprise products, you must use a Web browser that supports Unicode UTF-8 encoding. You must also configure your browser to use an appropriate font for the Unicode character set.

Setting the LANG variable for DB2

If you are using DB2® with Tivoli Enterprise products, you must set the LANG variable to the same value as the database territory field in the DB2 configuration. Use the following command to obtain the value of the database territory field:

```
db2 get database configuration for database_name | more
```

If you cannot set the LANG variable to the same value as the database territory field, set the RIM_NLS_LANG variable to the same value as the database territory field.

Set the variable (LANG or RIM_NLS_LANG) before starting the object dispatcher or by using the **odadmin environ set** command and then running the **odadmin reexec** command. For additional information about the **odadmin** command, refer to the *Tivoli Management Framework Reference Manual*.

Setting the NLS_LANG variable for Oracle

If you are using Oracle with Tivoli Enterprise products, you must set the NLS_LANG variable to the code page specified when you installed Oracle. For example, if you installed Oracle in a Japanese environment with the code page Japanese_Japan.JA16SJIS, you would set the NLS_LANG variable to Japanese_Japan.JA16SJIS in the object dispatcher environment.

To set the NLS_LANG variable in the object dispatcher environment, perform the following steps:

1. Run the following command to create a file that contains the object dispatcher environment variables:
   ```
   odadmin environ get > file_name
   ```
2. Add the following line to the *file_name* file:
   ```
   NLS_LANG=Oracle_code_page
   ```
3. Run the following command to set the object dispatcher environment with the new variables:
   ```
   odadmin environ set < file_name
   ```

Part 2. Tivoli server installation

Chapter 3. Installing a Tivoli server

This chapter contains instructions for installing a Tivoli server on a UNIX operating system or Windows operating system in your distributed network. Installing a Tivoli server is the first step in creating your Tivoli environment. Depending on whether you plan to use a UNIX or Windows Tivoli server, there are different installation procedures.

Note: You cannot use Tivoli Software Installation Service to install a Tivoli server, but you can use Tivoli Software Installation Service to upgrade a Tivoli server. For information about upgrading a Tivoli server, refer to "Upgrading Tivoli Management Framework" on page 234.

Overview of installing a Windows Tivoli server

To install a Tivoli server on a Windows operating system, you need to perform the following steps:

1. Install the Tivoli server as described in "Installing on a Windows operating system" on page 24.
2. If you plan to use the Tivoli desktop from this machine, install the Tivoli desktop as described in Chapter 8, "Tivoli Desktop for Windows," on page 65.

   Even if you do not install the Tivoli desktop, you can still use Tivoli commands on this machine after initializing the Tivoli environment variables. For additional information about initializing the environment variables, refer to "Setting Tivoli environment variables" on page 30.
3. Configure the Simple Mail Transfer Protocol (SMTP) service as described in Chapter 7, "Configuring for SMTP e-mail," on page 61.
4. Optionally, make a copy of the Tivoli Management Framework 1 of 2 CD as described in "Copying installation images" on page 221.
5. Perform a backup.

If you are upgrading your Tivoli server, you can use any of the installation mechanisms described in this guide. Upgrading a Tivoli server is the same as upgrading any Tivoli Enterprise product. For more information, refer to "Upgrading Tivoli Management Framework" on page 234.

Overview of installing a UNIX Tivoli server

To install a Tivoli server on a UNIX operating system, you need to perform the following steps:

1. Install the Tivoli server as described in "Installing on a UNIX operating system" on page 26.
2. Optionally, make a copy of the Tivoli Management Framework 1 of 2 CD as described in "Copying installation images" on page 221.
3. Perform a backup.

If you are upgrading your Tivoli server, you can use any of the installation mechanisms described in this guide. Upgrading a Tivoli server is the same as upgrading any Tivoli Enterprise product. For more information, refer to "Upgrading Tivoli products" on page 235.

Planning considerations

Before installing a Tivoli server, consider the following:

- The names of the managed nodes must be in the /etc/hosts file (UNIX operating system), LMHOSTS file (Windows operating system), the Network Information Services (NIS) host map, or the name server.
- All Tivoli administrators must have read access to the Tivoli directories.
- When installing a UNIX Tivoli server, the installation path that you specify cannot contain spaces.
- If you are using Dynamic Host Configuration Protocol (DHCP) on your Windows managed nodes, the Tivoli server must be on a Window operating system.

  Note: You can have DHCP managed nodes and a UNIX Tivoli server if you reserve a group of static IP addresses for these systems when they connect. However, these managed nodes cannot be gateways.
- If your Tivoli environment will contain Windows managed nodes, you must install Tivoli Remote Execution Service on at least one Windows operating system per domain. If your Tivoli server is running a Windows operating system, this service is installed automatically. For complete information and instructions, refer to "Using Tivoli Remote Execution Service" on page 341.

Installing on a Windows operating system

The Tivoli server must be installed before you can install any Tivoli managed resources.

Note: On Windows operating systems, the Tivoli graphical user interface (GUI) is provided by Tivoli Desktop for Windows. This product is included on the Tivoli Management Framework 1 of 2 CD and must be installed separately. You can install it before or after installing the Tivoli server. For complete installation instructions, refer to Chapter 8, "Tivoli Desktop for Windows," on page 65.

The following table provides the context and authorization role required for this task.

Activity	Context	Required role
Install a Tivoli server	Machine	Member of the Administrators group

Your login account can be any name, but the account must be a member of the Windows Administrators group. To install the Tivoli server, you must have full privileges for administering the system.

Note: If you change the account under which you installed the Tivoli server, you must use the **widmap** command to add the new account to root_user login map. Refer to "Accounts and user login maps" on page 344 for details.

To install Tivoli Management Framework on a Windows operating system, perform the following steps:

1. Insert the Tivoli Management Framework 1 of 2 CD into the CD-ROM drive or map the CD from a drive on a remote system.

2. From the taskbar, click **Start**, and then select **Run** to display the Run window.
3. In the **Open** field, type `x:\setup` where *x* is the CD-ROM drive or the mapped drive. The Welcome window is displayed.
4. Click **Next**. The License Agreement window is displayed.
5. Read the license agreement and click **Yes**. The Accounts and File Permissions window is displayed.
6. Click **Next**. The Installation Password window is displayed.
7. In the Installation Password window, perform the following steps:
   a. In the **Password** field, type an installation password, if desired. If you specify a password, this password must be used to install managed nodes, to create interregion connections, and to perform any installation using Tivoli Software Installation Service.

      Note: During installation the specified password becomes the installation and the region password. To change the installation password, use the **odadmin region set_install_pw** command. To change the region password, use the **odadmin region set_region_pw** command. If you change one of these passwords, the other password is not automatically changed.

   b. Click **Next**. The Remote Access Account window is displayed.
8. In the Remote Access Account window, perform the following steps:
   a. Type the Tivoli remote access account name and password through which Tivoli programs will access remote file systems. If you do not specify an account name and password and you use remote file systems, Tivoli programs will not be able to access these remote file systems. For additional information, refer to "Accessing remote systems" on page 340.

      Note: If you are using remote file systems, the password must be at least one character. If the password is null, the object database is created, but you cannot start the object dispatcher (the oserv service).

   b. Click **Next**. The Setup Type window is displayed.
9. In the Setup Type window, do the following:
   a. Select one of the following setup types:

      Typical
      Installs the Tivoli Management Framework product and its documentation library.

      Compact
      Installs only the Tivoli Management Framework product.

      Custom
      Installs the Tivoli Management Framework components that you select.

   b. Accept the default destination directory or click **Browse** to select a path to another directory on the *local* system.

      Note: Do not install on remote file systems or share Tivoli Management Framework files among systems in a Tivoli environment.

   c. Click **Next**. If you selected the **Custom** option, the Select Components window is displayed. If you selected **Compact** or **Typical**, go to step 11.
10. (Custom setup only) In the Select Components window, do the following:

    a. Select the components to install. From this window you can preview the disk space required by each component as well as change the destination directory.
    b. If desired, click **Browse** to change the destination directory.
    c. Click **Next**. The Choose Database Directory window is displayed.
11. In the Choose Database Directory window, do the following:
    a. Accept the default destination directory or click **Browse** to select a path to another directory on the *local* system.
    b. Click **Next**. The Enter License Key window is displayed.
12. In the Enter License Key window, click **Next**. The Start Copying Files window is displayed.

    Note: You do not need to provide a license key. This value is no longer used.
13. Click **Next**. The Setup Status window is displayed.
14. After installing the Tivoli Management Framework files, the setup program initializes the Tivoli object dispatcher server database. When the initialization is complete, you are prompted to press any key to continue.
15. If this is the first time you installed Tivoli Management Framework on this system, you are prompted to restart the machine.

    Note: Rebooting the system loads the TivoliAP.dll file.
16. After the installation completes, configure the Windows operating system for SMTP e-mail. From a command line prompt, enter the following commands:

    ```
    %SystemRoot%\system32\drivers\etc\tivoli\setup_env.cmd
    bash
    wmailhost hostname
    ```

    where *hostname* is the name of the network mail server. For more information, refer to Chapter 7, “Configuring for SMTP e-mail,” on page 61.

Installing on a UNIX operating system

The Tivoli server must be installed before you can install any Tivoli managed resources. You can install a Tivoli server on a UNIX operating system in one of the following ways:

- Using an X Window System environment, such as OpenLook, Motif, or the Common Desktop Environment

  Use this method for attended installations. A series of windows are displayed. You can accept the defaults or override them.
- Using only the UNIX command line

  Use this method if you want to use a script to install the Tivoli server. With this approach, you must enter the names of the installation variables correctly. The installation does not generate error messages if you specify invalid variable names.

Note: During the installation of a Tivoli server, the Tivoli desktop is displayed automatically. If you inadvertently close the Tivoli desktop, use the **tivoli** command to reopen it.

The following table provides the context and authorization role required for this task:

Activity	Context	Required Role
Install a Tivoli server on UNIX	Machine	root access

Installing in an X Window System Environment

To install Tivoli Management Framework on a UNIX Tivoli server while in an X Window System environment, perform the following steps:

1. Log in as root on the system that is to be the Tivoli server.
2. Ensure that the DISPLAY environment variable is set to display correctly to the terminal, and that the DOGUI environment variable is not set.
3. Ensure that adequate disk space is available in each file system where the files will be installed (refer to the *Tivoli Management Framework Release Notes* for disk space requirements). Directories are specified in the Install Tivoli Server window as part of step 10.

   Note: Disk space requirements are checked by the Tivoli installation. If the disk space is inadequate, messages are posted in the installation status window.
4. If required by your operating system, mount the Tivoli Management Framework 1 of 2 CD using the appropriate UNIX command. Refer to your operating system documentation for instructions on how to mount devices and how to determine the device name. In this example, the CD is mounted at /cdrom/cdrom0.
5. Create an installation directory (for example, /usr/local/Tivoli/install_dir) to contain the installation utilities and associated files. In this example, you would enter the following command:

   ```
   mkdir /usr/local/Tivoli/install_dir
   ```
6. Change to the installation directory. In this example, you would enter the following command:

   ```
   cd /usr/local/Tivoli/install_dir
   ```
7. Set the file creation mask to 022 using the following command:

   ```
   umask 022
   ```
8. Run the **WPREINST.SH** script that is located on the CD. This script creates links to the files used during the installation process. In this example, the following command is used:

   ```
   /cdrom/cdrom0/WPREINST.SH
   ```
9. Begin the installation using the command displayed at the end of the preinstallation script output. In this example, the following command is used:

   ```
   ./wserver -c /cdrom/cdrom0
   ```

   Two windows named Install Tivoli Server are displayed.
10. Complete the first Install Tivoli Server window.

    In this window, do the following:

    a. In the Specify Directory Locations section, specify the paths to the directories where the libraries, binaries, database, manual pages, X Window System resource files, and message catalogs are to be installed, or accept the defaults.

       Note: If you need to reinstall the Tivoli server, add an exclamation mark (!) at the end of the directory paths. (You can also replace a path name with the exclamation mark.) The exclamation mark forces

Tivoli to reinstall the directories if they already exist. You *must* overwrite the database, or the reinstalled Tivoli server will not work properly.

b. Set When installing, create "Specified Directories" if missing to create the specified directories if they do not already exist.

c. Set Arrange for start of the Tivoli daemon at system (re)boot time to have the object dispatcher (oserv) start when the system is started. The installation process updates the appropriate rc files.

d. Set Configure remote start capability of the Tivoli daemon to have the Tivoli installation change the /etc/inetd.conf file and add the Tivoli port number to the /etc/services file. This enables you to start the oserv from a remote managed node.

e. Click **Set** to set the options you specified.

11. Complete the second Install Tivoli Server window.

    In this window, do the following:

    a. Do not type anything in the **License Key** field. This value is no longer used.

    b. Choose an encryption level, **None**, **Simple**, or **DES**, from the Encryption Level options.

       Simple encryption is recommended, but DES encryption provides the highest level of security for network traffic. For a detailed explanation of the encryption levels, refer to the section on security and encryption levels in the *Tivoli Management Framework Planning for Deployment Guide*.

    c. Type an installation password in the Installation Password field, if desired. If you specify a password, this password must be used to install managed nodes, to create interregion connections, and to perform any installation using Tivoli Software Installation Service.

       Note: During installation the specified password becomes the installation and the region password. To change the installation password, use the **odadmin region set_install_pw** command. To change the region password, use the **odadmin region set_region_pw** command. If you change one of these passwords, the other password is not automatically changed.

    d. Type the name of the initial policy region in the **Region Name** field. This policy region will appear on the Tivoli desktop of the root administrator.

    e. The **TMR Server Name** field contains the host name of the UNIX machine. To perform a remote installation, type the host name of the remote machine. To perform a remote installation, you must have the appropriate trusted host access. Add an entry for your local system in the /.rhosts file of the remote system.

       Note: The login shell of the user must be Bourne or Korn shell.

    f. If you want to display or change the installation options, click **Install Options**. Refer to step 10 for instructions on specifying these options.

    g. Click **Install & Close** to begin the installation and close the Install Tivoli Server window when the installation is complete.

       The installation process prompts you with a TME® Install window. It provides the list of operations that take place during the installation and warns you of any problems that you might want to correct before installing this product.

12. Click **Continue Install** to continue the installation process. Another TME Install status window presents status information as the installation proceeds.

    The Tivoli desktop is automatically displayed before the installation is complete. Do not use it until the installation is complete.
13. Click **OK** when the TME Install status window indicates that the installation is complete.
14. When the installation is complete, remove the installation directory that was created in step 5. In this example, you would enter the following command:

    ```
    rm -r /usr/local/Tivoli/install_dir
    ```

Installing from the Command Line

To install the Tivoli server on a UNIX operating system using the UNIX command line, perform the following steps:

1. Log in as root on the system that is to be the Tivoli server.
2. Set the DOGUI environment variable to `no`, using the following commands:

   ```
   DOGUI=no
   export DOGUI
   ```
3. Ensure that adequate disk space is available in each directory where the files will be installed (refer to the *Tivoli Management Framework Release Notes* for disk space requirements). Directories are specified as options to the **wserver** command in step 9.

   Note: Disk space requirements are checked by the Tivoli installation. If the disk space is inadequate, messages are posted.
4. If required by your operating system, mount the Tivoli Management Framework 1 of 2 CD using the appropriate UNIX command. Refer to your operating system documentation for instructions on how to mount devices and how to determine the device name. In this example, the CD is mounted at /cdrom/cdrom0.
5. Create an installation directory (for example, /usr/local/Tivoli/install_dir) to contain the installation utilities and associated files. In this example, you would enter the following command:

   ```
   mkdir /usr/local/Tivoli/install_dir
   ```
6. Change to the installation directory. In this example, you would enter the following command:

   ```
   cd /usr/local/Tivoli/install_dir
   ```
7. Set the file creation mask to 022 using the following command:

   ```
   umask 022
   ```
8. Run the **WPREINST.SH** script from the Tivoli Management Framework 1 of 2 CD. This script creates links to the files used during the installation. In this example, you would enter the following command:

   ```
   /cdrom/cdrom0/WPREINST.SH
   ```
9. Begin the installation. The following example installs the Tivoli server on the local machine with the following characteristics:
   - `/cdrom/cdrom0`—The path to the CD-ROM image.
   - `/Tivoli/bin`—Where the binaries are installed.
   - `/Tivoli/lib`—Where the libraries are installed.
   - `/Tivoli/database`—Where the database is installed.
   - `/Tivoli/man`—Where the manual pages are installed.

- /Tivoli/X11—Where the X Window System application defaults are installed.
- /Tivoli/cat—Where the message catalogs are installed.
- NoonTide-Region—The name of the root policy region.
- AutoStart=1—The object dispatcher (the oserv service) is started at system boot time.
- SetPort=1—The remote start capability of the oserv is configured.
- CreatePath=1—Any specified directory that does not exist will be created.
- IP=Tivoli4Ever—The installation password.

```
./wserver -c /cdrom/cdrom0 BIN=/Tivoli/bin \
LIB=/Tivoli/lib ALIDB=/Tivoli/database \
MAN=/Tivoli/man APPD=/Tivoli/X11 CAT=/Tivoli/cat \
RN=NoonTide-Region AutoStart=1 \
SetPort=1 CreatePaths=1 IP=Tivoli4Ever
```

Note: If you need to reinstall the Tivoli server, add an exclamation mark (!) at the end of the directory paths. (You can also replace a path name with the exclamation mark.) The exclamation mark forces Tivoli to reinstall the directories if they already exist. You must overwrite the database, or the reinstalled Tivoli server will not work properly. The following command line reinstalls the Tivoli server installed with the preceding command, overwriting each directory:

```
./wserver -c /cdrom/cdrom0 BIN=! LIB=! \
ALIDB=! MAN=! APPD=! CAT=!  \
RN=NoonTide-Region AutoStart=1 SetPort=1 \
CreatePaths=1 IP=Tivoli4Ever
```

10. The command displays a list of actions that will take place during the installation and issues a confirmation prompt. To continue the installation process, type y and press Enter.

    Status information displays on your terminal as the installation proceeds.
11. If your DISPLAY environment variable is set, the Tivoli desktop is displayed partway through the installation process. Do not use it until the installation is complete. If the DISPLAY environment variable is not set, you can start the Tivoli desktop by performing the following steps:
    a. Set your DISPLAY environment variable.
    b. Initialize the Tivoli environment variables as described in “Setting Tivoli environment variables.”
    c. Run the **tivoli** command.

       For additional information about the **tivoli** command, refer to the *Tivoli Management Framework Reference Manual*.
12. When the installation of the Tivoli server is complete, remove the installation directory.

For additional information about the **wserver** command, refer to the *Tivoli Management Framework Reference Manual*.

Setting Tivoli environment variables

Before you can use the Tivoli desktop or commands, you must set up the Tivoli environment variables. You can manually run one of the scripts provided by Tivoli Management Framework or modify your initialization environment (UNIX operating systems only).

Setting environment variables for UNIX servers

For UNIX operating systems, the installation process creates the following setup scripts:

/etc/Tivoli/setup_env.csh

/etc/Tivoli/setup_env.sh

To set the Tivoli variables on a UNIX operating system, perform the following steps:

1. Log in to a UNIX Tivoli server or managed node either locally or using telnet.
2. For the Bourne (sh) or Korn (ksh) shells, enter the following command:

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   For the C (csh) shell, enter the following command:

   ```
   source /etc/Tivoli/setup_env.csh
   ```

Optionally, you can change your login initialization procedure to use the appropriate setup file so that the necessary environment variables and search paths are automatically set when you log in to the Tivoli server or managed node.

For example, you can add the following to your initialization procedure:

For sh or ksh shells:

```
if [ -f /etc/Tivoli/setup_env.sh ]; then
   . /etc/Tivoli/setup_env.sh
fi
```

For the csh shell:

```
if ( -f /etc/Tivoli/setup_env.csh ) then
   source /etc/Tivoli/setup_env.csh
endif
```

Setting environment variables for Windows servers

For Windows operating systems, the installation process creates the following setup scripts:

- %SystemRoot%\system32\drivers\etc\Tivoli\setup_env.cmd
- %SystemRoot%\system32\drivers\etc\Tivoli\setup_env.sh

To set the Tivoli variables on a Windows operating system, perform the following steps:

1. Log in to a Windows Tivoli server or managed node.
2. From a DOS command prompt, enter:

   ```
   %SystemRoot%\system32\drivers\etc\Tivoli\setup_env
   bash
   ```

   The **bash** command starts the bash shell, which is a variation of the Bourne shell.

Launching the Tivoli desktop

Before you launch the Tivoli desktop, you must initialize the Tivoli environment variables as described in “Setting Tivoli environment variables” on page 30.

To launch the Tivoli desktop on UNIX systems, perform the following steps:

1. Set your DISPLAY environment variable.
2. Initialize the Tivoli environment variables as described in “Setting Tivoli environment variables” on page 30.
3. Run the **tivoli** command.

   For additional information about the **tivoli** command, refer to the *Tivoli Management Framework Reference Manual*.

For the procedure to launch the Tivoli desktop on Windows systems, see Chapter 8, “Tivoli Desktop for Windows,” on page 65.

Enabling and disabling remote connections

During the installation of a Tivoli server or managed node, you specify whether to allow remote connections. After installation, you can modify this value to allow or disallow remote connections. Some products, such as the Tivoli desktop, cannot connect to a Tivoli server or managed node unless it allows remote systems to log in. You must therefore have remote connections enabled before attempting to start these products.

To determine whether a Tivoli server or managed node allows remote systems to log in, use the **odadmin** command on the host machine:

```
odadmin
```

If the **odadmin** command returns the one of the following messages, this system allows remote systems to log in:

```
Remote client login allowed = TRUE
Remote client login allowed = version_2
```

If the **odadmin** command returns the following message, this system does not allow remote systems to log in:

```
Remote client login allowed = FALSE
```

Enabling remote connections

To enable remote connections to a Tivoli server or managed node, set the `Remote client login allowed` option to `TRUE` by performing the following steps:

1. Log in to the managed node or Tivoli server as a Tivoli administrator.
2. Set the Tivoli environment variable following the procedure in “Setting Tivoli environment variables” on page 30.
3. From command line, enter the following command to determine the correct oserv number:

   ```
   odadmin odlist
   ```

   The output is similar to the following:

   ```
   Region     Disp Flags Port IP addr       Hostnames
   4062000000 1    ct-   94   145.80.21.37  alta.tivoli.com,alta
              2    ct-   94   145.80.21.48  newcastl.tivoli.com
              3    ct-   94   145.80.21.123 cygnus.tivoli.com
              4    ct-   94   145.80.21.34  djatzlau1.tivoli.com
   ```

The object dispatcher number is listed in the Disp column. In this example, the dispatcher number for the Tivoli server is always 1.

4. From the host machine command line, set the Remote client login allowed option to TRUE or version_2 by entering one of the following commands:

   ```
   odadmin set_allow_rconnect TRUE 1
   odadmin set_allow_rconnect version_2 1
   ```

   Notes®:

   - The values of TRUE and version_2 both allow for remote connections. The difference between these values is that TRUE uses a private key and version_2 uses a public key. The version_2 value provides for a more secure connection.
   - Refer to the *Tivoli Management Framework Reference Manual* for more information about the **odadmin** command.

Disabling remote connections

To prevent remote connections to a Tivoli server or managed node, set the Remote client login allowed option to FALSE by performing the following steps:

1. Log in to the managed node or Tivoli server as a Tivoli administrator.
2. Set the Tivoli environment variable following the procedure in "Setting Tivoli environment variables" on page 30.
3. From command line, enter the following command to determine the correct oserv number:

   ```
   odadmin odlist
   ```

   The output is similar to the following:

   ```
   Region     Disp Flags Port IP addr       Hostnames
   4062000000 1    ct-   94   145.80.21.37  alta.tivoli.com,alta
              2    ct-   94   145.80.21.48  newcastl.tivoli.com
              3    ct-   94   145.80.21.123 cygnus.tivoli.com
              4    ct-   94   145.80.21.34  djatzlau1.tivoli.com
   ```

   The object dispatcher number is listed in the Disp column. In this example, the object dispatcher number for the Tivoli server is always 1.
4. From the host machine command line, set the Remote client login allowed option to FALSE by entering the following command:

   ```
   odadmin set_allow_rconnect FALSE 1
   ```

   Note: Refer to the *Tivoli Management Framework Reference Manual* for more information about the **odadmin** command.

Chapter 4. Configuring SSH

Remote installations of managed nodes and endpoints require a connection to the machine on which the installation is being performed. When an rexec or rsh connection is used, data transmitted on the connection is not encrypted. Therefore, Tivoli Management Framework provides the option to use a secure shell (SSH) connection. An SSH connection encrypts all data sent over the connection.

Note: You cannot use SSH to perform installations on Windows targets.

If you want to perform installations using SSH, you must install and configure a supported version of SSH. For a list of versions of SSH that Tivoli Management Framework supports, see the *Tivoli Management Framework Release Notes*. You must install the SSH server on the machine on which the installation is being performed. To install managed nodes using SSH, you must install the SSH client on the Tivoli server. To install endpoints using SSH, you must install the SSH client on the managed node from which the SSH connection is made.

Tasks to configure SSH

This section describes tasks that you must complete before you use SSH to perform installations.

Installations to unfamiliar machines

An unfamiliar machine is a machine that is not listed in the $HOME/.ssh/known_hosts file that resides on the machine performing the installation. ($HOME is the home directory of root on UNIX systems or Administrator on Windows systems.) When SSH encounters an unfamiliar machine, it displays a prompt. This prompt causes Tivoli installations to fail. Therefore, you must configure the SSH client so that it does not prompt when it connects to an unfamiliar machine.

To configure the SSH client so that it does not prompt when it connects to an unfamiliar machine, perform one of the following tasks:

- Set the StrictHostKeyChecking parameter in the ssh_config file to `no`.
- Add each machine that is an installation target to the known_hosts file. This process can be automated using the ssh-scankey utility, which is included with OpenSSH Version 3.7.1.

  Note: If a machine has multiple host names, add all of the host names or SSH might not recognize the machine. For example, if a machine has the host names bpickin and bpickin.companyname.com, add both host names.
- Connect to the machine using SSH from either the root or Administrator account and respond manually to the prompt. After you respond to the prompt one time, the machine is added to the $HOME/.ssh/known_hosts file. SSH will no longer prompt when connecting to this machine.

Installations using Tivoli Software Installation Service

To use SSH with Tivoli Software Installation Service, you must set the `SSH` environment variable on the managed node on which the Tivoli Software Installation Service depot is installed. If the `SSH` environment variable is not set, or if a supported version of SSH is not installed, the SSH installation option is not available in the Tivoli Software Installation Service GUI. For more information about setting the SSH environment variable, see “Setting the SSH and SSH_AUTH_SOCK environment variables.”

Installations using the SSH Agent

The SSH agent enables you to use a trusted host access method. Trusted host access enables you to perform installations on multiple machines while having to enter your SSH password only one time. You run the SSH agent by launching the ssh-agent executable. If you intend to use public key authentication with the SSH agent, you must first set the SSH_AUTH_SOCK environment variable. For more information, see “Setting the SSH and SSH_AUTH_SOCK environment variables.”

Installations from Windows systems

To use SSH to perform an installation from a Windows system, you must ensure that the Windows system meets the requirements listed in this section. For managed node installations, these requirements apply to the Tivoli server. For endpoint installations, these requirements apply to the managed node from which you run the **winstlcf** command. The requirements for Windows systems are as follows:

- Cygwin is installed.
- The OpenSSH package, Version 3.6.1 or later, for Cygwin is installed.
- The Perl 5.*x* package for Cygwin is installed.

Note: To install an endpoint using SSH, you must launch a Cygwin command prompt and then run the **winstlcf** command from the version of Perl provided by Cygwin, as shown in the following example:

```
/usr/bin/perl -S winstlcf -j options
```

Before using Cygwin, review the Cygwin documentation for any issues that might affect your environment.

Setting the SSH and SSH_AUTH_SOCK environment variables

Before you use SSH, you must set the SSH environment variable to the fully qualified path of the SSH executable.

If you intend to use public key authentication with the SSH agent, you must set the SSH_AUTH_SOCK environment variable. (For more information about the SSH agent, see “Installations using the SSH Agent.”) You set the SSH_AUTH_SOCK environment variable to the value for SSH_AUTH_SOCK that is displayed when you launch the SSH agent. For example, when you launch the SSH agent, you see a message similar to the following:

```
SSH_AUTH_SOCK=/tmp/ssh-doM17436/agent.17436; export SSH_AUTH_SOCK;
SSH_AGENT_PID=15616; export SSH_AGENT_PID;
echo Agent pid 15616;
```

In this case, you would set the SSH_AUTH_SOCK environment variable to /tmp/ssh-doM17436/agent.17436.

The instructions to set the SSH and SSH_AUTH_SOCK environment variables are as follows:

- To install managed nodes using the Tivoli desktop or command line interface, set the SSH and SSH_AUTH_SOCK environment variables on the Tivoli server using the **odadmin environ set 1** command. Append the SSH and SSH_AUTH_SOCK environment variables to the list of environment variables that are displayed by the **odadmin environ get 1** command. For more information about the **odadmin** command, see the *Tivoli Management Framework Reference Manual*.

- To install managed nodes and endpoints using Tivoli Software Installation Service, set the SSH and SSH_AUTH_SOCK environment variables using the **odadmin environ set** command on the managed node on which the Tivoli Software Installation Service depot is installed. Append the SSH and SSH_AUTH_SOCK environment variables to the list of environment variables that are displayed by the **odadmin environ get** command. For more information about the **odadmin** command, see the *Tivoli Management Framework Reference Manual*.
- To install endpoints using the **winstlcf** command, set and export the SSH and SSH_AUTH_SOCK environment variables in the shell in which you run the **winstlcf** command. For example, if you are using bash, sh, or ksh, you can set the SSH environment variable by entering a command similar to the following example:

  ```
  SSH=/usr/local/bin/ssh; export SSH
  ```

Note: On Windows systems, you must run this command from a Cygwin command prompt. For more information, see "Installations from Windows systems" on page 36.

Chapter 5. Tivoli Management Framework in a Microsoft Cluster Server

To help ensure the availability of specific systems and applications, many vendors offer product solutions known as high-availability (HA) products. HA products are designed to monitor critical systems and applications and to restart those applications, possibly on a different physical system, in case of failure.

By implementing Tivoli Management Framework in an HA environment, you can more consistently ensure the capability to manage the resources in your network.

Installing on a Microsoft Cluster Server

You can install a Tivoli server on a Microsoft Cluster Server (MSCS).

To install the Tivoli server on a MSCS, perform the following steps:

1. Run the local installer in advanced mode on the primary node of the cluster.

   The installation is the same as the procedure in "Installing on a Windows operating system" on page 24, except for the following:

   - In the Advanced Options window:
     - Specify the cluster name in the **Host** text box
     - Ensure that the **Register services to start automatically** check box is not selected
   - In the Setup Type window, select a Destination Folder that is on an NTFS volume shared by all nodes in the cluster.

   On reboot, control is transfer to another node in the cluster.
2. After the original primary node reboots, perform the following steps on the new controlling node:
   a. Map a network drive to %SystemDrive% of the primary node.

   b. Copy the %SystemRoot%\system32\TivoliAP.dll file from the primary node to the new controlling node.

   c. Copy the %SystemRoot%\system32\drivers\etc\Tivoli directory from the primary node to the new controlling node.

   d. Set the Tivoli environment variables:

   ```
   %SystemRoot%\system32\drivers\etc\Tivoli\setup_env.cmd
   ```

   e. Create the accounts needed by Tivoli Management Framework:

   ```
   %BINDIR%\TAS\INSTALL\ntconfig -e
   ```

   f. Load the TivoliAP.dll file with the LSA:

   ```
   wsettap -a
   ```

   g. If a Tivoli remote access account was specified during installation, specify this domain account:

   ```
   wsettap -r domain\account
   ```

   h. Install Tivoli Remote Execution Service:

   ```
   trip -install -auto
   ```

   i. Install and configure the Tivoli object dispatcher:

   ```
   oinstall -install %BINDIR%\oserv.exe
   atinstall --quietcopy %BINDIR%\bin
   ```

   j. Reboot to return control to the primary node.

3. On the primary node, use the Cluster Administrator to creating the following cluster resources:
   - Tivoli Remote Execution Service:
     a. Specify the resource type as Generic Resource
     b. Set the following resource dependencies:
        - Cluster IP Address
        - Cluster Name
        - Disk
     c. Specify `trip` as the resource name. There are no parameters.
     d. Select the **User Network Name for computer name** check box.

     No registry keys need replication. Bring the resource online.
   - Tivoli Object Dispatcher:
     a. Specify the resource type as Generic Resource
     b. Set the following resource dependencies:
        - Cluster IP Address
        - Cluster Name
        - Disk
     c. Specify `oserv` as the resource name, and specify `"/-fTRUE /-Nali"` as the start parameters
     d. Select the **User Network Name for computer name** check box.
     e. Add SOFTWARE\Tivoli to the list of registry keys for replication.

     Bring the resource online.
4. On the non-primary node, perform the following:
   a. Use the **wsetadmin** command to add a login for the Administrator on the inactive node
   b. Use the **odadmin odlist change_ip** command to set the Tivoli server IP address to the MSCS IP address
   c. Use the **odadmin odlist add_hostname_alias** command to add the MSCS network name to the host name list on the Tivoli object dispatcher
   d. Use the **odadmin odlist delete_hostname_alias** command to remove node name from the host name list on the Tivoli object dispatcher

Microsoft Cluster Server installation scenario

Because of the complexity of this installation, the following scenario provides the a working model. In this scenario, command input is preceded by a hash character (#). The scenario is based on the following assumptions:

- The address of the network cluster is 192.168.16.100
- The name of the network cluster is WOLFPACK
- The primary (initially active) node is ROMULUS
- The second node is REMUS
- The domain account used by the Tivoli remote access account is ROMA\she_wolf
- Drive C is the %SystemDrive% for each node
- Drive D is reserved for the MSCS quorum
- Drive E is reserved the the NTFS share

- Drive R is the CD-ROM drive

To install the Tivoli server in a MSCS, perform the following steps:

1. Run the local installer process in advanced mode on ROMULUS. Advanced mode is activated by passing the **advanced** command line option:

   ```
   # R:\setup.exe advanced
   ```

   Follow the standard installation procedure except for the following windows:

   - In the Advanced Options window, specify the cluster name in the **Host** text box and ensure that the **Register servers to start automatically** check box is not set.
   - In the Setup Type window, select a Destination Folder that is located on the NTFS share:

     ```
     # E:\Tivoli
     ```

   When the installation completes, reboot ROMULUS to transfer control of the cluster to REMUS.

2. After ROMULUS has rebooted, perform the following steps on REMUS:

   a. Map a network drive to %SystemDrive% of ROMULUS:

      ```
      # net use F: \\ROMULUS C$
      ```

   b. Copy the TivoliAP.dll file from ROMULUS:

      ```
      # copy F:\winnt\system32\TivoliAP.dll C:\winnt\system32\TivoliAP.dll
      ```

   c. Copy the Tivoli system environment folder from ROMULUS:

      ```
      # copy F:\winnt\system32\etc\drivers C:\winnt\system32\etc\drivers /S
      ```

   d. Set the Tivoli environment variables:

      ```
      # C:\winnt\system32\drivers\etc\tivoli\setup_env.cmd
      ```

   e. Create the accounts needed by Tivoli Management Framework:

      ```
      # %BINDIR%\TAS\INSTALL\ntconfig -e
      ```

   f. Load the TivoliAP.dll file with the LSA:

      ```
      # wsettap -a
      ```

   g. Because a Tivoli remote access account was specified during installation, specify this domain account:

      ```
      # wsettap -r "ROMA\she_wolf"
      ```

   h. Install Tivoli Remote Execution Service:

      ```
      # trip -install -auto
      ```

   i. Install the Tivoli object dispatcher:

      ```
      # oinstall -install %BINDIR%\oserv.exe
      # atinstall --quietcopy %BINDIR%\bin
      ```

   j. Reboot REMUS to return control to ROMULUS.

3. After REMUS reboots, use the Cluster Administrator on ROMULUS to creating the following cluster resources:

   - Tivoli Remote Execution Service:

     a. Specify the resource type as Generic Resource

     b. Set the following resource dependencies:

        – Cluster IP Address

        – Cluster Name

        – Disk

     c. Specify `trip` as the resource name. There are no parameters.

     d. Select the **User Network Name for computer name** check box.

     No registry keys need replication. Bring the resource online.

- Tivoli Object Dispatcher:
  a. Specify the resource type as Generic Resource
  b. Set the following resource dependencies:
     - Cluster IP Address
     - Cluster Name
     - Disk
  c. Specify oserv as the resource name, and specify "/-fTRUE /-Nali" as the start parameters
  d. Select the **User Network Name for computer name** check box.
  e. Add SOFTWARE\Tivoli to the list of registry keys for replication.

  Bring the resource online.

4. On ROMULUS, perform the following:
   a. Set the Tivoli environment variables:
      ```
      # C:\winnt\system32\drivers\etc\tivoli\setup_env.cmd
      ```
   b. Add a login for the Administrator of REMUS:
      ```
      # wsetadmin -l REMUS\Administrator ROMULUS\Administrator
      ```
      Note: In some cases, the Administrator name might need to be fully qualified.
   c. Set the IP address of the Tivoli server to the cluster IP address:
      ```
      # odadmin odlist change_ip 1 192.168.16.100 TRUE
      ```
   d. Add the network name of the cluster to the hostname list in the Tivoli object dispatcher:
      ```
      # odadmin odlist add_hostname_alias 1 192.168.16.100 WOLFPACK
      ```
   e. Remove the physical node name to the hostname list in the Tivoli object dispatcher:
      ```
      # odadmin odlist delete_hostname_alias 1 192.168.16.100 ROMULUS
      ```

Chapter 6. Enabling Tivoli Web Interfaces

Tivoli Management Framework provides access to Web-enabled Tivoli Enterprise applications from a browser. When a browser sends an HTTP request to the Tivoli server, the request is redirected to a Web server. Tivoli Management Framework provides this Web access by using a collection of servlets and support files that are installed on the Web server. The servlets establish a secure connection between the Web server and the Tivoli server. The servlets and support files are known as the Tivoli Web interfaces.

To support Tivoli Web interfaces, your Tivoli environment must meet the following conditions:

- A computer system must have a Web server installed
- The object dispatcher on the Tivoli server must be configured to forward HTTP requests to the Web server

The Web server can be hosted on any computer system. The Tivoli Web interfaces were certified against the following Web servers, but you can use any Web server that supports the Servlet 2.2 specification:

- IBM WebSphere® Application Server, Advanced Single Server Edition, Version 4.0
- IBM WebSphere Application Server, Enterprise Edition, also called the Enterprise Application Server, Version 4.0.2
- IBM WebSphere Application Server – Express, Version 5
- IBM WebSphere Application Server, Version 5
- Jakarta Tomcat

Independent of the Web server, the Tivoli Web interface and support files are the same. Table 1 contains the files required by the Tivoli Web interfaces. These files are in /WEBAPP directory on the Tivoli Management Framework 1 of 2 CD.

Table 1. Tivoli Web interface access and support files

File name	Description
TivoliFRW.war	Tivoli Web interface Web archive (WAR) file.
jcf.jar, jsafe.zip, and jlog.jar	Tivoli Web interface support files.
cert.arm	Tivoli Web interface trust certificate file.
jlog.properties	Tivoli Web interface properties file used for configuring the logger.

To enable the Tivoli Web interfaces to operate in a non-English environment, the language bundles must be installed on the Web server. The language bundles are on the Tivoli Management Framework Language Support CD in the /WEBAPP directory. Table 2 on page 44 associates each Java™ archive (JAR) file with its associated language.

Table 2. Language support files for the Tivoli Web interfaces

File name	Language
TivoliFRW_zh_TW.jar	Chinese (Traditional)
TivoliFRW_zh_CN.jar	Chinese (Simplified)
TivoliFRW_ja.jar	Japanese
TivoliFRW_de.jar	German
TivoliFRW_es.jar	Spanish
TivoliFRW_fr.jar	French
TivoliFRW_it.jar	Italian
TivoliFRW_ko.jar	Korean
TivoliFRW_pt_BR.jar	Portuguese (Brazilian)

Installing Web access on Web application servers

To install Web Access on a Web application server, you must complete the following general steps:

1. Install the Web application server.
2. Enable the Web application server to support SSL.
3. Add the Tivoli trust certificate to a trust store file.
4. Enable logging by completing the following steps:
   a. Put the jlog.properties file in a directory accessible to the Web application server.
   b. Edit the jlog.properties file, uncommenting the lines that apply to the appropriate operating system.
   c. Create a directory for the log files.
5. Start the Web application server.
6. Add the logging properties file location and trust store file as system variables on the Web application server.
7. Place the Tivoli support files in the lib directory of the Web application server.
8. Deploy the TivoliFRW.war file.
9. Enable multi-language support.
10. Start the TivoliFRW application.

The sections that follow contain procedures to install Web access on specific Web application servers.

Installing Web access for WebSphere Advanced Single Server Edition, Version 4.0

This section describes how to install and uninstall Tivoli Web access on WebSphere Advanced Single Server Edition, Version 4.0.

Installing Web access

Installing Tivoli Web access on a WebSphere Advanced Single Server involves configuring the Web server to support the Tivoli servlets and installing the Tivoli

Web servlets and support files. To configure and install the Tivoli Web access on WebSphere Advanced Single Server Edition, Version 4.0, perform the following steps:

1. Install the WebSphere server, if not already installed, following the instructions in the product documentation.
2. Enable the WebSphere server to support SSL, if not already enabled, following the instructions in the product documentation.
3. Add the Tivoli trust certificate (cert.arm) to a trust store file using the import option of the Java Runtime Environment (JRE) keytool utility to create a trust store file and add the certificate.
4. Place the jlog.properties file on a directory accessible to the Web server.
5. To specify the directory to which log files will be written, edit the jlog.properties file as follows:
   - In the **FRW message logger file directory** setting section, uncomment the line for the FRW.handler.file.message.fileDir variable that applies to the appropriate operating system (Windows or UNIX).
   - In the **FRW trace logger file directory** setting section, uncomment the line for the FRW.handler.file.trace.fileDir variable that applies to the appropriate operating system (Windows or UNIX).

   There is one line to uncomment in each section for each operating system.
6. Create the TivoliCommon directory on the Web server and grant write permission. This directory must be the same directory specified in step 5.
   - For Windows operating systems, create the directory under the default drive.
   - For UNIX operating systems, create the directory under the /var directory.
7. Start the WebSphere server.
   - For Windows operating systems, enter the following command:
     ```
     WAS_HOME/bin/startServer.bat
     ```
   - For UNIX operating systems, enter the following command:
     ```
     WAS_HOME/bin/startServer.sh
     ```

   There is no visual indicator that the WebSphere server started. To determine whether the Web server started, examine the std.out file of the server. When the Web server starts, the log file contains an entry that states `Open for e-Business`.
8. Using the WebSphere Administration Console, modify the WebSphere configuration file to set Java system properties that locate the Tivoli Web interface trust store file and logging properties file.
   a. Start a Web browser and direct it to `http://`*`websphere-hostname`*`:9090/admin`, where *websphere-hostname* is the name of the machine running the WebSphere server.
   b. Enter a user ID in the **User ID** field of the form that is displayed in the browser.
   c. In the Configuration section in the main page window (bottom right frame) perform the following steps:
      1) Click the **Open a configuration file to edit with the console** link.
      2) Select the radio button next to **Select from files in “config” directory**.
      3) Select **server-cfg.xml** from the drop down list.
      4) Click **OK**.

d. Completely expand the **Nodes** tree in the left frame and click the **JVM Settings** link. A new screen is displayed.

e. Scroll down in the main page frame and click the **Systems Properties** link.

f. Add the system property for the logging properties:

   1) Click **New** to display the System Properties: New System Property screen.

   2) In the **Name** field, type `jlog.propertyFileDir`.

   3) In the **Value** field, type the directory containing the jlog.properties file. Type the directory using an explicit path or a system variable (${WAS_ROOT}) that represents the directory.

   4) Click **OK**.

g. Add the system property for the trust certificate:

   1) Click **New** to display the System Properties: New System Property screen.

   2) In the **Name** field, type `javax.net.ssl.trustStore`.

   3) In the **Value** field, type the path and file name for the trust certificate. Type the directory using an explicit path or a system variable-qualified path (${WAS_ROOT}\keystore) that represents the directory.

   4) Click **OK**.

h. Click the **Save** link in the WebSphere Application Server screen banner to begin saving the new properties in the configuration file. The Save Configuration screen is displayed.

i. Click **OK** to save the properties and values.

9. Copy the support files (jcf.jar, jsafe.zip, and jlog.jar) to the WAS_HOME/lib directory, where WAS_HOME is the installation directory for the WebSphere server.

10. Copy the TivoliFRW.war file to the WAS_HOME/installableApps directory.

11. Extract the TivoliFRW.war file:

    a. Enter the following command for Windows operating systems:

    ```
    WAS_HOME/bin/SEAppInstall.bat -install \
    WAS_HOME/installable/Apps/TivoliFRW.war
    ```

    Or, enter the following command for UNIX operating systems:

    ```
    WAS_HOME/bin/SEAppInstall.sh -Install \
     WAS_HOME/installable/Apps/TivoliFRW.war
    ```

    b. When prompted for an Application Display Name, enter `TivoliFRW`.

    c. When prompted for a Context Root, press Enter.

    d. When prompted to precompile all JSPs in the application, enter `no`.

    e. When prompted to precompile individual Web applications, enter `no`.

    f. When the prompt changes to `TivoliFRW`, press Enter.

12. Stop the WebSphere server.

    - For Windows operating systems, enter the following command:

      ```
      WAS_HOME/bin/stopServer.bat
      ```

    - For UNIX operating systems, enter the following command:

      ```
      WAS_HOME/bin/stopServer.sh
      ```

13. Optionally enable for non-English environments. Copy the Tivoli language bundles from the Tivoli Management Framework Language Support CD to the WAS_HOME/installedApps/TivoliFRW.ear/TivoliFRW.war/WEB-INF/lib directory.

14. If the port number used for the HTTPS protocol is not the default SSL communication port (port 443), edit the WAS_HOME/config/web.xml file to change the value of **https-port** to the correct port number.
15. Save the web.xml file as ASCII text.
16. Start the WebSphere server using the instructions for 7 on page 45.
17. Validate the installation.
    - For Windows operating systems, enter the following command:
      ```
      WAS_HOME/bin/SEAppInstall.bat -list apps
      ```
    - For UNIX operating systems, enter the following command:
      ```
      WAS_HOME/bin/SEAppInstall.sh -list apps
      ```

    The output should include `TivoliFRW`.
18. Using a Web browser, go to http*(s)*://*websphere-server*:*websphere-port*/TivoliFRW/webapp to start using the application.

Uninstalling Web access

To uninstall the Tivoli Web access from WebSphere Advanced Single Server Edition, Version 4.0, perform the following steps:

1. Uninstall the application WAR file.
   - For Windows operating systems, enter the following command, where WAS_HOME is the directory where the WebSphere server was installed:
     ```
     WAS_HOME/bin/SEAppInstall.bat -uninstall TivoliFRW.war
     -delete true
     ```
   - For UNIX operating systems, enter the following command:
     ```
     WAS_HOME/bin/SEAppInstall.sh -uninstall TivoliFRW.war
     -delete true
     ```
2. Stop the WebSphere server.
   - For Windows operating systems, enter the following command:
     ```
     WAS_HOME/bin/stopServer.bat
     ```
   - For UNIX operating systems, enter the following command:
     ```
     WAS_HOME/bin/stopServer.sh
     ```
3. Delete the WAS_HOME/temp/machine-name/DefaultServer/TivoliFRW directory, where WAS_HOME is the name of the directory where the WebSphere server was installed.
4. Delete the TivoliFRW.ear and TivoliFRW.war files from the WAS_HOME/installableApps directory.

Installing Web access for IBM WebSphere Application Server, Enterprise Edition, Version 4.0.2

This section describes how to install and uninstall Tivoli Web access on IBM WebSphere Application Server, Enterprise Edition, Version 4.0.2.

Installing Web Access

Installing Tivoli Web access on WebSphere Application Server, Enterprise Edition, Version 4.0.2 involves configuring the Web server to support the Tivoli servlets and installing the Tivoli Web servlets and support files. To configure and install the Tivoli Web access on a WebSphere Enterprise Edition server, perform the following steps:

1. Install the WebSphere server, if not already installed. Follow the instructions in the product documentation.

2. Enable the WebSphere server to support SSL, if not already enabled. Follow the instructions in the product documentation.
3. Add the Tivoli trust certificate (cert.arm) to a trust store file using the import option of the Java Runtime Environment keytool utility to create a trust store file and add the certificate.
4. Place the jlog.properties file in a directory that is accessible to the application server.
5. To specify the directory to which log files will be written, edit the jlog.properties file as follows:
   - In the **FRW message logger file directory** setting section, uncomment the line that applies to the appropriate operating system (Windows or UNIX).
   - In the **FRW trace logger file directory** setting section, uncomment the line that applies to the appropriate operating system (Windows or UNIX).

   There is one line to uncomment in each section for each operating system.
6. Create the TivoliCommon directory on the WebSphere server host. This must be the same directory specified in step 5.
   - For Windows operating systems, create the directory under the default drive.
   - For UNIX operating systems, create the directory under the /var directory.
7. Start the WebSphere server, if not already started.
   a. Open the WebSphere Advanced Administrative Console.
      - On Windows operating systems, from the **Start** menu click **Programs → IBM WebSphere → Application Server v4.0 AE → Administrator's Console**
      - On UNIX operating systems, run the WAS_HOME/bin/adminclient.sh script

   b. Expand **WebSphere Administrative Domain → Nodes →** *hostname* and click **Application Servers**

   c. Left click Default Server and click **Start**.
8. Modify the server configuration to set two Java system properties that locate the Tivoli Web logging properties file and the trust store file.
   a. In the WebSphere Advanced Administrative Console, expand **WebSphere Administrative Domain → Nodes →** *hostname* **→ Application Servers**

   b. Select the **JVM Settings** tab (on the right side of the console)

   c. Add the following logging property and trust store property to the system properties.
      1) In the **Systems Properties** section, click **Add**
      2) In the **Name** field, type `jlog.propertyFileDir`
      3) In the **Value** field, type the directory containing the jlog.properties file
      4) In the **Systems Properties** section, click **Add**.
      5) In the **Name** field, type `javax.net.ssl.trustStore`
      6) In the **Value** field, type the fully qualified path to the trust certificate (including the file name) created in step 3
      7) Click **Apply**

   d. Close the console.
9. Copy the following support files to the WAS_HOME/lib directory:
   - jcf.jar
   - jsafe.zip

- jlog.jar

10. Copy the TivoliFRW.war file to the WAS_HOME/installableApps directory

    Note: The TivoliFRW.war file has the default **https-port** value set to 443. If the HTTPS port for the WebSphere server is different, modify the **https-port** property in the TivoliFRW.war file as follows:

    a. Start the Application Assembly Tool.
       - On Windows operating systems, from the **Start** menu, click **Programs → IBM WebSphere → Application Server v4.0 AE** and click **Application Assembly Tool**.
       - On UNIX operating systems, run the WAS_HOME/bin/assembly.sh script.

    b. Select the **Existing** tab and click **Browse** to open the TivoliFRW.war file (that was placed in the WAS_HOME/installableApps directory).

    c. Expand the **TivoliFRW** tree and click **Context Parameters**.

    d. Select the https-port property and change the parameter value to the appropriate HTTPS port.

    e. Click **Apply**.

    f. From the **File** menu, click **Save**.

    g. Close the tool.

11. Extract the TivoliFRW.war file.

    a. Open the WebSphere Advanced Administrative Console.

    b. Expand **Console → Wizards** and click **Install Enterprise Application**.

    c. Select the **Install stand-alone module** check box.

    d. Point **Path** to the TivoliFRW.war file from step 10.

    e. Set the **Application name** to `TivoliFRW`.

    f. Set the **Context root for web module** to `/TivoliFRW`.

    g. Click **Next** to accept defaults on the next screens (until you reach the last panel).

    h. Click **Finish**.

12. Optionally enable for non-English environments. Copy the Tivoli language bundles from the Tivoli Management Framework Language Support CD to the WAS_HOME/installedApps/TivoliFRW.ear/TivoliFRW.war/Web-inf/lib directory.

13. Start the TivoliFRW application.

    a. In the WebSphere Advanced Administrative Console, expand **WebSphere Administrative Domain → Nodes →** *hostname* **→ Application Servers** and click **Default Server**.

    b. Select **Installed Web Modules**.

    c. Left click **TivoliFRW** (in the top right side of the console) and click **Start**.

    d. Close the console.

14. Using a Web browser, go to http*(s)*://*websphere-server*:*websphere-port*/TivoliFRW/webapp to start using the application.

Uninstalling Web access

To uninstall the Tivoli Web access from IBM WebSphere Application Server, Enterprise Edition, Version 4.0.2, perform the following steps:

1. Uninstall the application WAR file:

a. Open the WebSphere Advanced Administrative Console.
b. Expand the WebSphere Administrative Domain and click Enterprise Applications
c. Left click TivoliFRW and click **Stop**.
d. Left click TivoliFRW and click **Remove**.
e. Close the console.

2. Delete the WAS_HOME/temp/*machine_name*/Default_Server/TivoliFRW directory, where *machine_name* is the name of the system where the application server is running.
3. From the WAS_HOME/installableApps directory, delete the following files:
   - TivoliFRW_war.ear
   - TivoliFRW.war
4. Delete the trust store file created in installation step 3 on page 48.
5. Delete the jlog.properties file created in installation step 4 on page 48.
6. Delete the TivoliCommon directory created in installation step 6 on page 48.
7. Ensure that no other application is using the following files:
   - jcf.jar
   - jsafe.zip
   - jlog.jar

   If no other application is using these files, delete them.
8. Remove the system properties added to the server configuration in installation step 8 on page 48.
   a. Open the WebSphere Advanced Administrative Console.
   b. Expand **WebSphere Administrative Domain** → **Nodes** → *hostname* → **Application Servers** and click **Default Server**.
   c. Select the **JVM Settings** tab (on the right side of the console).
   d. In the **System Properties** section, select the `jlog.propertyFileDir` line and click **Remove**.
   e. In the **System Properties** section, select the `javax.net.sll.trustStore` line and click **Remove**.
   f. Close the console.

Installing Web access for WebSphere Application Server – Express, Version 5

This section describes how to install and uninstall Tivoli Web access on WebSphere Application Server – Express, Version 5.

Installing Web access

Installing Tivoli Web access on WebSphere Application Server – Express, Version 5 involves configuring the Web server to support the Tivoli servlets and installing the Tivoli Web servlets and support files. To configure and install the Tivoli Web access on WebSphere Application Server – Express, Version 5, perform the following steps:

1. Install the WebSphere server, if not already installed, following the instructions in the product documentation.
2. Enable the WebSphere server to support SSL, if not already enabled, following the instructions in the product documentation.

3. Add the Tivoli trust certificate (cert.arm) to a trust store file using the import option of the Java Runtime Environment keytool utility to create a trust store file and add the certificate.
4. Place the jlog.properties file in a directory accessible to the Web server.
5. To specify the directory to which log files will be written, edit the jlog.properties file as follows:
   - In the **FRW message logger file directory** setting section, uncomment the line for the FRW.handler.file.message.fileDir variable that applies to the appropriate operating system (Windows or UNIX).
   - In the **FRW trace logger file directory** setting section, uncomment the line for the FRW.handler.file.trace.fileDir variable that applies to the appropriate operating system (Windows or UNIX).

   There is one line to uncomment in each section for each operating system.
6. Create the TivoliCommon directory on the Web server and grant write permission. This must be the same directory specified in step 5.
   - For Windows operating systems, create the directory under the default drive.
   - For UNIX operating systems, create the directory under the /var directory.
7. Start the WebSphere server:
   - For Windows operating systems, go to **Start → Programs → IBM WebSphere Application Server – Express v5.0 → Start Application Server**.
   - For UNIX operating systems, enter the following command:

     ```
     WAS_HOME/bin/startServer.sh
     ```
8. Modify the configuration of the server to set two Java system properties that locate the Tivoli Web logging properties file and the trust store file:
   a. Open the Studio Site Developer console:
      - On Windows operating systems, go to **Start → Programs → IBM WebSphere Application Server – Express v5.0 → Studio Site Developer**.
      - On UNIX operating systems, run the following command:

        ```
        wsappdev
        ```
   b. Open the Server perspective by clicking **Window → Open Perspective → Server**.
   c. Open the server configuration editor. In the Server Configuration pane, located in bottom left corner, right-click the server where the TivoliFRW web application will be deployed and select **Open**. The server configuration editor opens in the top right corner of the Studio Site Developer console.
   d. In the server configuration editor, click the **Environment** tab on the bottom edge of the pane.
   e. Scroll down, if necessary, and expand the **System Properties** section, if not already expanded.
   f. Add the logging property to the system properties:
      1) Click **Add**.
      2) In the **Name** field, enter the following value:

         ```
         jlog.propertyFileDir
         ```
      3) In the **Value** field, enter the fully qualified directory that contains the jlog.properties file.
      4) Click **OK**.
   g. Add the trust certificate property to the system properties:

        1) Click **Add**.
        2) In the **Name** field, enter the following value:

           `javax.ssl.net.trustStore`

        3) In the **Value** field, type the fully qualified path to the trust certificate (including the file name).
        4) Click **OK**.

    h. Save the server configuration by clicking **File → Save** *server_name*.

9. Make the support files (jcf.jar, jsafe.zip and jlog.jar) accessible to the Web server:

    a. Open the Studio Site Developer console.

    b. Open the Server perspective by clicking **Window → Open Perspective → Server**.

    c. Open the server configuration editor: In the Server Configuration pane located in bottom left corner, right-click the server where the TivoliFRW Web application will be deployed and click **Open**. The server configuration editor opens in the top right corner of the Studio Site Developer console.

    d. In the server configuration view, click the **Paths** tab on the bottom edge of the pane.

    e. Expand the ws.ext.dirs section and click **Add External JARs**.

    f. Browse to and select the jcf.jar, jlog.jar and jsafe.zip files, and then click **Open**.

    g. Save the server configuration by clicking **File → Save** *server_name*.

10. Load the TivoliFRW web application into a project:

    a. Open the Studio Site Developer.

    b. Click **File → Import**.

    c. In the Import dialog box, select **WAR file** and then click **Next**.

    d. In the **WAR file** field, browse to the TivoliFRW.war file. Select the TivoliFRW.war and then click **Open**.

    e. In the **New project name** field, enter the following value:

       `TivoliFRW`

    f. Verify that the value of the **Context Root** field is `TivoliFRW`.

    g. In the Enterprise application project section, enter the following value in the **New project name** field:

       `TivoliFRW_ear`

    h. Click **Finish**.

    Note: The TivoliFRW.war file has the default **https-port** value set to 443. If the HTTPS port for the WebSphere server is different, modify the **https-port** property in the TivoliFRW.war file as follows:

        a. In the Navigator window located in the top left pane, expand **TivoliFRW → Web Content → WEB-INF**. Right-click **web.xml** and open the Web Deployment Descriptor editor. The Web Deployment Descriptor editor opens in the top right corner of the Studio Site Developer console.

        b. In the Web Deployment Descriptor editor, click the **Parameters** tab on the bottom edge of the pane.

        c. In the Parameters section, select **https-port**.

        d. In the Details section, enter the correct HTTPS port in the Value field.

  e. Save the changes by clicking **File → Save Web Deployment Descriptor**.

11. Deploy the TivoliFRW application:
    a. Open the Studio Site Developer console.
    b. In the Navigator window, located in the top left corner, right-click **TivoliFRW** and click **Run on Server**.
    c. In the Server Selection dialog box, specify the type of server on which to run TivoliFRW, if applicable.
    d. In the Server Selection dialog box, select an existing server or create a new server. Click **Finish** when you are done.
12. Optionally enable for non-English environments. Copy the Tivoli language bundles from the Tivoli Management Framework Language Support CD to the WAS_HOME/workplace/TivoliFRW/Web Content/WEB-INF/lib directory.
13. Using a Web browser, go to http*(s)*://*websphere-server*:*websphere-port*/ TivoliFRW/webapp to start using the application.

Uninstalling Web access

To uninstall the Tivoli Web access from WebSphere Application Server – Express, Version 5, perform the following steps:

1. Uninstall the TivoliFRW application from the server:
    a. Open the Studio Site Developer console.
    b. Open the Server perspective by clicking **Window → Open Perspective → Server**.
    c. In the Server Configuration pane on the left bottom corner, expand **Servers** and expand the server name where the TivoliFRW application is installed.
    d. Right-click **TivoliFRW_ear** and click **Remove**.
    e. Restart the server, if necessary. In the Servers panel in the bottom right corner, right-click the server name and click **Restart**.
2. Remove the TivoliFRW and TivoliFRW_ear projects:
    a. Open the Studio Site Developer console.
    b. In the Navigator pane on the left top corner, right-click **TivoliFRW** and click **Delete**. In the Confirm Project Delete dialog box, choose to also delete contents under the workplace and click **Yes**.
    c. In the same panel, right-click **TivoliFRW_ear** and click **Delete**. In the Confirm Project Delete dialog box, choose to also delete contents under the workplace and click **Yes**.
3. Delete the TivoliCommon directory created in installation step 6 on page 51.
4. Delete the trust store file created in installation step 3 on page 51 and the jlog.properties file from installation step 4 on page 51.
5. Remove the system properties added to the server configuration in installation step 8 on page 51:
    a. Open the Studio Site Developer console.
    b. Open the server configuration editor. In the Server Configuration dialog box, located in bottom left corner, right-click the server where the TivoliFRW web application was installed and then click **Open**.
    c. In the server configuration editor located in the top right, click the **Environment** tab on the bottom edge of the pane.
    d. Scroll down, if necessary, and expand the System Properties section.
    e. Choose the jlog.propertyFileDir property and click **Remove**.

f. Choose the javax.ssl.net.trustStore property and click **Remove**.

6. Make sure no other application is using the support files (jcf.jar, jsafe.zip and jlog.jar) and then remove the support files as follows:
   a. Open the Studio Site Developer console.
   b. Open the server configuration editor. In the Server Configuration dialog box, located in bottom left corner, right-click the server where the TivoliFRW web application was installed and then click **Open**.
   c. In the server configuration editor, located in the top right corner, click the **Paths** tab on the bottom edge of the pane.
   d. Expand the **ws.ext.dirs** section, select **jcf.jar** and then click **Remove**.
   e. Select **jsafe.zip** and then click **Remove**.
   f. Select **jlog.jar** and then click **Remove**.
   g. Save the server configuration by clicking **File → Save** *server_name*.

Installing Web access for WebSphere Application Server, Version 5

This section describes how to install and uninstall Tivoli Web access on WebSphere Application Server, Version 5.

Installing Web access

Installing Tivoli Web access on WebSphere Application Server, Version 5 involves configuring the Web server to support the Tivoli servlets and installing the Tivoli Web servlets and support files. To configure and install the Tivoli Web access on WebSphere Application Server, Version 5, perform the following steps:

1. Install the WebSphere server, if not already installed, following the instructions in the product documentation.
2. Enable the WebSphere server to support SSL, if not already enabled, following the instructions in the product documentation.
3. Add the Tivoli trust certificate (cert.arm) to a trust store file using the import option of the Java Runtime Environment keytool utility to create a trust store file and add the certificate.
4. Place the jlog.properties file in a directory accessible to the Web server.
5. To specify the directory to which log files will be written, edit the jlog.properties file as follows:
   - In the **FRW message logger file directory setting** section, uncomment the line for the FRW.handler.file.message.fileDir variable that applies to the appropriate operating system (Windows or UNIX).
   - In the **FRW trace logger file directory setting** section, uncomment the line for the FRW.handler.file.trace.fileDir variable that applies to the appropriate operating system (Windows or UNIX).

   There is one line to uncomment in each section for each operating system.
6. Create the TivoliCommon directory on the Web server and grant write permission. This must be the same directory specified in step 5.
   - For Windows operating systems, create the directory under the default drive.
   - For UNIX operating systems, create the directory under the /var directory.
7. Start the WebSphere server:
   - For Windows operating systems, click **Start → Programs → IBM WebSphere → Application Server v5.0 → Start the Server**.
   - For UNIX operating systems, enter the following command:

```
WAS_HOME/bin/startServer.sh
```

There is no visual indicator that the WebSphere server started. To determine whether the Web server started, examine the SystemOut file of the server. When the Web server starts, the log file contains an entry that states `Open for e-Business`.

8. Modify the configuration of the server to set two Java system properties that locate the Tivoli Web logging properties file and the trust store file:
   a. Open the WebSphere Administrative Console and then log in:
      - On Windows operating systems, click **Start** → **Programs** → **IBM WebSphere** → **Application Server v5.0** → **Administrative Console**.
      - On UNIX operating systems, direct a browser to http://*websphere-server*:9090/admin
   b. In the left pane of the console, expand **Servers** and click the **Application Servers** link.
   c. In the Application Servers frame, click the *hostname* link to go into the **Configuration** tab.
   d. Scroll down and click the **Process Definition** link.
   e. Scroll down and click the **Java Virtual Machine** link.
   f. Add the logging property to the system properties:
      1) Scroll down and click the **Custom Properties** link.
      2) Click **New**.
      3) In the **Name** field, enter the following value:
         ```
         jlog.propertyFileDir
         ```
      4) In the **Value** field, enter the directory that contains the jlog.properties file.
      5) In the **Description** field, enter the following value:
         ```
         JLOG Property File
         ```
      6) Click **OK**.
   g. Add the trust certificate property to the system properties:
      1) Scroll down and click the **Custom Properties** link.
      2) Click **New**.
      3) In the **Name** field, enter the following value:
         ```
         javax.net.ssl.trustStore
         ```
      4) In the **Value** field, type the fully qualified path to the trust certificate (including the file name) created in step 3 on page 54.
      5) In the **Description** field, enter the following string:
         ```
         Java TrustStore File
         ```
      6) Click **OK**.
   h. In the top banner titled Message(s), click the **Save** link.
   i. In the Save to Master Configuration box, click **Save**.
9. Copy the support files (jcf.jar, jsafe.zip and jlog.jar) to the WAS_HOME/lib directory.
10. Copy the TivoliFRW.war file to the WAS_HOME/installableApps directory.

    Note: The TivoliFRW.war file has the default **https-port** set to 443. If the HTTPS port for the WebSphere server is different, modify the **https-port** property in the TivoliFRW.war file as follows:
    a. Start the Application Assembly Tool.

- On Windows operating systems, click **Start → Programs → IBM WebSphere → Application Server v5.0 → Application Assembly Tool**
- On UNIX operating systems, run the following command:

```
WAS_HOME/bin/assembly.sh
```

b. In the Welcome to Application Assembly Tool dialog box, click the **Existing** tab and browse to open the TivoliFRW.war file that was placed in the WAS_HOME/installableApps directory.

c. After the TivoliFRW.war file is opened, expand the **TivoliFRW** tree and select **Context Parameters**.

d. Select the **https-port** property and change the parameter value to the appropriate HTTPS port.

e. Click **Apply**.

f. Click **File → Save** on the top menu to save the modified TivoliFRW.war file.

g. Close the TivoliFRW.war file if desired, by clicking **File → Close**.

11. Extract the TivoliFRW.war file:

    a. Open the WebSphere Administrative Console.

    b. On the left side of the console, expand **Applications** and click the **Install New Application** link.

    c. In the **Path** field, enter the fully qualified path and file name for the TivoliFRW.war file from step 10 on page 55.

    d. In the Context Root field, enter the following value:

    ```
    /TivoliFRW
    ```

    e. Click **Next**.

    f. In the Generate Default bindings screen, ensure that **Do not override existing bindings** is selected. Select the desired virtual host. Click **Next**.

    g. On Step 1, accept all defaults. Click **Next**.

    h. On Step 2, select the **TivoliFRW** web module and select a virtual host. Click **Next**.

    i. On Step 3, choose **TivoliFRW** module. Click **Next**.

    j. On Step 4, check the summary and click **Finish** if everything is correct.

    k. Click the **Save to Master Configuration** link.

    l. In the Save to Master Configuration box, click **Save**.

12. Optionally enable for non-English environments. Copy the Tivoli language bundles from the Tivoli Management Framework Language Support CD to the WAS_HOME/installedApps/*hostname*/TivoliFRW_war.ear/TivoliFRW.war/WEB-INF/lib directory.

13. Start the TivoliFRW application:

    a. Open the WebSphere Administrative Console.

    b. In the left pane of the console, expand **Applications** and click the **Enterprise Applications** link.

    c. In the Applications pane, choose the **TivoliFRW_war** application and then click **Start**.

14. Using a Web browser, go to http*(s)*://*websphere-server*:*websphere-port*/TivoliFRW/webapp to start using the application. The server might need to be restarted for all the new settings to take effect.

Uninstalling Web access

To uninstall the Tivoli Web access from WebSphere Application Server, Version 5, perform the following steps:

1. Uninstall the application WAR file:
   a. Open the WebSphere Administrative Console.
   b. In the left pane of the console, expand **Applications** and click the **Enterprise Applications** link.
   c. Select **TivoliFRW_war** and click **Stop**.
   d. Select **TivoliFRW_war** and click **Uninstall**.
   e. In the top banner titled Message(s), click the **Save** link.
   f. In the Save to Master Configuration box, click **Save**.
2. Delete the WAS_HOME/temp/*hostname*/server*n*/TivoliFRW_war directory, where *hostname* is the name of the computer system where the application server is running, and server*n* is the directory for the server instance, for example server1.
3. Delete the TivoliFRW.war file from the WAS_HOME/installableApps directory.
4. Delete the TivoliCommon directory created in installation step 6.
5. Delete the trust store file created in installation step 3 on page 54.
6. Delete the jlog.properties file from installation step 4 on page 54.
7. Remove the system properties added to the server configuration in installation step 8 on page 55:
   a. Open the WebSphere Administrative Console.
   b. In the left pane of the console, expand **Servers** and click the **Application Servers** link.
   c. In the Application Servers frame, click the *hostname* link to go into the **Configuration** tab.
   d. Scroll down and click the **Process Definition** link.
   e. Scroll down and click the **Java Virtual Machine** link.
   f. Scroll down and click the **Custom Properties** link.
   g. Select the **javax.net.ssl.trustStore** and the **jlog.propertyFileDir** properties and then click **Delete**.
   h. In the top banner titled Message(s), click the **Save** link.
   i. In the Save to Master Configuration box, click **Save**.
8. Make sure that no other application is using the support files (jcf.jar, jsafe.zip and jlog.jar) and then delete them from the WAS_HOME/lib directory.

Installing Web access for Tomcat

This section describes how to install and uninstall Tivoli Web access on a Tomcat server.

Installing Web access

Installing Tivoli Web access on a Tomcat server involves configuring the Web server to support the Tivoli servlets and installing the Tivoli Web servlets and support files. To configure and install the Tivoli Web access on a Tomcat server, perform the following steps:

1. Install the Tomcat server, if not already installed, following the product documentation.
2. On the Web server, install the Java Runtime Environment (JRE) 1.3

3. Enable the Tomcat server to support Direct SSL, if not already enabled, following the instructions in the product documentation.

   Note: Disable client authorization by setting `clientAuth` to `false` in the *tomcat-home*/conf/server.xml file.
4. Add the Tivoli trust certificate (cert.arm) to a trust store file. For your trust store file, you can use the one configured as part of the Direct SSL setup or create a new one. Use the import option of the JRE keytool utility to add a certificate. If you are creating a new trust store file, use a different alias name than the one used in step 3.
5. Place the jlog.properties file in a directory that is accessible to the application server.
6. To specify the directory to which log files will be written, edit the jlog.properties file as follows:
   - In the **FRW message logger file directory** setting section, uncomment the line that applies to the appropriate operating system (Windows or UNIX).
   - In the **FRW trace logger file directory** setting section, uncomment the line that applies to the appropriate operating system (Windows or UNIX).

   There is one line to uncomment in each section for each operating system.
7. Create the TivoliCommon directory on the Web server and grant write permission. This must be the same directory specified in step 6.
   - For Windows operating systems, create the directory under the default drive.
   - For UNIX operating systems, create the directory under the /var directory.
8. Modify the Tomcat startup script to set the properties that locate the Tivoli trust store file and logging properties file.
   - For Windows operating systems, add the following line to the *tomcat-home*/bin/tomcat.bat file:

     ```
     set TOMCAT_OPTS= \
       -Djavax.net.ssl.trustStore=trust-store
       -Djlog.propertyFileDir=logging-properties-directory
     ```
   - For UNIX operating systems, add the following line to the *tomcat-home*/bin/tomcat.sh file:

     ```
     TOMCAT_OPTS= \
       "-Djavax.net.ssl.trustStore=trust-store
       -Djlog.propertyFileDir=logging-properties-directory"
     ```

   where *trust-store* specifies the full path of the trust store file set in step 4 and *logging-properties-directory* is the full path to the directory containing the logging properties file set in step 5.
9. Copy the support jar files (jcf.jar, jsafe.zip and jlog.jar) to the *tomcat-home*/lib directory.
10. Copy the TivoliFRW.war file to the *tomcat-home*/webapps directory.
11. Start the Tomcat server to extract the contents of the TivoliFRW.war file to the *tomcat-home*/webapps/TivoliFRW directory.
12. Stop the Tomcat server.
13. Optionally enable for non-English environments. Copy the Tivoli application language bundles from the Tivoli Management Framework Language Support CD to the *tomcat-home*/webapps/TivoliFRW/WEB-INF/lib directory.

14. If the port number used for the HTTPS protocol (as set in step 3 on page 58) is not the default SSL communication port (port 443), edit the *tomcat-home*/webapps/TivoliFRW/WEB-INF/web.xml file to change the value of **https-port** to the correct port number.
15. Start the Tomcat server.
16. Using a Web browser, go to http://*tomcat-hostname*:*port*/TivoliFRW/webapp to start using the application.

Uninstalling Web access

To uninstall the Tivoli Web access from a Tomcat server, perform the following steps:

1. Stop the Tomcat server.
2. Delete the TivoliFRW.war file.
3. Delete the *tomcat-home*/webapps/TivoliFRW directory.
4. Delete the working files associated with the Tivoli Web interfaces from the *tomcat-home*/work directory.
5. Delete the TivoliCommon directory created in step 7 on page 58 of the installation procedure.
6. Delete the modifications to the startup script that were performed in step 8 on page 58 of the installation procedure.
7. Delete the support jar files copied in step 9 on page 58 of the installation procedure.
8. If applicable, delete the trust store file.

Redirecting HTTP requests

For Web-enabled Tivoli Enterprise applications that provide Web interfaces, redirect HTTP requests to the third-party Web server using the **whttpd** command. This command can be used to redirect HTTP requests to a third-party Web server, unregister the spider service, and remove the Spider.exe file (on Windows operating systems) and the Spider file (on UNIX operating systems).

For detailed information about the **whttpd** command, refer to the *Tivoli Management Framework Reference Manual*.

Chapter 7. Configuring for SMTP e-mail

Many Tivoli products generate e-mail for alerts and other messages. For example, Tivoli Software Distribution might send mail after a file distribution. Tivoli Management Framework does not include a proprietary e-mail server, but relies on the UNIX **sendmail** command. This command is based on the Simple Mail Transfer Protocol (SMTP).

Tivoli Management Framework assumes the **sendmail** command is running as an SMTP daemon in the environment. This assumption requires an SMTP server in operation somewhere in the network, usually on the machine designated as the network mail server. Tivoli Management Framework does not require the SMTP server to be installed on a Tivoli managed resource.

Use the information in the following sections to determine how to configure a mail server to correctly route the mail generated by Tivoli products.

UNIX-only installations

If your Tivoli environment consists only of UNIX operating systems, you do not need to take any configuration or installation steps to set up Tivoli Management Framework for e-mail connections. The Tivoli tools send e-mail, by default, through Transmission Control Protocol (TCP) port 25 to connect with the SMTP server and the SMTP daemon.

Windows-only installations

For a Tivoli environment that consists only of Windows operating systems, you need to install SMTP gateway software, which is freely available in the public domain, on each managed node, including the Tivoli server. Windows mail servers, such as Microsoft Exchange or Lotus Notes®, are based on the messaging applications programming interface (MAPI). The SMTP gateway that you install on the Windows mail server provides a kind of emulation that enables Tivoli tools to transfer e-mail among Microsoft Windows machines.

You also need to run the **wmailhost** command. Generally you run this command during the installation process of a managed node. The **wmailhost** command provides Tivoli tools with information about the name and location of the SMTP mail server. The **wmailhost** command enters the mail server name in the Windows operating system registry. Tivoli tools send e-mail through the specified SMTP gateway.

Refer to "Using the wmailhost command" on page 62 for information about using the **wmailhost** command to specify which machine hosts the SMTP mail server.

Windows and UNIX installations

If your Tivoli environment includes of a combination of UNIX operating systems and Microsoft Windows operating systems, you must configure them for SMTP communications. Windows operating systems use MAPI instead of SMTP to route e-mail between a mail server and its clients.

For a Tivoli environment that includes UNIX and Windows operating systems, you need to issue the **wmailhost** command on each managed node running a supported Windows operating system. The **wmailhost** command enters the name of the specified network mail server in the Windows registry. You can specify either a UNIX or Windows mail server. If you choose a host running the Windows operating system, the SMTP gateway software must already be installed on that system.

If you choose a UNIX mail server, the Microsoft operating systems communicate with the UNIX mail server using TCP port 25. The Tivoli tools installed on a Microsoft machine send e-mail as they do on a UNIX machine using the mail server specified with the **wmailhost** command.

If you choose a Microsoft mail server, Tivoli tools route e-mail from these systems through the SMTP gateway to the mail server.

Refer to "Using the wmailhost command" for information about using the **wmailhost** command to specify which machine hosts the SMTP mail server.

Using the wmailhost command

The syntax for the **wmailhost** command is as follows:

```
wmailhost host_name
```

where *host_name* is the host name of the mail server.

When you issue the **wmailhost** command, the mail server you specify is registered as the value for Tivoli\Platform\mailhost under HKEY_LOCAL_MACHINE\SOFTWARE in the Windows registry.

For additional information about **wmailhost**, refer to the *Tivoli Management Framework Reference Manual*.

Sending e-mail

To send e-mail on Microsoft managed nodes, use the **smtp_client** command located in $BINDIR/bin. This command is useful in scripts when you want to send an e-mail as the result of an action.

This command reads the message from standard input or from a file and relays the message to the SMTP server, which it finds by searching the following places in the following order:

1. The host specified in the e-mail address
2. The mail server specified with the **wmailhost** command
3. The local managed node
4. The Tivoli server

The syntax for the **smtp_client** command is as follows:

smtp_client [@*mail_server*[,...]:]*address* < *file_path*

where:

mail_server
A comma-separated list of mail hosts.

address The fully-qualified e-mail address.

file_path
The name of file, including the path, that contains the information to be sent. You can also include text redirected from the command line.

If you specify the **smtp_client** command without specifying a forwarding path, it looks at the value for Tivoli\Platform\mailhost under HKEY_LOCAL_MACHINE\SOFTWARE in the registry. Without designating the mail server, mail cannot be delivered to an external address.

Note: This key is set using the **wmailhost** command.

For example, to send the c:\tmp\comments.txt file to the Tivoli publication department through the my.net mail server, the script would include the following line:

```
smtp_client @my.net:pubs@tivoli.com < c:\tmp\comments.txt
```

Chapter 8. Tivoli Desktop for Windows

The Tivoli desktop is the graphical user interface (GUI) to managing Tivoli products. It is the system through which all Tivoli Enterprise products interact with Tivoli administrators. For OS/2 and Windows operating systems, you must install and use Tivoli Desktop for Windows. Tivoli Desktop for Windows functions exactly like the Tivoli desktop on UNIX operating systems.

Note: The system where you install and run Tivoli Desktop for Windows does not need to be a computer system that is a Tivoli Management Framework managed resource.

Tivoli Desktop for Windows is different from merely using an X Window System server on a PC and connection to a UNIX managed node. Although using an X Window System server enables a redirection from a UNIX managed node, there are the following advantages to using Tivoli Desktop for Windows:

- Better security—Guarantees the authenticity of the Tivoli administrator, which is a fundamental requirement for ensuring network security. Tivoli Desktop for Windows requires users to log in and provide DES encryption for the login password.
- Less resource consumption—Performs all user processing on the PC, which avoids any additional overhead on the UNIX managed node.

After starting the Tivoli desktop, it connects to a managed node that accepts *remote* connections. For additional information about enabling and disabling remote connections, refer to "Enabling and disabling remote connections" on page 32.

Notes:

- Tivoli Desktop for Windows cannot be installed on UNIX operating systems. Tivoli desktop is installed on UNIX operating systems during Tivoli server and managed node installation.
- For information about which OS/2 and Windows operating systems are supported, refer to the *Tivoli Management Framework Release Notes*.
- You can resolve many network connectivity issues related to the Tivoli desktop by setting the DESKTOPS_CHECK_TIMEOUT environment variable. For more information, see the section about the Tivoli desktop in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Installing Tivoli Desktop for Windows on Windows operating systems

You can install Tivoli Desktop for Windows on Windows systems using one of the following methods:

- From the Tivoli Management Framework 2 of 2 CD.
- From diskettes you create.
- Using InstallShield in silent mode.

Installing the Tivoli desktop on Windows operating systems from CD

To install Tivoli Desktop for Windows from the Tivoli Management Framework 2 of 2 CD, perform the following steps:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. Exit all Windows programs.
3. Run *x*:\desktop\nt_95\setup.exe, where *x*: is the drive letter of the CD-ROM drive.
4. Follow the instructions presented on the screen.

Installing the Tivoli desktop on Windows operating systems using a response file

You can install Tivoli Desktop for Windows in silent mode on a Windows operating system. In silent mode, the installation process does not prompt the user for input.

To do this, perform the following high-level actions:

1. Place the Tivoli Desktop for Windows installation media in a drive accessible as a drive letter by the systems on which you want to install Tivoli Desktop for Windows silently.
2. Install Tivoli Desktop for Windows on a system and record the installation in a silent InstallShield (.iss) file. Make sure that this file is on a drive accessible by the target systems.
3. Modify the .iss file as appropriate for the target systems.
4. On each target system, invoke the .iss file. This can be done manually from a command prompt or the Run window, or it can be automated using a login script or other method.

The following example demonstrates the process:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. Copy the \desktop directory from the CD to a network drive. For this example, assume that the \desktop directory is copied to `n:\network\installMedia\tivoli\desktop`.
3. From a command prompt, change directories to the \nt_95 directory of the installation media. For example:
   ```
   n:
   cd \network\installMedia\tivoli\desktop\nt_95
   ```
4. Run the following command to start the installation process and to record the following instructions presented on the screen to install the Tivoli Desktop for Windows on the current system and to create the silent InstallShield file in the file `n:\network\ISS\TivoliDesktop.iss`:
   ```
   setup -r -f1n:\network\ISS\TivoliDesktop.iss
   ```
   If you omit the **–f1** option, the file is created as %WINDIR%\setup.iss. By default, %WINDIR% is c:\winnt on Windows NT, Windows 2000, or Windows XP and c:\windows on Windows 98 operating systems.

   Note: This command does not work if you add a space between the **–f1** option and the path.
5. Follow the prompts in the installation windows to set installation options.
6. If you want to install to a different destination path or Start menu folder than you specified in step 5, use a text editor to modify the TivoliDesktop.iss response file. To change the directory in which files are installed, modify the `szDir=` statement. To change the name of the Start menu folder, modify the `szFolder=` statement.

For example, the following lines cause the files to be installed in c:\Tivoli\Desktop and the Start menu folder to be named **Tivoli Desktop**:

```
szDir=c:\Tivoli\Desktop
...
szFolder=Tivoli Desktop
```

7. On each machine where you want to install Tivoli Desktop for Windows silently, map to the network drives containing the installation media and the .iss file, and then run the setup program.

   For example, if the drive mapped as `n:` in step 2 is mapped as `t:` on the target system, run the following commands:

   ```
   t:
   cd \network\installMedia\tivoli\desktop\nt_95
   setup -s -f1t:\network\ISS\TivoliDesktop.iss
   ```

   Note: This command does not work if you add a space between the **–f1** option and the path.

Adding the icon to a Windows desktop

After you install Tivoli Desktop for Windows on a supported Windows system, you can add an icon to your Windows desktop to start the Tivoli desktop. To add this icon, perform the following steps:

1. Open Windows Explorer.
2. Navigate to the folder containing **tivoli.exe**. If you installed using the default installation location, this is c:\Program Files\Tivoli\Desktop.
3. Drag this executable to your desktop and drop it. The icon is now on your desktop. It is named **Shortcut to tivoli.exe**.
4. Rename the icon.

Installing Tivoli Desktop for Windows on OS/2

Before installing Tivoli Desktop for Windows on an OS/2 machine, you need to have the following software already installed:

- WIN-OS/2®
- TCP/IP for OS/2, Version 4.1

If your system does not have the WIN-OS/2 package installed, use **Selective Install** to add it. If your system does not have the appropriate OS/2 TCP/IP package installed, use **Selective Install for Networking** to install it or upgrade it to TCP/IP 4.1. TCP/IP 4.1 is available from the IBM PS Software Services Web site.

After installing these packages, install all fixes. Refer to “Using OS/2 systems” on page 347 for a list of required software. You must have these prerequisites installed before installing the Tivoli desktop.

To install the Tivoli desktop on an OS/2 machine, you must perform the following steps:

1. Run the preinstallation script.
2. Install the Tivoli desktop.
3. Add the program icons to the OS/2 desktop.

These procedures are described in the following sections.

Running the preinstallation script

Before installing the Tivoli desktop, you must run the preinstallation script to prepare your OS/2 system. This script copies migration data files to the appropriate location on your hard disk. To run the preinstallation script from the Tivoli Management Framework 2 of 2 CD, perform the following steps:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. Start a WIN-OS/2 session.
3. Run *x*:\desktop\os2\install\install.cmd, where *x* is the drive letter of your CD-ROM drive.
4. Follow the instructions presented on the screen.

Installing the Tivoli desktop on OS/2 operating systems

To install Tivoli Desktop for Windows from the Tivoli Management Framework 2 of 2 CD, perform the following steps:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. Start a WIN-OS/2 session.
3. Select **Run** from the **File** menu.
4. In the **Command Line** field, type `x:\desktop\os2\disk1\setup.exe`, where *x* is the drive letter of the CD-ROM drive.
5. Click **OK**.
6. Follow the instructions presented on the screen.

Adding the icons to the OS/2 desktop

To create the Tivoli program icons on an OS/2 desktop you must run the Add Programs migration utility to ensure correct configuration. To create the Tivoli icons, perform the following steps:

1. Right-click the OS/2 desktop and select **System Setup**.
2. From the **System Setup** folder, double-click the **Add Programs** icon. The Add Programs window is displayed.
3. Select **Search for and select programs to add** and click **OK**. The Search for Programs window is displayed.
4. Do the following:
   a. In the **Drives** list, select the drive where the Tivoli desktop was installed.
   b. In the **Use Database** field, change the database file name to `tivoli.dat`.
   c. Under **Program Type**, select the **Windows Programs** and **Windows Groups** check boxes. The other check boxes should not be selected.
   d. Click **OK**.

      After the hard disk drive is scanned, the Select Programs window displays a listing of the programs found on the drive.
   e. Find and select the following programs from this list:

      Tivoli (Windows)
      Tivoli Readme (Windows)
      Tivoli Uninstall (Windows)
   f. Click **OK**. A message is displayed when the program icons are successfully added to the desktop.
5. Click **Cancel** or **Exit** to close the Add Programs utility.

Starting the Tivoli desktop

Before starting the Tivoli desktop, ensure that the system that you want to connect to accepts remote connections. To determine whether a system accepts remote connections, follow the procedure in “Enabling and disabling remote connections” on page 32.

Use the following procedure to start the Tivoli desktop:

1. For Windows operating systems—From the **Start** menu, select **Programs** → **Tivoli** → **Tivoli**.

   For OS/2 operating systems—After adding the icons to your OS/2 desktop, click the **Tivoli (Windows)** icon.

   The Tivoli Management Environment window is displayed.
2. In this window, perform the following steps:
   a. In the **Host Machine** field, type the fully qualified host name of the managed node or type the host name of the Tivoli server. Do not type the IP address.
   b. In the **Log In As** field, type your domain-qualified user name (Windows) or the account name (UNIX) for the managed node. This field is case sensitive.
   c. In the **Password** field, type the password for the specified user or account. This field is case sensitive.

      Note: Ensure that there are no extra spaces in this field.
   d. Click **OK**.

Logging In to a Windows managed node

When you log in to a Windows managed node, the user name must be domain-qualified. If the machine is not the Tivoli server, the host name must be fully qualified. For example, you would type `djatzlau1.dev.tivoli.com` (the fully qualified name) in the **Host Machine** field and type `DJATZLAU1\debbiej` (the domain-qualified user name) in the **Log In As** field to connect to managed node djatzlau1.

Logging In to a UNIX managed node

When you log in to a UNIX system and this system is not the Tivoli server, the host name must be fully qualified. For example, you would type `cygnus` in the **Host Machine** field and `debbiej` in the **Log In As** field to connect to Tivoli server cygnus.

Note: Because cygnus is the Tivoli server, the host name is not fully qualified. Because cygnus is a UNIX operating system, the user name is not domain-qualified.

Using Pluggable Authentication Modules (PAM) on UNIX-like systems

On UNIX-based systems (AIX, Solaris, HP, Linux), the Tivoli management framework attempts to authenticate user logins, using Tivoli Desktop for Windows (DSWIN) or the Java client framework (JCF), by using one of the following techniques, in this order:

1. Using traditional authentication, using getpwnam() or getspnam(), which rely on /etc/passwd and /etc/shadow.

2. Using PAM authentication, which relies on configuation files in varying directories. For Linux, the configuration file is /etc/pam.d/oserv. For Solaris, HP, and AIX, the configuration file is /etc/pam.conf.

PAM authentication allows operating system (OS) administrators to use a variety of underlying authentication mechanisms, such as Kerberos and LDAP. However, the OS integration, with PAM and Kerberos, or PAM and anything else, is related to OS configuration. For detailed information about PAM and OS integration and PAM configuration, refer to the operating system documentation.

When /etc/pam.conf is configured correctly, the Tivoli management framework PAM authentication calls can successfully integrate with Kerberos, LDAP, and other authentication methods. But, there are some limitations; for example, multi-threaded PAM modules and modules that rely on conflicting libraries are not supported.

For AIX, Solaris, and HP systems, Tivoli management framework uses ″other″ entries in /etc/pam.conf . However, it is possible to add oserv-specific entries using the oserv keyword in /etc/pam.conf. (see AIX below for an example.) Note that all four attributes are marked as required; if one fails, the login is not successful.

```
Custom oserv entries for AIX added to /etc/pam.conf:
#
# Authentication
#
...
oserv   auth    required       /usr/lib/security/pam_aix

#
# Account Management
#
...
oserv   account required       /usr/lib/security/pam_aix

#
# Password Management
#
...
oserv   session required       /usr/lib/security/pam_aix

#
# Session Management
#
...
oserv   session required       /usr/lib/security/pam_aix
```

AIX has an additional requirement to enable PAM. Add the **auth_type** attribute under **usw** stanzas in the /etc/security/login.cfg file:

```
usw:
    ...
    auth_type = PAM_AUTH
```

Note: This attribute is the one that oserv will use to check to see if PAM is available and enabled on the AIX systems.

Starting the Tivoli desktop when not using port 94

If you did not use the default port number (94) when you installed your Tivoli region, the Tivoli desktop cannot connect to the Tivoli server or any managed node in the region without specifying the port number. You can specify the port number in the icon properties or from the command line.

To add this option to the Tivoli desktop icon, modify the properties of the icon. To modify, perform the following steps:

1. Create an icon as described in “Adding the icon to a Windows desktop” on page 67.
2. Right-click the **Shortcut to tivoli.exe** icon and select **Properties**. The Properties window is displayed.
3. Click the **Shortcut** tab.
4. In the **Target** field, add **–port** *port_number*, where *port_number* is the port for the Tivoli server and managed nodes. For example if you were using port 8765, the **Target** field would contain:

   ```
   "C:\Program Files\Tivoli\Desktop\tivoli.exe" -port 8765
   ```

To use this option when starting the Tivoli desktop from the command line, add **–p** *port_number* to the **tivoli** command. For example if you were using port 8765, you would perform the following steps:

1. Change to the directory containing **tivoli.exe** (the default is C:\Program Files\Tivoli\Desktop)
2. Enter the following command:

   ```
   tivoli -port 8765
   ```

Uninstalling Tivoli Desktop for Windows

This section contains instructions for uninstalling Tivoli Desktop for Windows on Windows and OS/2 operating systems.

Uninstalling the Tivoli desktop from Windows operating systems

To uninstall Tivoli Desktop for Windows, perform the following steps:

1. From the **Start** menu, select **Programs** → **Tivoli** → **Tivoli Uninstall**.
2. Follow the instructions presented on the screen.

Uninstalling the Tivoli desktop from OS/2 operating systems

To uninstall Tivoli Desktop for Windows, perform the following steps:

1. Select the **Tivoli Uninstall** icon from the **Tivoli Program** group.
2. Follow the instructions presented on the screen.

Part 3. Tivoli Software Installation Service

Chapter 13. Using response files 143

Chapter 14. Using Tivoli Software Installation Service log files 171

Chapter 15. Checking prerequisites 179

Chapter 9. Introduction to Tivoli Software Installation Service

This chapter contains the following information:

- An introduction to the parts and features of Tivoli Software Installation Service (SIS)
- Information required to plan for your deployment of Tivoli Software Installation Service

Overview of Tivoli Software Installation Service

Tivoli Software Installation Service provides an easy-to-use and fast process for installing Tivoli Enterprise software. It provides increased functionality over the installation mechanisms provided by Tivoli Management Framework.

This section contains the following information:

- The components of Tivoli Software Installation Service
- The features provided by Tivoli Software Installation Service

Components of Tivoli Software Installation Service

Tivoli Software Installation Service consists of depots and clients.

The *Tivoli Software Installation Service client* provides the user interface to Tivoli Software Installation Service. You can use the client either from the command line or through the Tivoli Software Installation Service console, a graphical user interface. You can perform most Tivoli Software Installation Service tasks using your choice of the commands or the console.

The *Tivoli Software Installation Service depot* manages the installation of managed nodes, endpoints, gateways, and Tivoli Enterprise products and patches. It contains the *install repository* where reusable installation images are stored. It also coordinates planning and performing installations by multiple clients.

For detailed information about planning for the depots and clients in your Tivoli region, refer to "Planning for Tivoli Software Installation Service" on page 78.

Features of Tivoli Software Installation Service

Tivoli Software Installation Service has the following features that help you quickly install Tivoli Enterprise software:

- Batch installation
- Enhancements to the dependency checks provided by Tivoli Management Framework
- Prerequisite checking to verify installation criteria
- Command line and graphical user interface for installation tasks
- Response files that facilitate large-scale installations from the command line
- An install repository for storing Tivoli product images
- Customizable, operating system-specific defaults for the installation options for Tivoli Enterprise software
- Customizable settings (preferences) for the depot and Tivoli Software Installation Service client

- The ability to install Tivoli Enterprise software to endpoints

The following sections provide a high-level description of these features and how to use them. Where necessary, references to more detailed information is provided.

Batch installation

An important benefit of Tivoli Software Installation Service is that it allows you to perform installations in groups, combining the installation of managed resources, products, and patches to a variety of machines. Very large installations of many products to many machines can be performed with little intervention on your part. Select all the products you want to install and the machines you want to install them on, start the installation, and wait for Tivoli Software Installation Service to provide a summary of installation results.

In contrast, the Tivoli Management Framework installation mechanisms require that you first create your managed resources, and then install each Tivoli product or patch to a group of machines separately. Installation is a serial process; you must wait for each installation to complete before starting the next installation.

Enhanced dependency checking

Tivoli Software Installation Service provides enhancements to the dependency checking performed by Tivoli Management Framework. Dependency checks ensure, for example, that you do not install a patch if you have not installed the product or patches on which it depends. Tivoli Software Installation Service checks these dependencies when you build the installation worksheet or install from a response file, and provides the following additional capabilities:

- Checks dependencies on products installed on the Tivoli server.

  For example, before installing any part of Inventory on a managed node, you must first install Inventory on the Tivoli server. Tivoli Management Framework cannot check this dependency.
- Checks dependencies on products installed on any managed node in the Tivoli region.

  For example, before installing IBM Tivoli Enterprise Console® event console on a managed node, you must first install event server on any managed node in the Tivoli region. Tivoli Management Framework cannot check this dependency.
- Automatically sequences the installation order of machines and products to meet dependency requirements.

  Tivoli Software Installation Service examines the dependencies among the products and machines in an installation set and performs the installations in the correct order. With Tivoli Management Framework, you must understand these dependencies and perform separate installations in the correct order.

Prerequisite checking

Tivoli Software Installation Service provides extensive prerequisite checking before installing each product. With this feature, you can check for commonly encountered causes of installation failure. By detecting and correcting these problems before starting an installation, particularly when it involves many products or machines, you can save considerable installation and troubleshooting time.

Tivoli Software Installation Service checks prerequisites before any installation, unless you explicitly disable them. You can also check prerequisites without installing any software.

For example, when creating a managed node on a machine running a Windows operating, the directory containing the Tivoli object database must be on a local hard drive. Tivoli Software Installation Service checks this prerequisite before attempting this machine. In addition to the prerequisite checks provided by Tivoli Software Installation Service, you can create prerequisite checks specific to your environment. For detailed information about the defined prerequisites and information about creating, modifying, and enabling prerequisite checks, refer to Chapter 15, "Checking prerequisites," on page 179.

When a prerequisite check detects a potential installation failure, a message is displayed and written to the log file for that machine. Installation on that machine stops, but installation on other machines continues.

The Tivoli Software Installation Service console and command line interface

You can perform most installation tasks using either the Tivoli Software Installation Service console or commands.

The console is a good choice in the following circumstances:

- You want a visual display of product dependencies, with specific explanations of unresolved dependencies.
- You want to install products to a few machines.
- You want to create a response file template based on machines that are currently available in your network.
- You prefer using a graphical user interface (GUI).

The command line is a good choice in the following circumstances:

- You want to automate installation tasks using scripts.
- You want to quickly define a large number of new machines.
- You want to quickly install to a large number of machines.
- You want to create a response file template for machines or products that are not currently available in your network.
- You prefer using a command line interface.

Response files

A *response file* is a text file containing information needed to install products. You can use a response file to install products from the command line or to add a large number of machines to Tivoli Software Installation Service. Response files are especially useful for installing products to a large number of machines. You can also import a response file into the Tivoli Software Installation Service console to quickly populate the installation worksheet.

For detailed information about response files, refer to Chapter 13, "Using response files," on page 143.

Install repository

The install repository, a part of the Tivoli Software Installation Service depot, is a directory structure that contains Tivoli Enterprise product and patch installation images. Centrally storing installation information provides a fast and easy method of installing products across a distributed environment of heterogeneous operating systems.

The install repository enables you to use a single, centralized repository that serves multiple Tivoli Software Installation Service clients. An install repository can be shared between depots in different Tivoli regions but not between depots in the same region. However, multiple depots in the same region can connect to one depot and use its install repository.

You can optionally restrict an install repository to read-only access. Refer to "Modifying preferences" on page 102 for information about making an install repository read-only.

Because the install repository stores installation images in the same format as the product CD, you can use the installation images in the install repository to perform an installation using the Tivoli desktop or the Tivoli Management Framework commands.

Customizable defaults for Tivoli Enterprise software

Tivoli Software Installation Service enables you to customize the default settings of installation options for Tivoli Enterprise software. You can set the defaults separately for each operating system supported by the software. For details, refer to "Viewing and customizing installation options" on page 119.

You can also easily override these defaults for a specific installation. This is described in "Overriding default installation options for one machine" on page 137 and in "[alias] section" on page 148.

Preferences

The Tivoli Software Installation Service preferences enable you to customize your Tivoli Software Installation Service depot and Tivoli Software Installation Service client configurations. You can modify these characteristics by changing values in the SIS Preferences window of the Tivoli Software Installation Service console or by using the **wsisprefs** command. Additional information about modifying preferences is provided in Chapter 11, "Configuring Tivoli Software Installation Service," on page 99.

Installing Tivoli Enterprise software on an endpoint

In most cases, Tivoli Enterprise software is not installed directly on an endpoint. Endpoint methods are stored on the gateway and downloaded to the endpoint as required. This reduces the footprint on the endpoint. To upgrade methods on an endpoint, upgrade the gateway, which then provides updated methods to the endpoint.

However, some Tivoli Enterprise software must be installed directly on an endpoint. The product documentation indicates when this is necessary.

Use Tivoli Software Installation Service to install products to endpoints. You cannot install a product to an endpoint using the Tivoli desktop or Tivoli Management Framework commands.

Planning for Tivoli Software Installation Service

Planning your deployment of Tivoli Software Installation Service involves the following considerations:

- The number and location of the Tivoli Software Installation Service depots
- The location of the install repository for each depot
- Authorizing the depot
- The number and location of the Tivoli Software Installation Service clients
- The location of the log directory for each client

Number and location of Tivoli Software Installation Service depots

To use Tivoli Software Installation Service, you must have at least one Tivoli Software Installation Service depot. You can install the depot on any managed node running a supported operating system, including the Tivoli server. Refer to the *Tivoli Management Framework Release Notes* for a list of supported operating systems. There can be one or more depots in your Tivoli environment. Generally, one depot is sufficient.

When initially deploying a new region, it is simplest to install the depot on the Tivoli server and use it to install managed nodes, gateways, endpoints, and Tivoli products on the other machines in your Tivoli environment.

However, because performing an installation of many products to many machines can require a lot of system and network resources, you might not want to perform installations using a depot on the Tivoli server of an active region. After using the depot on a new Tivoli server to do the initial deployment, you can create a depot on a managed node elsewhere in your region to reduce the load on the Tivoli server.

A depot can accept simultaneous connections from multiple Tivoli Software Installation Service clients. The clients can be on the different machines, as shown in Figure 1. Multiple clients can be started on a single machine, as shown in Figure 2. For example, this can happen when multiple users on a UNIX machine each start the Tivoli Software Installation Service console, or when a single user on a Windows operating system uses the command line interface at the same time he has the console open. The number of simultaneous connections is controlled by a preference that you can modify.

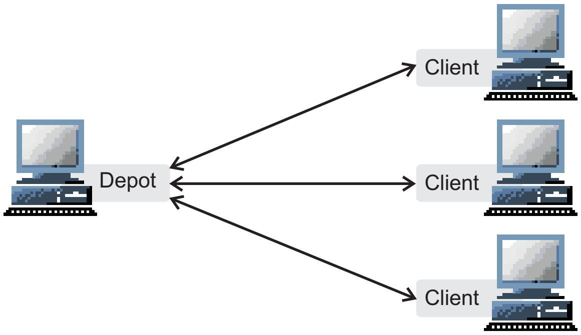


Figure 1. Clients on multiple machines simultaneously connecting to one depot

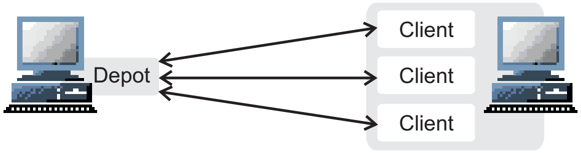


Figure 2. Multiple clients on one machine simultaneously connecting to one depot

A depot uses minimal system resources while connected to a client to plan an installation or manipulate the install repository. The depot process stops when it has not been used for five minutes. It is automatically started when contacted by a client. You cannot change the timeout value.

If you have slow network connections, place the depot on a machine on the same side of the slow link as the machines on which you want to install Tivoli Enterprise software. This can reduce network traffic caused by an installation and increase the speed with which products are installed. Place the client on a machine local to the administrator. This is illustrated in Figure 3.

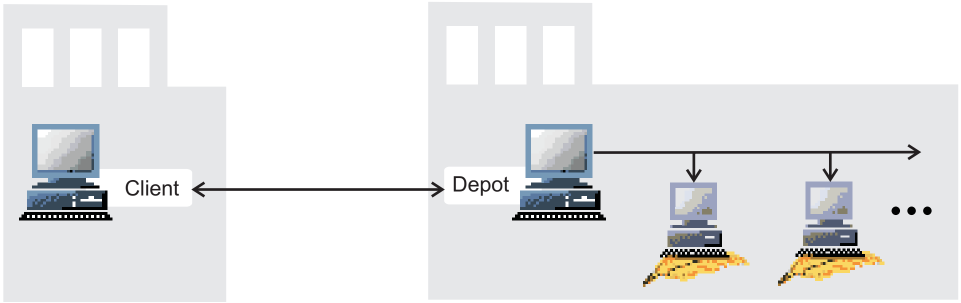


Figure 3. Placing a depot local to the machines it installs

Location of the Install Repository

Each Tivoli Software Installation Service depot has its own install repository, which is specified during the installation. Depots in different Tivoli regions can share an install repository.

An install repository can be in either a local or a remote file system. In general, it is preferable to place an install repository in a file system that is local to the machine hosting the depot to reduce the network traffic when planning or performing an installation.

It is a good idea to place the install repository in its own file system or disk partition.

Ensure that the file system in which you place an install repository has sufficient space to hold the required installation images. A Tivoli product CD contains files for installing on all supported operating systems. You can reduce the space required in the install repository by importing only the files needed to install a product on the operating systems available in your Tivoli region.

If you run out of space in the file system that contains the install repository, you can move the install repository to a new location. This is described in “Changing the location of the install repository” on page 109.

The following restrictions apply to the install repository:

- You cannot share an install repository between two depots in the same Tivoli region. However, you can share an install repository with a depot in another region, as illustrated in Figure 4 on page 81.

  To share an existing install repository with a depot in another Tivoli region, specify the local path to the install repository directory that was created when the other depot was installed. You do not need to explicitly allow sharing of an install repository. You need only ensure that both machines have physical access to the directory containing the install repository that you want to share.

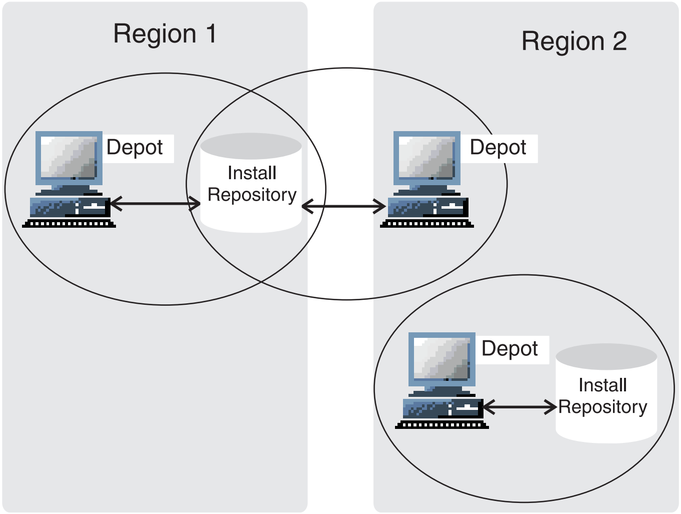


Figure 4. Sharing an install repository between depots in different Tivoli regions

- When you install the depot, you cannot reuse the install repository from a *previous*, non-client/server release of Tivoli Software Installation Service.

  You can reuse the installation images in the old install repository by copying them during the installation of the client (using the upgrade image) or by importing them after the client is installed.
- When you uninstall a depot, its install repository is not removed. Until you manually remove the install repository, you can reuse it as the install repository of another depot of the same version or to manually check the log files.

Authorizing the Tivoli Software Installation Service depot

The Tivoli Software Installation Service depot runs as root on UNIX operating systems and as Administrator on Windows operating systems. To provide the depot with the authorization it needs to access Tivoli resources, you must add that user as a login for a Tivoli administrator.

For example, if you have two depots, one on the UNIX machine cygnus and another on the Window machine djatzlau1, the following logins must be defined:

```
root@cygnus.dev.tivoli.com
DJATZLAU1\Administrator@djatzlau1.dev.tivoli.com
```

The logins do not all have to be created for the same Tivoli administrator, and it is not necessary to create them for the administrator that will use Tivoli Software Installation Service to perform installations.

The Tivoli Software Installation Service client uses the same authorization mechanisms as other Tivoli commands. The administrator using the clients must have a login for the user ID on the machine where the client is installed, and the appropriate roles must be assigned to that administrator.

Number and location of Tivoli Software Installation Service clients

You must have at least one Tivoli Software Installation Service client. You can install the client on any managed node running a supported operating system, including the Tivoli server. Refer to the *Tivoli Management Framework Release Notes* for a list of supported operating systems.

In general, you install the client on a managed node used by the Tivoli administrator who plans and deploys products in your Tivoli environment. For best performance of the Tivoli Software Installation Service console, do not separate the client and depot by a slow link or an overloaded network.

A client can connect to any depot in the Tivoli region, but can connect to only one depot at a time. If the region contains multiple depots, specify which one to use when you start the console or run a Tivoli Software Installation Service command.

The client can be on the same machine as the depot , or they can be on different machines.

A client can connect only to a depot in the same Tivoli region. If you connect regions, SisDepot objects in a remote region are not available to clients in the local region.

Multiple client can simultaneously connect to a depot. However, only one client at a time can perform an installation or import or remove products from the install repository. Additional clients can plan an installation, create or import response files, add or remove the definition of unmanaged machines, or change the default values of installation options.

Location of the client log directory

The Tivoli Software Installation Service client logs contain both log files generated on the client and cached copies of depot logs that are being viewed by the client. Select a directory with sufficient space. Periodically clean up the directory by removing old log files.

The client log directory is not deleted when you uninstall the client.

Chapter 10. Installing Tivoli Software Installation Service

This chapter provides the steps required to install the components of Tivoli Software Installation Service (SIS). It describes how to install each component using the following installation mechanisms:

- Tivoli desktop
- Tivoli command line
- An existing installation of Tivoli Software Installation Service

These instructions describe how to install or upgrade Tivoli Software Installation Service. Before installing Tivoli Software Installation Service, read the planning information in "Planning for Tivoli Software Installation Service" on page 78.

Installing Tivoli Software Installation Service

You must install each component of Tivoli Software Installation Service on a managed node that is running a supported operating system. Refer to the *Tivoli Management Framework Release Notes* for a list of supported operating systems. The steps for installing Tivoli Software Installation Service are as follows:

1. Ensure that the Java components required by the Tivoli Software Installation Service depot and client are installed. The following components must be installed on any system that will host a depot or client:
   - Java 1.3 for Tivoli
   - Tivoli Java Client Framework 4.1.1
2. Install one or more Tivoli Software Installation Service depots.
3. For each depot, add the user root or Administrator to any administrator. For more information, refer to "Authorizing the Tivoli Software Installation Service depot" on page 81.
4. Install one or more Tivoli Software Installation Service clients.

After you install Tivoli Software Installation Service, you can use it to create managed resources and to install Tivoli software in your Tivoli region.

The following table provides the context and authorization roles required to install any product.

Activity	Context	Required Role
Install using the Tivoli desktop or command line	Tivoli server	**install_product** or **super**
Install using Tivoli Software Installation Service	Tivoli server	**user** plus one of **super**, **senior**, **install_client**, or **install_product**

The following table describes the components of Tivoli Software Installation Service. This table lists the name of the index (.IND) file that you use to install the component, the name of the component, and the registered product tag that you use to uninstall the product. The SISDEPOT.IND and SISCLNT.IND files are shipped in the /JAVA subdirectory of the Tivoli Management Framework 2 of 2 CD. The SISD411U.IND and SISC411U.IND files are shipped in the /JAVA subdirectory of the Tivoli Management Framework upgrade CDs.

Table 3. Components of Tivoli Software Installation Service

.IND File Name	Component Name	Tag
SISDEPOT	Tivoli Software Installation Service Depot, Version 4.1.1	SISDepot
SISCLNT	Tivoli Software Installation Service Client, Version 4.1.1	SISCLNT
SISD411U	Tivoli Software Installation Service Depot Upgrade to 4.1.1	SISDPT411
SISC411U	Tivoli Software Installation Service Client Upgrade to 4.1.1	SISCLT411

The following tables provide the installation options that can be specified when you install the Tivoli Software Installation Service components. Detailed instructions for installing these components are provided in “Installing the Tivoli Software Installation Service depot” on page 88 and “Installing the Tivoli Software Installation Service client” on page 92.

Table 4. Installation options for the depot

	GUI Field Name	CLI Installation Option
	Description	
•	**Install Repository**	**IRDIR**
	Specifies the full path name of the install repository. **Note:** You must specify this variable. If you do not, the Tivoli Software Installation Service depot will appear to install but it will not work properly. To share an existing install repository with a depot in a different Tivoli region, specify the local path to the shared install repository. You do not need to explicitly allow sharing of an install repository. You need to ensure that both machines have physical access to the directory. For considerations and restrictions about where to place the install repository, refer to “Location of the Install Repository” on page 80.	

Table 5. Installation options for the client

	GUI Field Name	CLI Installation Option
	Description	
•	**Client Log Directory**	**CLIENTLOGDIR**
	Specifies the directory on the client in which to place client log files and cached copies of depot logs when they are accessed by the log viewer. If the directory does not exist, it is created. If you do not specify a directory, you are prompted the first time the client is started.	

The following sections describe how to install the components of Tivoli Software Installation Service using the Tivoli desktop, the **winstall** command, and Tivoli Software Installation Service, if another copy is already installed.

Note: Before you can install Tivoli Software Installation Service, you must install the required Java components.

Installing the required Java components

Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1 enable Java applications such as Tivoli Software Installation Service to communicate with Tivoli Management Framework. You must install both of these components on each system that will run the Tivoli Software Installation Service depot or the Tivoli Software Installation Service client. These components are located in the /JAVA subdirectory of the Tivoli Management Framework 2 of 2 CD. For additional details about the Java components, refer to “Installing Java components” on page 222.

These components are used by other Tivoli applications, so they might already be installed. If so, you do not need to reinstall them. To determine whether the components are already installed, use the following command:

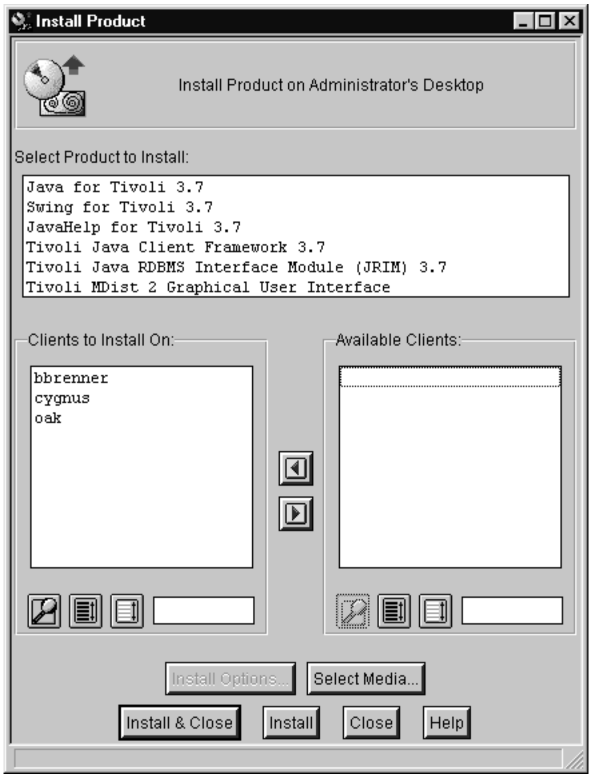
```
wlsinst -p -h
```

The following sections provide instructions for installing these Java components using the Tivoli desktop and the command line. You can also use an existing copy of Tivoli Software Installation Service to install the Java components.

Installing Java components from the Tivoli desktop

To install Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1 from the Tivoli desktop, perform the following steps from any Tivoli desktop:

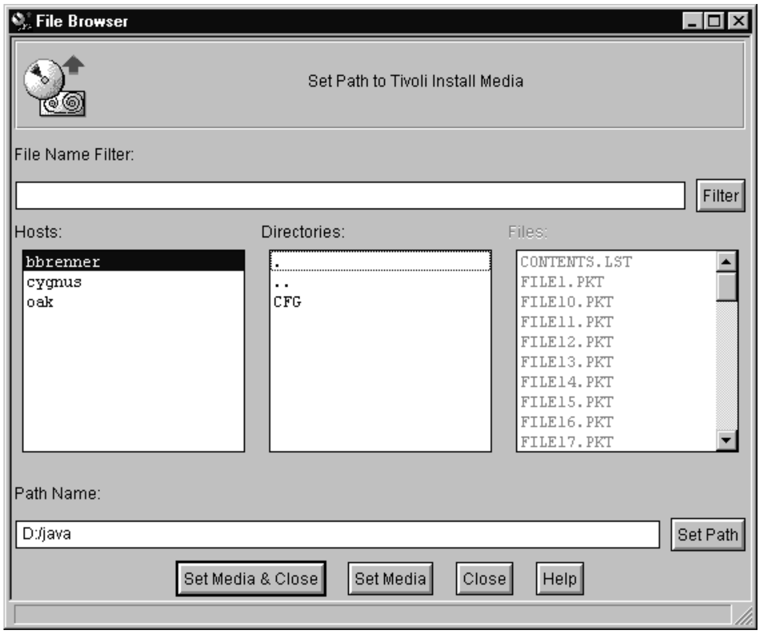
1. From the **Desktop** menu, select **Install → Install Product** to display the Install Product window.



Java for Tivoli 3.7
Swing for Tivoli 3.7
JavaHelp for Tivoli 3.7
Tivoli Java Client Framework 3.7
Tivoli Java RDBMS Interface Module (JRIM) 3.7
Tivoli MDist 2 Graphical User Interface

bbrenner
cygnus
oak

2. If the **Select Product to Install** list is empty or does not contain **Java 1.3 for Tivoli** and **Tivoli Java Client Framework 4.1.1**, continue with step 3. If the Java components are listed, skip to step 7.
3. Click **Select Media** to display the File Browser window. Use this window to specify the path to the installation image for the Java components.


File Browser

Set Path to Tivoli Install Media

File Name Filter:

Hosts:
bbrenner
cygnus
oak

Directories:
.
..
CFG

Files:
CONTENTS.LST
FILE1.PKT
FILE10.PKT
FILE11.PKT
FILE12.PKT
FILE13.PKT
FILE14.PKT
FILE15.PKT
FILE16.PKT
FILE17.PKT

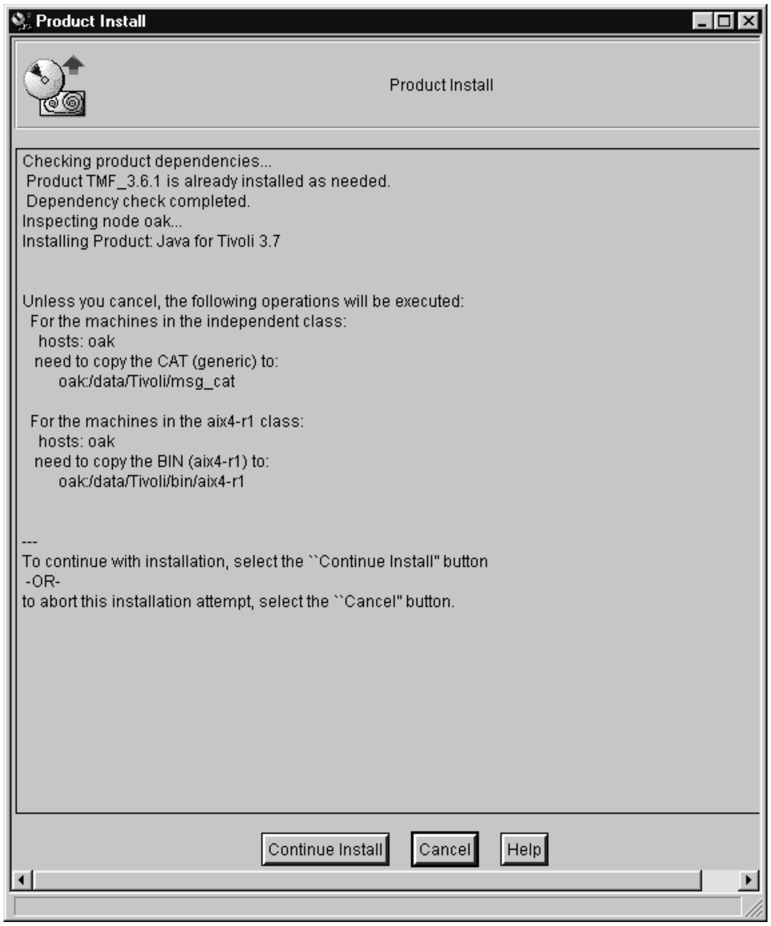
Path Name: D:/java

4. In the **Hosts** list, select the host on which the installation image is mounted.
5. Navigate to the directory that contains the installation image. The installation directory contains the product index (.IND) file.

   Double-click directory names in the **Directories** list until the installation images are shown in the **Files** list.

   Alternatively, if you know the path to the installation image, type the full path in the **Path Name** field. Click **Set Path** to list the contents of the specified directory.
6. Click **Set Media & Close** to save the path and return to the Install Product window. The window now lists **Java 1.3 for Tivoli** and **Tivoli Java Client Framework 4.1.1**, which indicates that they are available for installation.
7. In the **Select Product to Install** list, select **Java 1.3 for Tivoli**.
8. Use the arrow buttons to select the machines on which to install this product. Move the clients from one list (**Available Clients** or **Clients to Install On**) to the other by selecting the client name and clicking the left- or right-arrow button. **Java 1.3 for Tivoli** is installed on the clients in the **Clients to Install On** list.
9. Click **Install** to install **Java 1.3 for Tivoli**.

   The installation process displays a Product Install window similar to the following.



```
Checking product dependencies...
 Product TMF_3.6.1 is already installed as needed.
 Dependency check completed.
Inspecting node oak...
Installing Product: Java for Tivoli 3.7


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts: oak
   need to copy the CAT (generic) to:
        oak:/data/Tivoli/msg_cat

  For the machines in the aix4-r1 class:
    hosts: oak
   need to copy the BIN (aix4-r1) to:
        oak:/data/Tivoli/bin/aix4-r1


---
To continue with installation, select the ``Continue Install'' button
 -OR-
to abort this installation attempt, select the ``Cancel'' button.
```

This window lists the operations that will occur during the installation and any problems that you might want to correct before continuing the installation. If you want to correct any problems, click **Cancel**.

10. Click **Continue Install** to perform the installation. The Product Install window displays status information as the installation proceeds.

    When the installation is complete, the Product Install window displays a completion message and the **Cancel** button becomes the **Close** button.

11. Click **Close** to close the status window and return to the Product Install window.
12. In the **Select Product to Install** list, select **Tivoli Java Client Framework 4.1.1**.
13. Use the arrow buttons to select the Tivoli server or managed nodes on which to install this product. Move the clients from one list (**Available Clients** or **Clients to Install On**) to the other by selecting the client name and clicking the left- or right-arrow button. **Tivoli Java Client Framework 4.1.1** is installed on the clients in the **Clients to Install On** list.
14. Click **Install & Close** to install **Tivoli Java Client Framework 4.1.1** and close the Install Product window.

    The installation process displays a Product Install window, which lists the operations that will occur during the installation and any problems that you might want to correct before continuing the installation. If you want to correct any problems, click **Cancel**.

15. Click **Continue Install** to perform the installation. The Product Install window displays status information as the installation process proceeds.

When the installation is complete, the Product Install window displays a completion message and the **Cancel** button becomes the **Close** button.

16. Click **Close** to close the Product Install window.

Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1 are installed. You can now install the Tivoli Software Installation Service depot or client.

Installing Java components using the winstall command

To install Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1, use the **winstall** command as follows:

```
winstall -c source_dir -i JRE130 node
winstall -c source_dir -i JCF411 node
```

where:

–c *source_dir*
Specifies the path to the installation image. Both Java components are in the same directory on the CD.

–i JRE130
Specifies the index file for Java 1.3 for Tivoli, JRE130.IND.

–i JCF411
Specifies the index file for Tivoli Java Client Framework 4.1.1, JCF411.IND.

node Specifies the managed node on which to install these products.

The following example shows how to use the **winstall** command to install Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1 on managed node wave from the directory /cdrom/cdrom0/JAVA:

```
winstall -c /cdrom/cdrom0/JAVA -i JRE130 wave
winstall -c /cdrom/cdrom0/JAVA -i JCF411 wave
```

Refer to the **winstall** command in the *Tivoli Management Framework Reference Manual* for more information.

Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1 are installed. You can now install the Tivoli Software Installation Service depot or Tivoli Software Installation Service client.

Installing the Tivoli Software Installation Service depot

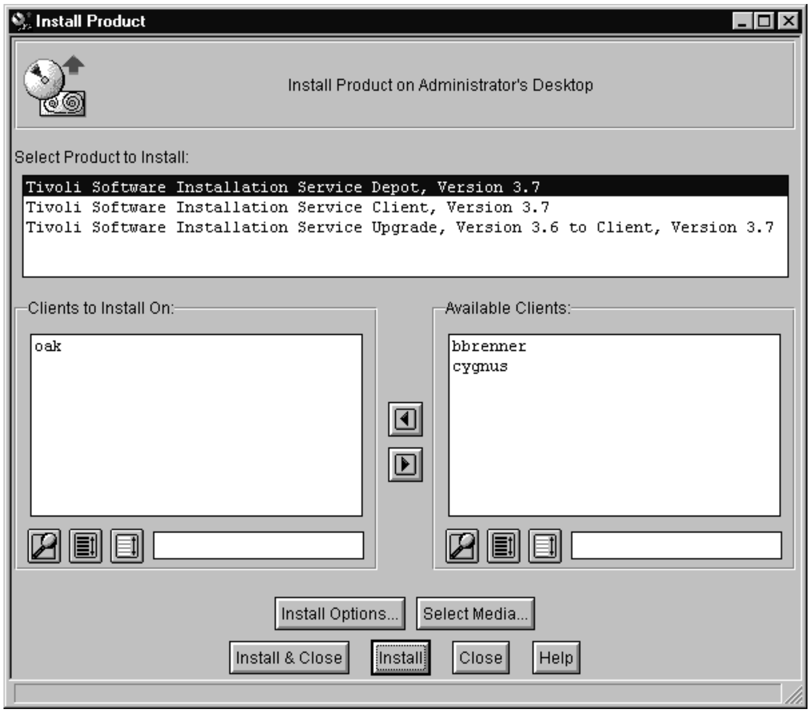
You can install the Tivoli Software Installation Service depot on any managed node in your Tivoli region, including the Tivoli server.

The following sections provide instructions for installing the depot using the Tivoli desktop and the command line. You can also use an existing copy of Tivoli Software Installation Service to install the depot.

Installing from the Tivoli desktop

To install the Tivoli Software Installation Service depot from the Tivoli desktop, perform the following steps from any desktop:

1. From the **Desktop** menu, select **Install → Install Product** to display the Install Product window.


Install Product

Install Product on Administrator's Desktop

Select Product to Install:

```
Tivoli Software Installation Service Depot, Version 3.7
Tivoli Software Installation Service Client, Version 3.7
Tivoli Software Installation Service Upgrade, Version 3.6 to Client, Version 3.7
```

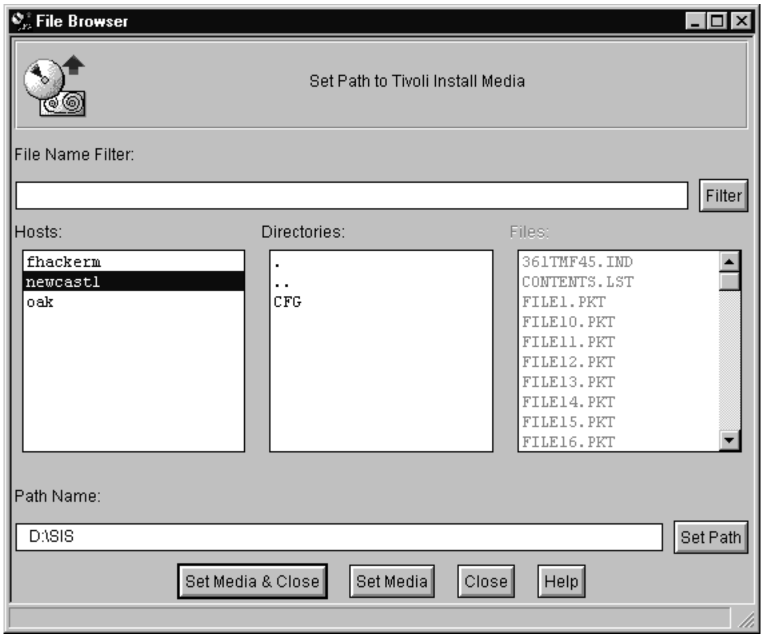
Clients to Install On:

```
oak
```

Available Clients:

```
bbrenner
cygnus
```

2. If the **Select Product to Install** list is empty or does not contain **Tivoli Software Installation Service Depot, Version 4.1.1**, continue with step 3. If the Tivoli Software Installation Service depot component is listed, skip to step 7.
3. Click **Select Media** to display the File Browser window. Use this window to specify the path to the installation image for the Tivoli Software Installation Service depot.


File Browser

Set Path to Tivoli Install Media

File Name Filter:

Hosts:

```
fhackerm
newcastl
oak
```

Directories:

```
.
..
CFG
```

Files:

```
361TMF45.IND
CONTENTS.LST
FILE1.PKT
FILE10.PKT
FILE11.PKT
FILE12.PKT
FILE13.PKT
FILE14.PKT
FILE15.PKT
FILE16.PKT
```

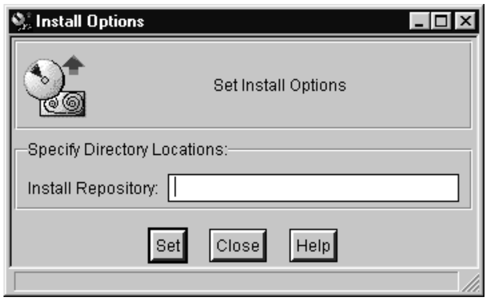
Path Name: D:\SIS

4. In the **Hosts** list, select the host on which the installation image is mounted.

5. Navigate to the directory that contains the installation image. The installation directory contains the product index (.IND) file.

   Double-click directory names in the **Directories** list until the installation images are shown in the **Files** list.

   Alternatively, if you know the path to the installation image, type the full path in the **Path Name** field. Click **Set Path** to list the contents of the specified directory.
6. Click **Set Media & Close** to save the path and return to the Install Product window. The window now lists **Tivoli Software Installation Service, Version 4.1.1**, which indicates that it is available for installation.
7. In the **Select Product to Install** list, select **Tivoli Software Installation Service Depot, Version 4.1.1**. The Install Options window is displayed.

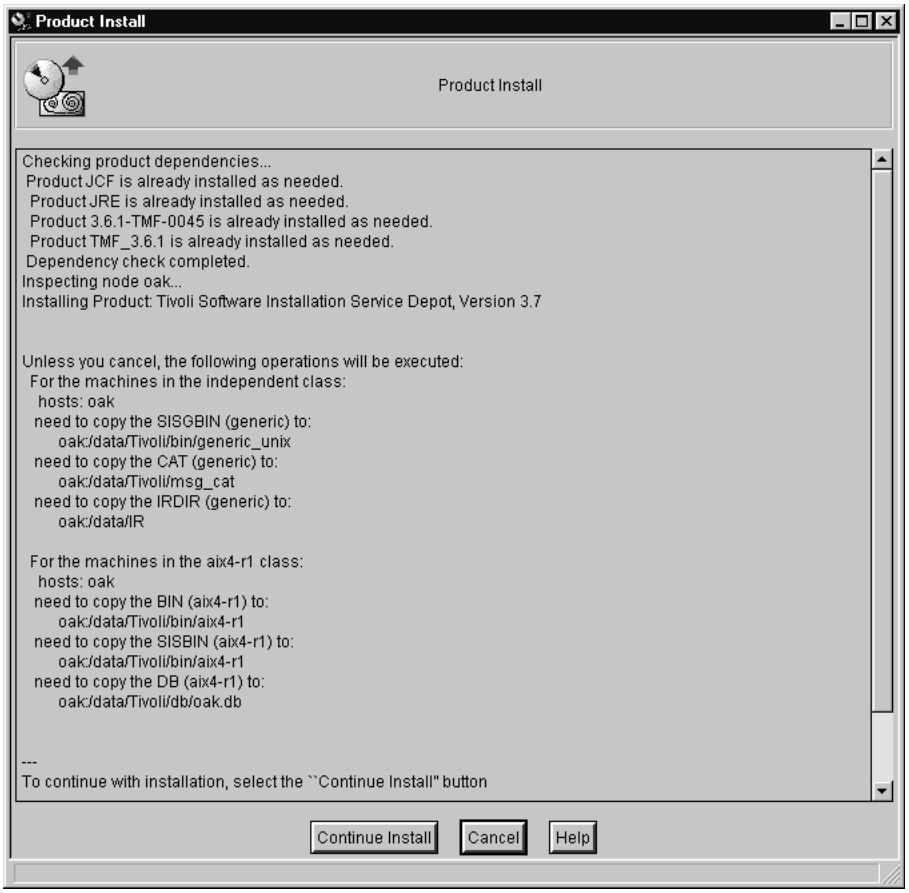


8. In the **Install Repository** field, type the path of the directory to contain the install repository. If the directory does not already exist, it is created during the installation. You must specify a value. If you do not, the depot will appear to install but will not work properly.

   To share an existing install repository with a depot in a different Tivoli region, specify the local path to the shared install repository directory. You do not need to explicitly allow sharing of an install repository, just ensure that both machines have physical access to the directory.

   For considerations and restrictions about where to place the install repository directory, refer to “Location of the Install Repository” on page 80.
9. Click **Set** to save the path information and return to the Install Product window.
10. Use the arrow buttons to select the machines on which to install this product. Move the clients from one list (**Available Clients** or **Clients to Install On**) to the other by selecting the client name and clicking the left- or right-arrow button. The depot is installed on the clients in the **Clients to Install On** list.
11. Click **Install & Close** to install the depot and close the Install Product window.

    The installation process displays a Product Install window similar to the following.



```
Checking product dependencies...
 Product JCF is already installed as needed.
  Product JRE is already installed as needed.
  Product 3.6.1-TMF-0045 is already installed as needed.
  Product TMF_3.6.1 is already installed as needed.
 Dependency check completed.
Inspecting node oak...
Installing Product: Tivoli Software Installation Service Depot, Version 3.7


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts: oak
   need to copy the SISGBIN (generic) to:
        oak:/data/Tivoli/bin/generic_unix
   need to copy the CAT (generic) to:
        oak:/data/Tivoli/msg_cat
   need to copy the IRDIR (generic) to:
        oak:/data/IR

  For the machines in the aix4-r1 class:
    hosts: oak
   need to copy the BIN (aix4-r1) to:
        oak:/data/Tivoli/bin/aix4-r1
   need to copy the SISBIN (aix4-r1) to:
        oak:/data/Tivoli/bin/aix4-r1
   need to copy the DB (aix4-r1) to:
        oak:/data/Tivoli/db/oak.db


---
To continue with installation, select the ``Continue Install'' button
```

This window lists the operations that will occur during the installation and any problems that you might want to correct before continuing the installation. If you want to correct any problems, click **Cancel**.

12. Click **Continue Install** to perform the installation. The Product Install window displays status information as the installation proceeds.

    When the installation is complete, the Product Install window displays a completion message and the **Cancel** button becomes the **Close** button.

13. Click **Close** to close the Product Install window.

The depot is installed and can be accessed by any Tivoli Software Installation Service client in this Tivoli region.

Installing using the winstall command

To install the Tivoli Software Installation Service depot, use the **winstall** command as follows:

```
winstall -c source_dir -i SISDEPOT IRDIR=IR_dir node
```

where:

–c *source_dir*
: Specifies the path to the installation image.

–i SISDEPOT
: Specifies the index file for the depot, SISDEPOT.IND.

IRDIR=*IR_dir*
: Specifies the full path name of the install repository.

Note: You must specify this variable. If you do not, the depot will appear to install but will not work properly.

To share an existing install repository with a depot in a different Tivoli region, specify the local path to the shared install repository directory. You do not need to explicitly allow sharing of an install repository, just ensure that both machines have physical access to the directory.

For considerations and restrictions about where to place the install repository directory, refer to "Location of the Install Repository" on page 80.

node Specifies the managed node on which to install the depot.

The following example uses the **winstall** command to install the depot on wave, using the installation image on /cdrom/cdrom0/SIS and placing the install repository in the /ir subdirectory:

```
winstall -c /cdrom/cdrom0/SIS -i SISDEPOT IRDIR=/ir wave
```

Refer to the **winstall** command in the *Tivoli Management Framework Reference Manual* for more information.

The depot is installed and can be accessed by any Tivoli Software Installation Service client in this Tivoli region.

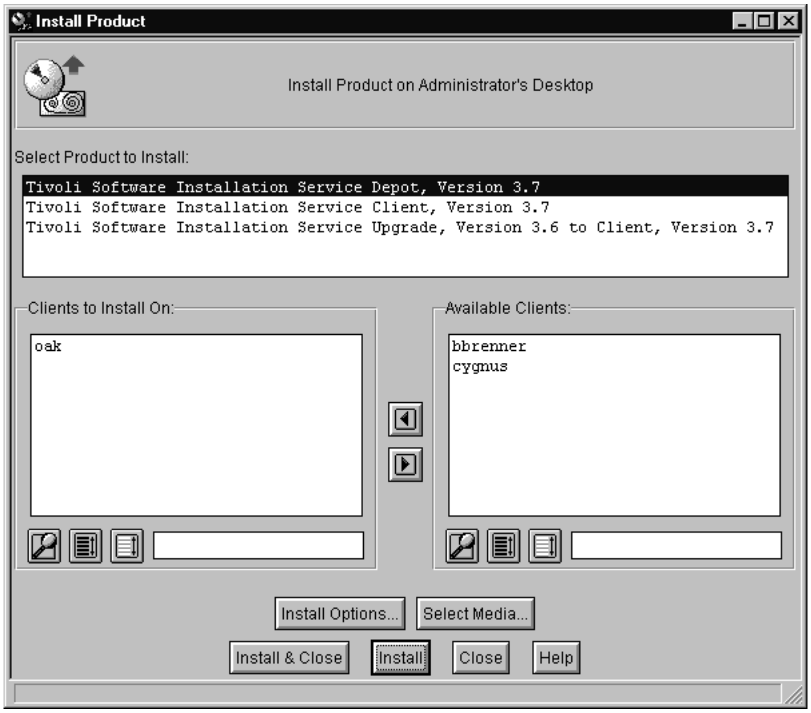
Installing the Tivoli Software Installation Service client

The following sections provide instructions for installing the Tivoli Software Installation Service client. They provide instructions for installing the client using the Tivoli desktop and the command line.

Installing from the Tivoli desktop

Complete the following steps to install the Tivoli Software Installation Service client from the Tivoli desktop:

1. From the **Desktop** menu, select **Install → Install Product** to display the Install Product window.



```
Tivoli Software Installation Service Depot, Version 3.7
Tivoli Software Installation Service Client, Version 3.7
Tivoli Software Installation Service Upgrade, Version 3.6 to Client, Version 3.7
```

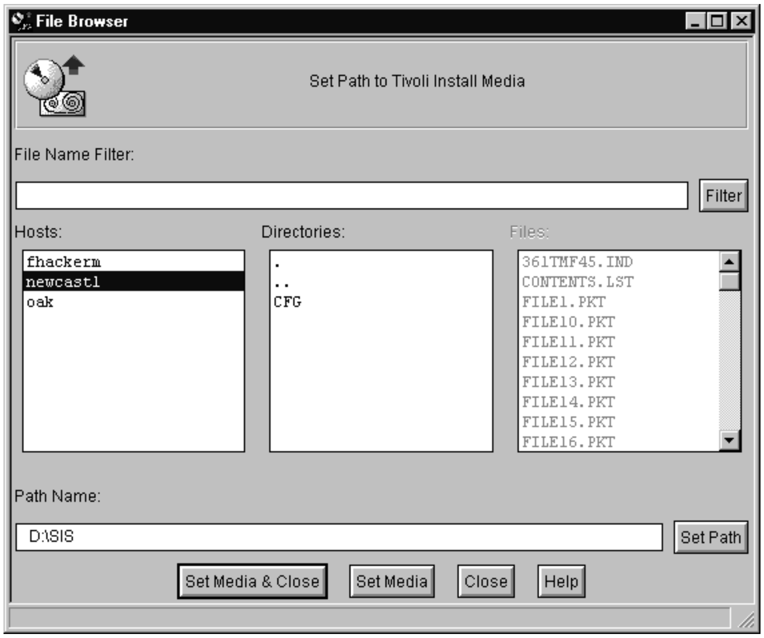
Clients to Install On:

```
oak
```

Available Clients:

```
bbrenner
cygnus
```

2. If the **Select Product to Install** list is empty or does not contain **Tivoli Software Installation Service Client, Version 4.1.1**, continue with step 3. If it is listed, skip to step 7.
3. Click **Select Media** to display the File Browser window. Use this window to specify the path to the installation image for the client.



Hosts:

```
fhackerm
newcastl
oak
```

Directories:

```
.
..
CFG
```

Files:

```
361TMF45.IND
CONTENTS.LST
FILE1.PKT
FILE10.PKT
FILE11.PKT
FILE12.PKT
FILE13.PKT
FILE14.PKT
FILE15.PKT
FILE16.PKT
```

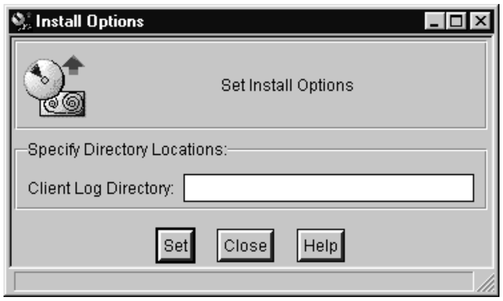
Path Name: D:\SIS

4. In the **Hosts** list, select the host on which the installation image is mounted.
5. Navigate to the directory that contains the installation image. The installation directory contains the product index (.IND) file.

Double-click directory names in the **Directories** list until the installation images are shown in the **Files** list.

Alternatively, if you know the path to the installation image, type the full path in the **Path Name** field. Click **Set Path** to list the contents of the specified directory.

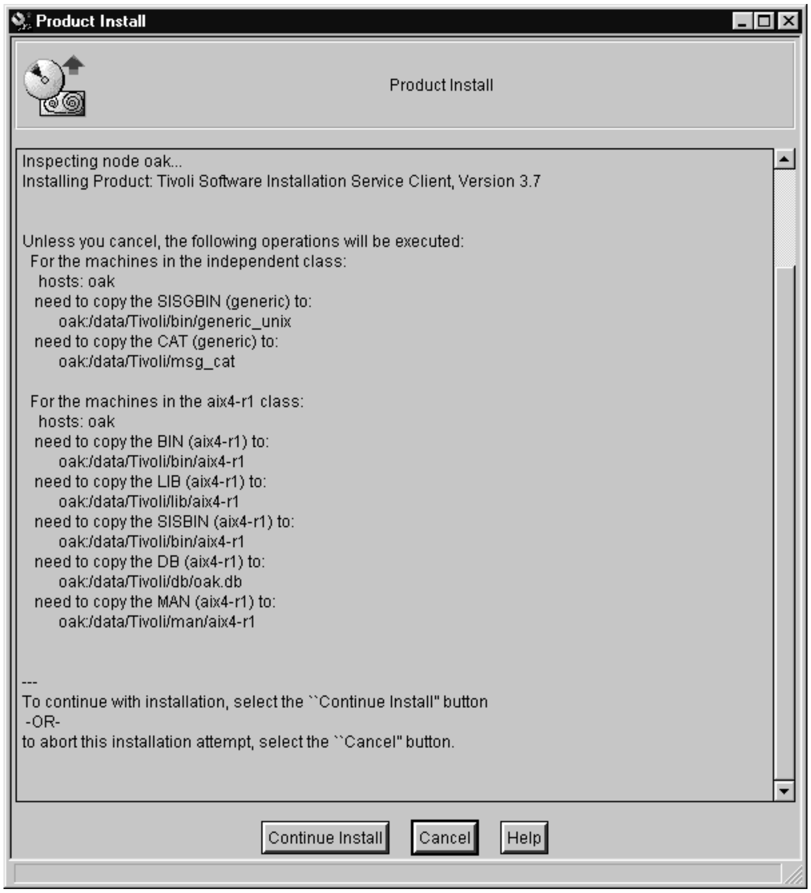
6. Click **Set Media & Close** to save the path and return to the Install Product window. The window now lists **Tivoli Software Installation Service Client, Version 4.1.1**, which indicates that it is available for installation.
7. In the **Select Product to Install** list, select **Tivoli Software Installation Service, Version 4.1.1**. The Install Options window is displayed.



8. In the **Client Log Directory** field, type the name of a directory on the client machine. This directory holds the client logs and cached copies of depot logs when they are accessed by the log viewer. If the directory does not already exist, it is created.

   If you do not specify a directory, you are prompted for it the first time the client is started.
9. Use the arrow buttons to select the machines on which to install this component. Move the clients from one list (**Available Clients** or **Clients to Install On**) to the other by selecting the client name and clicking the left- or right-arrow button. The component is installed on the clients in the **Clients to Install On** list.
10. Click **Install & Close** to install the component and close the Install Product window.

    The installation process displays a Product Install window similar to the following.



```
Inspecting node oak...
Installing Product: Tivoli Software Installation Service Client, Version 3.7


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts: oak
   need to copy the SISGBIN (generic) to:
         oak:/data/Tivoli/bin/generic_unix
   need to copy the CAT (generic) to:
         oak:/data/Tivoli/msg_cat

  For the machines in the aix4-r1 class:
    hosts: oak
   need to copy the BIN (aix4-r1) to:
         oak:/data/Tivoli/bin/aix4-r1
   need to copy the LIB (aix4-r1) to:
         oak:/data/Tivoli/lib/aix4-r1
   need to copy the SISBIN (aix4-r1) to:
         oak:/data/Tivoli/bin/aix4-r1
   need to copy the DB (aix4-r1) to:
         oak:/data/Tivoli/db/oak.db
   need to copy the MAN (aix4-r1) to:
         oak:/data/Tivoli/man/aix4-r1


---
To continue with installation, select the ``Continue Install'' button
 -OR-
to abort this installation attempt, select the ``Cancel'' button.
```

This window lists the operations that will occur during the installation and problems that you might want to correct before continuing the installation. If you want to correct any problems, click **Cancel**.

11. Click **Continue Install** to perform the installation. The Product Install window displays status information as the installation process proceeds.

    When the installation is complete, the Product Install window displays a completion message and the **Cancel** button becomes the **Close** button.

12. Click **Close** to close the Product Install window.

The client is installed. You can use the Tivoli Software Installation Service commands at any time and can start the Tivoli Software Installation Service console using the **wsisgui** command. If this is the first client to be installed, every administrator's Tivoli desktop must be closed before any administrator can access the console from a Tivoli desktop.

You can now use the client to create managed resources and install Tivoli software within your Tivoli region.

Installing using the winstall command

To install the Tivoli Software Installation Service client, use the **winstall** command as follows:

```
winstall -c source_dir -i SISCLNT CLIENTLOGDIR=log_dir node
```

where:

–c *source_dir*
: Specifies the path to the installation image.

–i SISCLNT
: Specifies the index file for the client, SISCLNT.IND.

CLIENTLOGDIR=*log_dir*
: Specifies the directory on the client in which to place client log files and cached copies of depot logs when they are accessed by the log viewer. If the directory does not exist, it is created. If you do not specify a directory, you are prompted for it the first time the client is started.

node Specifies the managed node on which to install the client.

The following example shows how to use the **winstall** command to install the client on cygnus, using the installation image in /cdrom/cdrom0/SIS and placing the install repository in the /ir subdirectory:

```
winstall -c /cdrom/cdrom0/SIS -i SISCLNT \
CLIENTLOGDIR=/ir cygnus
```

Refer to the **winstall** command in the *Tivoli Management Framework Reference Manual* for more information.

Uninstalling Tivoli Software Installation Service

You must uninstall the Tivoli Software Installation Service depot and the Tivoli Software Installation Service client separately. The install repository and client log directories are not removed; you must remove these directories manually after uninstalling the depot.

Notes:

- Before uninstalling each client, locate its log directory. This is the value of the **Client log directory** preference. To determine the value of this preference, follow the instructions in "Viewing preferences" on page 99.
- Before uninstalling the last client, make a note of the location of the install repository for each depot, which you can find by following the instructions in "Finding the install repository" on page 108.
- After uninstalling the last client, every Tivoli desktop must be closed (at the same time) before the client option is removed from the **Install** menu of any Tivoli desktop. After closing all of the desktops, you must wait a few minutes for the desktop change to complete.

Uninstall the components of Tivoli Software Installation Service in the following order:

1. Uninstall the client.
2. If you do not need the client logs, optionally remove that directory.
3. Uninstall the depot.
4. If the install repository of the depot you just uninstalled is not shared with a depot in another Tivoli region, optionally remove its directory.
5. If no other applications on the managed node require the Java components, uninstall Java 1.3 for Tivoli and Tivoli Java Client Framework 4.1.1. If you are not sure whether these components are required, do not uninstall them.

To uninstall these components, follow the instructions in Chapter 20, "Uninstalling a Tivoli environment," on page 255. The tag information needed to uninstall each component of Tivoli Software Installation Service is listed in Table 3 on page 84. The tag information for Java components is listed in Table 7 on page 222.

Chapter 11. Configuring Tivoli Software Installation Service

Tivoli Software Installation Service (SIS) allows you to customize the Tivoli Software Installation Service depot and the Tivoli Software Installation Service client using the Tivoli Software Installation Service preferences. You can change the settings of these preferences using either the Tivoli Software Installation Service console or the **wsisprefs** command.

The preferences for the depot are saved in the Tivoli object database, as attributes of the SisDepot object.

Preferences for the client are saved in the file $DBDIR/sisclnt.ini. When you start the client for the first time, it creates the sisclnt.ini file. Do not edit this file manually unless instructed to do so by your support provider.

This chapter describes the following tasks:

- Viewing the preferences
- Modifying the preferences
- Restoring the installation defaults for the preferences
- Stopping and restarting the depot
- Finding and changing the location of the install repository

In addition, this chapter describes each of the user-modifiable settings for the depot and the client.

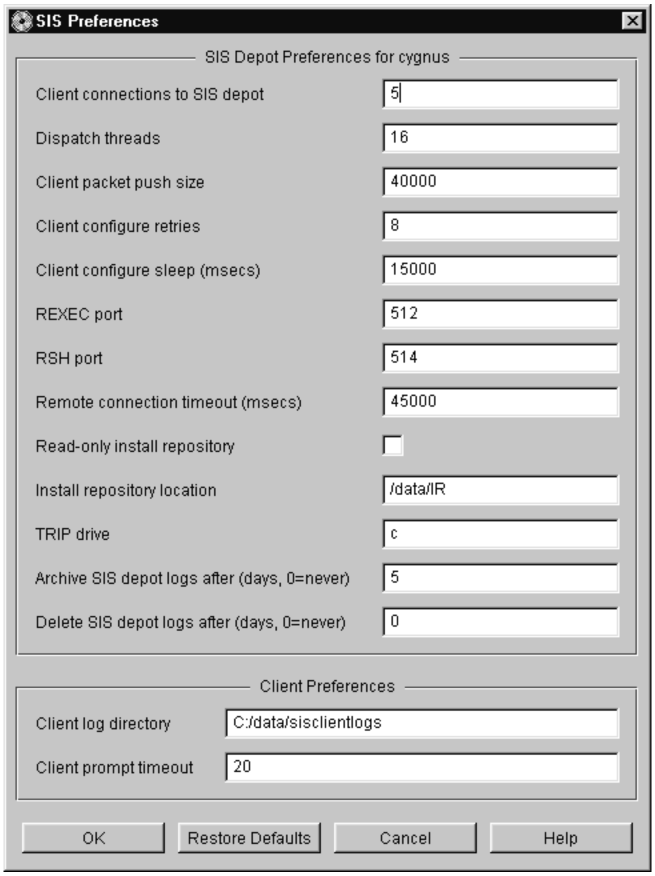
Viewing preferences

You can view the preferences using either the Tivoli Software Installation Service console or the **wsisprefs** command. The following sections provide instructions.

Viewing preferences using the Tivoli Software Installation Service console

To view the preferences using the Tivoli Software Installation Service console, perform the following steps:

1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113. The installation worksheet is displayed.
2. From the **Edit** menu of the installation worksheet, click **Preferences**. This opens the SIS Preferences window.



SIS Preferences

SIS Depot Preferences for cygnus

Client connections to SIS depot: 5

Dispatch threads: 16

Client packet push size: 40000

Client configure retries: 8

Client configure sleep (msecs): 15000

REXEC port: 512

RSH port: 514

Remote connection timeout (msecs): 45000

Read-only install repository: □

Install repository location: /data/IR

TRIP drive: c

Archive SIS depot logs after (days, 0=never): 5

Delete SIS depot logs after (days, 0=never): 0

Client Preferences

Client log directory: C:/data/sisclientlogs

Client prompt timeout: 20

The window contains sections for the Tivoli Software Installation Service depot and Tivoli Software Installation Service client preferences. For details about the preferences, including their minimum, maximum, and default values, click **Help**, or refer to “Tivoli Software Installation Service depot preferences” on page 105 and “Tivoli Software Installation Service client preferences” on page 107.

3. Click **OK** to close the SIS Preferences window.

Viewing preferences using the wsisprefs command

To view the preferences using the **wsisprefs** command, enter the following command:

```
wsisprefs -l
```

The command generates output similar to the following:

```
Connecting to SIS Depot
Successfully connected to SIS Depot cygnus
SIS Depot Preferences for cygnus

#  Name                                         Label                     Value
-- -------------------------------------------- ------------------------- -----
1  Client connections to SIS depot              client_connections        5
2  Dispatch threads                             dispatch_threads          16
3  Client packet push size                      client_pkt_push_size      40000
4  Client configure retries                     client_cfg_retries        8
5  Client configure sleep (msecs)               client_cfg_sleep          15000
6  REXEC port                                   rexec_port                512
7  RSH port                                     rsh_port                  514
8  Remote connection timeout (msecs)            remote_connection_timeout 45000
9  Read-only install repository                 read_only_ir              false
```

```
10 Install repository location                 depot_location      /data/IR/IR
11 TRIP drive                                  trip_drive                C
12 Archive SIS depot logs after (days, 0=never) log_archive_days          5
13 Delete SIS depot logs after (days, 0=never)  log_purge_days            0

Client Preferences

#  Name                  Label                 Value
-- --------------------- --------------------- ----------------------
14 Client log directory  client_data_dir       /data/clientlogs
15 Client prompt timeout client_prompt_timeout 20
```

The output displays the following information:

- The number of the preference. The number can be used to identify a preference when changing its value with the **wsisprefs** command.
- The name of the preference. This value is displayed in the language of the current locale.
- The internal label of the preference. This value can be used to identify a preference when changing its value with the **wsisprefs** command. This value is not translated.
- The current setting of the preference.

To view the minimum, maximum, and default value of each preference, use the following command:

```
wsisprefs -lv
```

This command generates output similar to the following:

```
Connecting to SIS Depot
Successfully connected to SIS Depot cygnus
SIS Depot Preferences for cygnus

#  Label                     Value       Default Min  Max
-- ------------------------- ----------- ------- ---- ----------
1  client_connections        5           5       1    10
2  dispatch_threads          16          16      1    100
3  client_pkt_push_size      40000       40000   1    2147483647
4  client_cfg_retries        8           8       1    1024
5  client_cfg_sleep          15000       15000   500  2147483647
6  rexec_port                512         512     1    65535
7  rsh_port                  514         514     1    1024
8  remote_connection_timeout 45000       45000   1    2147483647
9  read_only_ir              false       false   true false
10 depot_location            /data/IR/IR
11 trip_drive                C           c       a    z
12 log_archive_days          5           5       0    365
13 log_purge_days            0           0       0    365

Client Preferences

#  Label                 Value                  Default Min Max
-- --------------------- ---------------------- ------- --- ---
14 client_data_dir       /data/clientlogs
15 client_prompt_timeout 20                     20       1  100
```

The output displays the following information:

- The number of the preference. The number can be used to identify a preference when changing its value with the **wsisprefs** command.
- The internal label of the preference. This value can be used to identify a preference when changing its value with the **wsisprefs** command. This value is not translated.

- The current setting of the preference.
- The default value of the preference.
- The minimum value of the preference.
- The maximum value of the preference.

For additional information about the preferences, refer to “Tivoli Software Installation Service depot preferences” on page 105 and “Tivoli Software Installation Service client preferences” on page 107.

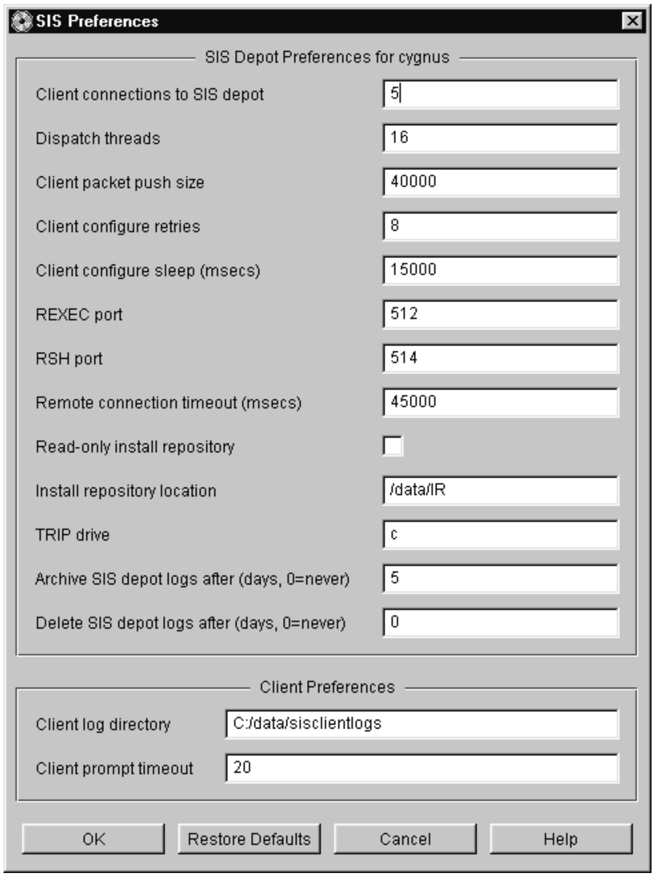
Modifying preferences

To modify the preferences, use the SIS Preferences window or the **wsisprefs** command. The following sections contain instructions.

Modifying preferences using the Tivoli Software Installation Service console

To modify preferences using the Tivoli Software Installation Service console, perform the following steps:

1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113. The installation worksheet is displayed.
2. From the **Edit** menu of the installation worksheet, click **Preferences**. The SIS Preferences window is displayed.



SIS Depot Preferences for cygnus

Client connections to SIS depot	5
Dispatch threads	16
Client packet push size	40000
Client configure retries	8
Client configure sleep (msecs)	15000
REXEC port	512
RSH port	514
Remote connection timeout (msecs)	45000
Read-only install repository	□
Install repository location	/data/IR
TRIP drive	c
Archive SIS depot logs after (days, 0=never)	5
Delete SIS depot logs after (days, 0=never)	0

Client Preferences

Client log directory	C:/data/sisclientlogs
Client prompt timeout	20

The window contains sections for the Tivoli Software Installation Service depot and Tivoli Software Installation Service client preferences. For details about the preferences, including their minimum, maximum, and default values, click

Help, or refer to “Tivoli Software Installation Service depot preferences” on page 105 and “Tivoli Software Installation Service client preferences” on page 107.

3. Modify the value of one or more preferences.
4. Click **OK** to save your changes and close the SIS Preferences window.

   Tivoli Software Installation Service validates the value of each preference at this time. If any preference is set to an incorrect value, a window displays the possible values for that preference. Correct the values and click then **OK** to process the new values.

   Some preferences do not take effect until the depot is stopped and restarted. When you modify these preferences, you are asked whether you want to stop and restart the depot immediately.

Modifying preferences using the wsisprefs command

To modify the preferences using the **wsisprefs** command, perform the following steps:

1. Determine the number of the preference you want to change by entering the following command:

   ```
   wsisprefs -l
   ```

   The command generates output similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   SIS Depot Preferences for cygnus

   #  Name                                         Label                     Value
   -- -------------------------------------------- ------------------------- -----
   1  Client connections to SIS depot              client_connections        5
   2  Dispatch threads                             dispatch_threads          16
   3  Client packet push size                      client_pkt_push_size      40000
   4  Client configure retries                     client_cfg_retries        8
   5  Client configure sleep (msecs)               client_cfg_sleep          15000
   6  REXEC port                                   rexec_port                512
   7  RSH port                                     rsh_port                  514
   8  Remote connection timeout (msecs)            remote_connection_timeout 45000
   9  Read-only install repository                 read_only_ir              false
   10 Install repository location                  depot_location      /data/IR/IR
   11 TRIP drive                                   trip_drive                C
   12 Archive SIS depot logs after (days, 0=never) log_archive_days          5
   13 Delete SIS depot logs after (days, 0=never)  log_purge_days            0

   Client Preferences

   #  Name                 Label                 Value
   -- -------------------- --------------------- ----------------------
   14 Client log directory client_data_dir       /data/clientlogs
   15 Client prompt timeout client_prompt_timeout 20
   ```

   The output displays the following information:

   - The number of the preference. The number can be used to identify a preference when changing its value with the **wsisprefs** command.
   - The name of the preference. This value is displayed in the language of the current locale.
   - The internal label of the preference. This value can be used to identify a preference when changing its value with the **wsisprefs** command. This value is not translated.
   - The current setting of the preference.

2. To determine the allowable values for a preference, run the following command:

   ```
   wsisprefs -lv
   ```

   The command generates output similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   SIS Depot Preferences for cygnus

   #  Label                     Value       Default Min  Max
   -- ------------------------- ----------- ------- ---- ----------
   1  client_connections        5           5       1    10
   2  dispatch_threads          16          16      1    100
   3  client_pkt_push_size      40000       40000   1    2147483647
   4  client_cfg_retries        8           8       1    1024
   5  client_cfg_sleep          15000       15000   500  2147483647
   6  rexec_port                512         512     1    65535
   7  rsh_port                  514         514     1    1024
   8  remote_connection_timeout 45000       45000   1    2147483647
   9  read_only_ir              false       false   true false
   10 depot_location            /data/IR/IR
   11 trip_drive                C           c       a    z
   12 log_archive_days          5           5       0    365
   13 log_purge_days            0           0       0    365

   Client Preferences

   #  Label                 Value                  Default Min Max
   -- --------------------- ---------------------- ------- --- ---
   14 client_data_dir       /data/clientlogs
   15 client_prompt_timeout 20                     20        1 100
   ```

   The output displays the following information:

   - The number of the preference. The number can be used to identify a preference when changing its value with the **wsisprefs** command.
   - The internal label of the preference. This value can be used to identify a preference when changing its value with the **wsisprefs** command. This value is not translated.
   - The current setting of the preference.
   - The default value of the preference.
   - The minimum value of the preference.
   - The maximum value of the preference.

3. Use the **wsisprefs** command to modify the value of a preference. For example, the following command sets the value of the `Read-only install repository` (preference number 9) to `true`:

   ```
   wsisprefs -s 9 true
   ```

   This could also be done using the label of the preference instead of its number. The following command has the same result:

   ```
   wsisprefs -s read_only_ir true
   ```

4. Some preferences do not take effect until the Tivoli Software Installation Service depot is stopped and restarted. When you modify these preferences, you are asked whether you want to stop and restart the depot immediately.

For additional information about the preferences, refer to “Tivoli Software Installation Service depot preferences” on page 105 and “Tivoli Software Installation Service client preferences” on page 107.

Resetting preferences to default values

You can reset the value of most preferences to their default values for the Tivoli Software Installation Service depot and Tivoli Software Installation Service client. The **Client log directory** and **Install repository location** preferences cannot be reset.

Resetting preferences using the Tivoli Software Installation Service console

To reset preferences to their initial values, click **Restore Defaults** on the SIS Preferences window. This resets the values for all preferences except **Client log directory** and **Install repository location**.

Some preferences do not take effect until the Tivoli Software Installation Service depot is stopped and restarted. When you modify these preferences, you are asked whether you want to stop and restart the depot immediately.

Resetting preferences from the command line

To reset preferences to their initial values, use the **wsisprefs** command as follows:

```
wsisprefs -r
```

This resets the values for all preferences except **Client log directory** and **Install repository location**.

Some preferences do not take effect until the Tivoli Software Installation Service depot is stopped and restarted. When you modify these preferences, you are asked whether you want to stop and restart the depot immediately.

Tivoli Software Installation Service depot preferences

This section describes the preferences you use to customize the Tivoli Software Installation Service depot.

Changes to some depot preferences require that the depot be stopped and restarted. This is noted in the description of those preferences in the following list. If you change these preferences using the Tivoli Software Installation Service console, you receive a window asking whether you want to restart the depot immediately. See “Stopping and restarting the Tivoli Software Installation Service depot” on page 108 for more instructions on stopping and restarting the depot.

Client connections to SIS depot

Specifies the number of clients that can connect to the depot at the same time. The value must be an integer between 1 and 10. The default is 5.

Changes do not take effect until the depot is stopped and restarted.

Dispatch threads

Specifies the number of machines that can have products installed on them at the same time. The value must be an integer between 1 and 100. The default is 16.

Tivoli recommends that you do not increase this value above the default. If Tivoli Software Installation Service is consuming too many system resources, such as memory or file descriptors, decrease this value.

Client packet push size

Specifies the size, in bytes, of packets that are used for communication between the depot and Tivoli Software Installation Service client, and for installing using the REXEC or RSH method. This preference does not affect the installation of products when a Tivoli connection is used. The value must be an integer greater than 1. The default is 40,000 bytes (39 KB).

When you change this preference, the depot and all clients are stopped and restarted automatically. A window asks if you want to do this before you are allowed to change the value.

Client configure retries

Specifies the number of times Tivoli Software Installation Service tests whether the object dispatcher is started on a newly installed managed node. The value must be an integer between 1 and 1,024. The default is 8.

Tivoli recommends that you do not reduce this value below the default. If the object ID of the managed node is in $DBDIR/oservlog on the client and Tivoli Software Installation Service fails to retrieve this object ID, increase either this value or the value of **Client configure sleep (msecs)**.

Client configure sleep (msecs)

Specifies the amount of time, in milliseconds, that Tivoli Software Installation Service waits between attempts to determine whether the object dispatcher has started on a newly installed managed node. The value must be an integer greater than or equal to 500. The default is 15,000.

Tivoli recommends that you do not reduce this value below the default. If the object ID of the managed node is in $DBDIR/oservlog on the client and Tivoli Software Installation Service fails to retrieve this object ID, increase either this value or the value of **Client configure retries**.

REXEC port

Specifies the port on a remote computer that Tivoli Software Installation Service uses for REXEC (account access) connections when installing managed nodes and endpoints. The port number can be an integer between 1 and 65,535. The default port number is 512, the well-known port number for REXEC.

Do not change this value if your Tivoli region contains Windows machines. REXEC on Windows operating system is provided by Tivoli Remote Execution Service, which requires port 512.

RSH port

Specifies the port on a remote computer that Tivoli Software Installation Service uses for RSH (trusted host) connections when installing managed nodes and endpoints. The port number must be an integer between 1 and 1,024. The default port number is 514, the well-known port number for RSH.

Remote connection timeout (msecs)

Specifies the time, in milliseconds, that Tivoli Software Installation Service waits for a new REXEC or RSH connection to be established. The timeout value must be an integer greater than or equal to 1. The default is 45,000 milliseconds (45 seconds).

Read-only install repository

Specifies that the depot has read-only access to the install repository. When an install repository is read-only, you can add machines, plan an installation, work with response files, change preferences, and install products, but you cannot do the following:

- You cannot import products into the install repository.

- You cannot remove products from the install repository.
- You cannot change defaults installation attributes associated with a product.
- You cannot change user-defined prerequisites.

Acceptable values are `true` and `false`.

Install repository location
Specifies the location of the install repository directory on the depot.

Change this value only if the install repository directory is moved. Changing this value does not move the install repository; it only tells the depot where to locate the install repository. For instructions for moving the install repository, refer to “Changing the location of the install repository” on page 109.

Changes do not take effect until the Tivoli Software Installation Service depot is stopped and restarted.

The value of this preference is not changed by the **Restore Defaults** button on the SIS Preferences window or the **–r** option of the **wsisprefs** command.

TRIP drive
For Windows targets, specifies the drive on which to install the Tivoli Remote Execution Service. The default is the C drive. The drive letter is specified without a colon; that is, do not use `C:`.

Archive SIS depot logs after (days, 0=never)
Specifies the number of calendar days after which the depot logs are automatically archived. Specify an integer between 1 and 365, or specify 0 to turn off automatic archiving. The default value is 5. For more information about archiving logs, refer to “Archiving logs” on page 175.

Changes do not take effect until the depot is stopped and restarted.

Delete SIS depot logs after (days, 0=never)
Specifies the number of calendar days after which the depot logs are automatically deleted. Specify an integer between 1 and 365, or specify 0 to turn off automatic deletion. The default value is 0. For more information about deleting logs, refer to “Deleting logs” on page 175.

Changes do not take effect until the depot is stopped and restarted.

Tivoli Software Installation Service client preferences

This section describes the preferences you use to customize the Tivoli Software Installation Service client.

Tivoli Software Installation Service commands begin to use the new value immediately. Any open Tivoli Software Installation Service console continues to use the original values until it is restarted.

Client log directory
Specifies the directory in which client logs are placed. This includes the client logs as well as cached copies of depot logs that are being displayed by the log viewer.

The value of this preference is not changed by the **Restore Defaults** button on the SIS Preferences window or the **–r** option of the **wsisprefs** command.

Client prompt timeout
Specifies how many seconds the client waits for a response to a prompt. The value must be an integer between 1 and 100 seconds. The default is 20.

Stopping and restarting the Tivoli Software Installation Service depot

Some of the preferences do not take effect until you stop and restart the Tivoli Software Installation Service depot. When you change these preferences, you can automatically restart the depot. If you chose not to restart the depot immediately, perform these steps to stop and restart it at a convenient time:

1. Close any open Tivoli Software Installation Service console sessions and wait for Tivoli Software Installation Service commands to complete. You cannot stop the depot if any Tivoli Software Installation Service client processes are running.
2. Determine the object ID of the depot using the following command:

   ```
   wlookup -ar SisDepot
   ```

   In a Tivoli region with two depots, the output is similar to the following:

   ```
       cygnus   1812582581.1.545#TMF_SisDepotServer::SISDepotServer#
       oak      1812582581.2.19#TMF_SisDepotServer::SISDepotServer#
   ```

   The object ID of the depot on cygnus is 1812582581.1.545.
3. Use the following command to stop the depot on cygnus:

   ```
   idlcall 1812582581.1.545 shutdown
   ```

   where 1812582581.1.545 is the object ID of the depot.

   If any client processes are still running, a message similar to the following indicates that you cannot stop the depot:

   ```
   { USER_EXCEPTION
    TMF_SisDepotServer::CannotExecuteException {
    "NLS_Normal_Shutdown" "" "" }}
   ```

   Note: Wait at least 10 seconds for the shutdown to complete before using any Tivoli Software Installation Service commands or starting the console.

The depot restarts automatically the next time a client initiates communication with it.

Finding the install repository

You can find the location of the install repository using either the console or the command line, as described in the following sections.

Finding the install repository using the Tivoli Software Installation Service console

To locate the install repository using the Tivoli Software Installation Service console, perform the following steps:

1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113. The installation worksheet is displayed.
2. From the **Edit** menu of the installation worksheet, click **Preferences**. This opens the SIS Preferences window.

   The location of the install repository is displayed in the **Install repository location** field.

Finding the install repository from the command line

To locate the install repository using the command line, use the **wsisprefs** command as follows:

```
wsisprefs -l
```

This commands displays the values of all preferences. The `Install repository location` line displays the name of the directory on the Tivoli Software Installation Service depot that contains the install repository.

Changing the location of the install repository

In normal operations, you do not need to change the install repository location. However, if the file system containing the install repository becomes full, you might need to move the install repository to another file system.

To change the location of the install repository, perform the following steps:

1. Copy the install repository directory, including all files and all subdirectories, to the new location. Make sure that the new directory structure has the same permissions as the original.

   If the install repository is shared by multiple depots, be sure that the new location is accessible by each depot.
2. Change the **Install repository location** preference to point to the new location. For information about changing preferences, refer to Chapter 11, “Configuring Tivoli Software Installation Service,” on page 99.

   If the install repository is shared by multiple depots, change the value of this preference for each of them.
3. Stop the depot as described in “Stopping and restarting the Tivoli Software Installation Service depot” on page 108.

   If the install repository is shared by multiple depots, stop each of them.
4. Test the new install repository by starting any client and connecting to that depot.

   If the install repository is shared by multiple depots, do this for each of them.
5. After all depots can connect to the new install repository, you can remove the original install repository.

The next time the depot starts, it will use the install repository in the new location.

Chapter 12. Using Tivoli Software Installation Service

You can use Tivoli Software Installation Service (SIS) to install Tivoli products and perform install repository maintenance tasks using either the Tivoli Software Installation Service console or commands. This chapter provides an overview of these procedures and provides detailed instructions for using Tivoli Software Installation Service to install Tivoli Enterprise software. This chapter contains the following sections:

- "Authorization roles"
- "Installation overview using the Tivoli Software Installation Service console"
- "Installation overview from the command line" on page 112
- "Using the Tivoli Software Installation Service console on UNIX operating systems" on page 113
- "Starting the Tivoli Software Installation Service console" on page 113
- "Working with products" on page 115
- "Working with machines" on page 130
- "Specifying products to install" on page 138
- "Installing products" on page 139

Authorization roles

The following table provides the context and authorization roles required to start and use Tivoli Software Installation Service.

Activity	Context	Required Roles
Start and use Tivoli Software Installation Service	Tivoli region	**user** plus one of **super**, **senior**, **install_client**, or **install_product**

Note: On Windows operating system, the user that starts the client must be in the Administrators group.

Installation overview using the Tivoli Software Installation Service console

To install Tivoli software using the console, complete these high-level steps:

1. Start the console (refer to "Starting the Tivoli Software Installation Service console" on page 113).
2. Complete the installation worksheet:
   a. Select products from the install repository:
      - Make sure that the products you want to install are in the install repository. If necessary, import the products (refer to "Importing products using the Tivoli Software Installation Service console" on page 115).
      - Add the products to the installation worksheet (refer to "Adding products to the installation worksheet" on page 127).
      - If necessary, specify default values for the installation options of each product you import (refer to "Viewing and customizing installation options" on page 119).

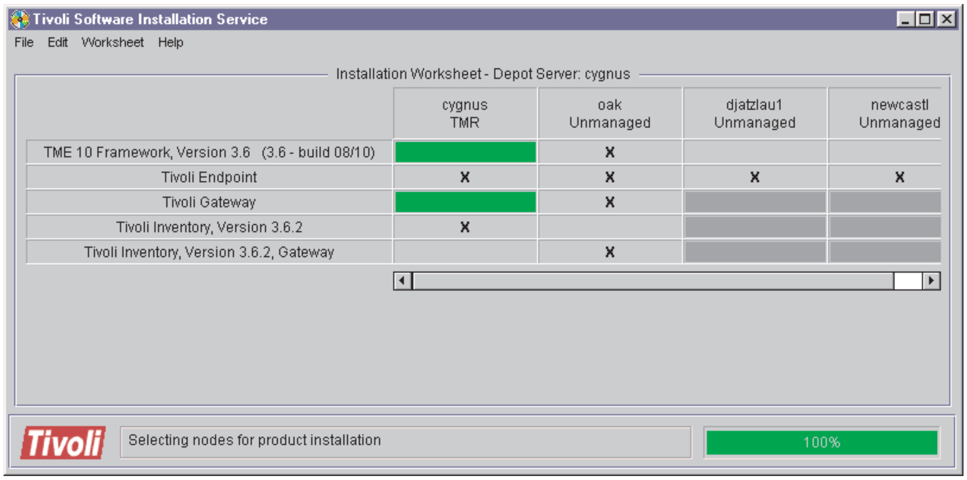
   b. Specify the machines on which to install:

- Make sure that the machines on which you want to install are already known to Tivoli Software Installation Service. If necessary, define the new machines (refer to "Adding machines to the Tivoli Software Installation Service depot" on page 130).
- Add the machines to the installation worksheet (refer to "Adding machines to the installation worksheet" on page 134).

c. Complete the installation worksheet to indicate which products are installed on which machines (refer to "Specifying products to install" on page 138).

d. If necessary, modify the installation options for a product on a specific machine (refer to "Overriding default installation options for one machine" on page 137).

The following figure shows an installation worksheet for installing multiple products to multiple machines.



Installation Worksheet - Depot Server: cygnus

	cygnus TMR	oak Unmanaged	djatzlau1 Unmanaged	newcastl Unmanaged
TME 10 Framework, Version 3.6 (3.6 - build 08/10)	🟩	X		
Tivoli Endpoint	X	X	X	X
Tivoli Gateway	🟩	X	▨	▨
Tivoli Inventory, Version 3.6.2	X		▨	▨
Tivoli Inventory, Version 3.6.2, Gateway		X	▨	▨

Selecting nodes for product installation

3. Start the installation (refer to "Using the Tivoli Software Installation Service console to install products" on page 139).

Installation overview from the command line

To install Tivoli software using the Tivoli Software Installation Service commands, perform the following steps:

1. If necessary, use the **wimport** command to import product images into the install repository (refer to "Importing products from the command line" on page 118).
2. If necessary, use the **wsisdefaults** command to set defaults for the installation options of each product (refer to "Setting product defaults from the command line" on page 121).
3. Create a response file containing the following information. Refer to "Creating a response file template" on page 157 for details. A response file specifies the following:

   a. The products to be installed.

   b. The machines on which to install each product. If a machine is not already known to Tivoli Software Installation Service, you can define it using the response file.

   c. The installation options to use when installing each product.

   You can create a response file template using either the Tivoli Software Installation Service console or the **–x** option of the **wsis** command. You can modify this template using any text editor.

4. Use the **wsis** command to start the installation (refer to “Using response files to install products” on page 141).

For additional information about using the Tivoli Software Installation Service commands to install products in your Tivoli environment, refer to the reference information about each command in Appendix A, “Installation Commands,” on page 361.

Using the Tivoli Software Installation Service console on UNIX operating systems

On UNIX operating systems, perform the following actions to allow the console to open on the X Window System display:

- Set the DISPLAY environment variable to the X Window System display on which to display the Tivoli Software Installation Service console.

  For example, to open the console on the X Window System display named cygnus:0.0, a Bourne or Korn shell user would enter the following commands:

  ```
  DISPLAY=cygnus:0.0
  export DISPLAY
  ```
- Enable remote connections to the X Window System.

  This step is necessary even if the console runs on the same machine as the X Window System display. For example, if the Tivoli Software Installation Service client is installed on cygnus, enter the following command to enable the console to start on the display:

  ```
  xhost +cygnus
  ```

Starting the Tivoli Software Installation Service console

The Tivoli Software Installation Service console can be started either from the Tivoli desktop or from the command line.

Starting the Tivoli Software Installation Service console from the Tivoli desktop

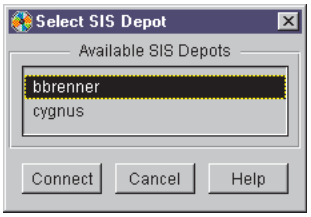
You can start the Tivoli Software Installation Service console from the Tivoli desktop only when the desktop is running on a machine where the Tivoli Software Installation Service client is installed, even though the menu option is visible on all desktops.

If you start the console from the Tivoli desktop, you cannot use the desktop while that console is open. If you want to run both the console and the Tivoli desktop at the same time, start the console using the **wsisgui** command as described in “Starting the Tivoli Software Installation Service console from the command line” on page 114.

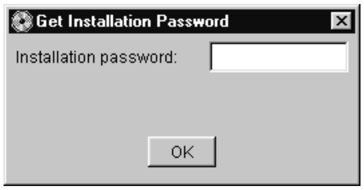
To start the console, perform the following steps from the Tivoli desktop:

1. From the **Desktop** menu, select **Install → SIS Client (local)**.

   If more than one Tivoli Software Installation Service depot exists in the Tivoli region, the Select SIS Depot window is displayed. If there is only one depot, the Get Installation Password window is displayed, as shown in step 2.

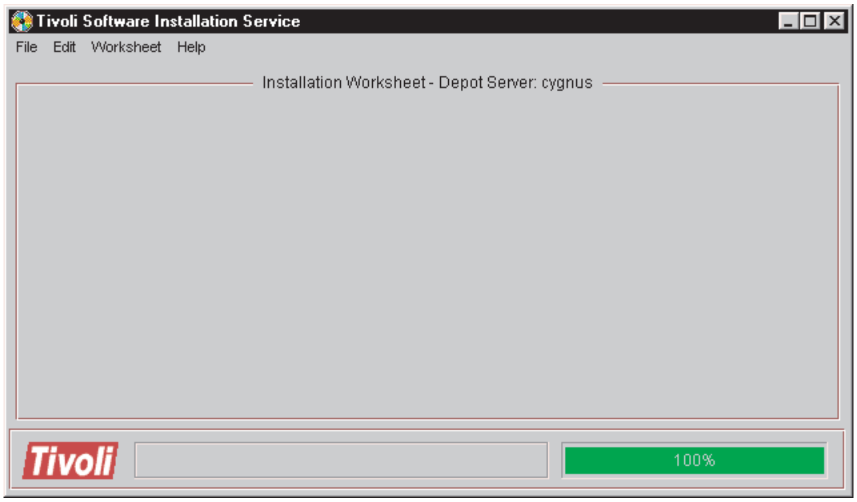


2. Select the appropriate depot and click **Connect**. The Get Installation Password window is displayed.



3. Type the installation password for the Tivoli region, if applicable. This is the password that was specified when the Tivoli server was installed or that was added later using the **odadmin set_install_pw** command. This password is required only when using Tivoli Software Installation Service to create managed nodes, not for other types of installations. Click **OK**. The Tivoli Software Installation Service main window is displayed.

   Use this window to create an installation worksheet. You use the installation worksheet to specify the products to install, the machines on which to install them, and values of the installation options.


Tivoli Software Installation Service
File
Edit
Worksheet
Help
Installation Worksheet - Depot Server: cygnus
Tivoli
100%

   The installation worksheet is initially empty unless there are unfinished installations from a previous session.

   Text at the top of the installation worksheet displays the name of the Tivoli Software Installation Service depot being used.

Starting the Tivoli Software Installation Service console from the command line

To start the Tivoli Software Installation Service console from the command line, perform the following steps:

1. Initialize the Tivoli environment variables as described in “Setting Tivoli environment variables” on page 30.

2. On UNIX systems, set the DISPLAY environment variable to the X Window System display on which you want to open the console.
3. Enter the following command to start the Tivoli Software Installation Service console:

```
wsisgui
```

Continue logging in to the console as described in “Starting the Tivoli Software Installation Service console from the Tivoli desktop” on page 113.

Working with products

This section describes the tasks relating to Tivoli products and patches:

- “Importing products into the install repository”
- “Removing products from the install repository” on page 123
- “Viewing and customizing installation options” on page 119
- “Adding products to the installation worksheet” on page 127

Note: Tivoli Software Installation Service treats products and patches in the same way. Unless otherwise stated, discussions about products pertain to patches as well.

Importing products into the install repository

The install repository stores all products available for installation on machines in the Tivoli region. Before you can use Tivoli Software Installation Service to install Tivoli products, you must import the images for each product into the install repository from the installation media, typically a CD. A product consists of one or more components, each of which can be imported. When imported, you can use these images to install the products on the machines you select.

You can import the components of a Tivoli product in one of the following ways:

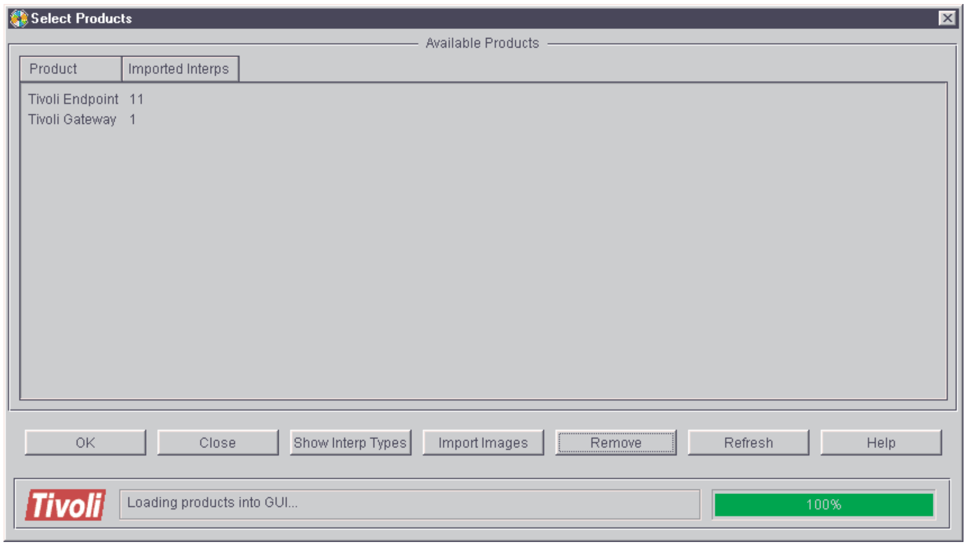
- Using the console, as described in “Importing products using the Tivoli Software Installation Service console”
- Using the **wimport** command, as described in “Importing products from the command line” on page 118

Typically, you import installation images from a product CD, or from a copy of a product CD. However, you can also import from another install repository. This is described in “Importing images from another install repository” on page 119.

Importing products using the Tivoli Software Installation Service console

To import products into the install repository using the console, perform the following steps:

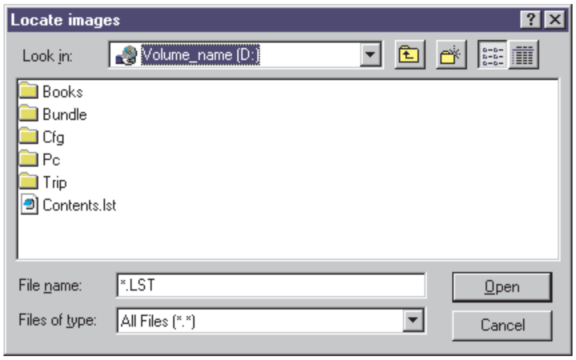
1. From the installation worksheet, select **Worksheet → Select products**. The Select Products window is displayed, which shows all products in the install repository.



Product	Imported Interps
Tivoli Endpoint	11
Tivoli Gateway	1

Loading products into GUI... 100%

2. Click **Import Images** to add products to the install repository. A window similar to the following is displayed:



Look in: Volume_name (D:)

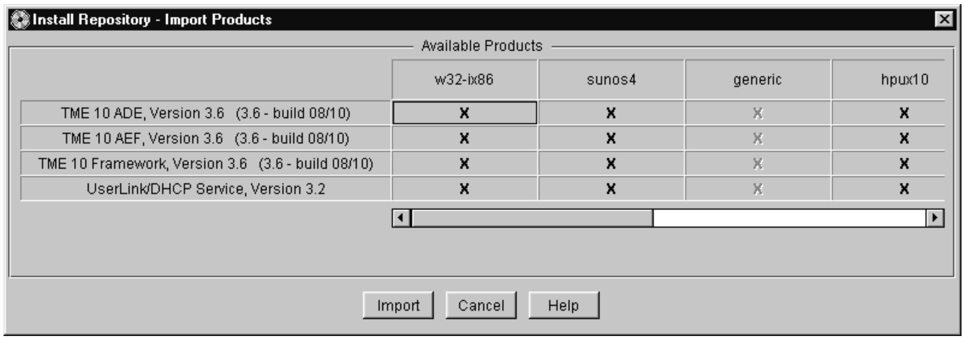
- Books
- Bundle
- Cfg
- Pc
- Trip
- Contents.lst

File name: *.LST

Files of type: All Files (*.*)

3. Navigate to the directory that contains the installation images. The installation directory contains a file named CONTENTS.LST for products and PATCHES.LST for patches.
4. Select any file in the directory and click **Open**.

   The Import Products window is displayed and contains all products and patches in the selected directory and its subdirectories. Each component corresponds to a product index (.IND) file.



	w32-ix86	sunos4	generic	hpux10
TME 10 ADE, Version 3.6 (3.6 - build 08/10)	X	X	X	X
TME 10 AEF, Version 3.6 (3.6 - build 08/10)	X	X	X	X
TME 10 Framework, Version 3.6 (3.6 - build 08/10)	X	X	X	X
UserLink/DHCP Service, Version 3.2	X	X	X	X

5. Deselect any components that you do not need in your Tivoli environment and deselect any interpreter types that you do not need for a specific component.

   By default, everything is selected. That is, all components and all interpreter types for each component are selected for import into the install repository.

However, to reduce the amount of data in your install repository or to shorten the time required to import products, you can import a subset of components and interpreter types.

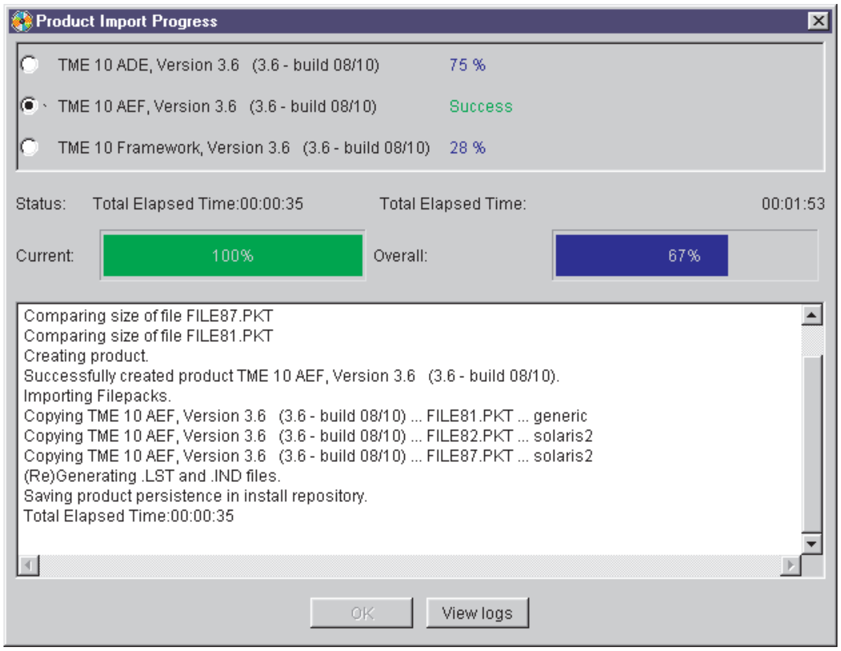
Select the items to be imported using these methods:

- To select or deselect all interpreter types for a component, click the name of the component.
- To select or deselect all components for an interpreter type, click the name of the interpreter type.
- To select or deselect a specific combination of component and interpreter type, click the corresponding cell in the installation worksheet.

You cannot deselect the **generic** interpreter type, but it is not imported unless other interpreter types are selected for that component.

Note: You can reimport a product to add interpreter types. Only interpreter types not currently in the install repository or interpreter types for which the installation images have changed are imported. Interpreter types previously imported are not removed from the install repository.

6. After you select the components and interpreter types that you want to import, click **Import**. The Product Import Progress window is displayed, which allows you to view the status of each component as it is imported into the install repository.


Product Import Progress
○ TME 10 ADE, Version 3.6 (3.6 - build 08/10) 75 %
● TME 10 AEF, Version 3.6 (3.6 - build 08/10) Success
○ TME 10 Framework, Version 3.6 (3.6 - build 08/10) 28 %
Status: Total Elapsed Time:00:00:35 Total Elapsed Time: 00:01:53
Current: 100%
Overall: 67%
OK
View logs

```
Comparing size of file FILE87.PKT
Comparing size of file FILE81.PKT
Creating product.
Successfully created product TME 10 AEF, Version 3.6   (3.6 - build 08/10).
Importing Filepacks.
Copying TME 10 AEF, Version 3.6   (3.6 - build 08/10) ... FILE81.PKT ... generic
Copying TME 10 AEF, Version 3.6   (3.6 - build 08/10) ... FILE82.PKT ... solaris2
Copying TME 10 AEF, Version 3.6   (3.6 - build 08/10) ... FILE87.PKT ... solaris2
(Re)Generating .LST and .IND files.
Saving product persistence in install repository.
Total Elapsed Time:00:00:35
```

The status bars are color-coded to indicate the progress of the product being imported into the install repository.

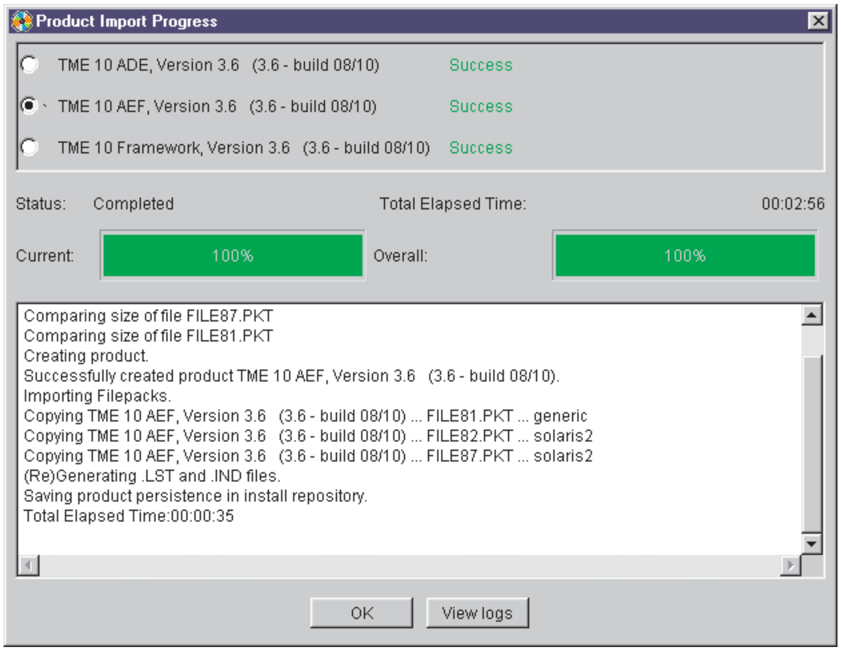
Blue Indicates that the product is being imported.

Green Indicates that the product was successfully imported.

Red Indicates that the import process failed. Click **View logs** at any time to view detailed HTML-based error messages. Refer to Chapter 14, “Using Tivoli Software Installation Service log files,” on page 171 for information about using the log files.

At any time during the import process, click the button to the left of the status bar to display the progress for that product. The Product Import Progress window displays the progress for only one product at a time.

7. Click **OK** when the process is complete.



○ TME 10 ADE, Version 3.6 (3.6 - build 08/10) Success

● TME 10 AEF, Version 3.6 (3.6 - build 08/10) Success

○ TME 10 Framework, Version 3.6 (3.6 - build 08/10) Success

Status: Completed Total Elapsed Time: 00:02:56

Current: 100% Overall: 100%

```
Comparing size of file FILE87.PKT
Comparing size of file FILE81.PKT
Creating product.
Successfully created product TME 10 AEF, Version 3.6  (3.6 - build 08/10).
Importing Filepacks.
Copying TME 10 AEF, Version 3.6  (3.6 - build 08/10) ... FILE81.PKT ... generic
Copying TME 10 AEF, Version 3.6  (3.6 - build 08/10) ... FILE82.PKT ... solaris2
Copying TME 10 AEF, Version 3.6  (3.6 - build 08/10) ... FILE87.PKT ... solaris2
(Re)Generating .LST and .IND files.
Saving product persistence in install repository.
Total Elapsed Time:00:00:35
```

After the import process completes, the imported components are listed in the **Available Products** list in the Select Products window.

8. Verify that the default values of installation options for newly imported components are correct for your Tivoli environment. This is described in “Viewing and customizing installation options” on page 119.

Importing products from the command line

As an alternative to the Tivoli Software Installation Service console, you can use the **wimport** command to import products into the install repository. You can import a complete product or selected interpreter types for a product.

- To import the managed node image for all interpreter types, enter the following command:

  ```
  wimport -c e:\\ -i TMF
  ```

  where e:\ represents the path to the managed node image, and TMF is the name of its index (.IND) file.

- To import only the interpreter types necessary to install managed nodes on interpreter types **w32-ix86** (Windows operating systems) and **solaris2** (Solaris operating environments), enter the following command:

  ```
  wimport -c e:\\ -i TMF w32-ix86 solaris2
  ```

  where e:\ represents the path to the managed node image, and TMF is the name of the index (.IND) file.

  Specifying w32-ix86 and solaris2 indicates that only those interpreter types are imported. The **generic** operating system type is automatically imported; you do not need to specify it.

After importing a new component, verify that the default values of its installation options are correct for your Tivoli environment. This is described in “Viewing and customizing installation options” on page 119.

Refer to Appendix A, “Installation Commands” for details on the **wimport** command.

Importing images from another install repository

If you have an install repository from another installation of Tivoli Software Installation Service in a file system accessible to the Tivoli Software Installation Service client, you can import installation images from it into the install repository of the Tivoli Software Installation Service depot. The source install repository can be from another depot.

To import images from another repository, follow the procedures in “Importing products into the install repository” on page 115, but specify the directory that contains the install repository from which you want to import. The Import Product window will list each product contained in the source install repository.

After importing, set the default values of installation options. The default values are not retained when copied from another repository. The procedures are described in the following section.

Viewing and customizing installation options

You can modify the installation options of a product for one machine or for all installations to better fit the needs of your environment. This section describes how to modify the default values of installation options for all installations of a product. For information about changing the values for a specific machine, refer to “Overriding default installation options for one machine” on page 137.

The installation options correspond to those in the Install Options window that is displayed when you install the product from the Tivoli desktop and to the installation variables you specify when you install the product from the command line. For details about the installation options for a specific product, refer to the product documentation. The installation options for managed nodes and gateways are described in “Installation options for managed nodes” on page 200 and “Installation options for gateways” on page 208. Information about installation options for endpoints is found in “Endpoint installation options” on page 149.

Tivoli Software Installation Service enables you to create default installation options that contain machine-specific information by providing variables that you can use to define an option. These include many of the variables described in “Using variables in prerequisite scripts” on page 193. For example, one of the installation options for a gateway is the gateway label, which defaults to its host name with the string `-gateway` appended (*`hostname`*`-gateway`). To specify this, use the @HostName@ variable and set that field to `@HostName@-gateway`.

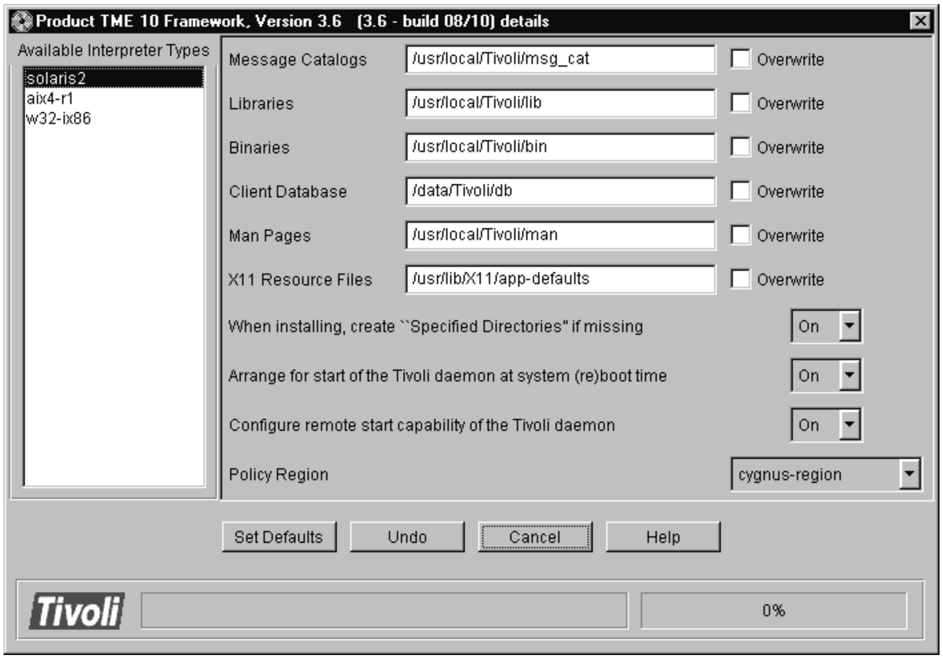
You can set the installation options for a product for all machines using either the Tivoli Software Installation Service console or commands. Using the console, you set defaults separately for each interpreter type. Using the commands, you can easily set the same defaults for multiple interpreter types. These topics are described in the following sections.

Note: Limit the number of administrators who can customize installation options. The values of the installation options are read from the Tivoli Software Installation Service depot at the time each cell is selected. You will get unexpected results if another administrator changes any installation option while you are completing an installation worksheet. If you change an option and another administrator changes its value before you select the cell, the last set value is used. After you select a cell, its current value is used, and any changes made by another administrator does not affect this installation.

Setting product defaults using the Tivoli Software Installation Service console

To set the default installation options for a product from the console, perform the following steps:

1. In the installation worksheet, right-click the product name to display the installation options for that product. A Product details window similar to the following is displayed.



Product TME 10 Framework, Version 3.6 (3.6 - build 08/10) details

Available Interpreter Types

solaris2
aix4-r1
w32-ix86

Message Catalogs	/usr/local/Tivoli/msg_cat	□ Overwrite
Libraries	/usr/local/Tivoli/lib	□ Overwrite
Binaries	/usr/local/Tivoli/bin	□ Overwrite
Client Database	/data/Tivoli/db	□ Overwrite
Man Pages	/usr/local/Tivoli/man	□ Overwrite
X11 Resource Files	/usr/lib/X11/app-defaults	□ Overwrite

When installing, create ``Specified Directories'' if missing: On

Arrange for start of the Tivoli daemon at system (re)boot time: On

Configure remote start capability of the Tivoli daemon: On

Policy Region: cygnus-region

2. Select an interpreter type from the **Available Interpreter Types** list. The default installation options for that interpreter type are displayed on the right-hand side of the window.
3. Type new values for the selected interpreter type. Path variables display with an **Overwrite** check box. If you are reinstalling a product, select **Overwrite** to force the files in that directory to be replaced.
4. Repeat steps 2 and 3 for each interpreter type. You do not need to save the default values before selecting the next interpreter type.

   To discard changes, click **Cancel**. This discards the new settings for all interpreter types.
5. Click **Set Defaults** to save the defaults for all machines installing this product from this install repository, or click **Cancel** to discard changes and close the window.

These settings are used as the default installation options each time this product is installed on a machine of that interpreter type.

Note: Setting the product defaults does not change the installation options for any machine already selected on the installation worksheet. If a cell in the installation worksheet is already selected, you must change the options for that machine individually, as described in “Overriding default installation options for one machine” on page 137.

Setting product defaults from the command line

To set the default installation options for a product for all machines from the command line, use the **wsisdefaults** command.

For example, to set the installation options for installing endpoints, perform the following steps:

1. Determine the exact name or number of the product in the install repository. To list each product in the install repository along with its default installation options, enter the following command:

   ```
   wsisdefaults -l
   ```

   The output of this command contains an alphabetic list of products and defaults, similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   Reading defaults from cygnus

   --------------Product 1--------------
   Product Name: GATEWAY-Tivoli_Gateway

        Interp Name: generic
   @GATE_NAME@ = @HostName@-gateway
   @GATE_PORT@ = 9494

   --------------Product 4--------------
   Product Name: TMA-Tivoli_Endpoint

        Interp Name: aix4-r1
   TMABIN = /opt/Tivoli/lcf
   @EndpointLabel@ = @HostName@
   @EndpointPort@ = 9495
   @EndpointStartupOpts@ =
   @GatewayName@ = Broadcast to Gateways
   @PolicyRegionName@ = None
   @CheckLogin@ = On
   @EndpointStartupTimeout@ = 300

        Interp Name: hpux10
   TMABIN = /opt/Tivoli/lcf
   @EndpointLabel@ = @HostName@
   @EndpointPort@ = 9495
   @EndpointStartupOpts@ =
   @GatewayName@ = Broadcast to Gateways
   @PolicyRegionName@ = None
   @CheckLogin@ = On
   @EndpointStartupTimeout@ = 300

   ... <additional interpreter types deleted> ...
   --------------Product 5--------------
   Product Name: TMF-client-3.7

        Interp Name: solaris2
   CAT = /usr/local/Tivoli/msg_cat
   LIB = /usr/local/Tivoli/lib
   BIN = /usr/local/Tivoli/bin
   DB = /data/Tivoli3.7
   MAN = /usr/local/Tivoli/man
   APPD = /usr/lib/X11/app-defaults
   @CreatePaths@ = On
   @AutoStart@ = On
   @SetPort@ = On
   PR_NAME = NoonTide-Region
   ```

2. Create a product defaults file that contains the current default values for the desired products. The following command creates a file in the current directory called moredefaults containing the current defaults for installing endpoints (`TMA-Tivoli_Endpoint`):

   ```
   wsisdefaults -x moredefaults TMA-Tivoli_Endpoint
   ```

   You can also use the product number. For example, the previous output shows product `TMA-Tivoli_Endpoint` as product `4`, therefore, the following command has the same results:

   ```
   wsisdefaults -x moredefaults 4
   ```
3. Modify the moredefaults file to change the installation options as needed, following the syntax described in "Syntax of the product defaults file."

   When you export a product defaults file, it contains a separate section for each interpreter type. To use identical default values for multiple interpreter types, delete all but one of the sections and specify multiple interpreter types in the remaining section.
4. To set the product option defaults using the information in the moredefaults file, enter the following command:

   ```
   wsisdefaults -i moredefaults
   ```

For more information about the **wsisdefaults** command, refer to "wsisdefaults" on page 370.

Syntax of the product defaults file

The product defaults file has the following syntax:

[**default** *interp*[,*interp*]... *product_ID*] *install_option=value*

Descriptions of the entries follow:

default
A keyword that identifies the section.

interp
The name of one or more interpreter types to which this section applies. To specify that multiple interpreter types use the same product defaults, use a comma-separated list such as `solaris2,w32-ix86,aix4-r1`. You can set default values for only those interpreter types in the install repository.

product_ID
The unique product ID. This must match the unique product ID that Tivoli Software Installation Service creates when importing the product into the depot. Use the **–l** option of the **wsisdefaults** command to determine the unique product ID.

install_option=value
An installation option specific to this product. Some installation option names are enclosed in at signs (@*install_option*@). Determine whether this is necessary for each option by examining the options that are exported by the **–x** option of the **wsisdefaults** command. You can have any number of these entries in the product defaults file.

Each *install_option* corresponds to an installation option in the Install Options window that is displayed when you install the product from the Tivoli desktop and to an installation variable you specify when you install the product from the command line. Refer to the installation instructions for each product for information about installation options and values, particularly if you create the product defaults file without using the **–x** option of the **wsisdefaults** command.

There are special entries for managed nodes, endpoints, and gateways. **wsisdefaults** uses the same options as are used in the [alias] section of a response file. For more information, refer to "Syntax—product-specific" on page 149.

The **–x** option of the **wsisdefaults** command creates a separate [default] section in the product defaults file for each operating system type in the Tivoli Software Installation Service depot. However, when creating your product defaults file or when modifying a template created by the **–x** option of the **wsisdefaults** command, you can specify multiple operating system types in one [default] section. This makes it easier to use one [default] section to set identical product defaults for multiple operating system types.

The following section of a product defaults file specifies the default installation options for managed nodes, whose unique product ID is `TMF-client-3.7`. The interpreter types `aix4-r1`, `solaris2`, and `hpux10` use the same values, but `w32-ix86` uses different values.

```
[default aix4-r1,solaris2,hpux10 TMF-client-3.7]
CAT=/Tivoli/msg_cat
LIB=/Tivoli/lib
BIN=/Tivoli/bin
DB=/Tivoli/db
MAN=/Tivoli/man
APPD=/usr/lib/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=UNIXMN-Region

[default w32-ix86 TMF-client-3.6]
CAT=c:/usr/local/Tivoli/msg_cat
LIB=c:/usr/local/Tivoli/lib
BIN=c:/usr/local/Tivoli/bin
DB=c:/Tivoli/db
MAN=c:/usr/local/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=NTMN-Region
tapUser=TivoliUser
TapPassword=tap$pw
Reboot=Yes
```

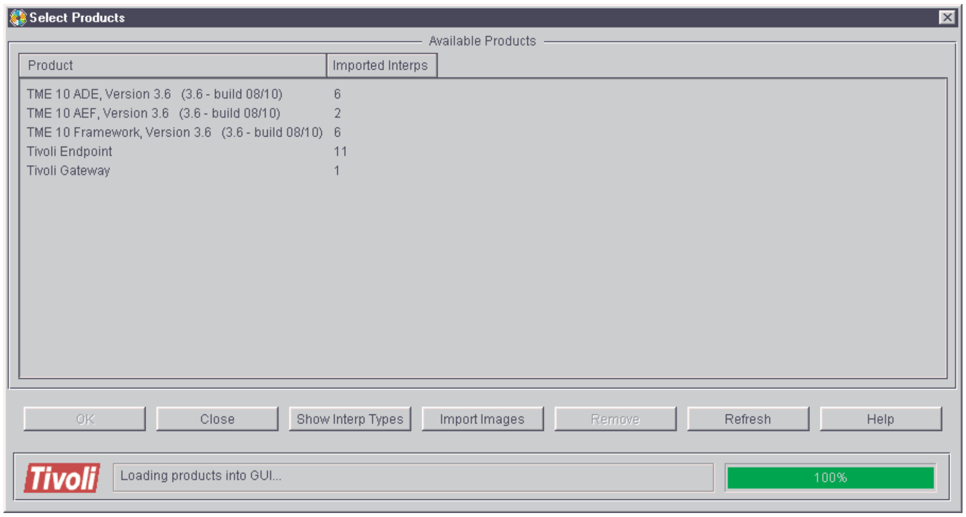
Removing products from the install repository

When Tivoli products are no longer needed in the install repository, or when you no longer need an interpreter type for one or more products, you can remove them using either the Tivoli Software Installation Service console or the command line. These procedures are described in the following sections.

Removing products using the Tivoli Software Installation Service console

To remove Tivoli products from the install repository using the console, perform the following steps:

1. From the installation worksheet, select **Worksheet → Select products** to open the Select Products window.

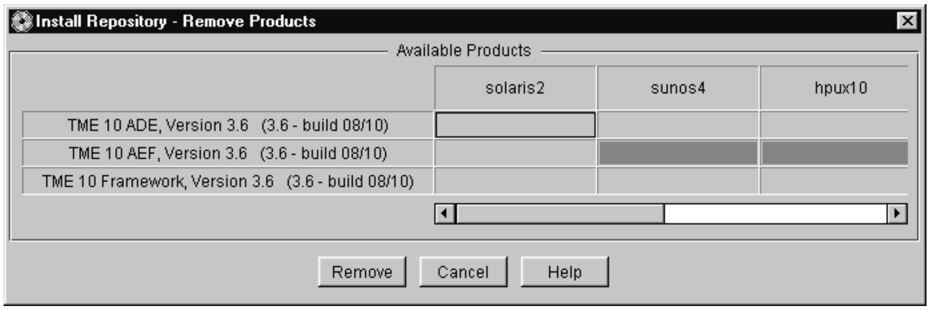


Product	Imported Interps
TME 10 ADE, Version 3.6 (3.6 - build 08/10)	6
TME 10 AEF, Version 3.6 (3.6 - build 08/10)	2
TME 10 Framework, Version 3.6 (3.6 - build 08/10)	6
Tivoli Endpoint	11
Tivoli Gateway	1

Loading products into GUI...

100%

2. Select one or more items from the **Product** list, and click **Remove** to open the Remove Products window.

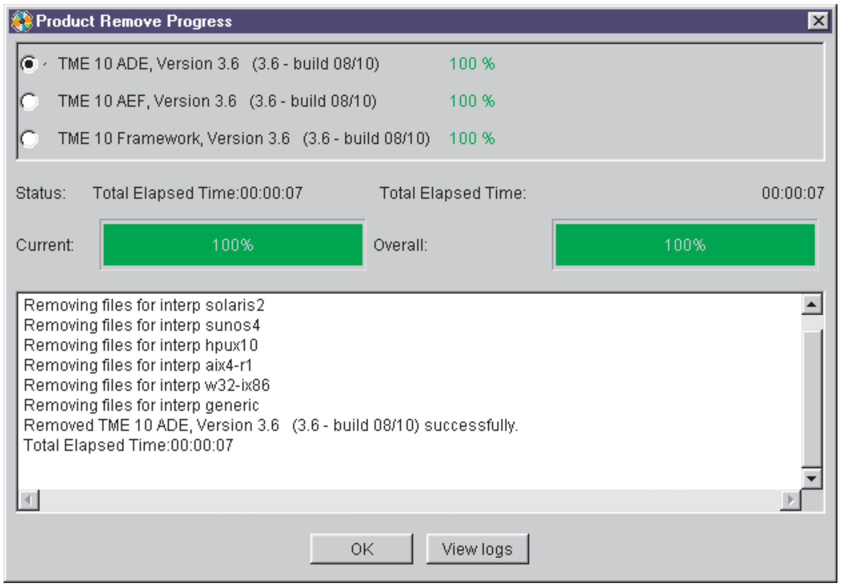


	solaris2	sunos4	hpux10
TME 10 ADE, Version 3.6 (3.6 - build 08/10)			
TME 10 AEF, Version 3.6 (3.6 - build 08/10)		▨	▨
TME 10 Framework, Version 3.6 (3.6 - build 08/10)			

3. Select the product and interpreter type combinations you want to remove. Selected cells are indicated by an **X**. Select the items to be removed using these methods:
   - To select or deselect all interpreter types for a product, click the name of the product.
   - To select or deselect an interpreter type for all products, click the name of the interpreter type.
   - To select or deselect a specific combination of product and interpreter type, click the corresponding cell.

   Note: The **generic** interpreter type is removed after all other interpreter types for a product are removed.

   Dark gray shading in a cell indicates that the install repository does not contain the combination of product and interpreter type.
4. Click **Remove** to remove the selected items. The Product Remove Progress window is displayed.


Product Remove Progress
● TME 10 ADE, Version 3.6 (3.6 - build 08/10) 100 %
○ TME 10 AEF, Version 3.6 (3.6 - build 08/10) 100 %
○ TME 10 Framework, Version 3.6 (3.6 - build 08/10) 100 %
Status: Total Elapsed Time:00:00:07 Total Elapsed Time: 00:00:07
Current: 100%
Overall: 100%
OK
View logs

```
Removing files for interp solaris2
Removing files for interp sunos4
Removing files for interp hpux10
Removing files for interp aix4-r1
Removing files for interp w32-ix86
Removing files for interp generic
Removed TME 10 ADE, Version 3.6 (3.6 - build 08/10) successfully.
Total Elapsed Time:00:00:07
```

Use this window to view the status of each component as it is removed from the install repository. The status bars are color-coded to indicate the progress of the product being removed from the install repository, as follows:

Blue Indicates that the product is being removed.

Green Indicates that the product was successfully removed.

Red Indicates that the remove process failed. Click **View logs** at any time to view detailed HTML-based error messages. Refer to Chapter 14, “Using Tivoli Software Installation Service log files,” on page 171 for information about using the log files.

At any time during the remove process, click the button to the left of the status bar for a product to display the progress for that product. The Product Remove Progress window displays the progress for only one product at a time.

5. Click **OK** when the process is completed.

After the removal process is complete, the Select Products window is updated to show the current list of available products and its number of interpreter types in the install repository.

Removing products from the command line

To remove a product or selected interpreter types for a product from the command line, use the **wimport** command. The **wimport** command requires that you specify the product to be removed by its unique product ID in the install repository.

Removing a product: To remove all interpreter types for a product from the install repository from the command line, perform the following steps:

1. Determine the name or number under which the product is stored in the install repository:

   ```
   wimport -l
   ```

   This command returns output similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus

   ==================================
   Products found in Depot:
   ```

```
--------------Product 1--------------
Product Name: Admin-3.6.2-Tivoli_User_Administration_3.6.2
Interp Name: generic
Interp Name: solaris2
Interp Name: aix4-r1
Interp Name: w32-ix86

--------------Product 2--------------
Product Name: GATEWAY-Tivoli_Gateway
Interp Name: generic

--------------Product 3--------------
Product Name: TMA-Tivoli_Endpoint
Interp Name: aix4-r1
Interp Name: hpux10
Interp Name: nw3
Interp Name: nw4
Interp Name: os2
Interp Name: os2-ix86
Interp Name: solaris2
Interp Name: w32-ix86
Interp Name: win95
Interp Name: nt
Interp Name: windows
Interp Name: netware
Interp Name: os400
```

You can remove a product by the number or product name displayed in the output.

2. To remove product number 1, named Admin-3.6.2-Tivoli_User_Administration_3.6.2, enter either of the following commands:

   ```
   wimport -r 1
   ```

   or

   ```
   wimport -r Admin-3.6.2-Tivoli_User_Administration_3.6.2
   ```

Refer to **wimport** for additional information about using this command.

Removing interpreter types for a product: To remove selected interpreter types from a product in the install repository from the command line, perform the following steps:

1. Determine the name or number under which the product is stored in the install repository:

   ```
   wimport -l
   ```

   This command returns output similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus

   ==================================
   Products found in Depot:

   --------------Product 1--------------
   Product Name: Admin-3.6.2-Tivoli_User_Administration_3.6.2
   Interp Name: generic
   Interp Name: solaris2
   Interp Name: aix4-r1
   Interp Name: w32-ix86

   --------------Product 2--------------
   Product Name: GATEWAY-Tivoli_Gateway
   Interp Name: generic
   ```

```
--------------Product 3--------------
Product Name: TMA-Tivoli_Endpoint
Interp Name: aix4-r1
Interp Name: hpux10
Interp Name: nw3
Interp Name: nw4
Interp Name: os2
Interp Name: os2-ix86
Interp Name: solaris2
Interp Name: w32-ix86
Interp Name: win95
Interp Name: nt
Interp Name: windows
Interp Name: netware
Interp Name: os400
```

You can remove a product by the number or product name displayed in the output.

2. To remove files for the **solaris2** and **w32-ix86** interpreter types for product number 1, named Admin-3.6.2-Tivoli_User_Administration_3.6.2, enter either of the following commands:

```
wimport -r 1 solaris2 w32-ix86
```

or

```
wimport -r Admin-3.6.2-Tivoli_User_Administration_3.6.2 \
solaris2 w32-ix86
```

In this example, because the install repository contains additional interpreter types for this product, the product remains available. If **solaris2** and **w32-ix86** were the only remaining interpreter types for this product, the generic interpreter type would be removed, which would remove the product from the install repository.

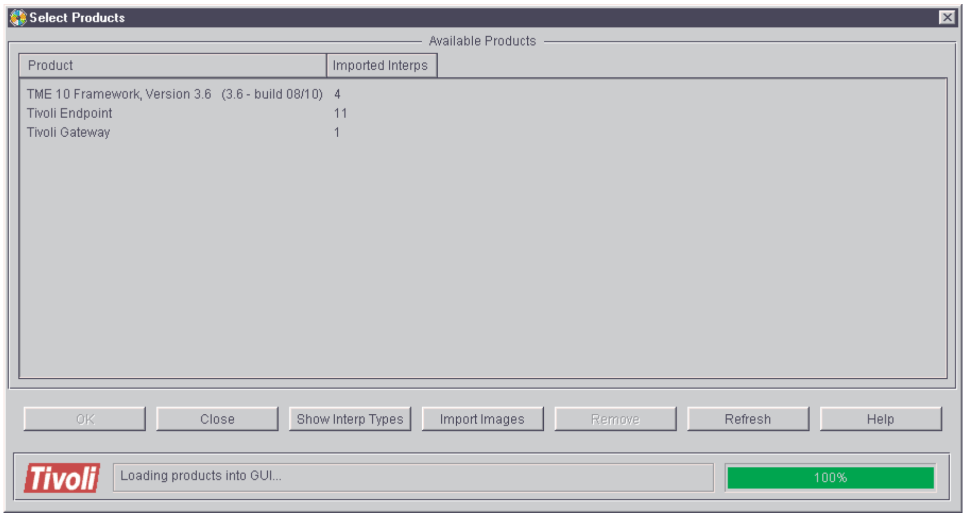
Refer to **wimport** for additional information about using this command.

Adding products to the installation worksheet

Because Tivoli Software Installation Service can install multiple products on multiple machines at the same time and keeps track of which products go on which machines, you can add *all* the products you will install on *any* machines.

To select the products that you want to install, perform the following steps:

1. From the installation worksheet, select **Worksheet → Select products** to open the Select Products window.

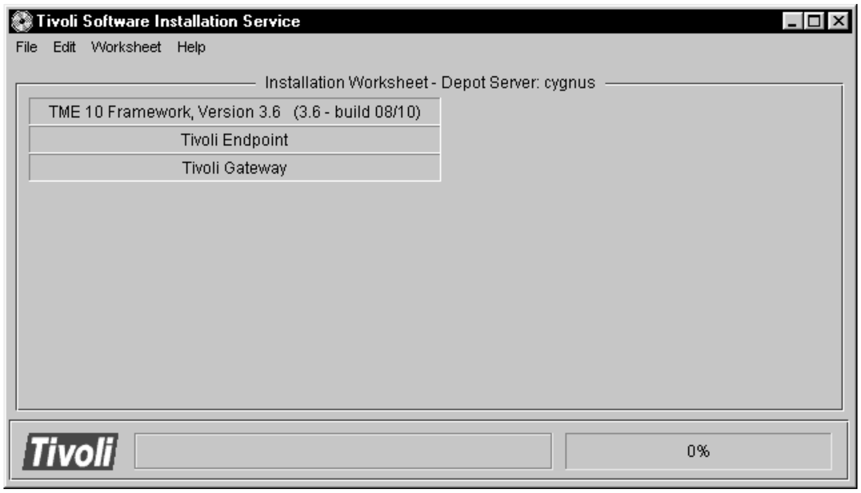


Product	Imported Interps
TME 10 Framework, Version 3.6 (3.6 - build 08/10)	4
Tivoli Endpoint	11
Tivoli Gateway	1

Loading products into GUI...

2. To sort the list by product name or by the number of interpreter types available in the install repository, click **Product** or **Imported Interps** at the top of the window.

   If you want to make sure that the install repository contains the necessary interpreter types for this product, follow the procedures described in "Viewing the available interpreter types."
3. Select one or more products from the **Product** list and click **OK**. The selected products are now listed in the installation worksheet.



Installation Worksheet - Depot Server: cygnus

- TME 10 Framework, Version 3.6 (3.6 - build 08/10)
- Tivoli Endpoint
- Tivoli Gateway

4. Check the installation options for each product that you imported, especially those that require a vendor database. Installation options vary by product and are described in the product documentation. If necessary, change the options as described in "Viewing and customizing installation options" on page 119.

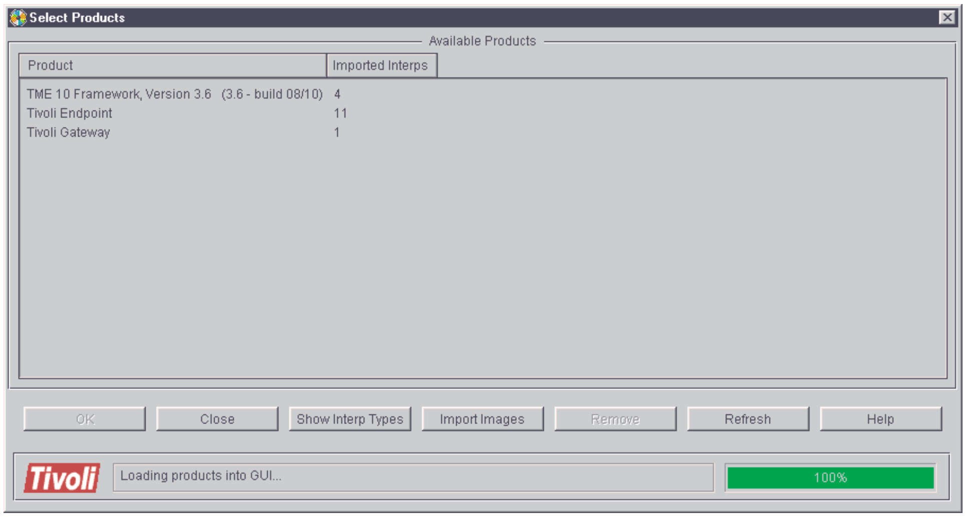
Viewing the available interpreter types

You can install a product only on machines whose interpreter types are in the install repository. You can check whether the necessary interpreter types are in the install repository using either the Tivoli Software Installation Service console or the command line.

Viewing interpreter types using the Tivoli Software Installation Service console

To determine which interpreter types are in the install repository for one or more products, perform the following steps:

1. From the installation worksheet, select **Worksheet → Select products** to open the Select Products window.



Product	Imported Interps
TME 10 Framework, Version 3.6 (3.6 - build 08/10)	4
Tivoli Endpoint	11
Tivoli Gateway	1

Loading products into GUI... 100%

2. From the Select Products window, click **Show Interp Types**. The Available Interpreter Types window is displayed, which lists the interpreter types that are available for each product in the install repository.

   An **X** in a cell indicates that the install repository contains the files that install the product on that interpreter type. A dark gray cell indicates that the interpreter type is not available, either because it was not imported into the install repository or because it is not supported by the product. This window lists only the interpreter types that are in the install repository for at least one product; additional interpreter types may be supported.

   If the interpreter type that you need is supported but not available in the install repository, import it as described in "Adding interpreter types."

3. Click **OK** to close the window and return to the Select Products window.

Viewing interpreter types from the command line

To determine which interpreter types are in the install repository for all products, enter the following command:

```
wimport -l
```

Adding interpreter types

You can add installation images for additional interpreter types at any time. This enables you to save time and disk space by importing only what you currently need, while providing the flexibility to add interpreter types as your environment changes.

To add installation images for additional interpreter types to the install repository, follow the procedures described in "Importing products into the install repository" on page 115, and import only the additional interpreter types that you need. You do not need to reimport the installation images for interpreter types already in your install repository.

Working with machines

The Tivoli Software Installation Service depot stores information about each machine in your Tivoli region. This information includes machine name, connection information, machine type, and installation options.

The following sections describe the tasks associated with the machine on which Tivoli Software Installation Service installs Tivoli products and patches:

- "Adding machines to the Tivoli Software Installation Service depot"
- "Removing machines from the Tivoli Software Installation Service depot" on page 134
- "Adding machines to the installation worksheet" on page 134
- "Viewing machine information" on page 136
- "Overriding default installation options for one machine" on page 137

For information about specifying which machines receive which products, refer to "Specifying products to install" on page 138.

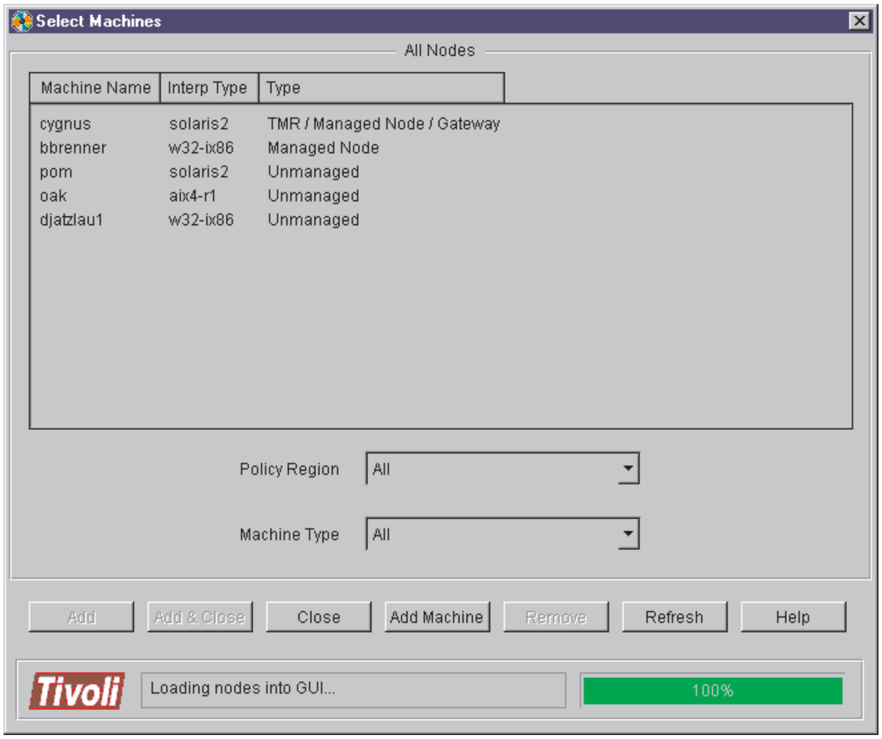
Adding machines to the Tivoli Software Installation Service depot

If you want to create Tivoli managed resources or install products on a machine that is not yet part of your Tivoli region, add the machine to the depot and provide the information needed to establish a remote connection to the machine. For a list of operating systems that can be added to Tivoli Software Installation Service, refer to the *Tivoli Management Framework Release Notes*.

Note: To add a large number of machines to your Tivoli region, use a response file to define these machines. The response file can be imported into the console or used from the command line. For details, refer to "Installing products from a response file" on page 159.

To add a machine to the list, use the following steps:

1. From the installation worksheet, select **Worksheet → Select machines** to open the Select Machines window, which lists the machines available for product installation.



Machine Name	Interp Type	Type
cygnus	solaris2	TMR / Managed Node / Gateway
bbrenner	w32-ix86	Managed Node
pom	solaris2	Unmanaged
oak	aix4-r1	Unmanaged
djatzlau1	w32-ix86	Unmanaged

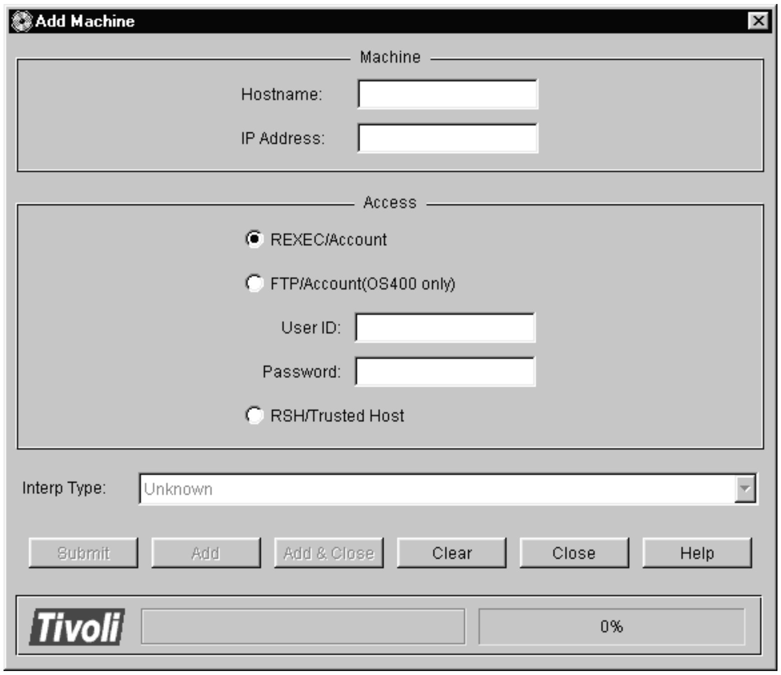
Policy Region: All

Machine Type: All

Loading nodes into GUI...

100%

2. From the Select Machines window, click **Add Machine**. The Add Machine window is displayed.



● REXEC/Account

○ FTP/Account(OS400 only)

○ RSH/Trusted Host

Interp Type: Unknown

0%

3. Type the machine name in the **Hostname** field.
4. Choose the appropriate access method for the machine that you want to add by selecting one of the buttons in the **Access** section of the window.

   The values for **User ID** and **Password** vary depending on the operating system of the machine that you are adding.

   - For REXEC on Windows machine, type the user ID and password for the Windows Administrator, not the Tivoli Administrator.

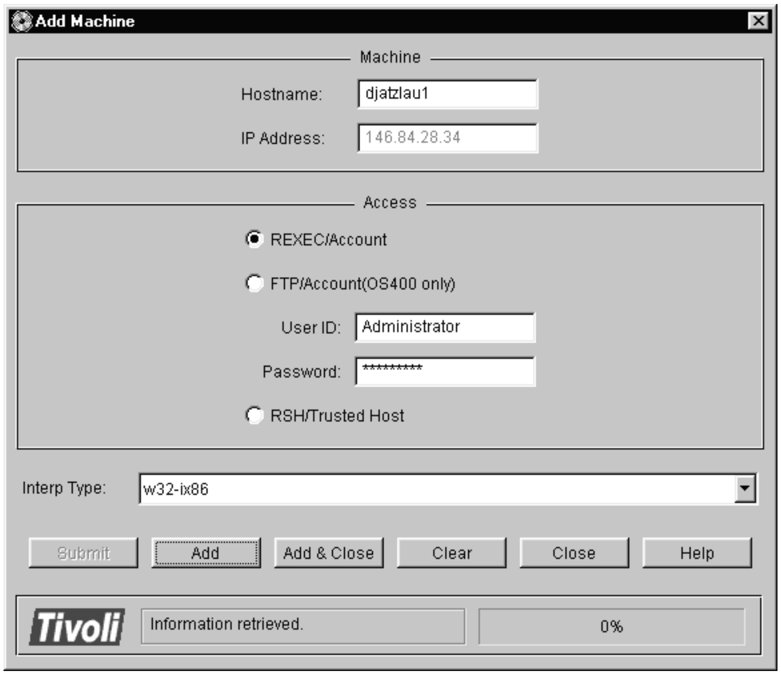
- For REXEC on UNIX machines, type the user ID and password for the root user. The login shell of this user must be Bourne or Korn shell.
- For REXEC on OS/2 machines, type the user ID and password account created with Security Enabling Services. For more information, refer to “Using OS/2 systems” on page 347.
- For file transfer protocol (FTP) on IBM OS/400 machines, type the user ID QSECOFR or another user ID with authority to issue OS/400 restore commands and with *SAVSYS authority.
- For RSH on UNIX machines, no user ID and password are required. You must ensure that the /.rhosts file on the target machine contains an entry for the user ID root (not another user with root authority) on the machine where the Tivoli Software Installation Service depot runs. The login shell of this user must be Bourne or Korn shell.

The access information is used only for the initial installation of managed nodes or endpoints on the machine. After the machine becomes a Tivoli client, Tivoli Software Installation Service uses a Tivoli connection to communicate with the machine.

5. Click **Submit**. Tivoli Software Installation Service uses the information you provided to make a test connection to the machine.

   If the test connection is successful, the `Information retrieved` message is displayed in the status area of the Add Machine window. The IP address of the machine is displayed in the **IP Address** field, and its interpreter type is displayed in the **Interp Type** field. (For most operating systems, the interpreter type is correct. If it does not match, you can change it.) You can now add the machine to the list of available machines.

   The following figure shows the window after a connection is established.



Hostname: djatzlau1

IP Address: 146.84.28.34

● REXEC/Account

○ FTP/Account(OS400 only)

User ID: Administrator

Password: *********

○ RSH/Trusted Host

Interp Type: w32-ix86

Information retrieved.

0%

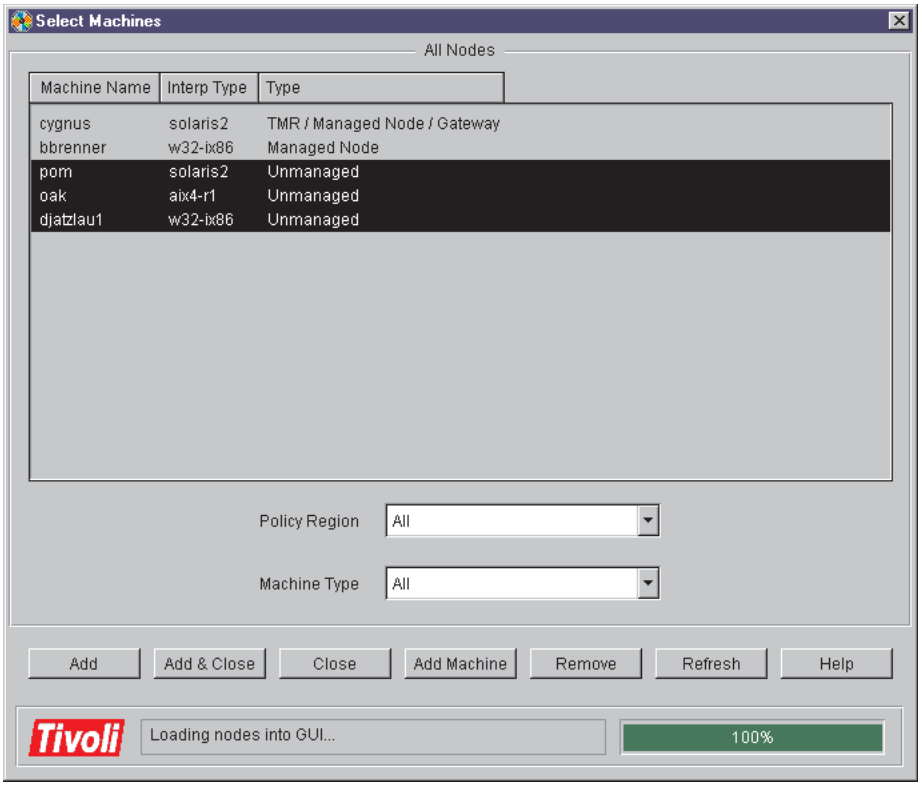
   If Tivoli Software Installation Service cannot contact the machine, the `Submission Failed` message is displayed in the status area of the Add Machine window, and a message window displays information about the failure. Common reasons for failure include the following:

   - The machine is not turned on, is not connected to the network, or is otherwise unreachable.

- The machine name is incorrect.
- The user ID or password is typed incorrectly.
- If you selected the RSH access method, the .rhosts file on that target machine does not have an entry for the user ID root on the machine running the Tivoli Software Installation Service depot.
- The machine is running the Windows operating system and Tivoli Remote Execution Service is not installed on it, and there is not another Windows machine in the Tivoli region that already has Tivoli Remote Execution Service installed on it. For more information, refer to “Using Tivoli Remote Execution Service” on page 341.

6. Click **Add** to add the machine to the list in the Select Machines window and continue adding additional machines.
7. Repeat steps 2 through 6 for each machine to be added to your Tivoli region. If desired, click **Clear** to clear the fields.
8. After establishing a test connection to the last machine that you want to add, click **Add & Close** to add the machine and close the Add Machine window.

The Select Machines window now displays all the machines, with the newly added ones already selected, as shown in the following figure.



Machine Name	Interp Type	Type
cygnus	solaris2	TMR / Managed Node / Gateway
bbrenner	w32-ix86	Managed Node
pom	solaris2	Unmanaged
oak	aix4-r1	Unmanaged
djatzlau1	w32-ix86	Unmanaged

Policy Region: All

Machine Type: All

Loading nodes into GUI...

100%

You can now select the machines on which you will install the products that you select. This is described in “Adding machines to the installation worksheet” on page 134.

Removing machines from the Tivoli Software Installation Service depot

You might need to remove a machine from the Tivoli Software Installation Service depot if the connection information provided on the Add Machine window is incorrect. You can remove a machine using only the Tivoli Software Installation Service console.

To remove a machine, perform the following steps:

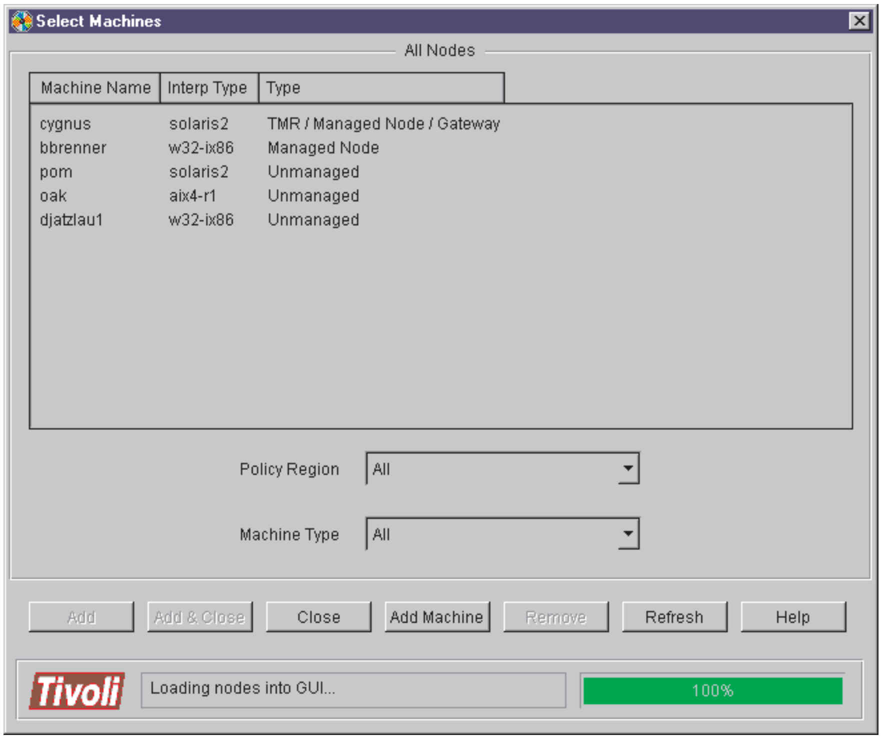
1. From the installation worksheet, select **Worksheet** → **Select machines** to display the Select Machines window, which lists the machines available for product installation.
2. In the **Policy Region** drop-down list, select **All**. In the **Machine Type** drop-down list, select **Unmanaged**. These selections display the machines that are known to Tivoli Software Installation Service but do not had a Tivoli product installed on them.
3. Select one or more machines to remove.
4. Click **Remove**.
5. Click **Close** to close the Select Machines window.

Adding machines to the installation worksheet

Because Tivoli Software Installation Service can install multiple products on multiple machines at the same time and keep track of which products go on which machines, you can add *all* the machines on which you want to install *any* software.

To specify the machines on which to install products, perform the following steps:

1. From the installation worksheet, select **Worksheet** → **Select machines** to display the Select Machines window, which lists the machines available for product installation.



Machine Name	Interp Type	Type
cygnus	solaris2	TMR / Managed Node / Gateway
bbrenner	w32-ix86	Managed Node
pom	solaris2	Unmanaged
oak	aix4-r1	Unmanaged
djatzlau1	w32-ix86	Unmanaged

Policy Region: All

Machine Type: All

Loading nodes into GUI...

100%

2. Click **Machine Name**, **Interp Type**, or **Type** at the top of the window to sort the list accordingly.

   To reduce the number of machines listed, use either or both of the **Policy Region** or **Machine Type** drop-down lists, as follows:

   - To filter by policy region, select the name of the policy region from the **Policy Region** drop-down list. The list of machines includes only the machines assigned to that policy region. Because endpoints and machines manually added to the install repository are not assigned to policy regions, select **All** to display them.
   - To filter by type of Tivoli managed resource, use the **Machine Type** drop-down list. The list of machines includes only the machines that have that type of Tivoli object installed on them. The following types are available:

     All Displays all machines in the install repository *except* endpoints.

     Unmanaged
     Displays machines that were manually added to the install repository. These machines are not yet Tivoli managed resources.

     Managed Nodes
     Displays managed nodes.

     Gateways
     Displays gateways.

     Endpoint Nodes
     Displays endpoints. Endpoints are not assigned to policy regions by default. To list all endpoints in your Tivoli region, use the policy region filter **All**.

     A machine can appear in more than one category if it has more than one type of Tivoli object installed on it. For example, it is possible to have a machine that is a managed node, a gateway, and an endpoint. Use the following guidelines to decide which category to use:

     – If a machine is both a managed node and an endpoint, select the one appropriate to the software you want to install. Typically, select the machine from the **Managed** Nodes list. Tivoli Enterprise software is not typically installed directly on an endpoint; for most products, endpoint methods are stored on the gateway and downloaded to the endpoint only as required.

       However, if the product documentation instructs you to install on an endpoint, select from the **Endpoint Nodes** list.

     – If the machine is both a managed node and a gateway, you can select the machine from **Gateways**. The **Gateways** category makes it easier to locate managed nodes when you are installing the gateway component of a product.

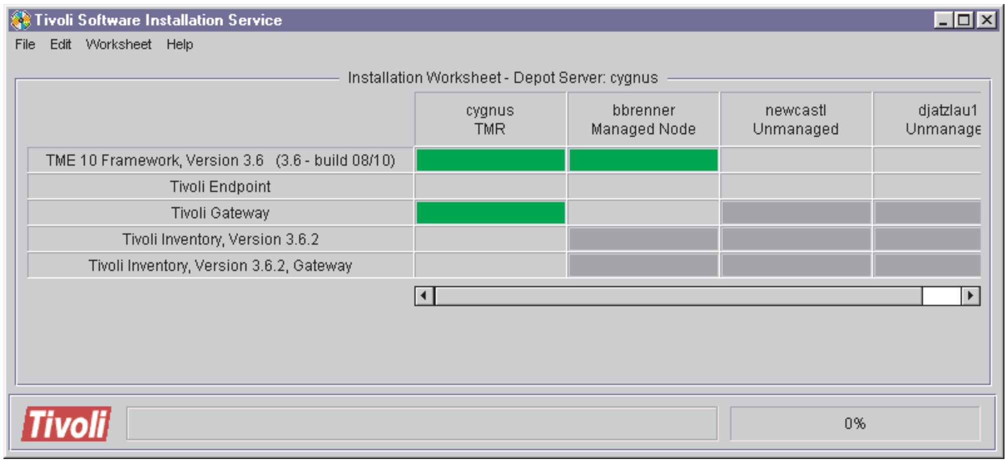
   For example, to display all machines in policy region cygnus-region, select cygnus-region from the **Policy Region** list and select **All** from the **Machine Type** list.

   To display only machines that are not currently managed resources (that is, machines that were manually added to the install repository), select **Unmanaged** from the **Machine Type** list and select **All** in the **Policy Region** field.

   To display only the managed nodes in policy region cygnus-region, select **Managed Nodes** from the **Machine Type** list and select cygnus-region from the **Policy Region** list.

3. From the Select Machines window, select one or more machines from the **Machine Name** list and click **OK**. The selected machines are added to the installation worksheet.

The following example shows an installation worksheet with multiple products and machines. You are now ready to specify which product to install on which machines.



Tivoli Software Installation Service
File
Edit
Worksheet
Help
Installation Worksheet - Depot Server: cygnus
Tivoli
0%

	cygnus TMR	bbrenner Managed Node	newcastl Unmanaged	djatzlau1 Unmanage
TME 10 Framework, Version 3.6 (3.6 - build 08/10)	🟩	🟩		
Tivoli Endpoint				
Tivoli Gateway	🟩		▨	▨
Tivoli Inventory, Version 3.6.2		▨	▨	▨
Tivoli Inventory, Version 3.6.2, Gateway		▨	▨	▨

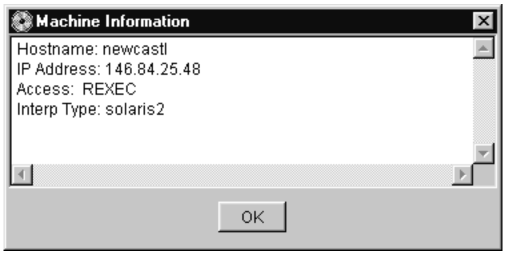
Viewing machine information

You can view information about a machine at any time. The information includes the host name, IP address, interpreter type, and the type of access Tivoli Software Installation Service will use to connect to the machine. You can obtain this information using either the Tivoli Software Installation Service console or commands.

Viewing machine information using the Tivoli Software Installation Service console

To view machine information using the console, perform the following steps:

1. In the installation worksheet, right-click the name of the machine. A Machine Information window similar to the following is displayed.



```
Hostname: newcastl
IP Address: 146.84.25.48
Access:  REXEC
Interp Type: solaris2
```

   The **Access** field of this window describes the connection method that Tivoli Software Installation Service will use to create the managed resource for the machine.

   An access type of **Tivoli** indicates that the machine is already a managed resource in this Tivoli region. In this case, a Tivoli connection is used and the IP address is not displayed. The access methods for unmanaged machines are described in step 4 on page 131.

2. Click **OK** to close the window and return to the installation worksheet.

Viewing machine information from the command line

You cannot display machine information from the command line. However, machine information is in the [machine] section of any response file that you export that has any product selected for installation. Refer to “Creating a response file template from the command line” on page 158 and “[machine] section” on page 146 for procedures.

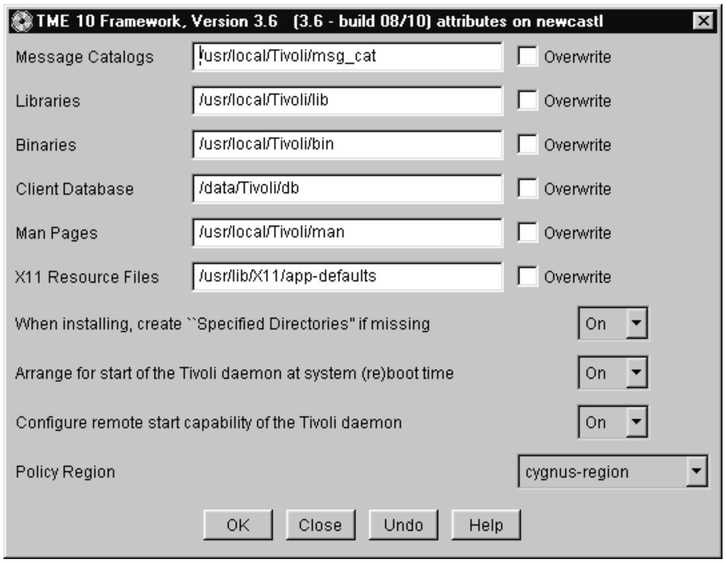
Overriding default installation options for one machine

You can override the installation options when installing a product on a specific machine. You can do this using either the Tivoli Software Installation Service console or commands, as described in the following sections.

Overriding default options using the Tivoli Software Installation Service console

To override the default installation options used when installing a product on a specific machine, perform the following steps using the console. You can do this when you initially select the cell or when it is already selected.

1. In the installation worksheet, right-click the cell corresponding to the desired product and machine. If the product has installation options, an attributes window is displayed. If the product has no installation options, the cell is selected and marked with an **X**.



TME 10 Framework, Version 3.6 (3.6 - build 08/10) attributes on newcastl

Message Catalogs /usr/local/Tivoli/msg_cat □ Overwrite

Libraries /usr/local/Tivoli/lib □ Overwrite

Binaries /usr/local/Tivoli/bin □ Overwrite

Client Database /data/Tivoli/db □ Overwrite

Man Pages /usr/local/Tivoli/man □ Overwrite

X11 Resource Files /usr/lib/X11/app-defaults □ Overwrite

When installing, create ``Specified Directories" if missing On

Arrange for start of the Tivoli daemon at system (re)boot time On

Configure remote start capability of the Tivoli daemon On

Policy Region cygnus-region

2. Type the values that you want to use when installing this product on the specified machine.

   The fields in this window correspond to those in the Install Options window that is displayed when you install the product from the Tivoli desktop and to the installation variables you specify when you install the product using the command line. For details about the installation option for a specific product, refer to the product documentation.

   Path variables are displayed with an **Overwrite** check box. If you are reinstalling a product, select **Overwrite** to force the files in that directory to be replaced.

   To return to the product defaults specified in the Product details window, click **Undo**.

3. Click **OK** to save the installation options for this machine, select the cell, and return to the installation worksheet.

If you deselect a cell after customizing the installation options for that machine, your changes are discarded. If you reselect the cell, its default installation options are used.

Overriding default options from the command line

To override the default installation options when installing a product on a specific machine from the command line, specify the desired values in the [alias] section of the response file. Refer to “[alias] section” on page 148 for more information.

Specifying products to install

The table cells in the installation worksheet are color-coded to indicate the status of the product on that machine.

Light Gray
Indicates that you can install the product.

Dark Gray
Indicates that you cannot install the product at this time. Either the dependencies are not met or the product files for the interpreter type of the specified machine are not in the install repository. Click the cell to display a message describing the specific reason.

Green Indicates that the product is already installed. You can select a green cell to reinstall the same version of a product. Be sure to specify that the installation should overwrite existing directories.

Yellow
Indicates that the product cannot be installed. Either the product is included in or overridden by a product that is selected or an equivalent product (that is, one with the same product tag) is already installed. Click the cell to display a message describing the specific reason.

Use any of the following methods to select which products and machines to install:

- To select or deselect all products on a machine, click the machine name.
- To select or deselect a product on all machines, click the product name.
- To select or deselect a specific product on one machine, click the light gray cell that corresponds to the desired product and machine combination. The installation uses the default values for the installation options.
- To select a specific product on one machine and override the default values of the installation options, right-click the cell. In the Product attributes window, modify the installation options and click **OK** to select the cell. (If you click **Close**, the cell is not selected.)
- To force the reinstallation of a product that is already installed, click the green cell and specify the overwrite option.

An **X** in the cell indicates that the product will be installed.

If you deselect a cell after customizing the installation options for that machine, your changes are discarded. If you reselect the cell, its default installation options are used.

Installing products

After you complete the installation worksheet, you can use it to install Tivoli software using either the Tivoli Software Installation Service console or the command line, as described in the following sections. Follow these same procedures if you want to check installation prerequisites without actually installing any products.

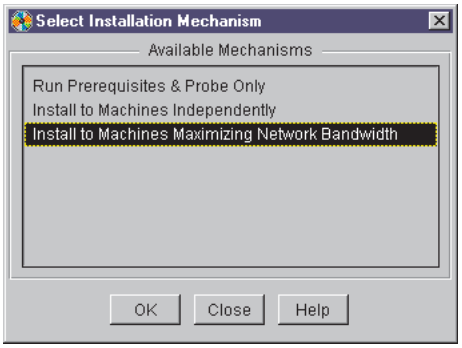
Using the Tivoli Software Installation Service console to install products

When the installation worksheet lists the Tivoli software that you want to install and the machines on which you want to install it, you can begin the installation process.

A completed installation worksheet looks similar to the following.

To install a product or patch on a machine, perform the following steps:

1. From the completed installation worksheet, select **Worksheet → Install** to begin the product installation. The Select Installation Mechanism window is displayed.


Select Installation Mechanism
Available Mechanisms
Run Prerequisites & Probe Only
Install to Machines Independently
Install to Machines Maximizing Network Bandwidth
OK
Close
Help

2. Select one of the following mechanisms in which the installation is to be performed:

   Run Prerequisites & Probe Only
   Without installing any products, runs the Tivoli- and user-defined prerequisites. The probe verifies that adequate disk space exists and that the directories are not read-only. This method is useful in identifying potential errors before installing products.

   Install to Machines Independently
   Installs products to machines independently. This method groups the installations by machines rather than by products, which allows for faster machines to complete the installation more quickly. However, this method uses more network bandwidth and is not recommended for installations over wide area networks (WANs).

   Installations to the Tivoli server occur first. If any installation to the Tivoli server fails, installations to other machines are not started.

   If the installation of the first product on a managed node fails, the installation of the remaining products is not attempted.

   Install to Machines Maximizing Network Bandwidth
   Installs products using the multiplexed distribution technology to optimize network use. This method groups installations by products, so

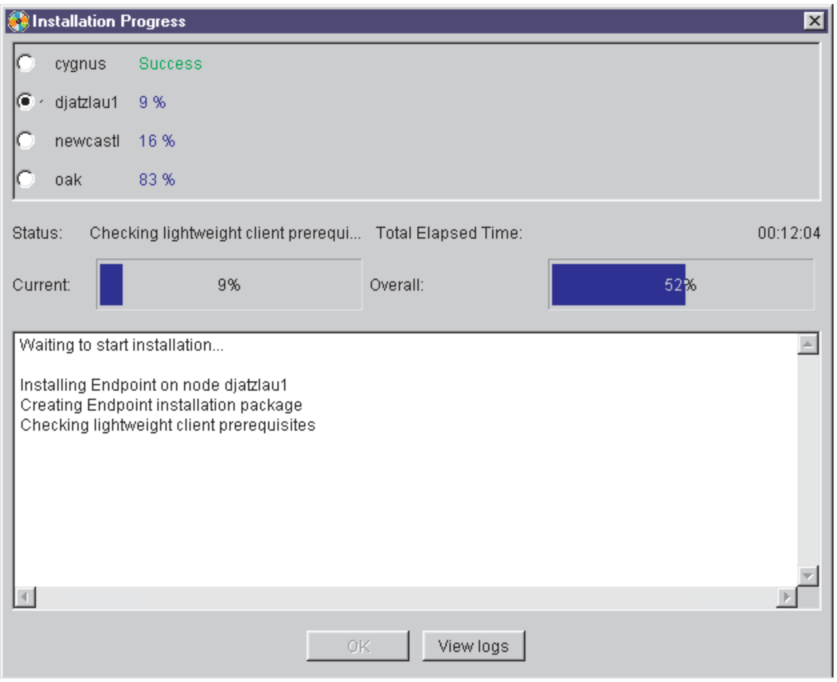
the speed of an installation is limited by the slowest machine. This method is recommended for mass installations. Refer to the *Tivoli Management Framework Planning for Deployment Guide* for more information about MDist.

If the installation of a product on the Tivoli server fails, installations to other machines continue.

If the installation of the first product on a machine fails, the installation of the remaining products on this machine is not attempted.

You can also specify this algorithm as **Install to Machine Maximizing Network Bandwidth**.

3. Click **OK** to start the installation or prerequisite check. The Installation Progress window is displayed.



```
Waiting to start installation...

Installing Endpoint on node djatzlau1
Creating Endpoint installation package
Checking lightweight client prerequisites
```

You can select the button to the left of the status bar to list the progress for each node. The status bars are color-coded to indicate the installation progress of each product.

Blue Indicates that the product is being installed.

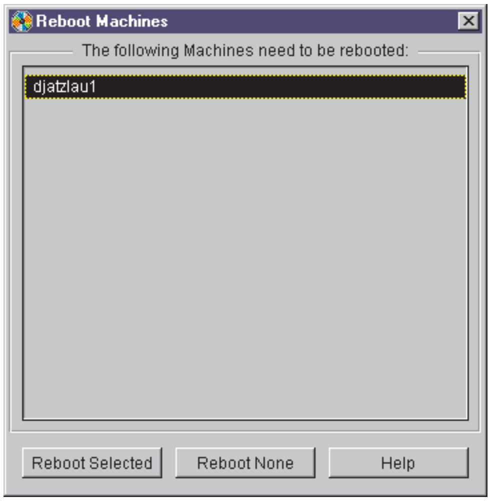
Green Indicates that the product was successfully installed.

Red Indicates that the installation failed. Click **View logs** to view HTML-based error messages. Refer to Chapter 14, “Using Tivoli Software Installation Service log files,” on page 171 for information about using logs.

An `Install Complete` message is displayed when the installation process is finished.

Note: If a particular product installation was not successful, an **X** remains in that cell of the installation worksheet. If you perform another installation, Tivoli Software Installation Service retries the failed installation unless you deselect the cells of the products and machines that were not successfully installed.

If Tivoli Authentication Package is installed or upgraded on a Windows operating system whose **Reboot** option is set to **No**, the Reboot Machines window is displayed.



4. Select the machines that you want to reboot and click **Reboot Selected**.

   The machines in this list must be rebooted before Tivoli Authentication Package can be used. If you deselect machines, you must manually reboot those machines.

   Note: A window is displayed on each Windows machine that is being rebooted to warn the user that the system will be rebooted in 30 seconds. If the user logs off during that time, Windows ends the reboot task and the machine does not reboot.

Using response files to install products

For information about using response files to install Tivoli software, refer to "Installing products from a response file" on page 159.

Chapter 13. Using response files

Tivoli Software Installation Service provides response files to streamline the installation of many products on many machines. Response files are text files that contain the product and machine information Tivoli Software Installation Service needs to perform an installation. You can think of a response file as the text file equivalent to the installation worksheet.

A response file is useful for performing unattended installations. You can use a response file to customize settings for the entire installation of multiple products on multiple machines. You can reuse this file for future installations, using a text editor to add machines or to customize product installation options.

You can automate installation tasks by using the **wsis** command and response files and by using the Tivoli scheduler service to perform these tasks at scheduled times.

You can start the installation by using a response file from either the Tivoli Software Installation Service console or the command line. The installation process reads the installation information from this file instead of prompting you for keyboard input.

This chapter contains the following sections:

- "Installing products from response files"
- "Syntax of a response file"
- "Creating a response file template" on page 157
- "Checking the syntax of a response file" on page 158
- "Installing products from a response file" on page 159
- "Sample response files" on page 161

In addition, this chapter contains a scenario-based example of how you create and modify a response file template, including the specific steps that you can do to modify it for a specific installation. This information is contained in "Scenario: Modifying an exported response file" on page 162.

Installing products from response files

To use a response file to install Tivoli Enterprise software. follow these basic steps:

1. Create a response file template.
2. Check the syntax of the response file.
3. Install from the response file.

For information about the structure and content of a response file, refer to "Syntax of a response file." Sample response files can be found in "Sample response files" on page 161.

Syntax of a response file

This section describes the syntax of a response file. A response file is a text file consisting of the following:

Comments

Begin with a pound sign (#) and *must* be the first character on the line. Anything on this line after the # character is ignored.

Section Headers

Enclosed in square brackets. A section header is the text between the first left square bracket ([) and the last right square bracket (]).

You can arrange the sections in any order. For example, you can group [machine] sections at the end of a response file. Alternatively, you can place each [machine] section near the [alias] sections that describe the products installed on that machine.

Entries

Items contained within a section. Within a section, Tivoli Software Installation Service looks for entries until it reaches the next section header or the end of the file. Blank lines are ignored.

Entries in response files use the following format *keyword=value* where *value* is optional.

A response file can contain the following sections:
- [globals]
- [machine]
- [alias]
- [byNode]
- [byProduct]

[globals] section

The [globals] section enables you to define global installation attributes.

Syntax

The [globals] section has the following syntax:

[globals
InstallPassword=[*password*]
InstallAlgorithm=[[**prereq** | **Run Prerequisite & Probe Only**] |
  [**independent** | **Install to Machines Independently**] |
  [**mdist** | **Install to Machines Maximizing Bandwidth**]]

Descriptions of these entries are as follows:

InstallPassword=[*password*]

Specifies the installation password used when you installed the Tivoli server, if a password was specified. Omit *password* if one was not specified during the installation of the Tivoli server.

If you omit the **InstallPassword=** entry, you are prompted for the installation password when you start the installation. The password is not displayed as you type it.

Note: The password is not encrypted in the response file. Ensure that directory and file permissions do not allow unauthorized users to access this information.

InstallAlgorithm=

Specifies the installation mechanism to be used when the response file is run. Valid values are as follows:

prereq | Run Prerequisites & Probe Only
Without installing any products, runs the Tivoli- and user-defined prerequisites. The probe verifies that adequate disk space exists and that the directories are not read-only. This method is useful in identifying potential errors before installing products.

independent | Install to Machines Independently
Installs products to machines independently. This method groups the installations by machines rather than by products, which allows for faster machines to complete the installation more quickly. However, this method uses more network bandwidth and is not recommended for installations over wide area networks (WANs).

Installations to the Tivoli server occur first. If any installation to the Tivoli server fails, installations to other machines are not started.

If the installation of the first product on a managed node fails, the installation of the remaining products is not attempted.

mdist | Install to Machine Maximizing Network Bandwidth
Installs products using the multiplexed distribution technology to optimize network use. This method groups installations by products, so the speed of an installation is limited by the slowest machine. This method is recommended for mass installations. Refer to the *Tivoli Management Framework Maintenance and Troubleshooting Guide* for more information about multiplexed distribution and its services.

If the installation of a product on the Tivoli server fails, installations to other machines continue.

If the installation of the first product on a machine fails, the installation of the remaining products on this machine is not attempted.

After generating a response file, check that the value is appropriate and edit it if it is not. The default is the installation method that was used most recently used by the Tivoli Software Installation Service client. If the Tivoli Software Installation Service client has not performed an installation, **wsis** exports the method most recently used in this depot session. When the depot starts, the default is **mdist**.

Example

The following example shows a [globals] section that runs prerequisites and probes for disk space and other installation requirements, but does not install anything. The Tivoli region has no installation password:

```
[globals]
InstallPassword=
InstallAlgorithm=Run Prerequisites & Probe Only
```

The following example is the same as the previous one, but uses the shorter name for the installation algorithm:

```
[globals]
InstallPassword=
InstallAlgorithm=prereq
```

[machine] section

The [machine] section enables you to import machine information into the depot. Importing a response file containing only the [machine] section provides a convenient way to add a large number of machines to Tivoli Software Installation Service. You must have a [machine] section for each machine that you define.

Syntax

The [machine] section has the following syntax:

[machine *machine_name***]**
access=[rexec | rsh | ftp | tivoli | lcf]
userid=*userid*
password=[*password*]
promptForPassword=[yes | no]
autoInstallTrip=[yes | no]
interp=*interpreter_type*

Descriptions of these entries are as follows:

machine_name
Specifies the name of the machine.

access=[rexec | rsh | ftp | tivoli | lcf]
Specifies the access method that Tivoli Software Installation Service will use to connect to the target machine. This information is used to verify connectivity when the response file is read. Valid values are as follows:

rexec For a UNIX and Windows system. Requires a **userid** entry and either a **password** or **promptForPassword** entry.

rsh For a UNIX system only. The /.rhosts file for the root account on the machine to be accessed must contain an entry specifying the user ID root on the machine running the depot. The **userid** and **password** entries are not required when you choose this access type.

ftp For an OS/400 system only. Requires a **userid** entry and either a **password** or **promptForPassword** entry.

tivoli Specifies that the machine is accessed through a Tivoli connection and is already recognized. The **userid** and **password** entries are not required.

lcf Specifies that the machine is already an endpoint. The **userid** and **password** entries are not required.

Use this access method only when the product documentation instructs you to install it directly to an endpoint or if the product index (.IND) file contains the **lcf_allow** tag.

userid=*userid*
Specifies the user name needed to contact the specified machine. This is required only for the **rexec** and **ftp** access methods.

On a UNIX machine, the login shell of this user must be Bourne, Korn, or bash. You cannot install Tivoli software using an account whose login shell is C or any of its variants.

password=[*password*]
Specifies the password of the specified user ID on the target machine. This entry is required only if you specify the **userid** entry and do not specify `promptForPassword=yes`.

If the specified user ID has a null password, specify **password=** without an option.

Note: The password is not encrypted in the response file. Ensure that directory and file permissions do not allow unauthorized users to access this information.

promptForPassword=[**yes** | **no**]
This optional entry specifies whether the user is prompted for a password when installing from a response file. This entry overrides the **password** entry. Valid values are as follows:

yes The user is prompted for a password. Use this value when you do not want to place unencrypted passwords in response files. The password does not displayed as you type it.

no The user is not prompted for a password.

autoInstallTrip=[**yes** | **no**]
This optional entry specifies whether Tivoli Remote Execution Service is installed on the specified Windows machine if it is required. Valid values are as follows:

yes Tivoli Remote Execution Service is installed. If you specify **yes** and the service is not already installed on the machine, it is installed when this section of the response file is processed. This happens when you import a response file into the Tivoli Software Installation Service console, when you check the syntax of a response file with the **–c** option of the **wsis** command, or when you install with the **–i** option of the **wsis** command.

no Tivoli Remote Execution Service is not installed. If you specify **no** and Tivoli Remote Execution Service is not already installed on the machine, the installation fails. Messages in the log explain the problem.

The default is **yes**.

For more information about Tivoli Remote Execution Service, refer to "Using Tivoli Remote Execution Service" on page 341.

interp=*interpreter_type*
This optional entry specifies the interpreter type of the machine. If you include this entry with a valid *interpreter_type* value, the value is not checked against the machine. If not specified, the value is determined internally.

Examples

The following [machine] section defines the Windows machine iandu-4, which will be accessed through rexec using the user ID Administrator and the password MyPassword. Tivoli Remote Execution Service will be installed if required.

```
[machine iandu-4]
access=rexec
userid=Administrator
password=MyPassword
autoInstallTrip=yes
interp=w32-ix86
```

The following [machine] section defines the Solaris machine sevenup, which already has a Tivoli connection. No user ID and password are required to access the machine.

```
[machine sevenup]
access=tivoli
```

The following [machine] section defines the UNIX machine oak. An rsh connection will be used.

```
[machine oak]
access=rsh
```

[alias] section

The [alias] section enables you to define shorter names for Tivoli products and their corresponding installation options.

Note: If a Tivoli product has different installation options for different machine types, you must define a separate [alias] section for each unique set of options.

Syntax—general

The [alias] section has the following general syntax:

[**alias** *aliasname productname*]
install_option=*value*
Overwrite=[*install_option*]...

Descriptions of these entries are as follows:

aliasname

Specifies the alias to be associated with the Tivoli product and its installation options. There must be a separate [alias] section for each unique set of installation options for a product. However, you can use a single [alias] section for multiple machines if they all use the same values for the installation options.

When a response file is exported, there is a separate [alias] section for the installation of each product on each machine. If the same options are used to install a product on more than one machine, you can create a single [alias] section.

When a response file is exported, the aliases are named *aliasname1* through *aliasnameN*. You can change the names to something more meaningful. For example, you might change aliasname2 to ManagedNode-NT for an alias used to create managed nodes on Windows NT machines by installing Tivoli Management Framework. You might change aliasname3 to Inventory362-Upgrade for an alias for installing the Tivoli Inventory product upgrade. If you do change the alias names in a response file exported, be sure to make the corresponding changes in the [byNode] and [byProduct] sections.

productname

Identifies the Tivoli product to be installed, using the unique product ID used in the install repository. To determine the unique product ID for a product, use the **–l** option of the **wimport** command.

install_option=value

Specifies the installation options specific to this installation. You can have multiple *install_option=value* specifications in the response file.

Each *install_option* entry corresponds to a field in the Install Options window that is displayed when you install the product using the Tivoli desktop or to an installation variable you specify when you install the product from the command line. For details about the installation options for a product, refer to the product documentation. This is particularly important if you create the response file manually instead of exporting a response file from the Tivoli Software Installation Service console.

When SIS exports installation options, some of them are enclosed in at signs (@). These symbols are not required. You can keep them or delete them.

Overwrite=[*install_option*]...

Specifies a space-separated list of installation options that are to be overwritten. Use this option when reinstalling a product to force existing files to be overwritten. Each installation option specified in this list must be listed in an *install_option=value* entry in this [alias] section or must specify one of the standard installation directories. For example, the following entry specifies that the BIN, LIB, and MAN directories are to be reinstalled:

```
Overwrite=BIN LIB MAN
```

Syntax—product-specific

In addition to the *install_option=value* entries that correspond to installation variables, the [alias] section also contains product-specific entries for installing managed nodes, gateways, and endpoints.

Endpoint installation options: The following entries specify the information needed to install endpoints. Some of these are specific to Tivoli Software Installation Service. The remaining ones correspond to options used on the **winstlcf** or **lcfd** commands. The description of each entry lists the corresponding command and option, if applicable. Refer to the *Tivoli Management Framework Reference Manual* for more information about these commands and the values for their options.

The following entries specify information required to install endpoints:

CheckLogin=[**On** | **Off**]
EndpointLabel=[*label*]
EndpointPort=[*port_number*]
EndpointStartupOpts=[*options*]
EndpointStartupTimeout=[*seconds*]
[**GatewayName=**[**Broadcast to Gateways** | *gateway_label*] |
  GatewayIP=*gateway_IP*]
GatewayPort=[*port_number*]
PolicyRegionName=[**None** | *policy_region*]
Reboot=[**yes** | **no**]

tapUser=[*userid*]
TapPassword=[*password*]
TMABIN=[*directory*]

Descriptions of the endpoint-specific entries are as follows:

CheckLogin=[**On** | **Off**]
Specifies whether Tivoli Software Installation Service checks to ensure that the endpoint successfully logs in to its gateway after installation. This entry corresponds to the **–a** option of the **winstlcf** command and to the **Check for TMA login to Gateway** installation option in the Tivoli Software Installation Service console. Valid values are as follows:

On Specifies that Tivoli Software Installation Service should verify that the endpoint logs in to its gateway. If the endpoint does not log in within the time limit specified by **EndpointStartupTimeout**, a connection failure is reported. This is the default.

Off Specifies that Tivoli Software Installation Service should not check whether the endpoint logs in to its gateway.

EndpointLabel=[*label*]
Specifies the label of the endpoint. This entry corresponds to the **–n** option of the **winstlcf** command and to the **Endpoint Label** installation option in the Tivoli Software Installation Service console. The default value is @HostName@.

EndpointPort=[*port_number*]
Specifies the port on which the endpoint listens. This entry corresponds to the **–l** option of the **winstlcf** command and to the **Optional port for endpoint** installation option in the Tivoli Software Installation Service console. The default value is 9495.

EndpointStartupOpts=[*options*]
Specifies optional configuration options to use when starting the endpoint. This entry corresponds to the **–L** option of the **winstlcf** command and to the **Additional Options for endpoint** installation option in the Tivoli Software Installation Service console.

For example, you might specify the following **EndpointStartupOpts=** entry to specify the gateway to which the endpoint sends its login packet and to prevent the endpoint from broadcasting at login:

```
@EndpointStartupOpts@=-D lcs.login_interfaces=cygnus -D bcast_disable=1
```

EndpointStartupTimeout=[*seconds*]
Specifies the number of seconds that Tivoli Software Installation Service waits for the endpoint to log in to its gateway. The default is 300 seconds. This entry is specific to installation using Tivoli Software Installation Service and corresponds to the **Seconds to wait for endpoint login** installation option in the Tivoli Software Installation Service console.

GatewayName=[**Broadcast to Gateways** | *gateway_label*]
Specifies the label of the gateway that the endpoint logs in to. This entry is specific to installation using Tivoli Software Installation Service and corresponds to the **Gateway Name** installation option in the Tivoli Software Installation Service console. You can specify only one of **GatewayIP=** or **GatewayName=**. Valid values are as follows:

Broadcast to Gateways
The endpoint broadcasts for a gateway. This is the default when

installing an endpoint using other methods but is the default in SIS only if no gateways exist in the Tivoli region.

If you specify **Broadcast to Gateways**, Tivoli Software Installation Service does not check that the endpoint successfully logs in to the gateway.

gateway_label

Specifies the name of a gateway. The default value is the first gateway in an alphabetic list of gateway labels.

If you specify a gateway label, Tivoli Software Installation Service checks that the endpoint is logged in to the gateway.

GatewayIP=*gateway_IP*

Specifies the IP address of the gateway that the endpoint logs in to. This entry corresponds to the **–g** option of the **winstlcf** command, but has no counterpart in the installation options in the Tivoli Software Installation Service console. You can specify only one of **GatewayIP=** or **GatewayName=**. If you specify **GatewayIP=**, you must also specify **GatewayPort=**.

Note: This entry is valid in a response file but not in a product defaults file, which uses the same *install_option=value* syntax.

GatewayPort=[*port_number*]

Specifies the TCP port number on which the gateway specified by **GatewayIP=** listens. This entry corresponds to specifying the option *+port* with the **–g** option of the **winstlcf** command, but has no counterpart in the installation options in the Tivoli Software Installation Service console. The default value is 9494.

Note: This entry is valid in a response file but not in a product defaults file, which uses the same *install_option=value* syntax.

PolicyRegionName=[None | *policy_region*]

Specifies the policy region for the endpoint. This entry corresponds to the **–r** option of the **winstlcf** command and to the **Add Endpoint Icon to Policy Region** installation option in the Tivoli Software Installation Service console. Valid values are as follows:

None The endpoint is not included in a policy region. This is the default. Endpoints are not typically assigned to policy regions.

policy_region

The name of the policy region for the endpoint.

Reboot=[yes | **no**]

Specifies whether to reboot the machine, if necessary. This entry is used only when installing an endpoint on a Windows machine. This entry corresponds to the **–R** option of the **winstlcf** command. Valid values are as follows:

yes Specifies that Tivoli Software Installation Service will reboot the machine.

no Specifies that Tivoli Software Installation Service will not reboot the machine. This is the default.

After all installations are complete, Tivoli Software Installation Service displays a summary of the machines that were rebooted (that is, Windows NT or Windows 2000 machines with **Reboot** specified as **yes**). Tivoli

Software Installation Service also issues a prompt that lists the machines that might need to be rebooted manually (that is, Windows NT or Windows 2000 machines with **Reboot** specified as **no**) and asks if you want to reboot them. If you do not respond to the prompt within the client prompt timeout period, the machines are not rebooted. If multiple products are selected for a machine, Tivoli Software Installation Service does not reboot the machine until all products are installed.

tapUser=[*userid*]
Specifies the user name for the Tivoli remote access account. This entry is used only when creating a endpoint on a Windows machine that does not have Tivoli Authentication Package already installed. This entry corresponds to the **–T** option of the **winstlcf** command and to the **TRAA Login** installation option in the Tivoli Software Installation Service console.

TapPassword=[*password*]
Specifies the password for the Tivoli remote access account. This entry is used only when creating an endpoint on a Windows machine that does not have Tivoli Authentication Package already installed. This entry corresponds to the **TRAA Password** installation option in the Tivoli Software Installation Service console.

TMABIN=*directory*
Specifies the directory in which the endpoint software is installed. This entry corresponds to the **–d** option of the **winstlcf** command and to the **Endpoint Binaries** installation option in the Tivoli Software Installation Service console.

Examples: If you export a response file from the installation worksheet containing a cell for the endpoint product on Windows NT, the [alias] section of the generated file is similar to the following:

```
[alias aliasname2 TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=bbrenner
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=
@PolicyRegionName@=
@CheckLogin@=On
@EndpointStartupTimeout@=300
tapUser=
TapPassword=
Reboot=No
Overwrite=
```

Remember that you can change the alias name, `aliasname2`, to a name that is more meaningful to you, such as `NT-endpoint`. You can also change the hard coded host name in the label, `bbrenner`, to the more general `@HostName@`, to make this alias install an endpoint on any Windows NT machine, and add a suffix (in this case, `-endpoint`) to make the label unique if a managed node is also installed on this machine. The resulting [alias] section looks similar to the following:

```
[alias NT-endpoint TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=@HostName@-endpoint
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=
@PolicyRegionName@=
@CheckLogin@=On
@EndpointStartupTimeout@=300
```

```
tapUser=
TapPassword=
Reboot=No
Overwrite=
```

Gateway installation options: When creating a gateway, the following installation options are available:

GATE_NAME=*gateway_label*
GATE_PORT=*port_number*
IPX=[On | Off]
GATE_PORT_IPX=*port_number*

Descriptions of the gateway-specific entries are as follows:

GATE_NAME=*gateway_label*
Specifies the label of the gateway. The default is `@HostName@-gateway`.

GATE_PORT=*port_number*
Specifies the TCP port number on which the gateway listens. The default is 9494. Tivoli recommends that all gateways in a Tivoli region use the same port number.

IPX=[On | Off
Specifies whether the gateway supports the IPX protocol. This entry is supported only for NetWare and is not displayed or exported for other operating systems.

On Specifies that the gateway can communicate with endpoints using the IPX/SPX protocol.

Off Specifies that the gateway does not communicate with endpoints using the IPX/SPX protocol. This is the default.

GATE_PORT_IPX=*port_number*
Specifies the socket number on which the gateway listens. This entry is supported only for NetWare and is not displayed or exported for other operating systems. The number specified is the IPX socket number (the default is 9494). The SPX socket number is *port_number*-1 (the default is 9493).

Tivoli recommends that all gateways in a Tivoli region use the same port number. The TCP/IP port number can be the same as the IPX port number.

Examples: If you export a response file to create a gateway on any interpreter type, the [alias] section of the generated file is similar to the following:

```
[alias aliasname7 GATEWAY-Tivoli_Gateway]
@GATE_NAME@=oak-gateway
@GATE_PORT@=9494
Overwrite=
```

The following [alias] section was changed so that you can use it to create a gateway on any managed node:

```
[alias gateway GATEWAY-Tivoli_Gateway]
@GATE_NAME@=@HostName@-gateway
@GATE_PORT@=9494
Overwrite=
```

The following [alias] section creates a NetWare gateway that accepts logins from endpoints using TCP/IP and IPX protocols. The gateway listens for both types of endpoints on port 9494. This [alias] section can be used only with a NetWare machine.

```
[alias NetWare-gateway GATEWAY-Tivoli_Gateway]
@GATE_NAME@=@HostName@-gateway
@GATE_PORT@=9494
IPX=On
GATE_PORT_IPX=9494
Overwrite=
```

Managed node installation options: For Tivoli Management Framework, which is installed to create managed nodes, the following installation options are available in addition to the *install_option=value* entries defined in the product index (.IND) file:

PR_NAME=[*policy_region*]
Reboot=[yes | no]
tapUser=[*userid*]
TapPassword=[*password*]

Descriptions of the manage node-specific entries are as follows:

PR_NAME=[*policy_region*]
Specifies the policy region in which to create the managed node.

Reboot=[yes | no]
Specifies whether to reboot the machine if necessary. This entry is used only when installing a managed node on a Windows machine. Valid values are as follows:

yes Specifies that Tivoli Software Installation Service will reboot the machine.

no Specifies that Tivoli Software Installation Service will not reboot the machine. This is the default.

After all installations are complete, Tivoli Software Installation Service displays a summary of the machines that were rebooted (that is, Windows NT or Windows 2000 machines with **Reboot** specified as **yes**). Tivoli Software Installation Service also issues a prompt that lists the machines that might need to be rebooted manually (that is, Windows NT or Windows 2000 machines with **Reboot** specified as **no**) and asks if you want to reboot them. If you do not respond to the prompt within the client prompt timeout period, the machines are not rebooted. If multiple products are selected for a machine, Tivoli Software Installation Service does not reboot the machine until all products are installed.

tapUser=[*userid*]
Specifies the user name for the Tivoli remote access account. This entry is used only when creating a managed node on a Windows machine that does not have Tivoli Authentication Package already installed. This entry corresponds to the **–T** option of the **wclient** command and to the **TRAA Login** installation option in the Tivoli Software Installation Service console.

TapPassword=[*password*]
Specifies the password for the Tivoli remote access account. This entry is used only when creating a managed node on a Windows machine that

does not have Tivoli Authentication Package already installed. This entry corresponds to the **TRAA Password** installation option in the Tivoli Software Installation Service console.

Example: If you export a response file from the installation worksheet containing a cell for Tivoli Management Framework, Version 3.7, on Windows NT, the [alias] section of the generated file is similar to the following:

```
[alias aliasname1 TMF-client-3.7]
CAT=c:/usr/local/Tivoli/msg_cat
LIB=c:/usr/local/Tivoli/lib
BIN=c:/usr/local/Tivoli/bin
DB=c:/data/Tivoli/db
MAN=c:/usr/local/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
tapUser=
TapPassword=
Reboot=No
Overwrite=
```

Remember that you can change the alias name, `aliasname1`, to a name that is more meaningful to you, such as `NT_managed_node-in-cygnus-region`.

[byNode] and [byProduct] sections

After you define aliases, you can create one of two special sections: [byNode] and [byProduct]. Use these sections to specify which products are to be installed on which machines. In terms of the installation worksheet, the [byNode] and [byProduct] sections determine which cells are selected.

Typically, you use only one of these sections, depending on whether you prefer to specify a list of products to install on each machine or a list of machines on which to install a product. However, response files exported from the Tivoli Software Installation Service console contain both sections so you can choose the one you prefer. You can delete the one you do not need.

The section you choose depends on how you model a specific installation. If you start with a list of nodes and specify which products to install on each, use the [byNode] section. If you start with a list of products and specify the nodes on which to install each, use the [byProduct] section. You can use both sections in a response file, if you have some installations you want to handle by node and others that you want to handle by product.

If you are using a response file to define new machines, you do not need either the [byNode] or [byProduct] section.

Syntax—[byNode]

The [byNode] section has the following syntax:

[byNode]
nodename=aliasname1...aliasnameN

A description of these entries follows:

nodename
The name of the machine on which products are to be installed. This name

is defined by the [machine] section for managed nodes and endpoints or by the [pcNode] section for PC managed nodes.

aliasname1...aliasnameN
: The short name or alias of one or more Tivoli products to be installed on *nodename*. Each alias is defined by an [alias] section in the response file.

Syntax—[byProduct]

The [byProduct] section has the following syntax:

[byProduct]
aliasname=nodename1...nodenameN

A description of these entries follows:

aliasname
: The short name or alias of the Tivoli product to be installed. The alias is defined by an [alias] section in the response file.

nodename1...nodenameN
: The name of one or more machines on which the product is to be installed. This machine name is defined by the [machine] section for managed nodes and endpoints.

Combining the [byNode] and [byProduct] sections

If a single machine is associated with two [alias] sections that define different installation options for the same product, the association defined by the last [byNode] or [byProduct] section in the response files takes precedence as in the following example:

```
[alias SomeProduct_on_UNIX SomeProduct]
BIN=/usr/local/bin
[alias SomeProduct_on_AIX SomeProduct]
BIN=/usr/share/bin

[byProduct]
SomeProduct_on_UNIX=machine1

[byNode]
machine1=SomeProduct_on_AIX
```

The two [alias] sections reference the same product, but have different attribute settings. Because the [byNode] section is last, `machine1` uses the installation options in `SomeProduct_on_AIX`, so the BIN attribute is set to /usr/share/bin.

Example

Suppose that you have two [alias] sections that define aliases `EP` and `MN`. Either of the following sections specifies that the product and installation options specified by the [alias] section for `EP` should be installed on machines iandu-4, kiwi, and pctmp83, and the product and installation options specified by alias `MN` should be installed on machines kiwi and pctmp83.

You can start with a list of nodes and specify which products to install on each using [byNode]:

```
[byNode]
iandu-4=EP
kiwi=EP MN
pctmp83=EP MN
```

Alternatively, you can start with a list of products and specify a list of nodes on which to install each using [byProduct]:

```
[byProduct]
EP=iandu-4 kiwi pctmp83
MN=kiwi pctmp83
```

Creating a response file template

The Tivoli Software Installation Service console and the **wsis** command are both quick and convenient ways to create a response file template. Although you can create a response file from scratch using any text editor, but by using a template created by Tivoli Software Installation Service ensures that you start with the correct sections, use the correct product names, and specify the correct installation options for products. The following sections provide instructions.

The value of the **InstallAlgorithm=** entry in the [globals] section of a response file depends on whether you export it using the console or the **wsis** command, as follows:

- When you export from the console, you select the installation method you want to use.
- When you export using the **wsis** command, check that the value is appropriate and edit the response file if it is not. The default is the installation method that was used most recently by the Tivoli Software Installation Service client. If it has not performed an installation, **wsis** exports the method most recently used in this depot session. When the depot starts, the default is **mdist**.

Creating a response file template using the Tivoli Software Installation Service console

The following procedure creates (exports) a response file using the Tivoli Software Installation Service console:

1. Start the Tivoli Software Installation Service console as described in “Starting the Tivoli Software Installation Service console” on page 113.
2. From the installation worksheet, select **Worksheet → Select products** to display the Select Products window. Use this window to select the products required for the installation you want to perform. Refer to “Adding products to the installation worksheet” on page 127 for more information.
3. From the installation worksheet, select **Worksheet → Select machines** to display the Select Machines window. Use this window to select representative machines for the installation you want to perform. Refer to “Adding machines to the installation worksheet” on page 134 for more information.

   If the machines to be installed are not listed in the Select Machines window, select another machine of the same interpreter type. If you are installing the same product to more than one machine of the same interpreter type, you can select only one of those machines.
4. In the installation worksheet, specify which products to install on which machines. If appropriate, specify installation options for each installation. For more information, refer to “Specifying products to install” on page 138 and “Overriding default installation options for one machine” on page 137.
5. From the installation worksheet, select **File → Export response file** to display the Select Installation Mechanism window.
6. Select the installation method. For a description of this window, refer to the 144 entry in “[globals] section” on page 144.

Note: Exporting a response file does not perform an installation; only the response file is created.

7. If you want the response file to include Tivoli installation password and the passwords to use when connecting to new machines, select **Include passwords in response file**.

   The passwords are displayed as clear text in the response file. Ensure that directory and file permissions do not allow unauthorized users to access this information.

   If you do not want the passwords included in the response file, the exported response file contains the statements necessary to prompt for passwords when you start the installation or import the response file into the installation worksheet.
8. Click **OK** to display the Save response file window.
9. Type a name and path for the response (.RSP) file. Tivoli Software Installation Service generates the response file and saves it in the file that you specified

At this point the response file template is created. Use any text editor to modify it.

Creating a response file template from the command line

The following procedure creates a response file using the **wsis** command:

1. Use the **wimport** command to add the necessary products to the install repository. Refer to "Importing products into the install repository" on page 115.
2. If desired, use the **wsisdefaults** command to set default values for the installation options of the products. This reduces the amount of change required in the [alias] sections of the response files. Refer to "wsisdefaults" on page 370 for more information.
3. Use the **–l** option of the **wimport** command to display the name or number of the required products in the install repository. Refer to "wimport" on page 363 for more information.
4. Use the **–x** option of the **wsis** command to export a response file for those products.

   For example, the following command creates a response file named my.rsp that installs the Tivoli Management Framework and an endpoint on the machine cygnus and an endpoint on the machine sevenup. If a machine is not already a managed resource in the Tivoli region and has not been added to Tivoli Software Installation Service, specify the interpreter type, for example, solaris2 or aix4-r1. Assume that output of the **wimport** command shows that the Tivoli Management Framework is named TMF-client-3.7 and the endpoint product is named TMA-Tivoli_Endpoint.

   ```
   wsis -x my.rsp \
    -p TMF-client-3.7 cygnus solaris2 \
    -p TMA-Tivoli_Endpoint cygnus solaris2 \
    -p TMA-Tivoli_Endpoint oak aix4-r1
   ```

Use any text editor to modify this response file template.

Checking the syntax of a response file

After you create or modify a response file, you need to check the syntax. You can do this from either the Tivoli Software Installation Service console or from the command line.

Notes:

1. Before using a response file, ensure that all products you want to install are already in the install repository. For details, refer to "Importing products into the install repository" on page 115.
2. Checking the syntax of the response file that includes Windows machines requires that Tivoli Remote Execution Service be installed on each Windows machine. If the [machine] section contains the entry `autoInstallTrip=no`, a prompt asks whether you want to install Tivoli Remote Execution Service. If the entry is omitted or specifies `autoInstallTrip=yes`, Tivoli Remote Execution Service is installed without a prompt.

Checking the syntax using the Tivoli Software Installation Service console

To check the syntax of a response file using the console, perform the following steps:

1. From an empty installation worksheet, select **File → Import response file** to display the Load response file window.
2. Select the response file that you want to verify, and click **OK** to import the response file into the installation worksheet.

Tivoli Software Installation Service reads the response file, contacts machines specified in the file, sets installation options, and adds this information to the depot. The installation worksheet is updated with the products and machines defined. If the response file contains syntax errors, a window is displayed that describes the errors. As much of the response file as possible is processed and placed in the installation worksheet.

You can select **Install** to begin product installations or select **Worksheet → Clear** to clear the installation worksheet.

Checking the syntax from the command line

To check the syntax of a response file, use the following command:

```
wsis -c response_file
```

where *response_file* is the name of the response file to verify.

Tivoli Software Installation Service validates the syntax of the response file and makes a test connection to the machines. It starts the installation based on the machine and installation options in the response file. Status is sent to standard output and standard error and HTML log files are created.

For additional information about the **wsis** command, refer to "wsis" on page 367.

Installing products from a response file

You can use a response file to install Tivoli products using either the console or the command line, as described in the following sections.

Note: Before using a response file, ensure that all products you want to install are in the depot. For details, refer to "Importing products into the install repository" on page 115.

Installing products from a response file using the Tivoli Software Installation Service console

To install from a response file from the console, perform the following steps:

1. From an empty installation worksheet, select **File → Import response file** to display the Load response file window.
2. Select the response file that you want to verify. Click **OK** to import the response file into the installation worksheet.

   Tivoli Software Installation Service reads the response file, contacts machines specified in the file, sets installation options, and adds this information to the depot. The installation worksheet is updated with the products and machines defined. If the response file contains syntax errors, a window is displayed that describes the errors. As much of the response file as possible is processed and placed in the installation worksheet.
3. To perform the installation, follow the procedures in "Using the Tivoli Software Installation Service console to install products" on page 139.

Installing products from a response file from the command line

To use the command line to perform an installation or add machines using with a response file, perform the following steps:

1. Use the **wsis** command as follows:

   ```
   wsis -i response_file
   ```

   where *response_file* is the name of the response file defining the installation.

   Tivoli Software Installation Service reads the response file, makes a test connection to the machines specified in the file, sets installation options, and adds this information to the depot. It starts the installation based on the machine information and product installation options in the response file. Status from the installation is sent to standard output and standard error and HTML log files are created.

   For additional information about the **wsis** command, refer to "wsis" on page 367.
2. After all installations are complete, Tivoli Software Installation Service displays a summary of the machines that were rebooted (that is, `Reboot=yes` machines on which Tivoli Authentication Package was installed). Tivoli Software Installation Service also issues a prompt that lists the machines that must be rebooted manually (that is, `Reboot=no` machines on which Tivoli Authentication Package was installed) and asks if you want to reboot them.

   To reboot any or all of the machines, type the names of the machines you want Tivoli Software Installation Service to reboot, separated by spaces, and press Enter. To reboot all the listed machines, type `all` and press Enter. If you do not respond to the prompt within the client prompt time out period, the machines are not rebooted.

   The machines in this list must be rebooted before the Tivoli Authentication Package can be used. If you do not allow Tivoli Software Installation Service to reboot them, you must manually reboot them.

   Note: A window is opened on each Windows machine that is being rebooted to warn a user that the system will be rebooted in 30 seconds. If the user logs off during that time, Windows ends the reboot task and the machine is not rebooted.

Sample response files

This section contains sample response files that perform the following tasks:

- Adding machines
- Installing and upgrading managed nodes

Sample 1: Adding machines

If you have many machines to add to Tivoli Software Installation Service, a response file provides an efficient way to define their connection information. The following response file provides connection information for the UNIX machines newcastl, sevenup, and shawnee, plus the Windows NT machine cribbage. Each of the UNIX machines has the same root user ID and root password, which is included in the file. You are prompted for the password for cribbage. The `interp=` entry is not required in any [machine] statement, but is included for cribbage as a reminder:

```
[machine newcastl]
access=rexec
userid=root
password=Root$$PW

[machine sevenup]
access=rexec
userid=root
password=Root$$PW

[machine shawnee]
access=rexec
userid=root
password=Root$$PW

[machine cribbage]
access=rexec
userid=Administrator
autoInstallTrip=yes
promptForPassword=yes
interp=w32-ix86
```

Sample 2: Installing and upgrading managed nodes

The following response file creates a Version 3.6.2 managed node on the Windows NT machines vwilburn, fhackerm, and bbrenner. To create a Version 3.6.2 managed node, install the Tivoli Management Framework, Version 3.6 and then install the patch that upgrades to Version 3.6.2.

```
[globals]
InstallPassword=
InstallAlgorithm=mdist

[byProduct]
ManagedNode-NT=vwilburn bbrenner fhackerm
FW362upgr=vwilburn bbrenner fhackerm

[alias ManagedNode-NT TMF-client-3.6]
CAT=c:/Tivoli/msg_cat
LIB=c:/Tivoli/lib
BIN=c:/Tivoli/bin
DB=c:/Tivoli/DB
MAN=c:/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
```

```
tapUser=
TapPassword=
Reboot=No
Overwrite=

[alias FW362upgr TMF_3.6.2-Tivoli_Mgt_FW_3.6.2_Maint_Rel]

[machine fhackerm]
access=rexec
userid=Administrator
password=pW4Admin
autoInstallTrip=yes
promptForPassword=no
interp=w32-ix86

[machine vwilburn]
access=rexec
userid=Administrator
password=pW4Admin
autoInstallTrip=yes
promptForPassword=no
interp=w32-ix86

[machine bbrenner]
access=rexec
userid=Administrator
password=pW4Admin
autoInstallTrip=yes
promptForPassword=no
interp=w32-ix86
```

Scenario: Modifying an exported response file

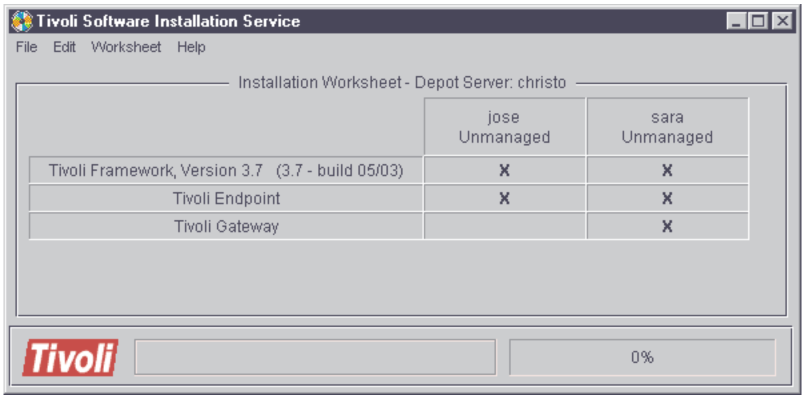
Suppose that you want to create a response file template to use when a new UNIX or Windows machine is added to your Tivoli environment. You plan to use this response file to install a managed node and an endpoint on each new machine, and to install a gateway on each new UNIX machine. Assume that you want to install Tivoli Enterprise products to the same directories on all UNIX platforms.

You need to export a response file that installs a managed node and an endpoint on each UNIX and Windows machine, and installs a gateway on the UNIX machine. Assume that you have a Windows NT machine named jose and a UNIX machine named sara. Neither machine is currently part of your Tivoli region.

Assume that you plan to use this template immediately to install the machines jose and sara, as well as two additional UNIX machines (basu and fadi) and one additional Windows NT machine (molly).

You can create the original template using either the console or the command line. This example uses the console. Perform the following steps to create and modify the response file:

1. Use the Tivoli Software Installation Service console to create the installation worksheet shown in the following figure.



Installation Worksheet - Depot Server: christo

	jose Unmanaged	sara Unmanaged
Tivoli Framework, Version 3.7 (3.7 - build 05/03)	X	X
Tivoli Endpoint	X	X
Tivoli Gateway		X

It is possible to override the default installation options using the installation worksheet before you export the response file. However, the purpose of this scenario is to demonstrate how to edit a response file, so do not change installation options at this time.

Although it appears easier to add several machines of the same type to the installation worksheet before you export a response file template, it generally is not. Use one machine for each unique combination of installation options and products. For example:

- Because installation options and file naming conventions differ between operating systems, include a machine of each operating system type for each product you want in your response file. Depending on the product, different UNIX platforms can have the same or different installation options.

  If the options are the same across platforms, you can combine them into a single [alias] section.
- Because installation options for products that require a RIM host vary with operating system, include a machine of each operating system on which you will install those products.
- Specify machines that are not currently defined in your Tivoli region. Some installation options apply only when the first managed resource is created on that machine. This type of option is not exported for machines that are currently in your Tivoli region.

  For example, if you export the endpoint product for a Windows NT machine that already has a managed node installed, the installation options relating to the Tivoli remote access account are not exported.

2. Export the installation worksheet as a response file. Specify the type of installation mechanism you want to use. In this example, `Install to Machines Maximizing Network Bandwidth` is selected. Include passwords when you export. Because the response file contains unencrypted passwords, be sure to export it to a directory that unauthorized users cannot access.

   The installation worksheet shown previously generates the following response file:

```
[alias aliasname4 TMA-Tivoli_Endpoint]
TMABIN=/opt/Tivoli/lcf
@EndpointLabel@=sara
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=cygnus-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
```

```
Overwrite=

[alias aliasname2 TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=jose
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=cygnus-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
tapUser=
TapPassword=
Reboot=No
Overwrite=

[byProduct]
aliasname1=jose
aliasname2=jose
aliasname3=sara
aliasname4=sara
aliasname5=sara

[alias aliasname3 TMF-client-3.7]
CAT=/usr/local/Tivoli/msg_cat
LIB=/usr/local/Tivoli/lib
BIN=/usr/local/Tivoli/bin
DB=/usr/local/Tivoli/database
MAN=/usr/local/Tivoli/man
APPD=/usr/lib/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-Region
Overwrite=

[alias aliasname1 TMF-client-3.7]
CAT=c:/usr/local/Tivoli/msg_cat
LIB=c:/usr/local/Tivoli/lib
BIN=c:/usr/local/Tivoli/bin
DB=c:/usr/local/Tivoli/database
MAN=c:/usr/local/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-Region
tapUser=
TapPassword=
Reboot=No
Overwrite=

[byNode]
jose=aliasname1 aliasname2
sara=aliasname3 aliasname4 aliasname5

[alias aliasname5 GATEWAY-Tivoli_Gateway]
@GATE_NAME@=sara-gateway
@GATE_PORT@=9494
Overwrite=

[machine sara]
access=rexec
userid=root
password=sara_root_pw
promptForPassword=no
interp=solaris2
```

```
[machine jose]
access=rexec
userid=Administrator
autoInstallTrip=yes
password=NT_password
promptForPassword=no
interp=w32-ix86

[globals]
InstallPassword=
InstallAlgorithm=mdist
```

`InstallPassword` does not have a value because, in this example, the Tivoli region does not have an installation password.

3. Using any text editor, edit the response file and make the appropriate changes. The following steps describe the changes required to use this response file to perform the installation scenario described above.
4. Because the response file in this scenario was exported from the Tivoli Software Installation Service console, `InstallAlgorithm` in the [globals] section is set to the value you specified.

   However, if the response file had been exported using the **wsis** command, `InstallAlgorithm` would specify the last method used by this client. You must edit the file to set `InstallAlgorithm` to the desired installation method.
5. Simplify the response file by removing one of the [byProduct] or [byNode] sections. This example uses the [byNode] section, so remove the [byProduct] section. That is, remove the following lines:

```
[byProduct]
aliasname1=jose
aliasname2=jose
aliasname3=sara
aliasname4=sara
aliasname5=sara
```

6. Rename the [alias] sections so that their names are more meaningful. Make the following changes throughout the response file:
   - Change `aliasname4` to `unix_endpoint`.
   - Change `aliasname2` to `NT_endpoint`.
   - Change `aliasname1` to `NT_managed_node`.
   - Change `aliasname3` to `unix_managed_node`.
   - Change `aliasname5` to `gateway`.

   The response file now looks like the following. Changes are indicated in **bold** text.

```
[alias unix_endpoint TMA-Tivoli_Endpoint]
TMABIN=/opt/Tivoli/lcf
@EndpointLabel@=sara
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=cygnus-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

[alias NT_endpoint TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=jose
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=cygnus-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
```

```
tapUser=
TapPassword=
Reboot=No
Overwrite=

[alias unix_managed_node TMF-client-3.7]
CAT=/usr/local/Tivoli/msg_cat
LIB=/usr/local/Tivoli/lib
BIN=/usr/local/Tivoli/bin
DB=/usr/local/Tivoli/database
MAN=/usr/local/Tivoli/man
APPD=/usr/lib/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-Region
Overwrite=

[alias NT_managed_node TMF-client-3.7]
CAT=c:/usr/local/Tivoli/msg_cat
LIB=c:/usr/local/Tivoli/lib
BIN=c:/usr/local/Tivoli/bin
DB=c:/usr/local/Tivoli/database
MAN=c:/usr/local/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-Region
tapUser=
TapPassword=
Reboot=No
Overwrite=

[byNode]
jose=NT_managed_node NT_endpoint
sara=unix_managed_node unix_endpoint gateway

[alias gateway GATEWAY-Tivoli_Gateway]
@GATE_NAME@=sara-gateway
@GATE_PORT@=9494
Overwrite=

[machine sara]
access=rexec
userid=root
password=sara_root_pw
promptForPassword=no
interp=solaris2

[machine jose]
access=rexec
userid=Administrator
autoInstallTrip=yes
password=NT_password
promptForPassword=no
interp=w32-ix86

[globals]
InstallPassword=
InstallAlgorithm=mdist
```

7. Make the [alias] sections apply to multiple machines by modifying the entries that contain the machine name. Change them to use the @HostName@ variable instead of hard coding the host name. That requires the following changes:

- In the [alias unix_endpoint TMA-Tivoli_Endpoint] section, change @EndpointLabel@=sara to @EndpointLabel@=@HostName@-ep. This sets the label of each new UNIX endpoint to the host name plus the suffix -ep.

  Note: It is recommended that you change the endpoint label to ensure that endpoints installed on machines that are also managed nodes have unique labels.
- In the [alias gateway GATEWAY-Tivoli_Gateway] section, change @GATE_NAME@=sara-gateway to @GATE_NAME@=@HostName@-gateway. This sets the label of each new gateway to *hostname*-gateway.
- In the [alias NT_endpoint TMA-Tivoli_Endpoint] section, change @EndpointLabel@=jose to @EndpointLabel@=@HostName@-ep. This sets the label of each new Windows NT endpoint to the host name plus the suffix -ep.

8. Set installation options for each [alias] as appropriate. In this example, make the following changes:
   - Change the name of the default login gateway for each endpoint to a gateway named initial-gateway that is used only for initial logins. To do that, change the @GatewayName@= entry of both endpoint [alias] sections to the following value:
     ```
     @GatewayName@=initial-gateway
     ```
   - Create each managed node in a policy region specific to its machine type. (Hint: This is for demonstration purposes only. It is not a recommended way to assign managed nodes to policy regions.) To do that, change the @PR_NAME@= entry of the [alias NT_managed_node TMF-client-3.7] section from cygnus-Region to NT-Region. Change the @PR_NAME@= entry of the [alias unix_managed_node TMF-client-3.7] section from cygnus-Region to UNIX-Region.
9. When installing managed nodes on Windows machines, you need to supply the Tivoli Authentication Package connection information. Add the Tivoli Authentication Package user ID and password to the tapUser= and TapPassword= lines in the [alias NT_managed_node TMF-client-3.7] section. For more information about Tivoli Authentication Package, refer to "Using Windows systems" on page 339.
10. Create additional [machine] sections for additional machines you want to install. In this example, assume that you currently have the new Windows NT machines jose and molly, and the new UNIX systems sara (a solaris2 machine), basu and fadi (new aix4-r1 machines). The response file already contains entries for jose and sara. Add the following sections:
    ```
    [machine basu]
    access=rexec
    userid=root
    password=basu_root_pw
    promptForPassword=no

    [machine fadi]
    access=rexec
    userid=root
    password=fadi_root_pw
    promptForPassword=no

    [machine molly]
    access=rexec
    userid=Administrator
    password=NT-password
    autoInstallTrip=yes
    promptForPassword=no
    ```

Tivoli Software Installation Service automatically determines the interpreter type of each new machine, so the interp= entry is not required. (You can remove them from the existing [machine] sections if you prefer.) Set the autoInstallTrip= entry as appropriate for your Windows NT [machine] entries.

11. Add the new machines to the [byNode] or [byProduct] section of the response file. In this example, after adding the new machines and inserting comments to aid maintainability, the [byNode] section looks similar to the following:

```
[byNode]
# NT machines
jose=NT_managed_node NT_endpoint
molly=NT_managed_node NT_endpoint
# UNIX machines
sara=unix_managed_node unix_endpoint gateway
basu=unix_managed_node unix_endpoint gateway
fadi=unix_managed_node unix_endpoint gateway
```

12. Add comments to the response file to aid in maintaining it. Comments are lines that begin with a pound sign (#).

13. Rearrange the sections to create the organization you prefer. In the example, the [byNode] section is at the beginning of the file, followed by machine definitions, product definitions, and the [globals] section. The response file now looks like the following:

```
# Response file template to create the following
# resources:
#   NT machines: managed node and endpoint
#   UNIX machines: managed node, endpoint, and gateway

[byNode]
# NT machines
jose=NT_managed_node NT_endpoint
molly=NT_managed_node NT_endpoint
# UNIX machines
sara=unix_managed_node unix_endpoint gateway
basu=unix_managed_node unix_endpoint gateway
fadi=unix_managed_node unix_endpoint gateway

# UNIX machines
[machine sara]
access=rexec
userid=root
password=sara_root_pw
promptForPassword=no

[machine basu]
access=rexec
userid=root
password=basu_root_pw
promptForPassword=no

[machine fadi]
access=rexec
userid=root
password=fadi_root_pw
promptForPassword=no

# NT machines
[machine molly]
access=rexec
userid=Administrator
password=NT-password
autoInstallTrip=yes
promptForPassword=no
```

```
[machine jose]
access=rexec
userid=Administrator
password=NT-password
autoInstallTrip=yes
promptForPassword=no

# Endpoints
#   All UNIX platforms install to the same directory.
[alias unix_endpoint TMA-Tivoli_Endpoint]
TMABIN=/opt/Tivoli/lcf
@EndpointLabel@=@HostName@-ep
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=initial-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

[alias NT_endpoint TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=@HostName@-ep
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=initial-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
tapUser=
TapPassword=
Reboot=No
Overwrite=

# Managed nodes
#   All UNIX platforms install to the same directories.
[alias unix_managed_node TMF-client-3.7]
CAT=/usr/local/Tivoli/msg_cat
LIB=/usr/local/Tivoli/lib
BIN=/usr/local/Tivoli/bin
DB=/usr/local/Tivoli/database
MAN=/usr/local/Tivoli/man
APPD=/usr/lib/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=UNIX-Region
Overwrite=

[alias NT_managed_node TMF-client-3.7]
CAT=c:/usr/local/Tivoli/msg_cat
LIB=c:/usr/local/Tivoli/lib
BIN=c:/usr/local/Tivoli/bin
DB=c:/usr/local/Tivoli/database
MAN=c:/usr/local/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=NT-Region
tapUser=TAPuser
TapPassword=TAP#@PW
Reboot=No
Overwrite=

# Gateways
[alias gateway GATEWAY-Tivoli_Gateway]
@GATE_NAME@=@HostName@-gateway
```

```
@GATE_PORT@=9494
Overwrite=

[globals]
InstallPassword=
InstallAlgorithm=mdist
InstallPassword=your_install_pw
```

14. The response file is now ready to use. Save the file and exit from the text editor.
15. Verify the syntax of the response file using the **–c** option of the **wsis** command.

Chapter 14. Using Tivoli Software Installation Service log files

Tivoli Software Installation Service (SIS) log files (logs) provide a record of activity for your client and depot sessions. A client session corresponds to one invocation of either the console or commands such as **wsis** or **wimport**. A depot session starts when the depot is first contacted by a client and continues until the depot stops because it is inactive or when it is stopped by an administrator.

Tivoli Software Installation Service provides the following logs:

- Logs for the current depot session
- Logs for the current client session
- Logs from previous client and depot sessions
- Depot logs that have been archived

The client logs contain information about the connection from the client to the depot from the client and high-level information about each installation request.

The depot logs contain information about activity involving the depot. This includes information about connections from all client consoles and commands, the importing of products, installation status, and troubleshooting information about the depot. The depot logs are your primary source of information about installation progress and problems.

The logs are provided only in English. However, much of the information in the logs is also displayed to the user in the current language in the Tivoli Software Installation Service console, in message windows, and as output from commands.

This chapter contains the following sections:

- "Viewing the logs"
- "Using the View Logs window" on page 172
- "Structure of log files" on page 174
- "Packaging logs for support" on page 175
- "Archiving logs" on page 175
- "Deleting logs" on page 175
- "Finding and changing the client log directory" on page 176

Viewing the logs

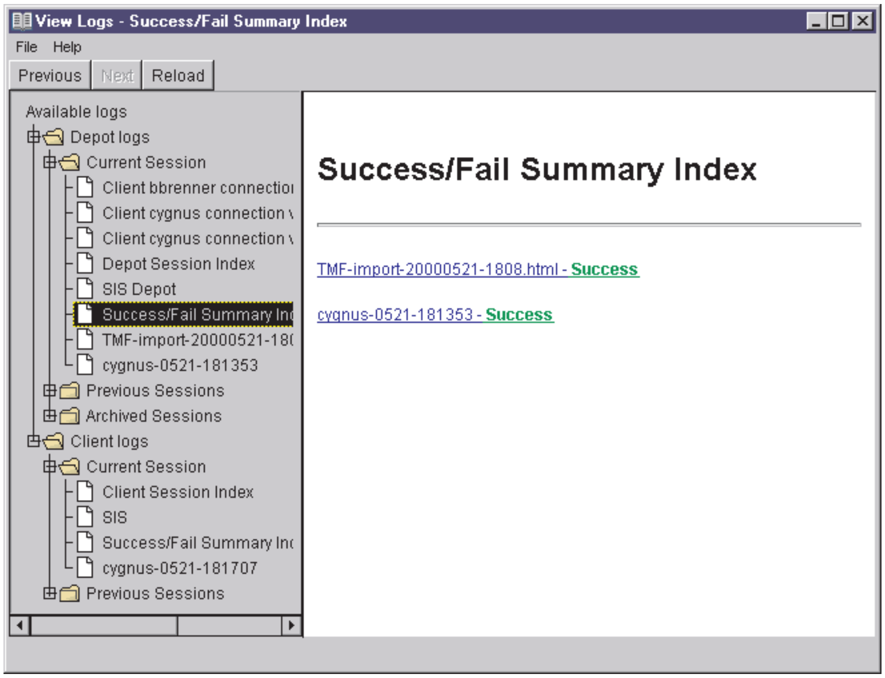
You can access the logs in one of the following ways:

- From the installation worksheet, select **File → View logs** to display the View Logs window.
- On any progress window, click **View logs**. This opens the View Logs window.

You cannot access the logs from the command line. If you have access to the machine on which the client or depot runs, you can use any browser to view the HTML log files. The client logs are located on the client in the directory specified by the `Client log directory` preference. The depot logs are located in the install repository.

Using the View Logs window

The following figure shows the View Logs window.



The View Logs window lists the available logs on the left and displays the contents of the selected log on the right. Click the plus sign [+] or minus sign [-] next to a folder name or double-click a folder name to expand or collapse its contents. Click any file in the navigator to display its contents on the right.

The Available logs folder contains the logs for the depot and client. The types of logs listed in each section are described in the following sections.

Depot logs

The depot logs contain detailed information about client sessions and installations. It contains diagnostic information about depot operations. The contents of the Depot logs folder are as follows:

Current Session

Contains the logs for the currently active depot. The following logs are found in this folder:

Client *name* **connection via [sisgui** | *command_name*]-*date-time*

Contains information about a client connection to the depot. There is one log for each client connection. A client connection is either a console session or the invocation of a Tivoli Software Installation Service command.

Note: The **wsisprefs** command does not create a connection log because it does not connect to the depot. It interacts directly with the SisDepot object.

Depot Session Index

Provides links to all depot logs for this session. Each log contains a return link to this file.

SIS Depot Server
Contains information about the operation of the depot. This log is primarily used when debugging problems with the depot.

Success/Fail Summary
Lists all installation attempts and whether each succeeded or failed. It also records attempts to import installation images and to import and export response files. This log provides links to the individual logs for each installation.

machine-date-time
Contains information about an installation to a machine. If you use a single client session to perform multiple installations to the same machine, there is a separate file for each attempt.

product-**import**-*date-time*
Contains information about each product that is imported.

rspfile-*date-time*
Contains information about each attempt to import or export a response file.

Previous Sessions
Contains a folder for each previous depot session. Each folder contains the same types of logs as the Current Session folder. A folder name has the following format `iu-`*`date-time`* where:

iu Indicates that this is a SIS log.

date Indicates the date on which the session was started. The date has the format *yyyymmdd*.

time Indicates the time at which the session was started. Time is indicated in 24-hour time in the format *hhmm*.

Archived Sessions
Contains a folder for each archive date. An archive folder can contain multiple folders, one for each session created on that date. The archive folder name has the following format: `iu-`*`date`*`.zip` where:

iu Indicates that this is a Tivoli Software Installation Service log.

date Indicates the date on which the sessions contained within the archive were started. The date has the format *yyyymmdd*.

.zip Indicates the archive is compressed into a zip format file.

When the depot starts, it examines the logs from previous sessions and selects those that match the archive criteria. For more information about the automatic archiving of logs, refer to “Archiving logs” on page 175.

Client logs

The client logs contain information about the client connection to the depot and the status of each installation. The contents of the Client logs folder are as follows:

Current Session
Contains the logs for the currently active client. The following logs are found in this folder:

SIS Contains information about the connection to the depot from the client.

Client Session Index
Provides links to all client logs for this session. Each log contains a return link to this file.

Success/Fail Summary
Provides a summary of all installation attempts and whether each succeeded or failed. This log provides a link to the log for each installation.

machine
Contains information about an installation on the specified machine. If you use a single client session to perform multiple installations to the same machine, there is a separate file for each attempt. To determine which log corresponds to a specific installation attempt, examine the time stamp inside the log file.

Previous Sessions
Contains a folder for each previous client session. Each folder contains the same types of logs as the Current Session folder. A folder name has the following format: `iu-`*`date-time`* where:

iu Indicates that this is a Tivoli Software Installation Service log.

date Indicates the date on which the session was started. The date has the format *yyyymmdd*.

time Indicates the time at which the session was started. Time is indicated in 24-hour time in the format *hhmm*.

Structure of log files

Each log file contains the following parts:

- A title.
- A link to the **Client Session Index** or **Depot Session Index**.
- A link to a list of errors contained in the log. This link exists even if no errors were recorded.
- A link to any warnings contained in the log. This link exists even if no warnings were recorded.
- The body of the log file, containing detailed log information. Within the body of the log, the color and weight of the text indicates whether the information describes the status of a normal operation, a warning, or a serious error. The following text conventions are used:

  Normal
  Text in normal typeface indicates status messages for a normal operation.

  Bold Text in bold typeface indicates text that is also displayed to the client, either through the Tivoli Software Installation Service console or as output from a command.

  Red Text in red indicates a problem. The text indicates whether a message is a warning or fatal error message.
- A list of error messages.
- A list of warning messages.

Packaging logs for support

If you encounter an installation problem or a problem with Tivoli Software Installation Service, you may need to send the logs to another Tivoli administrator or to your Tivoli support provider. To create a zip format compressed archive file of the logs for the current depot and client session, complete these steps:

1. From the **File** menu of the View Logs window, click **Zip Logs for Support**.
2. In the Save window, select or type the directory in which you want to save the compressed logs. If you type a directory name, click **Update**.
3. Type or select a name for the compressed logs.
4. Click **OK**.

The zip file is saved on the Tivoli Software Installation Service client machine.

Archiving logs

Depot logs can be automatically archived to free up disk space. When log files are archived, they are compressed into a zip file format. In this format, log files require less disk space but can still be displayed in the View Logs window.

To automatically archive depot log files, set the `Archive SIS depot logs after (days, 0=never)` preference to a nonzero number of days. If you set this preference to 0, the logs are never archived. The maximum value is 365. For more information about modifying preferences, refer to “Modifying preferences” on page 102.

When the depot starts, it archives any log files that were created the specified number of days ago. The creation date of a server log is the date on which the depot was started. One archive folder is created for each date; each archive folder contains a folder for each session started on that date.

Archiving is based on calendar days; for example, if you set the archive value to 1 day, when the depot starts, it archives any logs created the day before, regardless of whether 24 hours has passed.

Client logs are not automatically archived. The View Logs window does not read compressed client log files.

Deleting logs

Depot logs can be automatically deleted to free up disk space. To automatically delete log files, set the `Delete SIS depot logs after (days, 0=never)` preference to a nonzero number of days. If you set this preference to 0, the logs are never deleted. The maximum value is 365. For more information about modifying preferences, refer to “Modifying preferences” on page 102.

When the depot starts, it deletes any log files (archived or unarchived) that were created the specified number of days ago. The creation date of a server log is the date on which the depot was started.

Deleting is based on calendar days; for example, if you set the delete value to 1 day, when the depot starts, it deletes any logs created the day before, regardless of whether 24 hours has passed.

After a log file or log file archive is deleted, it cannot be restored.

Client logs are not automatically deleted. You must manually delete them.

Finding and changing the client log directory

To locate or change the client log directory, you can use the Tivoli Software Installation Service console or the **wsisprefs** command. The following sections provide instructions for each method.

Note: Changing the client log directory causes the client to create subsequent client logs and look for client logs in the new directory. Existing logs in the original directory are neither moved nor deleted, but they can no longer be viewed with the View Logs window.

If you want to use the View Logs window to view the existing client logs, you must manually move them to the new directory.

If you want to delete the existing log files and directories, delete them after changing the directory.

Finding and changing the client log directory using the Tivoli Software Installation Service console

To locate or change the directory containing the client logs using the console, perform the following steps:

1. From the installation worksheet, select **Edit → Preferences** to display the SIS Preferences window.
2. View or modify the value of the **Client log directory** field. To change the value, specify the name of a directory on the system that is running the client. If the directory does not already exist, it is created the next time a client starts.

If you change the value, you must close and reopen the Tivoli Software Installation Service console to start using the new directory. Commands start using the new location immediately.

Finding and changing the client log directory from the command line

To locate the directory containing the client logs using the **wsisprefs** command, perform the following steps:

1. Enter the following command to determine the current location of the client log directory and to determine the exact label or number you will use to change the `Client log directory` preference:

   ```
   wsisprefs -l
   ```

   The bold text in the following output fragment shows the preference:

   ```
   Client Preferences

   #  Name                 Label                 Value
   -- -------------------- --------------------- -----------
   14 Client log directory client_data_dir       /data/c_logs
   15 Client prompt timeout client_prompt_timeout 20
   ```

2. To change the location, use the **wsisprefs** command. For example, to set the client log directory to /tmp/clientlogs, enter the following command:

   ```
   wsisprefs -s 14 /tmp/clientlogs
   ```

   If the directory does not already exist, it is created the next time a client starts.

If you change the value, you must close and reopen the Tivoli Software Installation Service console to start using the new directory. Commands start using the new location immediately.

Chapter 15. Checking prerequisites

Prerequisites are system and machine checks that Tivoli Software Installation Service (SIS) performs before installing software on a machine. When a failure is encountered, a brief message is displayed and detailed information is written to a log.

An initial set of prerequisite checks is provided with Tivoli Software Installation Service. These *Tivoli-defined prerequisites* test for common causes of installation failure. For example, they ensure that network connections are working and that two-way communications are enabled.

You can also create your own prerequisites, called *user-defined prerequisites*. Each prerequisite has a shell script that runs on the target machine to test conditions specific to your Tivoli environment.

The following types of installations are supported:

Client These prerequisites are run before installing Tivoli Management Framework to create a managed node (client).

Product
These prerequisites are run before installing a Tivoli product on a managed node. These prerequisites are not checked when installing Tivoli products to endpoints.

Patch These prerequisites are run before applying a patch to a Tivoli product on a managed node. These prerequisites are not checked when applying patches to Tivoli products installed on endpoints.

Endpoint
These prerequisites are run before installing an endpoint on a managed node or new machine. These prerequisites are not checked when installing Tivoli Enterprise software to existing endpoints.

Note: Special prerequisites are run before installing an endpoint on an OS/400 machine. You cannot enable or disable these prerequisites or create user-defined prerequisites for OS/400 endpoints.

If an install repository is shared by depots in multiple Tivoli regions, the prerequisites apply only to the region in which they were defined.

You can work with prerequisites using either the Tivoli Software Installation Service console or the command line. The following sections describe:

- "Creating an installation prerequisite" on page 180
- "Removing an installation prerequisite" on page 184
- "Modifying an installation prerequisite" on page 186
- "Tivoli-defined prerequisites" on page 189
- "Format of the prerequisite definition file" on page 191
- "Using variables in prerequisite scripts" on page 193

Creating an installation prerequisite

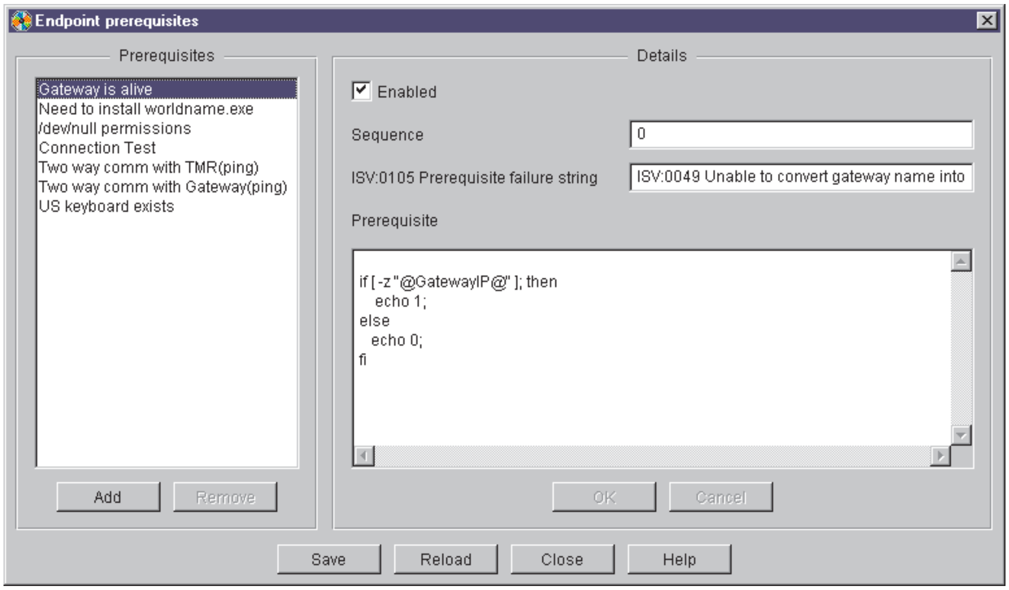
Each prerequisite you create is used in every installation performed by the Tivoli Software Installation Service depot until you remove or disable it as described in "Removing an installation prerequisite" on page 184 and "Modifying an installation prerequisite" on page 186.

You can create a prerequisite using either the Tivoli Software Installation Service console or the command line. These tasks are described in the following sections.

Creating a prerequisite using the Tivoli Software Installation Service console

To create an installation prerequisite using the console, perform the following steps:

1. From the **Edit** menu of the installation worksheet, select the type of prerequisite you want to create. In the following example, **Edit → Endpoint prerequisites** was selected to display the Endpoint Prerequisites window.



Gateway is alive
Need to install worldname.exe
/dev/null permissions
Connection Test
Two way comm with TMR(ping)
Two way comm with Gateway(ping)
US keyboard exists

☑ Enabled

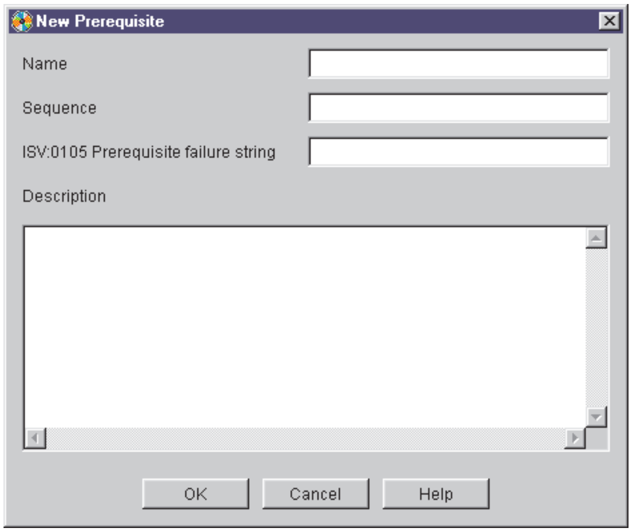
Sequence: 0

ISV:0105 Prerequisite failure string: ISV:0049 Unable to convert gateway name into

Prerequisite

```
if [ -z "@GatewayIP@" ]; then
   echo 1;
else
   echo 0;
fi
```

2. Click **Add** to add a prerequisite to the **Prerequisites** list. The New Prerequisite window is displayed. Initially, all fields in the window are blank.

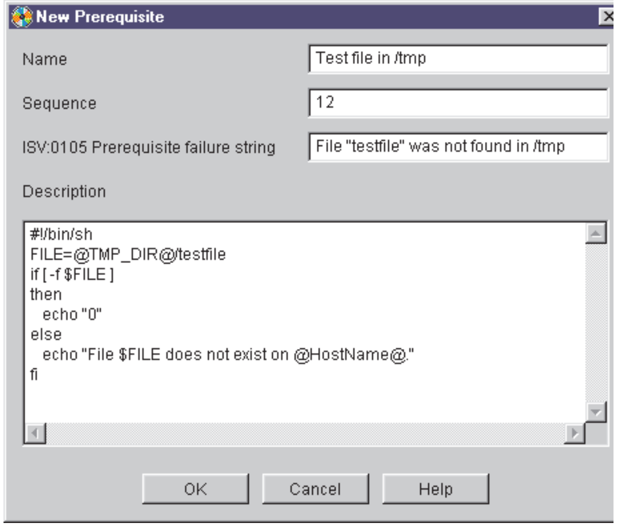


3. In the **Name** field, type a descriptive name for the prerequisite.
4. In the **Sequence** field, type a number indicating the order in which you want the prerequisite to run.

   Prerequisites are run in numerical order, beginning with sequence number **0**. You can skip or duplicate sequence numbers. The order in which prerequisite checks run when they have the same sequence number is undefined.
5. In the **Prerequisite failure string** field, type the error message that is written to the depot log if the prerequisite script fails.
6. In the **Description** field, type the shell script that runs on the target system to test the prerequisite. To indicate success, the last text written to standard output must be the single character "0" (zero). Any non-zero value in the last text written to standard output (for example, a text message) indicates failure. Any other information written to standard output, and any information written to standard error, is recorded in the depot logs.

   For example, the following New Prerequisite window defines an over-simplified prerequisite that tests for the existence of a file, testfile, in the temporary directory appropriate to the platform of the target machine (@TMP_DIR@), on the machine to which software is to be installed (@HostName@). If the file exists, the script writes the number 0 to standard output to indicate that the prerequisite succeeded. If the file does not exist on the target machine, the script stops the installation process and writes an error message that is recorded in the depot logs.

   For additional information about special variables such as @TMP_DIR@ and @HostName@, refer to "Using variables in prerequisite scripts" on page 193.



Name: Test file in /tmp

Sequence: 12

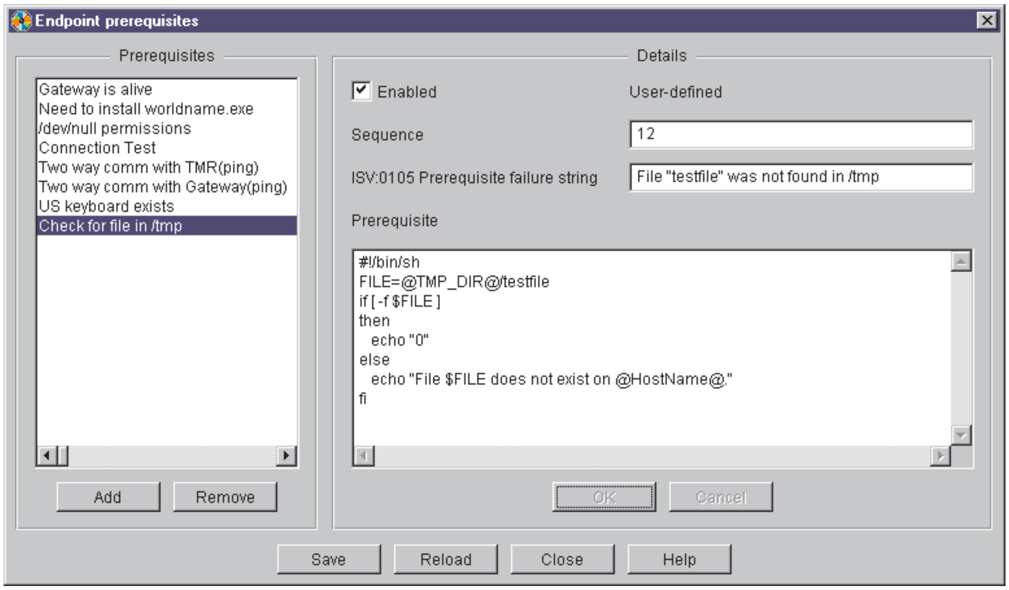
ISV:0105 Prerequisite failure string: File "testfile" was not found in /tmp

Description

```
#!/bin/sh
FILE=@TMP_DIR@/testfile
if [ -f $FILE ]
then
  echo "0"
else
  echo "File $FILE does not exist on @HostName@."
fi
```

7. Click **OK** to add the prerequisite to the **Prerequisites** list and close the New Prerequisite window.
8. In the Endpoint Prerequisites window, select the name of the new prerequisite.
9. Click **Enabled** to enable the prerequisite.
10. Click **OK** to record the change.

    The following example shows the new prerequisite added to the **Prerequisites** list and enabled.



Prerequisites:

- Gateway is alive
- Need to install worldname.exe
- /dev/null permissions
- Connection Test
- Two way comm with TMR(ping)
- Two way comm with Gateway(ping)
- US keyboard exists
- Check for file in /tmp

☑ Enabled   User-defined

Sequence: 12

ISV:0105 Prerequisite failure string: File "testfile" was not found in /tmp

Prerequisite

```
#!/bin/sh
FILE=@TMP_DIR@/testfile
if [ -f $FILE ]
then
  echo "0"
else
  echo "File $FILE does not exist on @HostName@."
fi
```

11. Click **Save** to save the new prerequisite. (Although you added the new prerequisite to the list in the previous step, you must save the updated list.)

    Note: If you do not click **Save** before clicking **Close**, the new prerequisite is not saved.
12. Click **Close** to return to the installation worksheet.

Creating a prerequisite from the command line

1. Create and test the prerequisite script. For this example, the following script will run to test the prerequisite:

   ```
   #!/bin/sh
   FILE=@TMP_DIR@/testfile
   if [ -f $FILE ]
   then
     echo "0"
   else
     echo "Cannot find file $FILE on @HostName@."
   fi
   ```

   This script tests for the existence of a file, testfile, in the temporary directory appropriate to the platform of the target machine (@TMP_DIR@), on the machine to which software is to be installed (@HostName@). If the file exists, the script writes the number 0 to standard output to indicate that the prerequisite succeeded. If the file does not exist on the target machine, the script writes an error message that is recorded in the Tivoli Software Installation Service depot logs and stops the installation.
2. Export the existing user-defined prerequisites for an endpoint installation. For example, to export into the file /tmp/ep.prereqs, enter the following command:

   ```
   wsisprereq -x /tmp endpoint ep.prereqs
   ```

   The exported file contains comments explaining the format of the file and the definition of each existing user-defined prerequisite.
3. Add the new prerequisite to the exported file, following the format described in “Format of the prerequisite definition file” on page 191. For example, to create an enabled prerequisite named **/tmp/testfile exists** that runs at sequence 12, and issues the message **Can’t find /tmp/testfile** if the script fails, add the following lines:

   ```
   [Look for file in /tmp]
   sequence=12
   enabled=true
   failStr=Can't find /tmp/testfile
   description=#!/bin/sh
   description=FILE=@TMP_DIR@/testfile
   description=if [ -f $FILE ]; then
   description=  echo "0"
   description=else
   description=  echo "Cannot find file $FILE on @HostName@."
   description=fi
   ```
4. Import the prerequisite definition file. For example, the following command imports prerequisites from the file **/tmp/ep.prereqs**:

   ```
   wsisprereq -i /tmp/ep.prereqs endpoint
   ```
5. Verify the prerequisite. For example, the following command lists the resulting prerequisites:

   ```
   wsisprereq -l endpoint
   ```

   The following sample output shows that the new prerequisite was added as prerequisite number 8:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   ---------- Prerequisites for endpoint ----------

   Number Name                           Enabled User-defined Sequence
   ------ ------------------------------ ------- ------------ --------
   1      Gateway is alive               true    false        0
   2      Need to install worldname.exe  true    false        1
   3      /dev/null permissions          true    false        2
   ```

```
4       Connection Test                 true    false       3
5       Two way comm with TMR(ping)     true    false       4
6       Two way comm with Gateway(ping) true    false       5
7       US keyboard exists              true    false       6
8       Look for file in /tmp           true    true        12
```

For more information about the **wsisprereq** command, refer to "wsisprefs" on page 374.

Removing an installation prerequisite

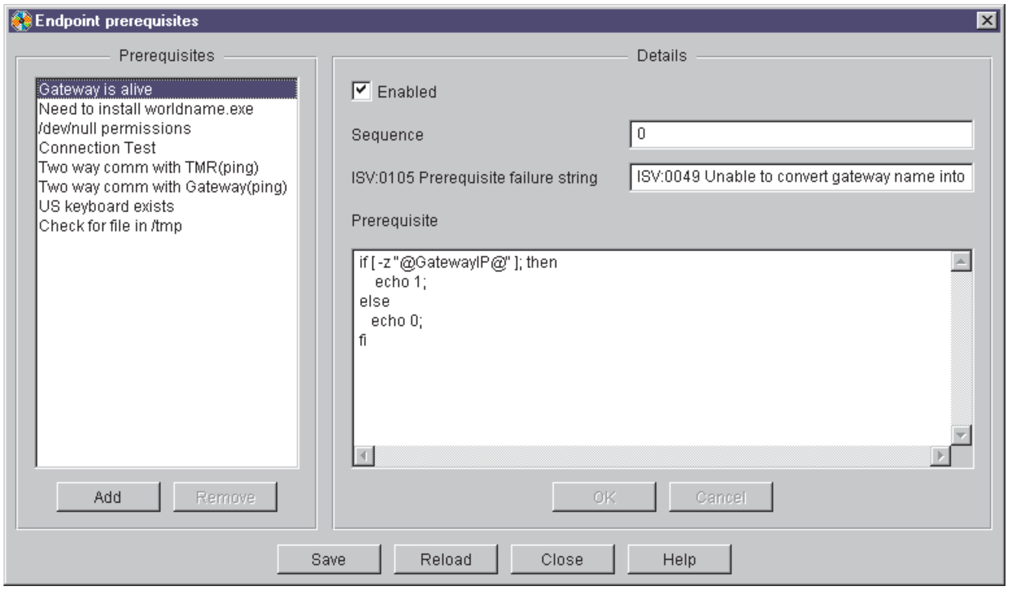
In general, it is better to disable a prerequisite than to remove it. For information about disabling prerequisites, refer to "Modifying an installation prerequisite" on page 186.

You can remove only user-defined prerequisites. Tivoli-defined prerequisites cannot be removed. To prevent a Tivoli-defined prerequisite from being tested, disable it. For a list of Tivoli-defined prerequisites, refer to "Tivoli-defined prerequisites" on page 189.

Removing a prerequisite using the Tivoli Software Installation Service console

To remove an installation prerequisite, perform the following steps:

1. From the **Edit** menu of the installation worksheet, select the type of prerequisite you want to create. In the following example, **Edit → Endpoint prerequisites** was selected to display the Endpoint Prerequisites window.



Prerequisites:
- Gateway is alive
- Need to install worldname.exe
- /dev/null permissions
- Connection Test
- Two way comm with TMR(ping)
- Two way comm with Gateway(ping)
- US keyboard exists
- Check for file in /tmp

☑ Enabled

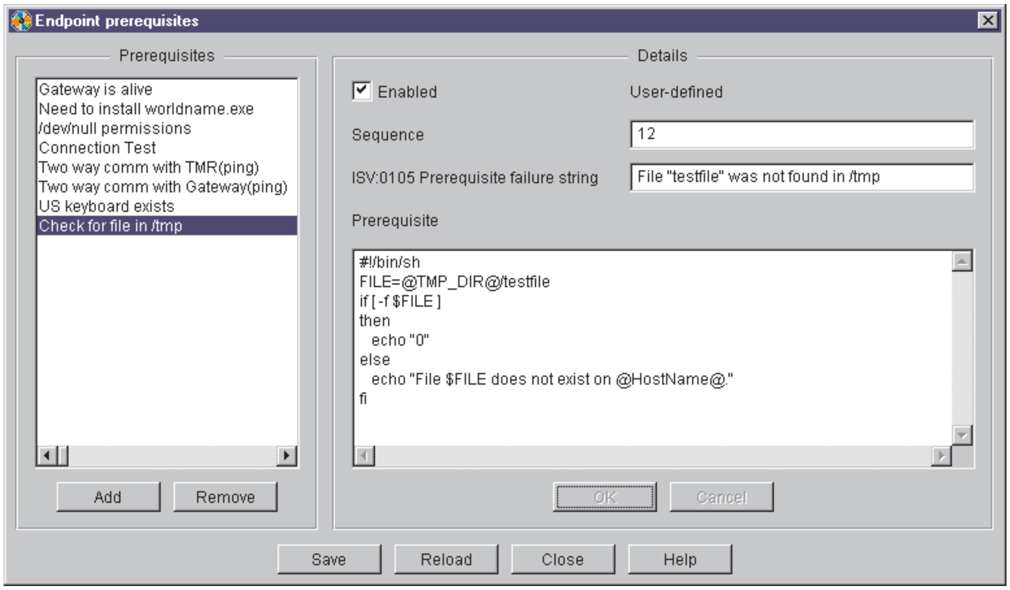
Sequence: 0

ISV:0105 Prerequisite failure string: ISV:0049 Unable to convert gateway name into

Prerequisite:

```
if [ -z "@GatewayIP@" ]; then
   echo 1;
else
   echo 0;
fi
```

2. In the **Prerequisites** list, select the name of the prerequisite to be removed. In the following example, the endpoint prerequisite **Check for file in /tmp** is selected.



Gateway is alive
Need to install worldname.exe
/dev/null permissions
Connection Test
Two way comm with TMR(ping)
Two way comm with Gateway(ping)
US keyboard exists
Check for file in /tmp

☑ Enabled User-defined

Sequence: 12

ISV:0105 Prerequisite failure string: File "testfile" was not found in /tmp

Prerequisite

```
#!/bin/sh
FILE=@TMP_DIR@/testfile
if [ -f $FILE ]
then
  echo "0"
else
  echo "File $FILE does not exist on @HostName@."
fi
```

3. Click **Remove** to remove the selected prerequisite from the **Prerequisites** list.
4. Click **Save** to record that the prerequisite was deleted, then click **Close** to return to the installation worksheet.

   Note: If you do not click **Save** before clicking **Close**, the prerequisite is not removed.

Removing a prerequisite from the command line

To remove the user-defined endpoint prerequisite named `Check for file in /tmp`, perform the following steps:

1. Determine the prerequisite number using the following command:

   ```
   wsisprereq -l endpoint
   ```

   The following sample output shows that the desired prerequisite is prerequisite number 8. The value `true` in the `User-defined` column indicates that this is a user-defined prerequisite, which can be removed.

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   ---------- Prerequisites for endpoint ----------

   Number Name                           Enabled User-defined Sequence
   ------ ------------------------------ ------- ------------ --------
   1      Gateway is alive               true    false        0
   2      Need to install worldname.exe  true    false        1
   3      /dev/null permissions          true    false        2
   4      Connection Test                true    false        3
   5      Two way comm with TMR(ping)    true    false        4
   6      Two way comm with Gateway(ping) true   false        5
   7      US keyboard exists             true    false        6
   8      Check for file in /tmp         true    true         12
   ```

2. Remove the prerequisite using the following command:

   ```
   wsisprereq -r -t endpoint -p 8
   ```

For more information about the **wsisprereq** command, refer to “wsisprefs” on page 374.

Modifying an installation prerequisite

You can change a user-defined prerequisite in the following ways:

- Enable or disable it to control whether it runs at all.
- Change the sequence in which it runs.
- Change the string that is recorded in the logs if it fails.
- Change the script that runs to check it.

For Tivoli-defined prerequisites, you can change the sequence in which the prerequisite runs and whether the prerequisite is enabled or disabled.

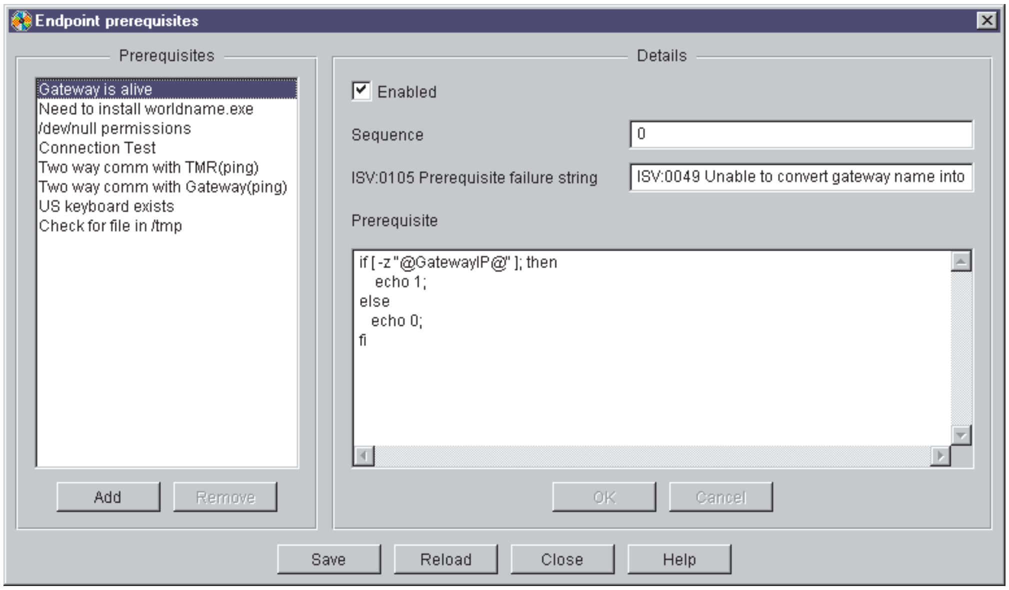
Enabling a prerequisite allows it to run when Tivoli Software Installation Service performs prerequisite tests for the corresponding type of installation. *Disabling* a prerequisite prevents it from being run. You can do this using either the console or the command line.

The following sections describe how to modify a prerequisite using the Tivoli Software Installation Service console and the Tivoli Software Installation Service commands.

Modifying a prerequisite using the Tivoli Software Installation Service console

To modify an installation prerequisite using the console, perform the following steps:

1. From the **Edit** menu of the installation worksheet, select the type of prerequisite you want to create. In the following example, **Edit → Endpoint prerequisites** was selected to display the Endpoint Prerequisites window.



Gateway is alive
Need to install worldname.exe
/dev/null permissions
Connection Test
Two way comm with TMR(ping)
Two way comm with Gateway(ping)
US keyboard exists
Check for file in /tmp

☑ Enabled

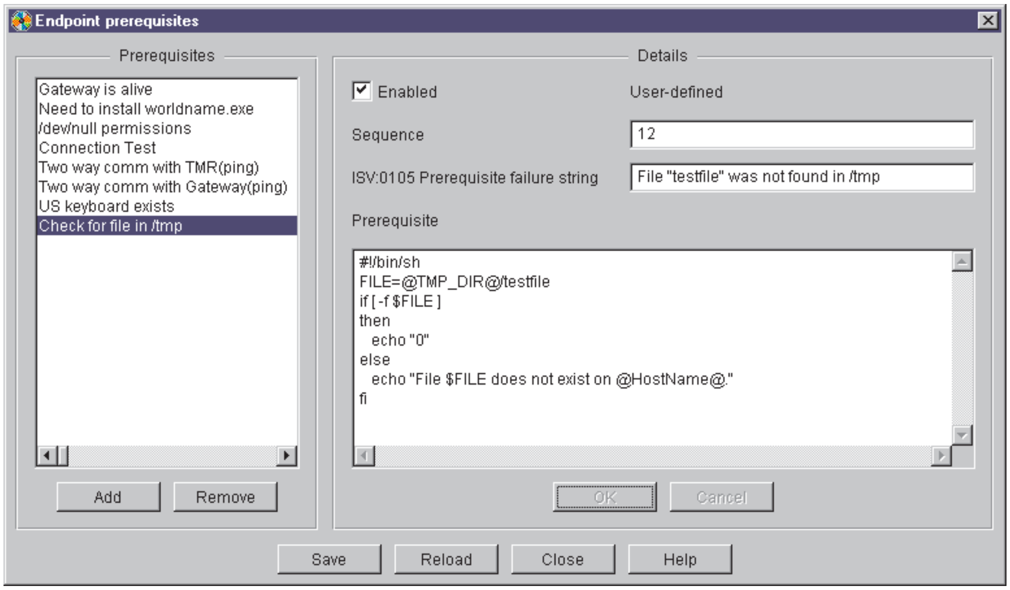
Sequence: 0

ISV:0105 Prerequisite failure string: ISV:0049 Unable to convert gateway name into

Prerequisite

```
if [ -z "@GatewayIP@" ]; then
    echo 1;
else
   echo 0;
fi
```

2. In the **Prerequisites** list, select the name of the prerequisite to modify. In the following example, the endpoint prerequisite **Check for file in /tmp** is selected.



Gateway is alive
Need to install worldname.exe
/dev/null permissions
Connection Test
Two way comm with TMR(ping)
Two way comm with Gateway(ping)
US keyboard exists
Check for file in /tmp

☑ Enabled User-defined

Sequence: 12

ISV:0105 Prerequisite failure string: File "testfile" was not found in /tmp

Prerequisite

```
#!/bin/sh
FILE=@TMP_DIR@/testfile
if [ -f $FILE ]
then
  echo "0"
else
  echo "File $FILE does not exist on @HostName@."
fi
```

3. Make the appropriate changes to the prerequisite. The procedures for making those changes are described here:
   - To enable or disable a prerequisite, click **Enabled**.
   - To change the sequence in which the prerequisite runs, modify the **Sequence** field. Prerequisites at sequence **0** are the first to run. The order in which prerequisite checks run when they have the same sequence number is undefined.
   - To change the message that displays in the log when the prerequisite fails, change the text in the **Prerequisite failure string** field.
   - To change the shell script that runs to test the prerequisite, modify the script in the **Description** field.

     The output displays the following information:
     - The number of the preference. The number can be used to identify a preference when changing its value with the **wsisprefs** command.
     - The name of the preference. This value is displayed in the language of the current locale.
     - The internal label of the preference. This value can be used to identify a preference when changing its value with the **wsisprefs** command. This value is not translated.
     - The current setting of the preference.
4. Click **OK** to record the changes.
5. Click **Save** to save the changes and then click **Close** to return to the installation worksheet.

   Note: If you do not click **Save** before clicking **Close**, your changes are not saved.

Modifying a prerequisite from the command line

The following sections describe how to use the commands to modify a prerequisite:

- “Enabling or disabling a prerequisite” on page 188
- “Changing the sequence of a prerequisite” on page 188

- “Modifying prerequisite attributes” on page 189

Enabling or disabling a prerequisite

The following procedures describe how to disable and enable a SIS prerequisite using the command line.

- To disable the Tivoli-defined product prerequisite named `Need to install worldname.exe`, perform the following steps:
  1. Determine the prerequisite number using the following command:

     ```
     wsisprereq -l product
     ```

     The following sample output shows that the desired prerequisite is prerequisite number 2:

     ```
     Connecting to SIS depot
     Successfully connected to SIS depot cygnus
     ---------- Prerequisites for product ----------

     Number Name                             Enabled User-defined Sequence
     ------ -------------------------------- ------- ------------ --------
     1      Connection to target             true    false        0
     2      Need to install worldname.exe    true    false        2
     ```
  2. To disable prerequisite number 2 for all product installations, use this command:

     ```
     wsisprereq -e no product 2
     ```
- To enable the Tivoli-defined product prerequisite named `Need to install worldname.exe`, perform the following steps:
  1. Determine the prerequisite number using the following command:

     ```
     wsisprereq -l product
     ```

     The following sample output shows that the desired prerequisite is prerequisite number 2.

     ```
     Connecting to Tivoli Software Installation Service depot
     Successfully connected to Tivoli Software Installation Service depot cygnus
     ---------- Prerequisites for product ----------

     Number Name                             Enabled User-defined Sequence
     ------ -------------------------------- ------- ------------ --------
     1      Connection to target             true    false        0
     2      Need to install worldname.exe    true    false        1
     ```
  2. To enable prerequisite number 2 for all product installations, use either of these commands:

     ```
     wsisprereq -e product 2
     wsisprereq -e yes product 2
     ```

For more information about the **wsisprereq** command, refer to “wsisprefs” on page 374.

Changing the sequence of a prerequisite

To change the sequence in which the endpoint prerequisite named `Check for file in /tmp` runs, perform the following steps:

1. Determine the prerequisite number using the following command:

   ```
   wsisprereq -l endpoint
   ```

   The following sample output shows that the desired prerequisite is prerequisite number 8.

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   ---------- Prerequisites for endpoint ----------

   Number Name                             Enabled User-defined Sequence
   ```

```
   ------ ------------------------------ ------- ------------ --------
   1      Gateway is alive               true    false        0
   2      Need to install worldname.exe  true    false        1
   3      /dev/null permissions          true    false        2
   4      Connection Test                true    false        3
   5      Two way comm with TMR(ping)    true    false        4
   6      Two way comm with Gateway(ping) true   false        5
   7      US keyboard exists             true    false        6
   8      Check for file in /tmp         true    true         12
```

2. To make the prerequisite number 8 run at sequence 0, use the following command:

   ```
   wsisprereq -s 0 Endpoint 8
   ```

   Note: Resequencing a prerequisite changes its prerequisite number and the numbers of other prerequisites. Use the **–l** option of the **wsisprereq** command to determine the new numbers.

For more information about the **wsisprereq** command, refer to "wsisprefs" on page 374.

Modifying prerequisite attributes

You can modify any attributes of a user-defined prerequisite using a prerequisite definition file. This includes its name, enabled status, sequence number, failure message, and the script that runs to check the prerequisite. Perform the following steps:

1. Export the user-defined prerequisites. The following command exports the user-defined prerequisites for endpoint installations:

   ```
   wsisprereq -x /tmp endpoint ep.prereqs
   ```

   This creates the file /tmp/ep.prereqs that contains all the user-defined prerequisites for endpoint installations.
2. Use any text editor to modify the prerequisite definition file to change the desired attributes of one or more prerequisites. For more information about changing this file, refer to "Format of the prerequisite definition file" on page 191 and "Using variables in prerequisite scripts" on page 193.

   Removing a prerequisite or one of its attributes from the exported file removes that prerequisite from Tivoli Software Installation Service or resets its attribute.
3. Import the prerequisite definition file to modify the prerequisites. The following command imports the file /tmp/ep.prereqs, which sets user-defined prerequisites as described in that file:

   ```
   wsisprereq -i /tmp/ep.prereqs endpoint
   ```

For more information about the **wsisprereq** command, refer to "wsisprefs" on page 374.

Tivoli-defined prerequisites

This section describes the Tivoli-defined prerequisites for the following types of installation:

- Managed node
- Product
- Patch
- Endpoint

Managed node prerequisites

Tivoli Software Installation Service checks the following Tivoli-defined prerequisites before installing the Tivoli Management Framework to create a managed node (client):

Bourne shell
For UNIX targets, verifies that the root shell on the target is either Bourne or Korn shell.

Connection to target
Verifies that the depot can run a script on the target machine.

Database directory not already installed
Verifies that the Tivoli Management Framework is not already installed on the target.

This prerequisite is checked only if you did not specify that the install should overwrite an existing client database (by selecting the **Overwrite** option for the **Client Database** field in the installation options in the **Product details** or Product attributes window). If the target is already a managed node and you did not select **Overwrite**, this prerequisite fails.

DBDIR is local and not FAT
For Windows and OS/2 targets, verifies that the client database directory is in a local file system and is not in a FAT file system. A managed node must be installed in a file system that supports long file names.

/dev/null permissions
For UNIX targets, verifies that you have write access to the /dev/null directory.

Target/server connection
Verifies that the target machine can ping the Tivoli server.

US keyboard exists
For Windows targets, verifies that the U.S. keyboard DLL file (kbdus.dll) is available. This file is required even if the target is not using the US keyboard.

Valid license key
Verifies that the license key is valid and has not expired.

Refer to “Creating an installation prerequisite” on page 180 for instructions for creating your own prerequisites.

Product prerequisites

Tivoli Software Installation Service checks the following prerequisites before installing a Tivoli product on an existing managed node. These prerequisites do not apply when installing products on an endpoint.

Connection to target
Verifies that the depot can run a script on the target machine.

Need to install worldname.exe
On Windows targets, determines whether the file worldname.exe exists. This prerequisite is successful if the file does not exist; in this case, the file is created when an endpoint is installed. Failure of this prerequisite indicates that the file already exists on the target; the installation does not terminate (as when other prerequisites fail), but continues without installing worldname.exe.

Refer to “Creating an installation prerequisite” on page 180 for instructions for creating your own prerequisites.

Patch prerequisites

Tivoli Software Installation Service checks the following prerequisites before installing a Tivoli patch on an existing managed node. These prerequisites do not apply when installing patches to Tivoli products installed on an endpoint.

Connection to target
Verifies that the depot can run a script on the target machine.

Refer to “Creating an installation prerequisite” on page 180 for instructions for creating your own prerequisites.

Endpoint prerequisites

Tivoli Software Installation Service checks the following prerequisites before creating an endpoint on a managed node or new machine. These prerequisites are not checked when installing Tivoli Enterprise software to existing endpoints:

Connection Test
Verifies that the depot can run a script on the target machine.

/dev/null permissions
For UNIX targets, verifies that you have write permission in the /dev/null directory.

Gateway is alive
Verifies that the default gateway for this endpoint is running.

Need to install worldname.exe
On Windows targets, determines whether the file worldname.exe exists. This prerequisite is successful if the file does not exist; in this case, the file is created when an endpoint is installed. Failure of this prerequisite indicates that the file already exists on the target; the installation does not terminate (as when other prerequisites fail), but continues without installing worldname.exe.

Two way comm with Gateway (ping)
Verifies that the target machine can ping the gateway to which it will log in.

Two way comm with TMR (ping)
Verifies that the target machine can ping the Tivoli server.

US keyboard exists
For Windows targets, verifies that the U.S. keyboard DLL file (kbdus.dll) is available. This file is required even if the target is not using the US keyboard.

Refer to “Creating an installation prerequisite” on page 180 for instructions for creating your own prerequisites.

Format of the prerequisite definition file

The format of the prerequisite definition file is as follows:

[*name_of_prerequisite***]**
sequence=*number*
enabled=[true | false]

failStr=*text_message*
description=*first_line_of_script*
description=*second_line_of_script*
...
description=*last_line_of_script*

where:

[*name_of_prerequisite*]
Specifies the name of the prerequisite. The name must be enclosed in square brackets ([ ]).

sequence=*number*
Specified the sequence in which the prerequisite is run. Prerequisites are run in numerical order, beginning with sequence number 0. You can skip or duplicate sequence numbers. The order in which prerequisite checks run when they have the same sequence number is undefined.

enabled=[**true** | **false**]
Specifies whether the prerequisite is checked before an installation.

failStr=*text_message*
Specifies the message that is written to the log when the prerequisite fails. This message is in addition to any output written by the prerequisite script, which is also collected in the log.

description=*line_of_script*
Specifies one line of the shell script that runs on the target machine to test the prerequisite. For information about special variables available for use in the prerequisite script, refer to "Using variables in prerequisite scripts" on page 193.

To indicate success, the last text written to standard output must be the single character "0" (zero). Any non-zero value in the last text written to standard output (for example, a text message) indicates failure. Any other information written to standard output, and any information written to standard error, is recorded in the depot logs.

Include **description=** at the beginning of each line of the script. When you export a prerequisite definition file using the **–x** option of the **wsisprereq** command, the description statements are numbered, as shown in this example:

```
description1=#!/bin/sh
description2=if [ -f /tmp/testfile ]; then
description3=   echo 0
description4=else
description5=   echo "ERROR: File not found."
description6=fi
```

When you import a prerequisite definition file, the description numbers are ignored. You do not need to number the statements in your description. If you are modifying an exported file, you do not need to modify or remove the existing numbers. You may skip description numbers or specify them out of sequence.

Note: Unrecognized keywords in a prerequisite definition file are ignored. Make sure that you spell the names of the keywords correctly (for example, **description=**) to prevent prerequisite errors.

Using variables in prerequisite scripts

This section describes variables you can use in the shell script that tests your prerequisite. When you use these variables in a script, the name must start and end with an at sign (@). The case of the variable name is significant. Some variables provide a platform-independent way to issue commands, for example, `@mkdirCommand@` and `@ping@`.

The following variables are available:

installation_options

You can test installation options for all products selected for install on the target machine. For example, if the depot is selected, you can test @IRDIR@, the target directory for the install repository.

When installing to a managed node, the Tivoli Management Framework directories are available, including the following:

@ALIDB@

The directory on the Tivoli server where the Tivoli object database resides ($DBDIR).

@CAT@

The directory where product message catalogs are installed.

@BIN@

The directory where architecture-specific product binaries are stored ($BINDIR).

@LIB@

The directory where product libraries are installed.

@MAN@

The directory where product man pages are installed.

@DB@ The directory on managed nodes where the Tivoli object database resides ($DBDIR).

When installing to an endpoint, the following directory variable is available:

@TMABIN@

The directory containing the endpoint files.

When installing an endpoint, managed node, or gateway, you can test additional installation options. For a description of these options, refer to "Syntax—product-specific" on page 149.

@Arch@

The interpreter type of the target machine. For example, `generic`, `w32-ix86` or `aix4-r1`. For a complete list of interpreter types and the operating systems they represent, refer to the *Tivoli Management Framework Release Notes*

@BIN+ROOT@

The root directory containing binary files for all interpreter types. That is, $BINDIR/...

@DevNull@

/dev/null for UNIX operating systems, "null" Windows and OS/2 operating systems.

@HostName@
The host name of target machine.

@mkdirCommand@
A platform-independent command that creates a directory. Before using this variable, set the $MDIR variable to the name of the directory you want to create. For example, the following lines in your prerequisite script create the /new/directory directory:

```
MDIR=/new/directory
@mkdirCommand@
```

@ping@
A platform-independent command to ping a host. Before using this variable, set the $PINGHOST variable to the name of the host you want to ping. For example, the following statements in your prerequisite script cause the target machine to ping the Tivoli server and use the result to determine success or failure of the prerequisite:

```
PINGHOST=@ALI_NAME@
@ping@
echo $?
```

@Port@
The port number on which the Tivoli server listens ($o_dispatch).

@ProductName@
The name of the product being installed. This comes from the `description` statement in the product .IND file.

@ProductRevision@
The revision number of the product being installed. This comes from the `revision` statement in the product .IND file.

@ProductTag@
The registered product tag of the product being installed. This is the first value on each line of the product .IND file.

@Region@
Region number of the Tivoli region, for example, 1979132042. This is the Tivoli environment variable $TMR.

@Shell@
The path to the shell. For example, /bin/sh.

@SISHostName@
The host name of the machine on which the depot is running.

@TmpDir@
The directory used for temporary file storage on this operating system. Typically, this is /tmp for UNIX operating systems and $DBDIR/tmp for Windows and OS/2 operating systems.

@TMP_DIR@
Same as @TmpDir@.

@TMRInterp@
The interpreter type of Tivoli server.

A complete list of variables can be found in the install repository.

Part 4. Resource and product installation

Chapter 16. Installing resources managed by Tivoli Management Framework

This chapter contains instructions for installing the following Tivoli Management Framework managed resources:

- Managed nodes
- Gateways
- Endpoints

For each of the previously listed resources, the following installation mechanisms are available:

- Tivoli Software Installation Service console
- Tivoli Software Installation Service commands
- Tivoli Management Framework commands
- Tivoli desktop

Notes:

1. To install any of the listed resources, that system must be reachable on the network.
2. When you create any of the listed resources on a Windows operating system, you need the password for the Administrator.
3. You cannot install a managed resource across Tivoli region boundaries.

Authorization roles for creating managed resources

The following table provides the context and authorization roles required for creating managed resources.

Activity	Context	Required role
Create a managed node	Tivoli region	**super** or **install_client**
Create a gateway	Tivoli region	**senior**
Install an endpoint	Machine	root or Administrator access
Create a RIM object	Tivoli region	**senior** or **super**

Creating managed nodes

After you install the Tivoli server, create managed nodes by installing Tivoli Management Framework.

You can create managed nodes individually or by using a list. If you specify a list of systems and any of them is unavailable, the installation process skips that system and proceeds to the next available. At a later time, you can perform the installation on any system that was skipped.

This section contains information that you need to know before installing a managed node, lists the Tivoli accounts created on the target system, explains the installation options available when creating a managed node, and describes the process of creating a managed node using the following mechanisms:

- Tivoli Software Installation Service
- Tivoli desktop
- Tivoli Management Framework commands

Available mechanisms for creating managed nodes

There are several ways to create managed nodes. The following table lists each installation method, provides a brief description of the method, and suggests when to use each method.

Method	Description	Situation
SIS	Creates managed nodes on multiple systems in parallel. For details, refer to "Creating managed nodes using Tivoli Software Installation Service" on page 201.	When you are uncertain of the status of the target systems and want to perform prerequisite checks, or when you want to create multiple managed nodes at one time
wclient command	Creates one or more managed nodes. For details, see "Creating managed nodes from the command line" on page 202.	When you have a small number of systems on which to create managed nodes and you do not want to perform prerequisite checks
wstandalone command	Creates a managed node without requiring that the Tivoli server initiate a connection to the local machine. For more information, see the *Tivoli Management Framework Reference Manual*.	When initiating a connection from the Tivoli server presents a security risk

You can create managed nodes on a Windows operating system using any installation method listed in this table. However, all of the installation methods except the **wstandalone** command require that Tivoli Remote Execution Service be installed on the Windows system.

Before creating a managed node

Before you create any managed node, consider the information in the following sections.

Network considerations

To create a managed node, the name of the target system must be in the /etc/hosts file (Linux or UNIX operating system), LMHOSTS file (Windows), the Network Information System (NIS) host map, or the name server.

Network security setting of the Tivoli server

The installation of a managed node is not supported when the network security setting on a Tivoli server is set to FORCE_SSL. Before you the create the managed node, you must temporarily set the network security setting on the Tivoli server to SSL using the **odadmin set_network_security** command. After the managed node has been created, you must set the security level of the managed node to SSL or FORCE_SSL. When both of these steps are complete, you can reset the network security setting on the Tivoli server to FORCE_SSL.

Enabling remote access

When you create a managed node, the Tivoli server or another managed node must create a connection to the remote system. To create this connection, the managed node performing the installation must have privileged access to the remote system. (An exception is when you use the **wstandalone** command to create a managed node.) The managed node performing the installation connects to the target system using one of the following methods, depending on the installation options you specify:

- trusted host access (**rsh** on Linux and UNIX operating system)
- account access (**rexec** on UNIX and Windows operating systems
- SSH (secure shell) on Linux, UNIX and Windows operating systems

Note: Generally, Linux operating systems do not support rexec. To use rexec on Linux operating systems, you might need to manually install and configure this service. For details, refer to your operating system documentation or contact your operating system provider.

For trusted host access, the /.rhosts file on the target machine must have an entry for the user **root** on the managed node performing the installation. This entry must be the user **root** and not another privileged user. The login shell of this user must be Bourne, Korn, or bash.

For account access on Windows operating system, specify the user name and password of a user in the Administrators group. For account access on UNIX operating system, specify the user name and password of a user with **root** authority. The login shell of this user must be Bourne, Korn, or bash. To install a managed node using Tivoli Management Framework, this user must have a password that is at least one character long. To install a managed node using Tivoli Software Installation Service, this user can have a zero-length password.

SSH can use both account and trusted host access methods. Account access methods require a password; trusted host methods do not. If the SSH client that is installed on the Tivoli server or managed node performing the installation requires a password, you must use an account access method with SSH. If the SSH client does not require a password, you must use a trusted host access method with SSH.

Special Considerations for Windows Operating Systems

If your Tivoli server is *not* a Windows operating system but you plan to create managed nodes or endpoints on Windows operating systems, you must manually install Tivoli Remote Execution Service on at least one of the target systems. If your Tivoli server is running Windows, this service is already available.

The first managed node that you install on a Windows operating system becomes the **CurrentNtRepeat** object from which other Window managed nodes are installed.

For information about Tivoli Remote Execution Service, refer to "Using Tivoli Remote Execution Service" on page 341.

Accounts Created during Managed Node Installation

During installation, Tivoli Management Framework creates the tmersrvd account. For additional information on accounts created during the installation of a Windows managed node, refer to "Accounts created during installation" on page 345.

Installation options for managed nodes

The following table provides the installation options that can be specified when you create a managed node.

Note: Do not specify directory names that contain a space, even on Windows operating system. For example, do not specify the **BIN** option as C:\Program Files\Tivoli. If you need to specify this directory, use the shortened format as C:\progra~1\Tivoli.

<table>
<tr><th rowspan="2"></th><th>GUI field name</th><th>CLI attribute</th></tr>
<tr><th colspan="2">Description</th></tr>
<tr><td rowspan="2">•</td><td>Policy Region</td><td><b>–p</b> <i>policy_region</i></td></tr>
<tr><td colspan="2">Specifies the name of the policy region where the managed node will be installed.
<b>Notes:</b>
• From the Tivoli desktop, creating managed nodes is performed from the Policy Region window.
• To create a managed node in a policy region other than the initial policy region, ensure that the ManagedNode resource is available in the target policy region. For details on setting resource types for a policy region, refer to <i>Tivoli Management Framework User’s Guide</i>.</td></tr>
<tr><td rowspan="2">•</td><td>Binaries</td><td><b>BIN</b>=<i>binaries</i></td></tr>
<tr><td colspan="2">Overrides the default installation path (/usr/local/Tivoli/bin) for the Tivoli Management Framework binaries.</td></tr>
<tr><td rowspan="2">•</td><td>Libraries</td><td><b>LIB</b>=<i>libraries</i></td></tr>
<tr><td colspan="2">Overrides the default installation path (/usr/local/Tivoli/lib) for the Tivoli Management Framework libraries.</td></tr>
<tr><td rowspan="2">•</td><td>Client Database</td><td><b>DB</b>=<i>client_database</i></td></tr>
<tr><td colspan="2">Overrides the default installation path (/var/spool/Tivoli) for the Tivoli Management Framework client database.</td></tr>
<tr><td rowspan="2">•</td><td>Man Pages</td><td><b>MAN</b>=<i>manpages</i></td></tr>
<tr><td colspan="2">Overrides the default installation path (<b>/usr/local/Tivoli/man</b>) for the Tivoli Management Framework man pages. This option is required only for Linux and UNIX operating system.</td></tr>
<tr><td rowspan="2">•</td><td>X11 Resource Files</td><td><b>APPD</b>=<i>X11_defaults</i></td></tr>
<tr><td colspan="2">Overrides the default installation path (/usr/lib/X11/app-defaults) for the X11 application defaults. This option is available only for UNIX operating system.</td></tr>
<tr><td rowspan="2">•</td><td>Message Catalogs</td><td><b>CAT</b>=<i>message_catalog</i></td></tr>
<tr><td colspan="2">Overrides the default installation path (/usr/local/Tivoli/msg_cat) for the Tivoli Management Framework message catalogs.</td></tr>
<tr><td rowspan="2">•</td><td>Arrange for start of the Tivoli daemon at system (re)boot time</td><td><b>@AutoStart@=1 | 0</b></td></tr>
<tr><td colspan="2">Indicates whether the object dispatcher (oserv) should be started (1) at system boot time. By default, the oserv is not started (0).</td></tr>
<tr><td rowspan="2">•</td><td>Configure remote start capability of the Tivoli daemon</td><td><b>@SetPort@=1 | 0</b></td></tr>
<tr><td colspan="2">Indicates whether to configure (1) or not configure (0) the remote start capability of the oserv daemon. Enabling remote start changes /etc/inetd.conf and adds the port number to /etc/services. These changes enable you to start the oserv on this managed node from another managed node. By default, this capability is configured.</td></tr>
</table>

	GUI field name	CLI attribute
	Description	
•	When installing, create "Specified Directories" if missing	**@CreatePaths@=0 | 1**
	Indicates whether the directories should be created (1) or not created (0) if they do not already exist. The default is to create them.	
•	Not available from the GUI.	**@ClientAddNoTrans@=yes | no**
	Indicates whether managed nodes should be added using a transaction. Adding managed nodes without a transaction can save significant time when installing over slow links. If an error occurs, however, run the **wchkdb** command to verify the state of the database.	

Creating managed nodes using Tivoli Software Installation Service

To create a managed node using Tivoli Software Installation Service, perform the follow high-level steps:

1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113.
2. Complete the installation worksheet as follows:
   a. Select the Tivoli Management Framework product from the install repository:
      - If necessary, import the product into the install repository, as described in “Importing products using the Tivoli Software Installation Service console” on page 115.
      - Add the Tivoli Management Framework product to the installation worksheet as described in “Adding products to the installation worksheet” on page 127.
      - If desired, specify default values for the installation options for managed nodes as described in “Viewing and customizing installation options” on page 119.

   b. Specify the machines on which to install managed nodes:
      - Make sure that the machines on which you want to install managed nodes are already known to Tivoli Software Installation Service. If necessary, define the new machines as described in “Adding machines to the Tivoli Software Installation Service depot” on page 130.
      - Add the machines to the installation worksheet as described in “Adding machines to the installation worksheet” on page 134.

   c. Mark the installation worksheet to indicate on which machines to install Tivoli Management Framework. You may mark other products to be installed at the same time. This is described in “Specifying products to install” on page 138.

   d. If desired, modify the installation options for Tivoli Management Framework on a specific machine. This is described in “Overriding default installation options for one machine” on page 137.
3. Start the installation as described in “Using the Tivoli Software Installation Service console to install products” on page 139.
4. Perform post-installation work required for Windows managed nodes:

- If you did not allow Tivoli Software Installation Service to reboot a Windows operating system after creating a managed node, reboot it manually to complete the installation.
- Run the **wmailhost** command to configure the SMTP mail service. For additional information about configuring the mail service on Windows operating system, refer to Chapter 7, “Configuring for SMTP e-mail.”
- You can resolve many network connectivity issues related to the Tivoli desktop by setting the DESKTOPS_CHECK_TIMEOUT environment variable. For more information, see the section about the Tivoli desktop in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

A command line interface to Tivoli Software Installation Service is available as well. For more information, refer to “Installation overview from the command line” on page 112.

Creating managed nodes from the command line

To create one or more managed nodes from the command line, use the **wclient** command. The following example creates managed nodes cook and everest in the austin-pr policy region:

```
wclient -c F:\cdrom -d -I -p austin-pr cook everest
```

In the previous example, the options are as follows:

–c F:\cdrom
Specifies the path to the installation image.

–d
Specifies that this installation uses the variable values from the previous installation.

–I
Specifies that the installation password prompt is displayed.

–p austin-pr
Specifies that the managed nodes are for the austin-pr policy region.

cook everest
Specifies the host names of the new managed nodes.

After creating a managed node on a Windows operating system, you must do the following:

- Reboot the system to complete the installation.
- Run the **wmailhost** command to configure the SMTP mail service. For additional information about configuring the mail service on Windows operating system, refer to Chapter 7, “Configuring for SMTP e-mail,” on page 61.

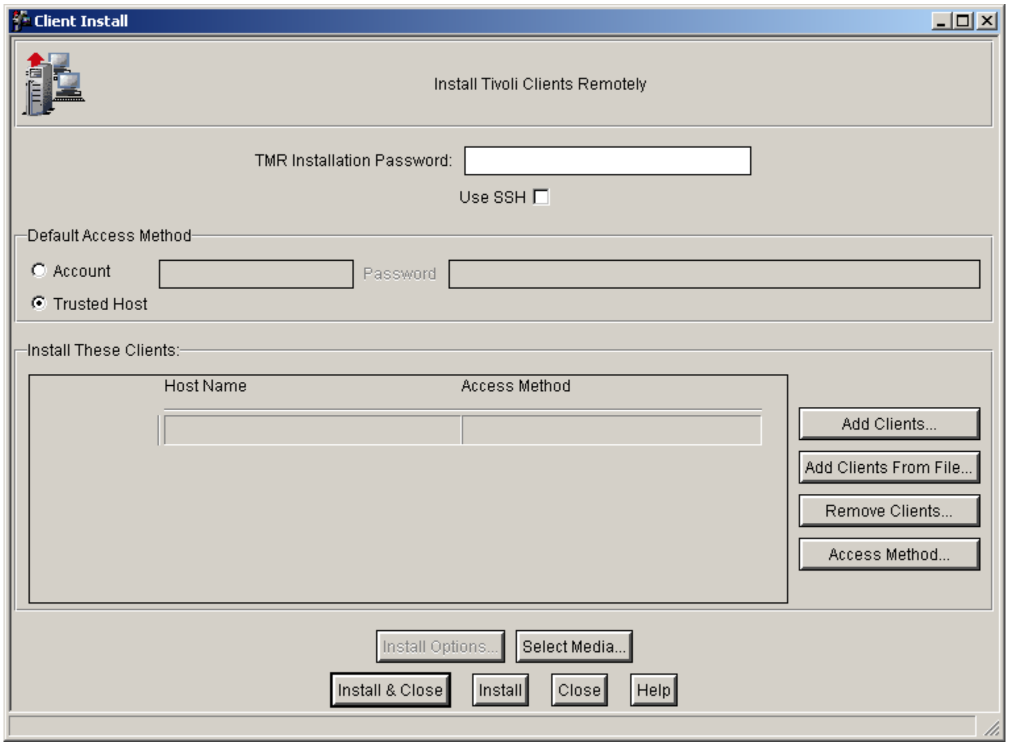
For detailed information about using the **wclient** command to create a managed node, refer to *Tivoli Management Framework Reference Manual*.

You can resolve many network connectivity issues related to the Tivoli desktop by setting the DESKTOPS_CHECK_TIMEOUT environment variable. For more information, see the section about the Tivoli desktop in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Creating managed nodes from the Tivoli desktop

To create a managed node on one or more systems, perform the following steps:

1. From within a policy region, select **Create → Managed Node**. The Client Install window is displayed.



2. In the **TMR Installation Password** field, type the installation password. If a password was not specified when the Tivoli server was created, leave this field empty. If a password was specified, you must use this password for each managed node.
3. Click the SSH check box to use a secure shell (SSH) connection from the Tivoli server to the system on which the managed node will be created. When you select this option, all data transmitted over the connection is encrypted.

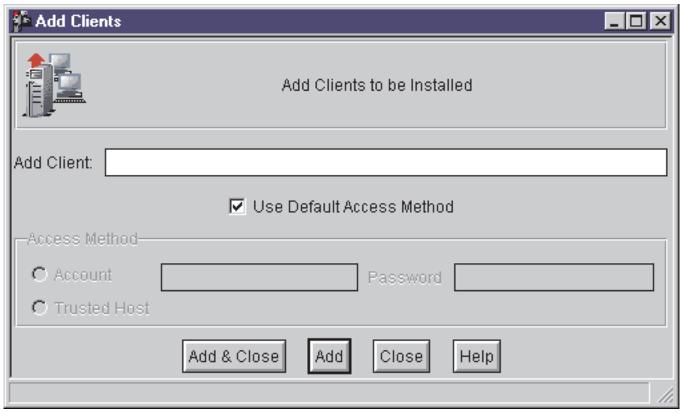
   Note: You cannot use SSH to perform installations on Windows targets.
4. In the **Default Access Method** group box, select the radio button to indicate how each system should be contacted:

   Account
   : In the **Account** field, type the name and password of the privileged user to contact the system. In the **Password** field, type the password for this user. This option applies to UNIX and Windows operating systems.

   Trusted Host
   : For trusted host access, the /.rhost file on each system must grant access to the system performing the installation by adding and entry for `root@`*`hostname`*, where *hostname* is the host name of the Tivoli server or managed node performing the installation. This option is applicable only for Linux and UNIX operating systems.

   Note: You can override the access method when you add a single system, load systems from a file, or after the systems are added to the list.
5. To add a single system, perform this step. To load multiple systems from a text file, continue with step 6.

To add a single system, click **Add Clients**. The Add Clients window is displayed.



Add Clients

Add Clients to be Installed

Add Client:

☑ Use Default Access Method

Access Method

○ Account Password

○ Trusted Host

In this window, perform the following steps:

a. In the **Add Client** field, type the name of the system you want to make a managed node.

b. Specify which access method the Tivoli server will use to contact the system:

- Set the **Use Default Access Method** check box to use the default access method that you specified in step 4.
- Set the **Account** radio button to use account access instead of the default access method. For details on appropriate values, refer to Account on page 203.
- Set the **Trusted Host Access** radio button to use trusted host access instead of the default access method. For details, refer to Trusted Host on page 203.

c. Click **Add & Close** to add the system to the list and return to the Client Install window.

Repeat this step for each machine that you want to add.

6. To add multiple systems from a text file, perform this step. If you do not need to add systems from a file, continue with step 7 on page 205.

   a. Create a text file containing the systems. This file contains one line per system using the *hostname*,*user*,*password* format. If *user* is not specified, the default is root or Administrator. The content of each line determines the access method.

   - For the default, each entry contains only the system name.
   - For account access, each entry contains the host name followed by a comma (,) the user ID followed by a comma (,), followed by the password.

     Note: Passwords are not encrypted. Anyone with access to this file can see the passwords.
   - For trusted host access, each entry contains the host name followed by a comma (,).

   The following shows an example of a text file containing all of these formats:

```
undead
vampire,lefanu,paris
werewolf,
```

In this example, undead is added using the default access method. vampire is added using account access and the user name of lefanu and a password of paris. werewolf is added using trusted host access.

Note: You can add the majority of systems from a text file and then use the **Add Clients** button to add systems not listed in the file.

b. Load the list of systems from a text file by clicking **Add Clients From File**. A file browser is displayed.

Within the file browser, navigate to the correct file and click **Set File & Close** to return to the Client Install window.

Note: When installing from a remote drive on a Windows operating system, your Tivoli remote access account must have access to this drive.

c. If you need to add systems not in the file, go to step 5.

7. If you need to remove a system from the list, perform this step. If systems listed in the **Install these Clients** list is correct, continue with the next step.

   To remove systems from the **Install these Clients** list:

   a. Select one or more managed nodes to remove.

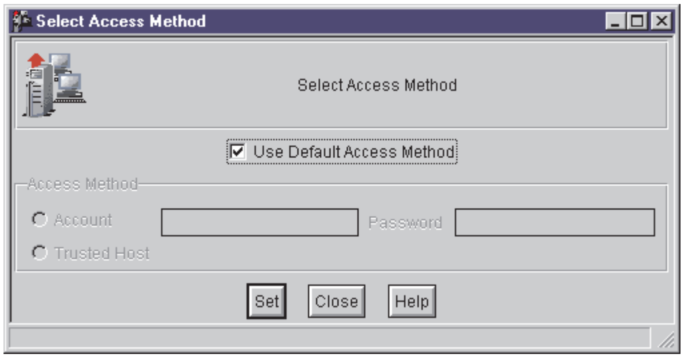
   b. Click **Remove Clients**. The selected systems are removed from the list.

8. If you need to change the access method for any of the listed systems, perform this step. If the access method for all the listed systems is correct, continue with the next step.

   To change the access method for a listed system:

   a. Select the system to change.

   b. Click **Access Method**. The Select Access Method window is displayed.



☑ Use Default Access Method

   c. Modify the access method as needed.

   d. Click **Set** to return to the Client Install window.

9. Click **Select Media** to specify the path to the directory containing the installation images. The File Browser window is displayed.

10. Use the File Browser window to navigate to the correct file, then click **Set Media & Close**. The Install Options window is displayed.

    Refer to the *Tivoli Management Framework User's Guide* for instructions on how to use the File Browser window.

Note: Generally, the Tivoli server and managed nodes are installed from the same image.

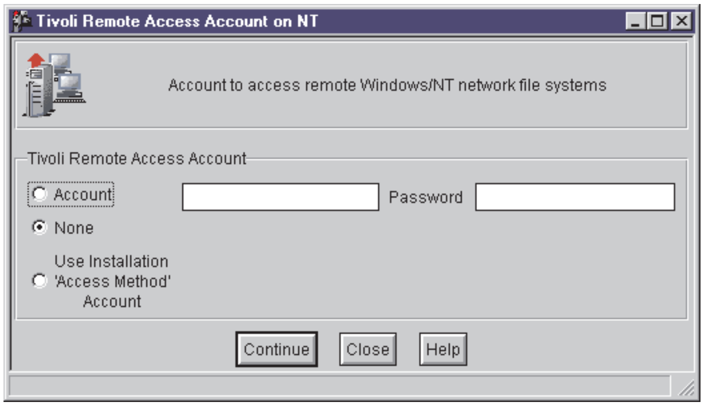
11. In the Install Options window, perform the following:
    a. Modify the installation options as necessary. The options in this window apply to all listed systems.

       For details on the fields available in this window, refer to "Installation options for managed nodes" on page 200.

       Note: To reinstall a managed node, set the value for the path installation options to an exclamation mark (!), which forces the files to be overwritten. If you are reinstalling, you *must* overwrite the **Client Database** field or the reinstalled managed node will not work properly.

    b. Click **Set** to save your changes and return to the Client Install window.
12. Click **Install & Close** to start the installation process for the managed nodes. The Client Install window is displayed. This window lists operations that will take place during the installation and any problems that you might want to correct before continuing the installation.
13. Decide whether to cancel or continue the installation based on the messages in the Client Install window.
    - If there are problems that you need to correct before continuing the installation, click **Cancel**.
    - To perform the installation, click **Continue Install**.

      After you click this button, the **Cancel** button changes to the **Close** button. If any of the managed nodes will be installed on Windows operating system, the Tivoli Remote Access Account window is displayed.



Tivoli Remote Access Account on NT

Account to access remote Windows/NT network file systems

Tivoli Remote Access Account

○ Account
● None
○ Use Installation 'Access Method' Account

Password

In this window, perform the following steps:

a. Select the appropriate radio button.

Account
Defines an account other than the default access account.

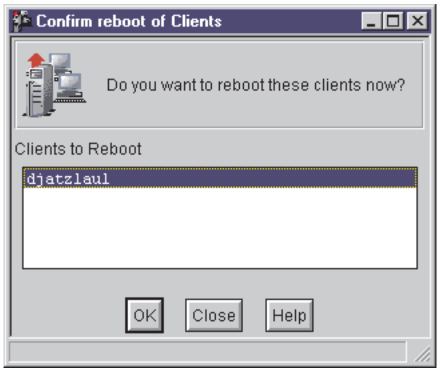
None
Specifies that there is no Tivoli remote access account. If reinstalling and there is a previous Tivoli remote access account, this option does not overwrite the original account.

Use Installation 'Access Method' Account
Specifies that Tivoli remote access account will be the default access account. This selection grants the full rights that the default access account has.

For more information about Tivoli remote access account, refer to “Tivoli Remote Execution Service” on page 340.

b. Click **Continue**.

The Client Install window returns status information as the installation proceeds. When the installation is almost complete, the Confirm Reboot of Clients window is displayed. This window lists the Windows operating system that need to be rebooted.


Confirm reboot of Clients
Do you want to reboot these clients now?
Clients to Reboot
djatzlaul
OK
Close
Help

Click **OK** to reboot these systems or click **Close** to reboot later. You need to reboot these systems for Tivoli Management Framework to work. If you click **Close**, you must reboot these systems manually.

14. When the installation completes, the Client Install window ends with the following completion message:

```
Finished client install
```

15. Click **Close**. The Client Install window is closed.

    The newly installed managed nodes should now be displayed in the policy region.

16. Configure each Windows managed node for Simple Mail Transfer Protocol (SMTP) e-mail. On each of these managed nodes, enter the following command:

    `wmailhost` *`hostname`*

    where *hostname* is the name of the network mail server. For additional information, refer to Chapter 7, “Configuring for SMTP e-mail,” on page 61.

Creating gateways

Note: NetWare and OS/2 gateways are hosted on ManagedNode objects. Resources that require a managed node cannot be installed or created on these systems unless explicitly stated.

You can create a gateway on an existing managed node or on the Tivoli server. This includes NetWare and OS/2 managed nodes. The gateway is designed primarily to provide access to methods that must be passed to the endpoint. The gateway also provides the necessary communication that endpoints occasionally require with the Tivoli server.

Notes:

- Before installing a gateway on an OS/2 system, read the information in “Using OS/2 systems” on page 347.

- Before installing a gateway on a NetWare system, read the information in "Using NetWare systems" on page 351.

When you create a gateway, the httpd service is disabled by default. To access gateway information from a browser, you need to use the **wgateway** command to define the access account and password and enable the gateway to accept http request. For information about enabling and disabling http request and for information about setting the access account, refer to *Tivoli Management Framework Reference Manual*.

Installation options for gateways

The following table provides the installation options that can be specified when you create a gateway.

	GUI field name	**CLI attribute**
	Description	
•	Name	**–n** *gateway_name*
	Specifies the name of the gateway. If this option is not used, the gateway is named *managed_node_name*-gateway.	
•	Port	**–p** *TCPIP_port*
	Specifies the TCP port on which the gateway will communicate with its assigned endpoints. The default TCP port is 9494.	
•	Socket	**–i** *IPX_socket*
	Specifies the IPX port on which the gateway will communicate with its assigned endpoints. The SPX port is always assigned *IPX_socket***–1**. The default IPX port is 9494, and the default SPX port is 9493.	
•	Protocol List	**–P** *protocol_list*
	Specifies which protocols are supported by the gateway. Valid values are TCPIP and IPX. To specify both TCP/IP and IPX protocols, specify this option as `-P TCPIP,IPX`. If you do not specify TCPIP, it is automatically supported. The TCP/IP protocol cannot be removed.	
•	Managed Node Proxy	**–h** *managed_node*
	Specifies the name of the managed node on which the gateway is created. If this option is not used, the gateway is created on the managed node where the command was run.	
•	Not available from the GUI.	**–p** *default_session_timeout*
	Specifies how long the gateway waits for a response from the endpoint.	

Note: Tivoli recommends that all gateways in a Tivoli region use the same port number. The TCP/IP port number can be the same as the IPX port number.

The following sections provide instructions for creating a gateway using the following methods:

- Tivoli Software Installation Service
- The Tivoli desktop
- The **wcrtgate** command

Creating a gateway while BDT is enabled

Generally, gateways are created before bulk data transfer (BDT) is enabled on the Tivoli server or managed nodes. If BDT is enabled (**Single Port BDT=TRUE**) and you try to create a gateway on one of these resource, the creation appears to hang. Therefore, to create a gateway after BDT is enabled, perform the following steps:

1. Change the BDT port for this resource using the **odamin set_bdt_port** command.
2. Create the gateway using the **wcrtgate** command.

Alternatively, to create a gateway after BDT is enabled, perform the following steps:

1. Temporarily disable BDT for this resource using the **odadmin single_port_bdt** command.
2. Create the gateway using the **wcrtgate** command.
3. Re-enable BDT for this resource using the **odadmin single_port_bdt** command.

If you attempted to create a gateway and did not disable BDT or change the BDT port, the creation of the gateway appears to hang. Before cancelling this process, check the gateway log ($DBDIR/gatelog) to see whether the gateway process started. In the gatelog file, look for the following message:

```
gateway boot: BDT Service started.
```

If the file contains this message, the gateway was successfully created. You can cancel the process and recycle to object dispatcher on this resource using the **odadmin reexec** command.

For information about the **odadmin** command, refer to *Tivoli Management Framework Reference Manual*. For information about creating a gateway using the **wcrtgate** command, refer to “Creating a gateway from the command line” on page 211.

Creating a gateway using Tivoli Software Installation Service

To create a gateway using Tivoli Software Installation Service, perform the follow high-level steps:

1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113.
2. Complete the installation worksheet as follows:
   a. Select the Tivoli Gateway product from the install repository:
      - Add the Tivoli Gateway product to the installation worksheet as described in “Adding products to the installation worksheet” on page 127.
      - If desired, specify default values for the installation options for gateways as described in “Viewing and customizing installation options” on page 119.

   b. Specify the machines on which to install the gateway:
      - Make sure that the machines on which you want to install the gateway are already known to Tivoli Software Installation Service. If necessary, define the new machines as described in “Adding machines to the Tivoli Software Installation Service depot” on page 130.
      - Add the machines to the installation worksheet as described in “Adding machines to the installation worksheet” on page 134.

   c. Mark the installation worksheet to indicate on which machines to install the gateway. You may mark other products to be installed at the same time. This is described in “Specifying products to install” on page 138.

d. If desired, modify the gateway installation options on a specific machine. This is described in "Overriding default installation options for one machine" on page 137.

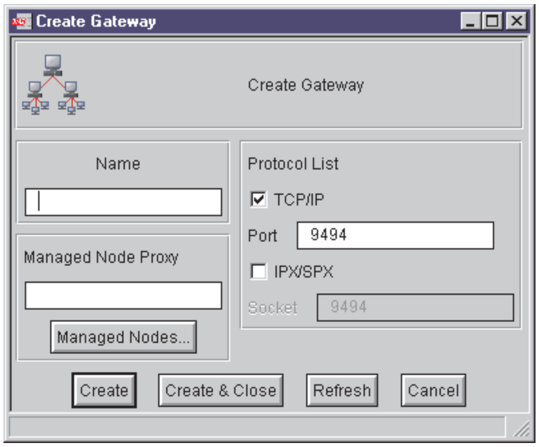
3. Start the installation as described in "Using the Tivoli Software Installation Service console to install products" on page 139.

A command line interface to Tivoli Software Installation Service is available as well. For more information, refer to "Installation overview from the command line" on page 112.

Creating a gateway from the Tivoli desktop

To create a gateway from the Tivoli desktop, perform the following procedure:

1. Start the Tivoli desktop.
2. Right-click the **EndpointManager** icon and then select **Create Gateway**. The Create Gateway window is displayed.



3. In the **Name** field, type the name of the gateway. This name will be the label for the gateway icon that will be added to the Tivoli desktop.
4. In the **Port** field, type the port number through which the gateway will communicate with its endpoints in TCP/IP.

   Note: For NetWare gateways that need IPX support, perform the following steps:

   a. Set the **IPX/SPX** check box.
   b. In the **Socket** field, type the port number through which the gateway will communicate with its endpoints in IPX/SPX.

   All gateways in a Tivoli region should use the same port number. The default port number for TCP/IP and IPX communication is 9494, and the default port number for SPX communication is 9493.
5. In the **Managed Node Proxy** field, type the name of the managed node on which this gateway will reside. If you do not know the label for the managed node, perform the following steps:

   a. Click **Managed Nodes**. The Available Managed Nodes window is displayed.
   b. In the Available Managed Nodes window, select a name from the list.
   c. Click **Select & Close** to return to the Create Gateway window.

6. If you are creating multiple gateways, click **Create** to create a gateway and clear this window. Repeat steps 2 through 5 for each gateway.

   If this is your last or only gateway, click **Create & Close** to create the gateway and return to the Tivoli desktop.

Creating a gateway from the command line

To create a gateway from the command line, use the **wcrtgate** command. The following command creates a gateway on managed node vernon using the default settings.

```
wcrtgate -h vernon
```

After creating the gateway, you can tune its repeater functions using the **wmcast**, **wmdist**, and **wrpt** commands. For information about using these commands, refer to *Tivoli Management Framework Reference Manual*. For information about tuning gateways and repeaters, refer to *Tivoli Management Framework User's Guide*.

Installing endpoints

Before you install endpoints, you need to have at least one gateway in the Tivoli region. For instructions on creating gateways, refer to "Creating gateways" on page 207.

Note: When you have a gateway that supports IPX/SPX protocol, endpoints connecting to this gateway can use either IPX or TCP/IP.

If you install an endpoint with broadcast mode disabled, you must assign a gateway to the endpoint. For example, you can specify a gateway during endpoint creation using the **winstlcf –g** command. Or, you can specify a gateway after creating the endpoint using the **lcfd –g** command.

For more information about broadcast mode, see the *Tivoli Management Framework Planning for Deployment Guide*. For more information about Tivoli commands, see the *Tivoli Management Framework Reference Manual*.

Accounts created during endpoint installation

When you install an endpoint, the tmersrvd account should be created, if not already on this computer system. This account is used on HP-UX and Windows operating systems. This account have limited access and allow methods with the ID of nobody to execute. For Windows operating systems, this account requires the Log on locally privilege, but no special group membership.

Ensure that the /etc/passwd file on the system has one of the following accounts defined:

For AIX operating systems

```
nobody:*:4294967294:4294967294::/:
```

For HP-UX operating systems

```
tmersrvd:*:59999:59999:Reserved forTME:/:/bin/false
```

For Solaris operating environments

```
nobody:*:60001:60001::/:
```

When this account does not exist, Tivoli Management Framework randomly selects an account ID under which it runs required operations. As a rule, Tivoli

Management Framework selects an account ID high enough so that conflicts do not occur. To ensure that these conflicts do not arise, add the appropriate line to the /etc/password file.

Note: The tmersrvd is required by Tivoli Management Framework. If you disable or delete this account, Tivoli Management Framework will not operate as expected.

Endpoint installation setup procedure

For Windows operating systems, if you are using the Tivoli remote access account, the Act as part of the operating system right must be granted to the current user. To grant this right, perform the following steps:

1. From the **Start** menu, select **Programs → Administrative Tools (Common) → User Manager** to display the User Manager window.
2. From the **Polices** menu, select **User Rights** to display the User Rights Policy window.
3. In this window, perform the following:
   a. Select the **Show Advanced User Rights** check box.
   b. From the **Rights** drop down list, select **Act as part of the operating system**.
   c. Click **Add** to display the Add Users and Groups window.
4. In this window, perform the following:
   a. Click **Show Users**.
   b. From the **Names** list, select the user you are currently logged in with.
   c. Click **Add** to add the user to the **Add Names** list
   d. Click **OK** to close the window and return to the Add Users and Groups window.
5. Click **OK**. The new user now has the Act as part of the operating system right.
6. Log off of the operating system and log in again as the same user.
7. Run the endpoint setup procedure and provide the appropriate Tivoli remote access account user name and password.

Note: For additional information about adding, deleting, or modify user rights, consult your operating system documentation.

Authorization for installing endpoints

Installing an endpoint does not require a Tivoli authorization role. However, the following operating systems require the specified authorization for installing an endpoint:

Windows NT, Windows 2000, and Windows XP
Access as a member of the Administrator group.

Linux and UNIX
Access as root.

NetWare
The appropriate NetWare context as Admin (for NDS without bindery) or supervisor (for NDS with bindery). For additional information about NetWare accounts used within a Tivoli environment, refer to “NetWare accounts” on page 351.

Available mechanisms for installing endpoints

There are several ways to install endpoints. The following table lists each installation method, provides a brief description of the method, and suggests when to use each method.

Method	Description	Situation
SIS	Installs and starts multiple endpoints on multiple systems in parallel. For details, refer to "Installing endpoints using Tivoli Software Installation Service."	When you are uncertain of the status of the target systems and want to perform prerequisite checks, or when you want to install multiple endpoints at one time
winstlcf command	Installs and starts an endpoint on supported Linux, NetWare, Windows, and UNIX workstations. For details, refer to "Installing endpoints from the command line" on page 214.	When you have a small number of systems on which to install endpoints and you do not want to perform prerequisite checks
InstallShield or OS/2 Installer	Locally installs and starts an endpoint on supported systems. For details, refer to "Installing endpoints using InstallShield" on page 215 and "Installing endpoints using OS/2 Installer" on page 218.	When SIS or the **winstlcf** command does not work
Logon script	Installs and starts an endpoint on systems when it logs in to the Windows domain or NetWare server. For details, refer to "Installing endpoints using logon scripts" on page 219.	When you want to add new systems to the network automatically

You can install endpoints on a Windows operating system using any method listed in this table. For any Windows endpoint to work in a Tivoli region, that system must have Tivoli Remote Execution Service installed. When you use the **winstlcf** command to install Windows endpoints, at least one endpoint must be manually installed to serve as the proxy for other Windows endpoints. When you use Tivoli Software Installation Service to install endpoints, at least one Windows managed node must be available to serve as the Windows repeater (that is, the CurrentNtRepeat object).

Depending on which installation mechanism you choose when installing endpoints, the destination directory differs. When you install an endpoint using the InstallShield image on a Windows operating system, the destination directory is %SystemRoot%\Program Files\Tivoli\lcf. When you install an endpoint using any other installation mechanism or using the InstallShield image on another operating system, the destination directory is the one shown in "Default directory structure" on page 392. Independent of how you install the endpoint, you can change the destination directory.

Installing endpoints using Tivoli Software Installation Service

Before you can use Tivoli Software Installation Service to install an endpoint on any Windows NT, Windows 2000, or Windows XP system, there must be at least one Windows managed node in the Tivoli region. When you install a managed node on a Windows system, this system becomes a repeater. For additional information, refer to "Windows repeaters" on page 342.

Note: If there are no Windows repeaters in the Tivoli region, you cannot use Tivoli Software Installation Service to install endpoints on a Windows system. If you do not want to install a Windows managed node, you must use another mechanism to install endpoints on Windows machines.

To create an endpoint using Tivoli Software Installation Service, perform the follow high-level steps:

1. Start the console as described in "Starting the Tivoli Software Installation Service console" on page 113.
2. Complete the installation worksheet as follows:
   a. Select the Tivoli Endpoint product from the install repository:
      - Add the Tivoli Management Framework product to the installation worksheetas described in "Adding products to the installation worksheet" on page 127.
      - If desired, specify default values for the installation options for endpoints as described in "Viewing and customizing installation options" on page 119.

   b. Specify the machines on which to install the endpoint:
      - Make sure that the machines on which you want to install the endpoint are already known to Tivoli Software Installation Service. If necessary, define the new machines as described in "Adding machines to the Tivoli Software Installation Service depot" on page 130.
      - Add the machines to the installation worksheet as described in "Adding machines to the installation worksheet" on page 134.

   c. Mark the installation worksheet to indicate on which machines to install endpoints. You may mark other products to be installed at the same time. This is described in "Specifying products to install" on page 138.

   d. If desired, modify the endpoint installation options on a specific machine. This is described in "Overriding default installation options for one machine" on page 137.
3. Start the installation as described in "Using the Tivoli Software Installation Service console to install products" on page 139.

A command line interface to Tivoli Software Installation Service is available as well. For more information, refer to "Installation overview from the command line" on page 112.

Installing endpoints from the command line

To install endpoints from the command line, use the **winstlcf** command. This command allows you to remotely install endpoints on all supported operating systems except AS/400® and OS/2. With **winstlcf**, you can install multiple endpoints either by specifying system names on the command line or by using a text file as input.

Notes:

- To install Windows endpoints using the **winstlcf** command, you must first install one Windows endpoint in the domain or trust manually using the InstallShield image. This endpoint is used as a proxy to remotely install all additional Windows endpoints in the domain or trust. After you have installed the proxy endpoint, install all additional Windows endpoints in that domain or trust using the **winstlcf** command and **–N** option.

- Before installing Linux endpoints, make sure that the the access method used (exec or shell) is enabled on the target. The exec access method is the default installation option; the shell access method is specified using the **–e** option.
- To use an encrypted secure shell (SSH) connection when installing an endpoint, specify the **–j** option. For more information about using this option, see the *Tivoli Management Framework Reference Manual*.

The following example installs an endpoint on a UNIX operating system named vernon using default settings:

```
winstlcf vernon
```

The following example installs an endpoint on managed node satellite using vernon as the installation source. The endpoint will use gateway LUX using port 9999 and communicate in IPX:

```
winstlcf -N vernon -x IPX -g LUX+9999 satellite
```

Note: The NetWare server name for the gateway must always be in uppercase for the installation to be successful.

The following example installs an endpoint on a Windows NT system named garcia, using singh as a proxy.

```
winstlcf -N singh garcia
```

The following example installs endpoints (using default settings) on the systems listed in the endpoint.txt file. The **–P** option enables you to specify a global password that applies to all systems.

```
winstlcf -f endpoint.txt -P
```

For detailed information about the **winstlcf** command, refer to the *Tivoli Management Framework Reference Manual*.

Installing endpoints using InstallShield

You can use InstallShield to install an endpoint on a local Windows operating system or mapped NetWare operating system. When using the InstallShield locally, you can perform the installation in either unattended (silent) mode or in interactive mode. If you want to create a response file for the InstallShield installation, refer to the information in Chapter 19, “Creating InstallShield response files,” on page 253.

Before installing the endpoint on a Windows operating system, ensure that the Act as part of the operating system user right is granted to the Administrator. For detailed information, refer to “Tivoli Remote Execution Service” on page 340.

Note: Tivoli does not recommend installing an endpoint on a Windows operating system in a file allocation table (FAT) file system. If you do this, you are allowing anyone to run any program as $root_user on this system.

Using InstallShield in unattended mode

You can install endpoints in unattended mode to Windows operating system using InstallShield. To install in unattended mode, perform the following steps:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. From the taskbar, click **Start**, and then select **Run** to display the Run window.
3. In the **Open** field, type the following command:

   ```
   x:\directory\setup rebootok tapuser=login tappass=password -s icons
   ```

where:

x:\directory\\setup
: Specifies the directory where the installation image for the supported operating system is.

rebootok
: Reboots the system after installation, if required.

tapuser=*login*
: Specifies the Tivoli remote access account.

tappass=*password*
: Specifies the password for the Tivoli remote access account.

–s
: Installs the endpoint in silent mode, without displaying the setup windows. This option must follow the **tapuser** and **tappass** options.

icons
: If you specify this option, Tivoli options are added to the **Start** menu.

Note: When you install in unattended mode, the destination directory for the endpoint is %SystemRoot%\Program Files\Tivoli\lcf.

Using InstallShield in interactive mode

You can use InstallShield in interactive mode to install endpoints as follows:

- Locally on Windows operating system
- Remotely to a NetWare server from a Windows operating system

To install an endpoint, perform the following steps:

1. If installing directly on a Windows operating system, continue to step 2. If installing to a remote NetWare, perform the following steps:
   a. On the local Windows operating system, open its Explorer program.
   b. From Explorer, select the **Tools → Map Network Drive**. The Map Network Drive window is displayed.
   c. In the **Drive** field, select the drive that will represent the networked system. By default, this field contains the next available logical drive letter.
   d. In the **Path** field, type the network path using the universal naming convention (UNC) format. For example:
      ```
      \\NetWare_server_name\SYS_volume_name
      ```
      Note: You must have Admin or supervisor permissions on the remote system.
   e. In the **Connect As** field, type your login name. The remote system might prompt you for a password.

   For more information about mounting a remote drive, refer to the documentation for the operating system.
2. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive or insert the installation diskette into the drive.
3. From the taskbar, click **Start**, and then select **Run** to display the Run window.
4. In the **Open** field, type the following command:
   ```
   x:\directory\setup icons
   ```
   where:

   x:\directory\\setup
   : Specifies the directory where the installation image for the supported operating system is.

icons Adds endpoint icons to the desktop, the menu, or both, depending on the operating system. By default, icons are not created.

5. Click **OK** to run the setup program and display the welcome window. This program leads you through the endpoint installation process, displaying option and information windows that provide you with the necessary information. The **Back**, **Next**, and **Cancel** buttons allow you to navigate through these windows.
6. Click **Next**. The License Agreement window is displayed.
7. Read the agreement and click **Yes**.

Depending on your operating system, perform one of the following windows is displayed:

- (For Windows operating systems only) The Accounts and File Permissions window is displayed. Type the name and password of the account through which Tivoli Management Framework will access remote file systems. If you do not need remote access, click **Next** to skip this step.
- (For NetWare operating systems only) You are asked whether you need to map to the NetWare system.
  - If you do not have a current map to the NetWare system, click **Yes** and map the drive.
  - If you have a current map to the NetWare system, click **No**.
- The Select NetWare Mapped Drive window is displayed where you can select the mapped drive.

8. Click **Next**. The Choose Destination Folder window is displayed.
9. In this window, perform the following steps:
   a. Make sure the endpoint component is selected.
   b. If you do not want to use the default directory, click **Browse** to change the destination directory.
10. (For Windows operating system only) Click **Next**. The Remote Access Information window is displayed.

    Type the name and password of the account through which Tivoli Management Framework will access remote file systems. If you do not need remote access, click **Next** to skip this step.
11. Click **Next**. The Advanced Configuration window is displayed. Use this window to specify startup and configuration options.

Field name	**Use when ...**
Gateway port	The gateway uses a port number other than the default 9494.
Endpoint port	You want to specify a port for the endpoint to use. The default is 9495.
Options	You want to specify other lcfd configuration options. For example, use the **–g** *hostname* option to specify the intercepting gateway. Refer to the **lcfd** command in the *Tivoli Management Framework Reference Manual* for a list of valid options.

12. Click **Next** to continue the installation process.
13. (For NetWare server only) The Directory Services context window is displayed.

    Type the NDS name for the context in which the system resides. Leave the field empty if working in Bindery emulation.

Note: If you are installing from a Windows operating system with Novell Requester running, the account is created automatically. If not, you must manually create this account using the Tivoli-supplied **addadmin** utility. For additional information, refer to "NetWare accounts" on page 351.

14. Click **Next**. When installation is completed, a window stating this fact is displayed.
15. Click **Finish**.
16. Repeat this procedure as appropriate.

The endpoint is installed and logged in to a gateway. The installation process configures the endpoint system to automatically start the endpoint service when the system boots. The technique depends on the platform type:

Windows NT, Windows 2000, or Windows XP
Adds the Tivoli Endpoint as a service with the startup type set to automatic.

NetWare
Modifies the AUTOEXEC.NCF file to call the LCF.NCF startup file.

Installing endpoints using OS/2 Installer

You can use the OS/2 Installer to install endpoints on OS/2 systems in unattended (or silent) mode or in interactive mode.

Installing OS/2 endpoints in unattended mode

To install an endpoint on an OS/2 system in unattended mode, perform the following steps:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. From the DOS command prompt, enter the following command to change to the directory on the CD that contains the OS/2 image:

   ```
   cd x:\lcf\os2\cdrom
   ```

   where *x* is the CD-ROM drive.
3. Enter the **install** command with options from the command prompt. The following command installs the endpoint without displaying the installation windows:

   ```
   INSTALL /X /R:C:\LCFSAMP.RSP /L1:C:\INSTERR.LOG
    /L2:C:\INSTHIST.LOG
   ```

   where:

   /X Indicates that the installation is in unattended mode.

   /R:C:\LCFSAMP.RSP
   Specifies the absolute path to the response file. This response file contains the default installation information for OS/2 endpoints, and it is on the Tivoli Management Framework 2 of 2 CD.

   /L1:C:\INSTERR.LOG
   Specifies the absolute path to the error log file.

   /L2:C:\INSTHIST.LOG
   Specifies the absolute path to the installation history log.

Installing OS/2 Endpoints in interactive mode

To install an endpoint on an OS/2 system in interactive mode, perform the following steps:

1. Insert the Tivoli Management Framework 2 of 2 CD into the system CD-ROM drive.
2. From a DOS command prompt, enter the following commands:

   ```
   cd x:\lcf\os2\cdrom
   install
   ```

   where *x* is the CD-ROM drive.

   A series of windows lead you through the installation process. During the installation, you can stop the process by clicking **Cancel** or **Stop** on any of the windows.
3. Click **Continue** to begin the installation process and display the first installation window.

   This window provides the product number and version of the endpoint being installed. The CONFIG.SYS file is always updated to enable the endpoint to start automatically when the system is rebooted. The update adds a new Run line to call the **STARTLCF.EXE** command. The **Options** check box is always unavailable (or inactive).
4. Click **OK**. The Install - directories window is displayed.
5. In the **Installation directory** field, type the name of the directory in which you want to install the endpoint.
6. If you need to check disk space, perform the following steps. Otherwise, continue with step 7.
   a. Click **Disk Space**. A window showing the disk space requirements for the endpoint and the disk space available on the local system is displayed.
   b. Click **OK** to return to the Install - directories window.
7. Click **Install** to continue to the Install - progress window.

   A window that contains the following fields is displayed. Use this window to specify startup and configuration options as shown in the following table.

Field name	Use when ...
Gateway port	The gateway uses a port number other than the default.
Endpoint port	You want to specify a port for the endpoint to use.
Options	You want to specify other lcfd configuration options. For example, use the **–g** *hostname* option to specify the intercepting gateway. Refer to the **lcfd** command in the *Tivoli Management Framework Reference Manual* for a list of valid options.

8. Click **OK** to complete the installation and start the endpoint.

Several completion messages are displayed to let you know when the endpoint is fully installed and logged in to a gateway.

Installing endpoints using logon scripts

Endpoints can be automatically installed on Windows operating system using logon scripts. When a user logs on, the logon script checks for an existing endpoint directory and, if necessary, launches the endpoint installation process. Endpoints are installed with the default configuration.

Note: You must be a member of the Windows Administrators group to use logon scripts.

Chapter 17. Installing Tivoli products and patches

This chapter provides the procedures required to install or upgrade Tivoli products in your Tivoli environment. Before installing or upgrading any product, read the release notes and other documentation for that product.

Note: Unless otherwise stated, this chapter uses the term *patch* to refer to maintenance releases and individual product patches.

Before and after installing any Tivoli product, back up the Tivoli object database for all machines in the Tivoli region. This backup enables you to return the object database to a known working state if you encounter a problem during an installation.

To back up the Tivoli object database from the Tivoli desktop, select **Desktop → Backup**. This action backs up the object database on the Tivoli server and the managed nodes in the region. If you are using the command line, use the **wbkupdb** command. For additional information about the **wbkupdb** command, refer to *Tivoli Management Framework Reference Manual*.

Note: To install a product to a managed resource, that computer system must be reachable on the network.

The following table provides the authorization roles required to perform an installation.

Table 6. Authorization roles for installation

Activity	Context	Authorization Role
Install or upgrade a product using Tivoli Management Framework	Tivoli region	**super** or **install_product**
Install or upgrade a product using Tivoli Software Installation Service	Tivoli region	**user** plus one of **super**, **senior**, **install_client**, or **install_product**

Products and patches available

Each Tivoli product ships with at least one CD. This CD generally contains multiple product installation images that can be installed on different systems in the Tivoli environment. For additional information about the files contained on a product CDs, refer to “Packaging of installation images” on page 7.

Copying installation images

If you want to copy any Tivoli Enterprise CD, you must use the **wcpcdrom** command. Use this command instead of copying the directory or folder to ensure that you can use the directory to install Tivoli software from any supported operating system.

Although making a copy is not required, the copy enables you to perform the following tasks at a later time without finding and mounting the CD:

- Creating Tivoli managed resources or installing Tivoli products using the Tivoli desktop or the Tivoli commands.

- Importing product installation image into the Tivoli Software Installation Service install repository.

For example, to copy the Tivoli Management Framework 1 of 2 CD, complete these steps:

1. Initialize the Tivoli environment variables, as described in “Setting Tivoli environment variables” on page 30.
2. Use **wcpcdrom** to copy the files. The following example copies the installation files on the Tivoli Management Framework 1 of 2 CD mounted on /cdrom/cdrom0 to the directory /usr/local/Tivoli/FW_install_media:

   ```
   wcpcdrom -c /cdrom/cdrom0 /usr/local/Tivoli/FW_install_media
   ```

To reduce the space required, you can copy only the files needed for the operating systems in your Tivoli environment. For complete information about the **wcpcdrom** command, refer to *Tivoli Management Framework Reference Manual*.

Installing Java components

The Tivoli Management Framework 2 of 2 CD shipped with Tivoli Management Framework contains product installation images that allow the Tivoli Management Framework and other Tivoli products to run Java programs. Refer to the product documentation to determine whether any of these Java components are required.

The following table describes the Java components provided on the Tivoli Management Framework 2 of 2 CD. The table contains the name of the index (.IND) file that you use to install the component from the command line, the name of the component that you use to install the component using either the Tivoli desktop or the Tivoli Software Installation Service console, and the registered product tag that you use to uninstall the component.

Table 7. Java components

.IND File Name	Component Name	Tag
JRE130	Java 1.3 for Tivoli	JRE130
JCF411	Tivoli Java Client Framework 4.1.1	JCF411
JHELP41	JavaHelp 1.0 for Tivoli	JavaHelp
MDIST2GU	Distribution Status Console, Version 4.1.1	mdist2gui
JRIM411	Tivoli Java RDBMS Interface Module (JRIM) 4.1.1	JRIM411

None of these Java components has installation options that you must specify during installation.

Note: These Java components are for use by Tivoli products in your Tivoli environment. Although other applications installed on your systems have similar Java components, you must install these Java components to have Tivoli products correctly work in your Tivoli environment.

Purpose of Java components

The following list describes each Java component shipped with Tivoli Management Framework, Version 4.3.1:

Java 1.3 for Tivoli

The Java Runtime Environment (JRE)—contains all the files required to run a Java applications. For AIX, this includes the Java Development Kit (JDK) files.

Tivoli Java Client Framework 4.1.1

The Tivoli Java Client Framework (JCF)—contains only Java executable code. This component is used by Tivoli products that make requests to Tivoli Management Framework.

JavaHelp 1.0 for Tivoli

The Sun Microsystems JavaHelp classes—contains only Java executable code. This component is used by Tivoli products that implement JavaHelp.

Distribution Status Console, Version 4.1.1

The graphical user interface (GUI) for MDist 2, a multiplexed distribution service.

Tivoli Java RDBMS Interface Module (JRIM) 4.1.1

The Tivoli Java RDBMS Interface Module (JRIM)—contains only Java executable code. This component is used by the Distribution Status console to store information in the relational database management system (RDBMS).

Where to install Java components

The following list explains where each Java component should be installed in your Tivoli environment:

Java 1.3 for Tivoli

This component must be installed on any Tivoli server or managed node where the Distribution Status console, Tivoli Software Installation Service depot, or Tivoli Software Installation Service client will be installed.

Tivoli Java Client Framework 4.1.1

This component must be installed on any Tivoli server or managed node where the Tivoli Software Installation Service depot or Tivoli Software Installation Service client, or Distribution Status console will be installed.

JavaHelp 1.0 for Tivoli

This component must be installed on any Tivoli server, managed node, or endpoint where the Distribution Status console will be installed.

Distribution Status Console, Version 4.1.1

This component must be installed on any supported Tivoli server or managed node where you want to run the Distribution Status console.

Tivoli Java RDBMS Interface Module (JRIM) 4.1.1

This component must be installed on any machine where the Distribution Status console will be installed.

These Java components might need to be installed on other systems hosting different Tivoli Enterprise products. For details, refer to the documentation for that product.

Installing Java components for other products

Some Tivoli products that run on Tivoli Management Framework, Version 4.1.1 require back-level versions of Java components. For these products, you must install the versions of the Java components required by the product as well as the versions required by Tivoli Management Framework. New Java components do not supersede or overwrite old Java components. They install separately and they can coexist on the same system with back-level components. For example, Tivoli Java Client Framework 4.1.1 does not supersede Tivoli Java Client Framework 4.1.

To determine which versions of Java components a Tivoli product requires, see the product-specific documentation.

The Tivoli Management Framework Compatibility CD supplies the following back-level versions of Java components:

- JavaHelp for Tivoli 3.7
- Tivoli Java Client Framework 3.7
- Tivoli Java Client Framework 3.7.1
- Tivoli Java Client Framework 4.1
- Java for Tivoli 3.7
- Java for Tivoli 3.7.1
- Tivoli Java RDBMS Interface Module (JRIM) 3.7
- Tivoli Java RDBMS Interface Module (JRIM) 3.7.1
- Tivoli Java RDBMS Interface Module (JRIM) 4.1
- Swing for Tivoli 3.7

Installing MDist 2 components

Both the MDist and MDist 2 services are installed with Tivoli Management Framework. However, if your Tivoli application uses MDist 2, you must install the following components to take full advantage of its functionality:

- Java 1.3 for Tivoli
- Java Client Framework for Tivoli
- Install an RDBMS

To use the MDist 2 service from the command line or through the Distribution Status console, you must perform the following tasks:

1. Configure the RDBMS that is supported by Tivoli Management Framework. Either the RDBMS client or the RDBMS server must be on a system managed by Tivoli Management Framework. This system becomes your RDBMS Interface Module (RIM) host.
2. Create the MDist 2 tables. For instructions, refer to “Database tables required for MDist 2” on page 225.
3. Create a RIM object named mdist2 on the RIM host. For general instructions, refer to “Creating the RIM object for MDist 2” on page 225. For detailed instructions, refer to Chapter 18, “Using RIM objects,” on page 243.

For more information about MDist 2 features, refer to *Tivoli Management Framework Planning for Deployment Guide*.

The following graphical user interfaces (GUIs) are available for use with MDist 2:

Distribution Status console

Enables a Tivoli administrator to monitor status and control distributions using MDist 2.

To use MDist 2 through the Distribution Status console, you must separately install the console on one or more managed nodes. For instructions, refer to “Installing the Distribution Status console” on page 226.

You do not need to install the console to control distributions using only the command line interface, the **wmdist** command.

Mobile Computing console
Provides end users with notification of pending distributions and enables them to control distributions sent to their computers. Install this program on each endpoint that will operate in mobile mode. For instructions, refer to “Installing the Mobile Computing console” on page 227.

For an overview about how to use the consoles, refer to *Tivoli Management Framework User’s Guide*.

Database tables required for MDist 2

When you install MDist 2, the following SQL scripts are written to $BINDIR/TME/MDIST2/sql:

- mdist_*vendor*_admin.sql
- mdist_*vendor*_schema.sql

where *vendor* is the name of a supported database vendor.

The mdist_*vendor*_admin.sql scripts are sample structured query language (SQL) scripts that can be used as templates. Modify the script to comply with your environment. The default database user name and password is mdstatus.

When you run the mdist_*vendor*_schema.sql script, the following tables are created in the associated database:

DIST_STATE
This table contains a description and status summary for each distribution. Each distribution creates a new row in the table. The exact amount of space needed for each entry depends on the length of user-specified strings, such as the name of the distribution. As an estimate, each entry requires 350 bytes.

DIST_NODE_STATE
This table contains the node statuses. A new row is created for each node in each distribution. A node can be either an endpoint or a repeater. The exact amount of space needed for each entry depends on the length of user-specified strings. As an estimate, each entry requires 130 bytes.

For example, each distribution requires approximately 350 bytes plus 130 bytes for each endpoint or repeater in the distribution. If 20 endpoints and repeaters are involved in the distribution, the space required in the tables is 2950 bytes (350+(130*20)).

To modify the amount of time a completed distribution remains in the tables, use the **wmdist –T** command. The removal of the entries occurs when the next distribution is started.

Creating the RIM object for MDist 2

To create the RIM object, use the **wcrtrim** command.

For example, to create a RIM object on managed node reality to support Sybase, enter the following command:

```
wcrtrim -v Sybase -h reality -d mdist2 -u tivoli -H /data/sybase -s reality mdist2
```

where:

–v Sybase
Specifies that Sybase is the database vendor.

–h reality
Creates the RIM host on managed node reality.

–d mdist2
Specifies that mdist2 is the name of the database to which the RIM object connects.

–u tivoli
Specifies that tivoli is the name of the database user.

–H /data/sybase
Specifies that /data/Sybase is the path to the database home directory. Generally, this is the value of the SYBASE variable created during Sybase installation.

–s reality
Specifies that reality is the server ID for the database. Generally, this is the value of the DSQUERY variable created during Sybase installation.

mdist2
Specifies that mdist2 is the label of the created RIM object.

The label for the RIM object must be mdist2. If you use a label other than mdist2, you must use the **wmdist –R** command to rename the RIM object.

For additional information about creating RIM objects, refer to Chapter 18, “Using RIM objects,” on page 243. For additional information about configuring the database in your Tivoli environment refer to *Tivoli Management Framework Release Notes*.

Installing the Distribution Status console

The following table describes the component of Distribution Status console, which is shipped in the \JAVA subdirectory of the Tivoli Management Framework 2 of 2 CD. The table lists the name of the index (.IND) file that you use to install the component, the name of the component, and the registered product tag that you use to uninstall the product.

Table 8. Components of Distribution Status console

.IND File Name	Component Name	Tag
MDIST2GU	Distribution Status Console, Version 4.1.1	mdist2gui

Before installing the Distribution Status console, install the Java components in the following order:

1. Java 1.3 for Tivoli
2. JavaHelp 1.0 for Tivoli
3. Tivoli Java Client Framework 4.1.1
4. Tivoli Java RDBMS Interface Module (JRIM) 4.1.1

After the Java prerequisites are installed, install the Distribution Status console using the component name listed in Table 7 on page 222. The Distribution Status console has no installation options.

Installing the Mobile Computing console

This section describes actions you must take in your Tivoli environment to prepare to use the Mobile Computing console and the steps necessary to install the console on each mobile endpoint.

Before installing the Mobile Computing console

Before installing the console, perform the following tasks:

- Make sure that your Tivoli environment has been upgraded to Tivoli Management Framework, Version 4.3.1.
- Use the **wep** command to allow the desired endpoints to use mobile computing. Although you can install the Mobile Computing console on an endpoint in the default desktop mode, the console will not function.

  You can enable mobile computing using one of these approaches, which are described in the following sections:

  – Manually specify the login mode for each endpoint.

  – Allow each endpoint to specify its own login mode as either mobile or desktop.

Manually specifying the endpoint login mode: Follow these steps to explicitly enable Mobile Computing on each endpoint that you want to operate in mobile mode:

1. Enter the following command on the Tivoli server or any managed node to enable an endpoint to operate in mobile mode:

   ```
   wep ep_label set login_mode -m mobile
   ```

   where *ep_label* is the label of the endpoint you want to become mobile.
2. Synchronize the change with the gateway by entering the following command:

   ```
   wep sync_gateway
   ```

   If you use a script to set multiple endpoints to mobile mode, you can synchronize once after the login mode has been set for all endpoints.

For more information about the **wep** command, refer to *Tivoli Management Framework Reference Manual*.

Allowing an endpoint to set its own login mode: The Mobile Computing console installation program automatically configures the endpoint to request mobile mode. However, if a Tivoli administrator has not enabled the endpoint to change its own login mode, the request is ignored when the endpoint logs in.

Follow these steps for each endpoint that you want enabled to change its own login mode:

1. Enter the following command on the Tivoli server or any managed node to enable an endpoint to change its own login mode:

   ```
   wep ep_label login_mode -s variable
   ```

   where *ep_label* is the label of the endpoint you want to control its own login mode.
2. Synchronize the change with the gateway by entering the following command:

   ```
   wep sync_gateway
   ```

   If you are using a script to enable multiple endpoints to change their login mode, you can synchronize once after changing all endpoints.

For more information about the **wep** command, refer to *Tivoli Management Framework Reference Manual*.

Installing the Tivoli Mobile Computing console

To install the Mobile Computing console on an endpoint, perform the following steps on the endpoint:

1. Make sure that an endpoint has been created on each system where you want to run the Mobile Computing console.
2. Run the **setup.exe** script appropriate to the operating system of the endpoint. The installation programs are located on the Tivoli Management Framework 2 of 2 CD in the following locations:

   For Windows 98 operating systems:
   /mobile/win95/setup.exe

   For Windows operating system, except Windows 98:
   /mobile/winnt/setup.exe
3. Follow the instructions in the installation program to install the Mobile Computing console on the endpoint.
4. To add support for languages other than English, use the **setup.exe** script in the \mobile directory of the Version 4.3.1 language support CD. This installation program contains all of the support languages.

You can also record response files to enable you to install the Mobile Computing console silently, without user interaction. For an example demonstrating how to set up for and perform a silent installation, refer to "Installing the Tivoli desktop on Windows operating systems using a response file" on page 66.

In this release (Version 4.3.1), the Mobile Computing console installer has been modified such that if you run the `setup.exe` program to install it, it is installed and runs using the Windows LocalSystem account. This means that now, any user, whether in the Administrators group, or not, can accept distributions sent from a Tivoli Administrator, when the mobile endpoint is not connected. If you want to require only Administrators to be able to accept distributions while a mobile endpoint is not connected (the behavior of the console prior to this release), use the `-noservice` option when you run the `setup.exe` program. The syntax to use is as follows:

```
setup.exe -noservice
```

Uninstalling the Mobile Computing console

To uninstall the Mobile Computing console, use the Windows Add/Remove Programs window.

Installing Secure Sockets Layer

The Secure Socket Layer (SSL) packages provided by and used with Tivoli Management Framework provide support for SSL security in your Tivoli environment. On all non-Linux managed nodes, you must install the SSL-A package to enable SSL connections. On Linux managed nodes, SSL is enabled by default. You only need to install the SSL-B package for keystore management. If the SSL-A package is not installed on a managed node, the managed node can only accept non-SSL connections from clients. For detailed information about how to enable SSL in your Tivoli environment, refer to *Tivoli Management Framework User's Guide*.

The following table describes the components for SSL-A and SSL-B, which are shipped in the /SSL subdirectory of the Tivoli Management Framework 1 of 2 CD. The table lists the name of the index (.IND) file that you use to install the component, the name of the component, and the registered product tag that you use to uninstall the product.

Table 9. Components for installing or upgrading SSL packages

.IND File Name	Component Name	Tag
SSLA	Tivoli Management Framework SSL-A, Version 1.1	SSLA
SSLA_UPG	Tivoli Management Framework SSL-A, Version 1.1	SSLA
SSLB	Tivoli Management Framework SSL-B, Version 1.1	SSLB
SSLB_UPG	Tivoli Management Framework SSL-B, Version 1.1	SSLB

Installing documentation

The documents in Tivoli Management Framework library are in Portable Data Format (PDF) versions and can be installed on any machine in your environment. The documentation files are in the root directory of the Tivoli Management Framework Version 4.3.1 Documentation CD. To install any document, copy the file from the CD to any directory on any machine in your environment. The optimal place to copy the documentation to is a common file server where all Tivoli administrators can access this information.

This library contains the following documents:

- *Tivoli Enterprise Installation Guide* (Installation_Guide)
- *Tivoli Management Framework Planning for Deployment Guide* (TMF_Planning_Guide)
- *Tivoli Management Framework User's Guide* (TMF_Users_Guide)
- *Tivoli Management Framework Reference Manual* (TMF_Reference_Manual)
- *Tivoli Management Framework Maintenance and Troubleshooting Guide* (TMF_Maint_Troubleshoot)

Before installing a product

Before installing or upgrading components of a specific Tivoli product, you must prepare your Tivoli environment to function correctly after the product is installed or upgraded. This information is contained in the documentation for that product.

Note: Do not install products across Tivoli region boundaries.

Refer to the product documentation to determine where the product can be installed. Most products are installed on the Tivoli server or managed nodes. Some products can be installed on endpoints. To install a product on an endpoint you must use Tivoli Software Installation Service or an operating-specific installation mechanism. The Tivoli desktop and **winstall** command cannot install on an endpoint.

Installing a product using Tivoli Software Installation Service

To install a Tivoli product using Tivoli Software Installation Service, perform the following high-level steps:

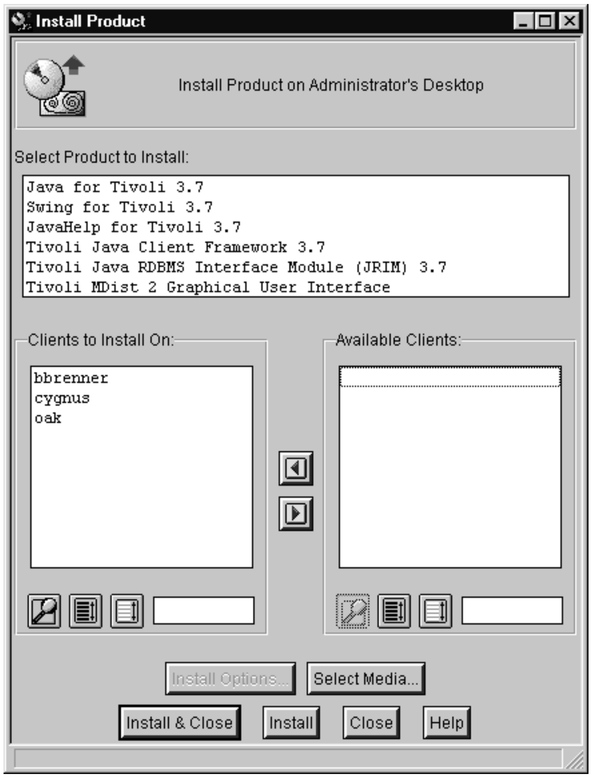
1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113.
2. Complete the installation worksheet as follows:
   a. Select one or more products from the install repository:
      - If necessary, import the products into the install repository, as described in “Importing products using the Tivoli Software Installation Service console” on page 115.
      - Add the products to the installation worksheet as described in “Adding products to the installation worksheet” on page 127.
      - If desired, specify default values for the installation options for the products as described in “Viewing and customizing installation options” on page 119.
   b. Specify the machines on which to install:
      - Make sure that the machines on which you want to install are already known to Tivoli Software Installation Service. If necessary, define the new machines as described in “Adding machines to the Tivoli Software Installation Service depot” on page 130.
      - Add the machines to the installation worksheet as described in “Adding machines to the installation worksheet” on page 134.
   c. Mark the installation worksheet to indicate on which machines to install each product. This is described in “Specifying products to install” on page 138.
   d. If desired, modify the installation options for a product on a specific machine. This is described in “Overriding default installation options for one machine” on page 137.
3. Start the installation as described in “Using the Tivoli Software Installation Service console to install products” on page 139.

A command line interface to Tivoli Software Installation Service is available as well. For more information, refer to “Installation overview from the command line” on page 112.

Installing a product from the Tivoli desktop

To install a Tivoli product from the Tivoli desktop, perform the following steps:

1. Start the Tivoli desktop on the Tivoli server by issuing the **tivoli** command or double-clicking the Tivoli icon (Windows and OS/2). For additional information about the **tivoli** command, refer to *Tivoli Management Framework Reference Manual*.
2. From the **Desktop** menu, select **Install → Install Product**. The Install Product window is displayed.



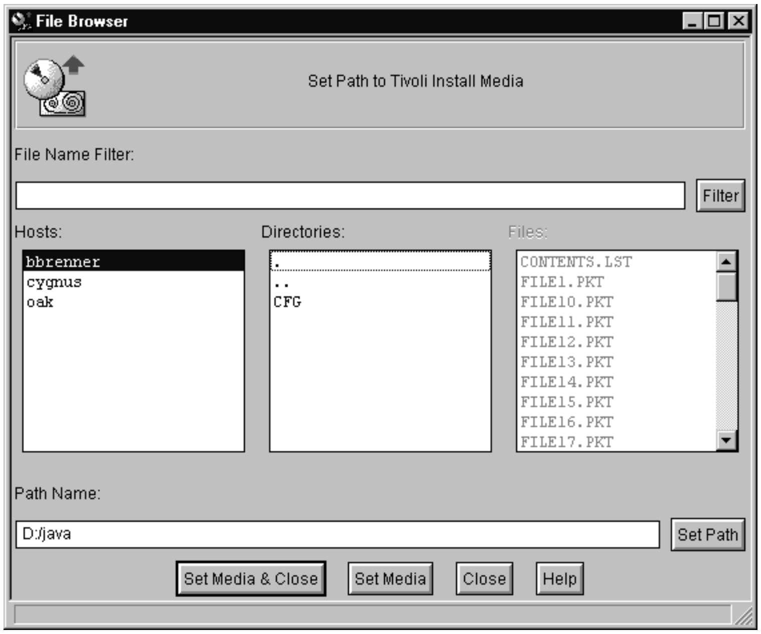
```
Java for Tivoli 3.7
Swing for Tivoli 3.7
JavaHelp for Tivoli 3.7
Tivoli Java Client Framework 3.7
Tivoli Java RDBMS Interface Module (JRIM) 3.7
Tivoli MDist 2 Graphical User Interface
```

Clients to Install On:

```
bbrenner
cygnus
oak
```

If the **Select Product to Install** list contains the Tivoli product, go to step 6; otherwise, continue to step 3.

3. Click **Select Media**. The File Browser window is displayed. This window enables you to specify the path to the installation images.



Hosts:

```
bbrenner
cygnus
oak
```

Directories:

```
.
..
CFG
```

Files:

```
CONTENTS.LST
FILE1.PKT
FILE10.PKT
FILE11.PKT
FILE12.PKT
FILE13.PKT
FILE14.PKT
FILE15.PKT
FILE16.PKT
FILE17.PKT
```

Path Name: D:/java

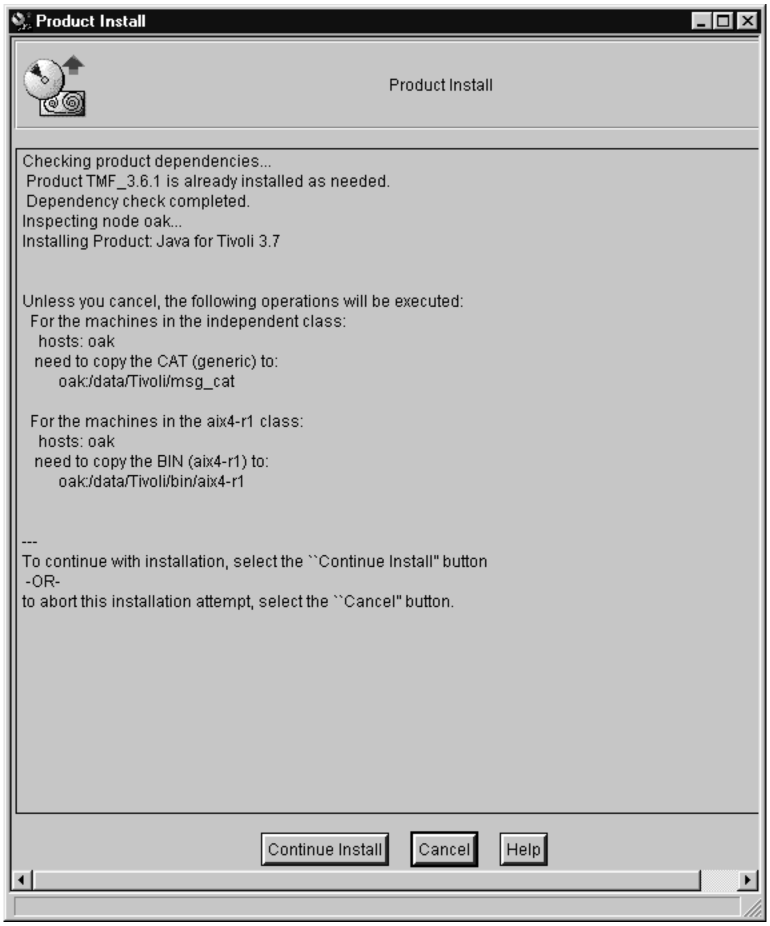
4. In the **Hosts** list, select the host on which the installation image is mounted.
5. Navigate to the directory that contains the installation image. The installation directory contains the product index (.IND) file.

   Double-click directory names in the **Directories** list until the installation images are shown in the **Files** list.

Alternatively, if you know the path to the installation image, type the full path in the **Path Name** field. Click **Set Path** to list the contents of the specified directory.

6. Click **Set Media & Close** to save the new media path and return to the Install Product window. The window now contains a list of products that are available for installation.
7. From the **Select Product to Install** list, select the Tivoli products to install. If the product has installation options, the Install Options window is displayed.

   Refer to the product documentation for details on the information required for this window. Each Tivoli product has different installation options.
8. Click **Set** to save the values in the Install Options window and return to the Install Product window.
9. Select the systems on which you want to install this Tivoli product from the **Available Clients** list. Only supported managed resources are displayed.
10. Use the left-arrow button to move the selected managed resources to the **Clients to Install On** list.
11. Click **Install** to start installing the Tivoli product. The Product Install window provides a list of the operations as they occur and informs you of any problems that you might want to correct before installing.



```
Checking product dependencies...
 Product TMF_3.6.1 is already installed as needed.
 Dependency check completed.
Inspecting node oak...
Installing Product: Java for Tivoli 3.7


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts: oak
   need to copy the CAT (generic) to:
        oak:/data/Tivoli/msg_cat

  For the machines in the aix4-r1 class:
    hosts: oak
   need to copy the BIN (aix4-r1) to:
        oak:/data/Tivoli/bin/aix4-r1


---
To continue with installation, select the ``Continue Install'' button
 -OR-
to abort this installation attempt, select the ``Cancel'' button.
```

12. Choose one of the following options:
    - Review the status information and click **Continue Install**. The Product Install window informs you when the installation is complete.
    - Click **Cancel** to install the Tivoli product later.

13. When the installation is complete, click **Close** to close the Product Install window and return to the Install Product window.
14. Click **Close** to close the Install Product window.

Installing a product from the command line

This section contains detailed information about using the **winstall** command to install Tivoli products.

Installing a product using the winstall command

To install Tivoli products from the command line, use the **winstall** command. The general form of the **winstall** command is as follows:

winstall –c *source_dir* **–i** *product* [*install_variables*]... *managed_node*

where:

–c *source_dir*
: Specifies the complete path to the directory containing the installation image.

–i *product*
: Specifies the product index file from which the product is installed. Index files have an IND extension. You can specify this value without the file extension.

install_variables
: Specifies product-specific installation variables. For details about the installation options for a specific product, refer to the product documentation.

managed_node
: Specifies the managed node on which to install this Tivoli product. You can specify multiple managed nodes. If you do not specify a managed node, the product is installed on all managed nodes.

For more information about the **winstall** command, refer to *Tivoli Management Framework Reference Manual*.

Examples of using the winstall command

To install the Tivoli Software Installation Service depot from the command line on managed node oak, enter the following command:

```
winstall -c /cdrom -i SISDEPOT IRDIR=/ir oak
```

where:

–c /cdrom
: Specifies the path to the installation image.

–i SISDEPOT
: Specifies the SISDEPOT product index file.

IRDIR=/ir
: Specifies that the install repository directory is /ir. If this directory does not exist, it is created for you. This directory cannot be the install repository directory from a 3.6 or 3.6.1 version of Tivoli Software Installation Service.

oak Indicates that the depot will be installed on the managed node oak.

To install the Tivoli Software Installation Service client from the command line on managed node oak and place the client logs in the /data/sis_logs directory, enter the following command:

```
winstall -c /cdrom -i SISCLNT CLIENTLOGDIR=/data/sis_logs oak
```

where:

–c /cdrom
Specifies the path to the installation image.

–i SISCLNT
Specifies the SISCLNT product index file.

CLIENTLOGDIR=/data/sis_logs
Specifies that the logs will be placed in the /data/sis_logs directory.

oak Indicates that the client will be installed on the managed node oak.

After installing a product

After installing specific Tivoli products, there might be additional steps you need to perform so that the product will function correctly. This information is contained in the documentation for that product.

Upgrading Tivoli Management Framework

This section contains information about upgrading the Tivoli server, managed nodes, and endpoints.

Upgrading Tivoli servers and managed nodes

Before upgrading a Tivoli server, ensure that the object dispatcher (oserv) process is running on the Tivoli server and managed nodes.

To upgrade the Tivoli server and managed nodes in your region, follow the procedures in “Upgrading Tivoli products” on page 235 to apply the appropriate upgrade or patch image.

Note: After upgrading Tivoli Management Framework, you need to run the **odadmin reexec** command to recycle the oserv processes on the upgraded systems.

Upgrading gateways

Although you do not explicitly upgrade a gateway, the phrase is commonly used to refer to upgrading Tivoli Management Framework on the managed node that hosts a gateway. If a Tivoli product requires that one of its components be installed on a gateway, upgrading the gateway may refer to upgrading those components as well.

You do not need to perform any gateway-specific actions after upgrading the managed node that hosts a gateway, except on the following operating systems:

OS/2 operating systems
After upgrading Tivoli Management Framework on an OS/2 managed node that hosts a gateway, reboot the OS/2 system.

NetWare operating systems

After upgrading a NetWare Gateway, run the following script on the Tivoli server to complete the upgrade of the NetWare Gateway upgrade:

```
bash nw_Upgrade.sh host_name
```

where *host_name* is the TCP/IP host name of the NetWare system.

Upgrading endpoints

If you have a previous version of the Tivoli endpoint software installed, you can use one of the following methods to upgrade the endpoint:

- The **wepupgd** command
- The **login_policy** script

After you upgrade the endpoint images on the Tivoli server, managed nodes, and gateways in your region by upgrading Tivoli Management Framework, you can use the **wepupgd** command to manually upgrade endpoints. For more information about the **wepupgd** command, refer to *Tivoli Management Framework Reference Manual*.

The **login_policy** script automates the upgrade process for you. After you configure this policy to call the **upgrade.sh** script, all endpoints are upgraded when each logs in to their assigned gateway. For instructions on how to configure the **login_policy** script, refer to the section on Tivoli-defined policies in *Tivoli Management Framework Reference Manual*.

Upgrading Tivoli products

Tivoli patches provide software corrections, and sometimes enhancements, to a previous release of Tivoli products.

Note: If you upgrade any localized installation of any Tivoli product, you must apply the language pack for that product after upgrading. Otherwise, the upgraded installation might operate in English.

You can install patches using Tivoli Software Installation Service, the Tivoli desktop, or the command line.

In some cases, the final step of installing a patch is restarting the object dispatcher (oserv) on the Tivoli server, its managed nodes, or both. Refer to the release notes or the readme file for the actions required for a particular patch.

Upgrading a product using Tivoli Software Installation Service

To install a Tivoli patch using Tivoli Software Installation Service, perform the following high-level steps:

1. Start the console as described in “Starting the Tivoli Software Installation Service console” on page 113.
2. Complete the installation worksheet as follows:
   a. Select one or more patches from the install repository:
      - Tivoli Software Installation Service considers each installation image, whether a product installation image or a patch installation image, to be a product installation image.

- If necessary, import the products into the install repository, as described in "Importing products using the Tivoli Software Installation Service console" on page 115.
- Add the products to the installation worksheet as described in "Adding products to the installation worksheet" on page 127.
- If desired, specify default values for the installation options for the products as described in "Viewing and customizing installation options" on page 119.

b. Specify the machines on which to install:

- Make sure that the machines on which you want to install are already known to Tivoli Software Installation Service. If necessary, define the new machines as described in "Adding machines to the Tivoli Software Installation Service depot" on page 130.
- Add the machines to the installation worksheet as described in "Adding machines to the installation worksheet" on page 134.

c. Mark the installation worksheet to indicate on which machines to install each product. This is described in "Specifying products to install" on page 138.

d. If desired, modify the installation options for a product on a specific machine. This is described in "Overriding default installation options for one machine" on page 137.

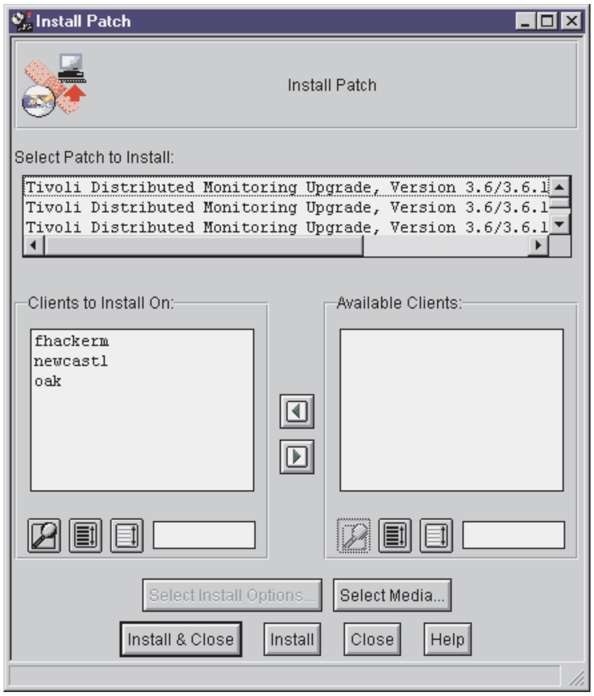
3. Start the installation as described in "Using the Tivoli Software Installation Service console to install products" on page 139.

A command line interface to Tivoli Software Installation Service is available as well. For more information, refer to "Installation overview from the command line" on page 112.

Upgrading a product from the Tivoli desktop

To install patches from the Tivoli desktop, perform the following steps:

1. Start the Tivoli desktop by issuing the **tivoli** command (UNIX or Windows) or by clicking the Tivoli icon (Windows and OS/2). For additional information about the **tivoli** command, refer to *Tivoli Management Framework Reference Manual*.
2. From the **Desktop** menu, select **Install → Install Patch**. The Install Patch window is displayed.



Select Patch to Install:

```
Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1
Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1
Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1
```

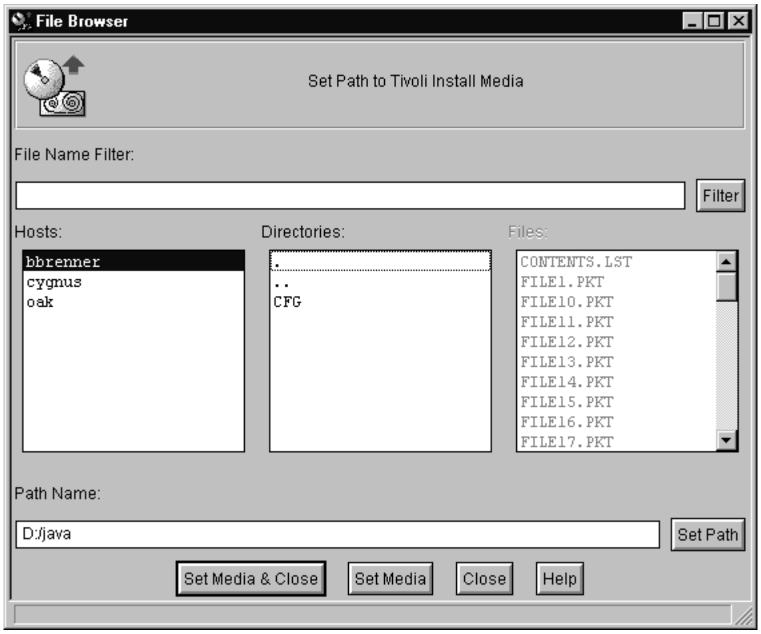
Clients to Install On:

```
fhackerm
newcastl
oak
```

Available Clients:

If the patch you want to install is not listed in the **Select Patch to Install** list, proceed to step 3. If the correct patch is listed, skip to step 6.

3. Click **Select Media**. The File Browser window is displayed. This window enables you to specify the path to the installation images.



File Name Filter:

Hosts:

```
bbrenner
cygnus
oak
```

Directories:

```
.
..
CFG
```

Files:

```
CONTENTS.LST
FILE1.PKT
FILE10.PKT
FILE11.PKT
FILE12.PKT
FILE13.PKT
FILE14.PKT
FILE15.PKT
FILE16.PKT
FILE17.PKT
```

Path Name: D:/java

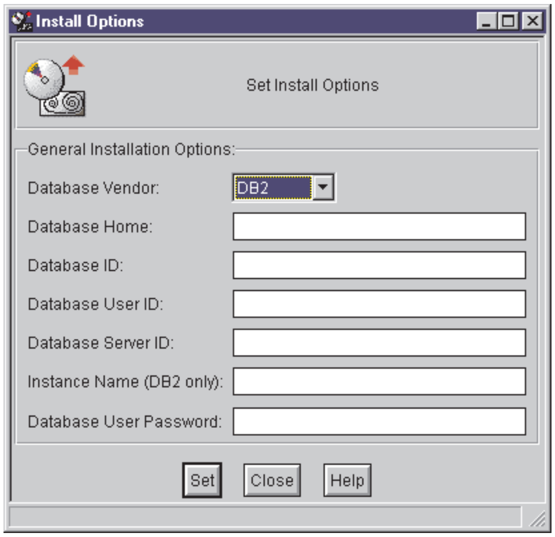
4. In the **Hosts** list, select the host on which the installation image is mounted.
5. Navigate to the directory that contains the installation image. The installation directory contains a file named PATCHES.LST and the index (.IND) file for the patch.

   Double-click directory names in the **Directories** list until the installation images are shown in the **Files** list.

Alternatively, if you know the path to the installation image, type the full path in the **Path Name** field. Click **Set Path** to list the contents of the specified directory.

6. Click **Set Media & Close** to save the new media path and return to the Install Product window. The window now contains a list of products that are available for installation.
7. Select the patch you want to install from the **Select Patch to Install** list.

   If the patch you select includes a product enhancement, an Install Options window similar to the following might be displayed.



Install Options

Set Install Options

General Installation Options:

Database Vendor: DB2

Database Home:

Database ID:

Database User ID:

Database Server ID:

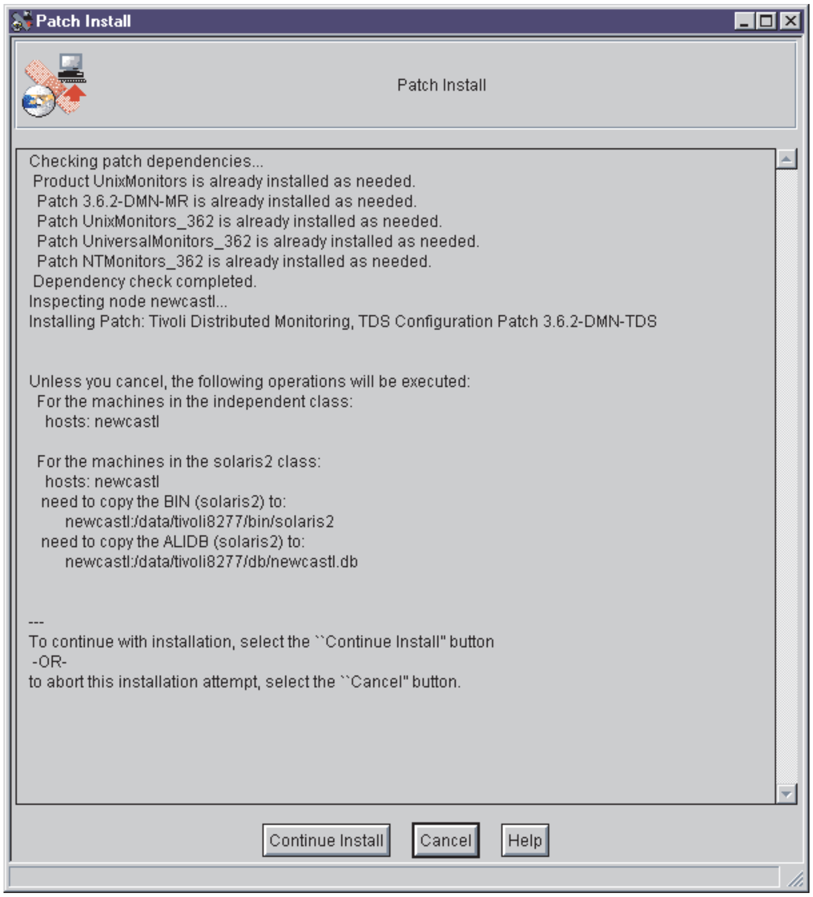
Instance Name (DB2 only):

Database User Password:

   Fill in this window according to the information in the product documentation for that specific Tivoli product.
8. To specify the clients on which this product will be installed, use the arrow keys to move machine names between the **Clients to Install On** list and the **Available Clients** list.

   By default, all machines in the current TMR are listed in the **Clients to Install On** list. To move a machine name to the **Available Clients** list, choose one or more clients from the **Clients to Install On** list and press the right-arrow button. The chosen clients are moved from the **Clients to Install On** list to the **Available Clients** list.
9. Click **Install & Close** to install the patch and close the Install Patch window.

   The installation process prompts you with a Patch Install window similar to the following. This window provides the list of operations that will take place during the installation process. This window also warns you of any problems that you may want to correct before you install the patch.



```
Checking patch dependencies...
 Product UnixMonitors is already installed as needed.
  Patch 3.6.2-DMN-MR is already installed as needed.
  Patch UnixMonitors_362 is already installed as needed.
  Patch UniversalMonitors_362 is already installed as needed.
  Patch NTMonitors_362 is already installed as needed.
 Dependency check completed.
Inspecting node newcastl...
Installing Patch: Tivoli Distributed Monitoring, TDS Configuration Patch 3.6.2-DMN-TDS


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts: newcastl

  For the machines in the solaris2 class:
    hosts: newcastl
   need to copy the BIN (solaris2) to:
         newcastl:/data/tivoli8277/bin/solaris2
   need to copy the ALIDB (solaris2) to:
         newcastl:/data/tivoli8277/db/newcastl.db


---
To continue with installation, select the ``Continue Install'' button
 -OR-
to abort this installation attempt, select the ``Cancel'' button.
```

10. Click **Continue Install** to begin the installation process and display the Patch Install status window. The Patch Install status window contains status information as the installation proceeds.

    When the installation is completed, the Patch Install window returns a completion message.
11. Click **Close** when the Patch Install window indicates that the installation is completed.
12. If the release notes or readme file indicates that you need to restart the object dispatcher (oserv), enter the following command:

    ```
    odadmin reexec all
    ```

Upgrading a product from the command line

This section contains detailed information about using the **wpatch** command to upgrade Tivoli products.

Upgrading a product using the wpatch command

To upgrade products from the command line, use the **wpatch** command. The general form of the **wpatch** command is as follows:

wpatch –c *source_dir* **–i** *patch* **–n –y** [*install_variables*]... *managed_node*

where:

–c *source_dir*
: Specifies the complete path to the directory containing the installation image.

–i *patch*
Specifies the index file from which the patch is installed. Index files have an IND extension. You can specify this value without the file extension.

–n Installs the patch on all managed nodes that do not currently have the patch installed. This option is ignored if a managed node is specified.

–y Specifies that the installation should proceed without confirmation. By default, this command identifies the actions that must be taken to perform the installation and requests confirmation before continuing. Using this option, **wpatch** identifies the actions and performs the installation without requesting the confirmation.

install_variables
Specifies patch-specific installation variables. These generally take the form `@`*`variable`*`@=`*`value`*.

For details about the installation options for a specific product and its patches, refer to the product and patch documentation.

managed_node
Specifies the managed node on which to install this patch. You can specify multiple managed nodes. If you do not specify a managed node, the patch is installed on all managed nodes where the base product is currently installed in the Tivoli region. In most cases, this option will not be specified.

For more information about the **wpatch** command, refer to *Tivoli Management Framework Reference Manual*.

Examples of using the wpatch command

To upgrade Tivoli Inventory, Version 3.6 to Version 3.6.2 from the command line on managed nodes babel, elvis, and sleepy, enter the following command:

```
wpatch -c /cdrom -i 362_INV_ babel elvis sleepy
```

where:

–c /cdrom
Specifies the complete path to the installation image.

–i 3.6.2_INV_
Specifies the name of the index file for the patch being installed.

babel elvis sleepy
Specifies the systems on which patch is installed.

To upgrade the Tivoli Inventory gateway from Version 3.6 to Version 3.6.2 from the command line on managed nodes elvis and sleepy, enter the following command:

```
wpatch -c /cdrom -i 362_GW_U elvis sleepy
```

where:

–c /cdrom
Specifies the complete path to the installation image.

–i 3.6.2_GW_U
Specifies the name of the index file for the patch being installed.

elvis sleepy
Specifies the systems on which the patch is installed.

After upgrading a product

After upgrading specific Tivoli products, there might be additional steps you need to perform so that the product will function correctly. This information is contained in the documentation for that product.

Chapter 18. Using RIM objects

For some Tivoli Enterprise applications, you must create and configure a RDBMS Interface Module (RIM) object. Before creating a RIM object, you must install and configure the relational database management system (RDBMS). The RIM object is hosted on a managed node called the RIM host. If the product that requires a RIM object does not create and configure a RIM object during the installation process, you must create it by using the **wcrtrim** command.

Note: You cannot create a RIM object using the Tivoli desktop.

The RIM object provides communication between Tivoli Enterprise applications and the RDBMS. Tivoli Management Framework supports the following databases:

- DB2
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

For information about which versions are supported for each database, refer to the *Tivoli Management Framework Release Notes*.

RIM host considerations

A RIM host is a managed node where the RIM object is created. Generally, RIM objects are created during the installation process. When deciding on which managed nodes should be RIM hosts, consider the following:

- It must be a managed node running a supported version of AIX, HP-UX, Linux, Solaris, or Windows operating system. For a list of supported managed nodes, refer to the *Tivoli Management Framework Release Notes*.
- The managed node must be local to the Tivoli region.
- The manage node must be preconfigured with the RDBMS client or server software.

  Note: Do not install the RDBMS server software on the RIM host unless this computer system is designated solely for RDBMS usage and is not designated for any other Tivoli operation.
- You can use a single managed node to host multiple RIM objects.
- Although you can configure multiple Tivoli Enterprise products to use a single database instance or schema, do not use this configuration. If you choose to use this configuration, consult a qualified database administrator on the proper use of tuning options to ensure optimal performance of all products involved. Additionally, you should thoroughly test this configuration in a lab to ensure that it works properly.
- After a RIM host is created, you can move it to another managed node, or you can change the associated RDBMS. To move the RIM object, use **wmvrim** command. To change the RDBMS, you need to delete the RIM object and recreated it using the **wdel** and **wcrtrim** commands, respectively. If you change the RDBMS, you need to recreate and repopulate the database tables in the RDBMS. For detailed information about these commands, refer to the *Tivoli*

Management Framework Reference Manual. For details about recreating and repopulating the database tables, refer to the application documentation.

RDBMS server considerations

The RDBMS server contains the repositories used by Tivoli Enterprise applications. When deciding which computer system should be for the RDBMS server, consider the following requirements:

- There must be a TCP/IP connection between the RIM host and the RDBMS server.
- The RDBMS server does not need to be on a computer system managed by Tivoli Management Framework.
- The RDBMS server must be in the same network as the Tivoli server.
- There must be enough disk space in the RDBMS to handle the amount of data stored in the repositories.
- Multiple Tivoli regions can use a single RDBMS.

DB2 considerations

The following sections list the requirements for IBM DB2 and provide information to help you install and configure DB2 for use with Tivoli Enterprise applications.

Requirements for DB2

The following requirements apply to DB2 RIM hosts:

- The DB2 user ID for the RIM object must have the correct permissions to access the server instance and database.
- One of the following must be true:
  - The client instance name must match the DB2 user ID for the RIM object, or
  - the instance name attribute for the RIM object (set with the **wsetrim –t** command) must be set to the client instance name.

Installing and configuring DB2

From a DB2 perspective, the RIM host is a database client. Therefore, the RIM host must have access to a DB2 server. Either a DB2 server or a DB2 client must be installed on the RIM host.

This information provides an overview of the steps required to configure DB2 for use with the RIM host. The installation process and considerations for DB2 are independent of this documentation. Therefore, you should follow the installation and configuration information provided in the IBM DB2 documentation.

Note: When you install DB2, the following environment variables are created in the .profile file of the instance owner:

DB2DIR
The directory where the DB2 software is installed.

INSTHOME
The home directory of the instance owner.

DB2INSTANCE
The active instance. This is the login name of the instance owner.

For UNIX operating systems, these variables can be set by running the $DB2INSTANCE/sqllib/db2profile script (Bourne, bash, or Korn shell) or the $DB2INSTANCE/sqllib/db2cshrc script (C shell).

Considerations for the DB2 user name

For UNIX systems, log in as root to install DB2 and create a DB2 instance, and then use the DB2 instance owner to configure the instance.

For Windows systems, log in as Administrator to install DB2 and create a DB2 instance, and then log in as db2admin to configure the instance.

The RIM object uses the DB2 user name as the client DB2 instance name unless the instance name attribute is set on the RIM object using the **wcrtrim** or **wsetrim –t** command. It is recommended that the DB2 user name match the client DB2 instance name. If the user name does not match the client instance name, then you must set the instance name attribute on the RIM object to the client instance name.

Configuring the DB2 server for remote client access

If the RIM host is the same system as your DB2 server, no additional configuration is required. If you want remote DB2 managed nodes to access the DB2 server, perform the following steps on the DB2 server:

1. Add the DB2COMM variable with the value `tcpip`.
2. Configure the DB2 server to accept TCP/IP clients. Edit the /etc/services file (UNIX operating systems) or %systemroot%\system32\drivers\etc\service file (Windows operating systems) and add an entry for the connection port. The connection port name must be the same as the instance name. For example, add the following line to the services file:

   ```
   TEC 3700/tcp #DB2 connection service port
   ```
3. Update the service name with the connection port name (in this case the instance name):

   ```
   db2 update database manager configuration using svcename TEC
   ```
4. Issue a **db2stop** command and then a **db2start** command to enable listening on the service port previously defined.

Configuring the DB2 client

To configure the DB2 client, perform the following steps:

1. Configure the DB2 client to use TCP/IP to connect to the DB2 server. Edit the /etc/services file (UNIX operating systems) or %systemroot%\system32\drivers\etc\service file (Windows operating systems) and add an entry for the connection port. The connection port name must be the same as the instance name. For example, add the following line to the services file:

   ```
   TEC 3700/tcp #DB2 connection service port
   ```
2. Add the DB2COMM variable with the value `tcpip`.
3. Catalog the server node and the remote database:
   a. Start the DB2 interactive session:

      ```
      db2
      ```
   b. Catalog the server node:

      `catalog tcpip node` *`db2node`* `remote` *`hostname`* `server` *`service_name`*

      where:

      db2node

      Specifies the name of the system where the DB2 client is installed. This name must be unique in your node directory list.

hostname
Specifies the name of the system where the DB2 server is installed.

service_name
Specifies the connection port name as defined in the services file.

c. Catalog the remote database:

```
catalog database db_name as alias_name at node db2node
```

where:

db_name
Specifies the name of the remote database.

alias_name
Specifies the name of the remote database or an alias for the remote database. This name must be unique in your database directory list.

db2node
Specifies the name of the system where the DB2 client is installed.

4. Verify that the node and database were created:

```
list node directory
list database directory
```

For the first command, the name, host name, and service name you created should be displayed. For the second command, the alias name, its database, and the node hosting the DB2 server should be displayed.

5. Verify connectivity to the database:

```
connect to db_name user db_user using db_user_password
```

where:

db_name
Specifies the name of the remote database.

db_user
Specifies the name of the database user.

db_user_password
Specifies the password of the database user.

Note: Create the client instance with server authentication. RIM does not support the default authentication type of CLIENT. You must create the client instance with an authentication type of SERVER or SERVER_ENCRYPT.

Informix considerations

The following sections list the requirements for Informix and provide information to help you install and configure Informix for use with Tivoli Enterprise applications.

Requirements for Informix

With each Informix server, you can use either unbuffered or buffered logging. For Tivoli Enterprise applications that use Informix repositories, database logging *must* be American National Standards Institute (ANSI) unbuffered logging. With buffered logging, you can experience problems with database locks. To switch to unbuffered logging after running in a buffered mode, you *must* back up, delete, and re-create all databases used by Tivoli Enterprise applications.

Refer to the Informix documentation for complete instructions about switching to unbuffered logging. The following steps provide a high-level description of the procedure:

1. Create a level 0 archive of the databases to replace.
2. Export existing tables using the **dbexport –ss** command.
3. Drop current databases configured with buffered logging.
4. Create new databases configured with unbuffered logging.
5. Import the exported tables from step 2 into the new databases created in step 4 by using the **dbimport** command.

Note: In a high-transaction or high-volume Tivoli environment, concurrency problems can occur. Under these conditions, tune your Informix server carefully to ensure that you do not have locking problems that result in data loss. Consult a qualified Informix database administrator to determine the best lock mode for your environment.

Installing and configuring Informix

Install the Informix RDBMS server. For installation details, refer to your Informix documentation.

Before creating an Informix repository, review the following prerequisites:

- The user ID for an Informix RIM object must be informix. This user ID must be a valid UNIX or Windows system account name on the computer system hosting the Informix server and clients.
- The Informix client must be installed on the RIM host and the .odbc.ini and .odbcinst.ini files must be installed in the directory where the Informix software is installed ($INFORMIXDIR).
- You must run the Informix server in ANSI unbuffered mode.
- The Informix client SDK is installed.

For details, see the Informix documentation.

Microsoft SQL Server considerations

The following sections list the requirements for Microsoft SQL Server and provide information to help you install and configure Microsoft SQL Server for use with Tivoli products.

Requirements for Microsoft SQL Server

There are no known requirements.

Installing and configuring Microsoft SQL Server

Install Microsoft SQL Server on the RDBMS server. For installation details, refer to your Microsoft SQL Server documentation.

Note: The Install Options window displayed during Microsoft SQL Server installation includes a **Sort Order** option. To use Microsoft SQL Server for your repository, you must select the **Dictionary order**, case-sensitive option at installation.

You must use a Windows managed node to be the RIM host with this database.

After your client is installed, verify connectivity using the osql client application.

Creating an ODBC data source for Microsoft SQL Server

RIM uses ODBC to connect to a Microsoft SQL Server database. You must create an ODBC data source to enable RIM to connect to a Microsoft SQL Server database. To create an ODBC data source, perform the following steps:

1. In Windows, click **Start** → **Settings** → **Control Panel**.
2. Perform one of the following steps depending on your operating system:
   - Double-click **ODBC Data Sources**.
   - Or, double-click **Administrative Tools**, then in the Administrative Tools window click **Data Sources (ODBC)**.
3. In the ODBC Data Source Administrator window, click the **System DSN** tab and then click **Add**.
4. In the Create New Data Source window, double-click **SQL Server**.
5. In the Create a New Data Source to SQL Server window, enter a name for the ODBC data source in the Name field, select the Server from the Server pull-down menu, and then click **Next**.
6. Select the radio button **With SQL Server authentication using a login ID and password entered by the user**. Deselect the **Connect to SQL Server to obtain default settings for the additional configuration options** check box. Then click **Next**.
7. Accept the defaults and click **Next**.
8. Accept the defaults and click **Next**.
9. Click **OK**.

Note: When you create a RIM object to be used with a Microsoft SQL Server database, set the database name (using the **wcrtrim –d** option) to the name of the data source that you specified in step 5 in the preceding procedure.

Oracle considerations

The following sections list the requirements for Oracle and provide information to help you install and configure Oracle for use with Tivoli products.

Requirements for Oracle

The following requirements apply to Oracle RIM hosts:

- For Japanese, Simplified Chinese, or Korean environments, you must install the Oracle client. If you do not fully install the Oracle client on your UNIX RIM host, you might experience connectivity problems.
- The TNS_ADMIN and LOCAL variables are not supported.
- Ensure that the ORACLE_HOME variable is set to the directory where the Oracle software is installed.
- Ensure that the TWO_TASK variable is set to extract information from the tnsnames.ora file.
- If you need to store your tnsnames.ora file in a location other than $ORACLE_HOME/network/admin directory, create a link or copy the file to that directory.

Installing and configuring Oracle

Install the Oracle RDBMS on the RDBMS server. For installation details, refer to your Oracle documentation.

Complete the following steps to configure the RIM host as an Oracle client:

1. Install Oracle client software, which includes SQL*Plus, on the RIM host.
2. Use FTP in text mode to copy the tnsnames.ora file from the %ORACLE_HOME%\network\admin directory on the RDBMS server to the %ORACLE_HOME%\network\admin directory on the RIM host. %ORACLE_HOME% is the environment variable equating to the directory where your Oracle server or client installation resides.
3. Verify that the tnsnames.ora file is properly updated to reflect your configuration. In other words, check that the host name of the server, the Oracle instance ID, the port number, and the communication protocol are accurate.
4. Verify connectivity using the Oracle SQL*Plus client application.
5. After any RIM objects configured to use the Oracle client have been created, you can verify connectivity by using the **wrimtest** command. It is possible that permissions issues in the Oracle client tree may cause connectivity with **wrimtest** to fail. If it does, see the following Note.

   Note: The RIM agent runs as the user `nobody` on AIX, Solaris, and Linux, and as the user `tmersrvd` on HP-UX and Windows. Recent versions of Oracle and DB2 have tightened the security of default file permissions which prevent the `nobody` and `tmersrvd` users from accessing important runtime libraries. The solution to this problem is to change the file permissions to allow the `nobody` and `tmersrvd` users access to the files. To do this, perform the following platform-dependent steps.

UNIX interps:

For Oracle:

a. Source the Oracle environment.

b. Change the access permissions for the $ORACLE_HOME directory:

```
chmod -R 755 $ORACLE_HOME
```

For DB2:

1) Source the DB2 environment.

2) Change the access permissions for the DB2 installation directory ($DB2DIR). Note that this is not the DB2 instance directory. For example:

```
chmod -R 755 /opt/IBM/db2/V8.1
```

Windows interps:

For Oracle and DB2:

a. Right-click on the Oracle or DB2 directory and select **Properties**.

b. Select the **Security** tab.

c. Click the **Add** button.

d. Type `tmersrvd`.

e. Click **OK** (add the tmersrvd user).

f. Click **OK** again (this time, to update the properties).

g. Verify that the tmersrvd user has access to all subdirectories under the Oracle or DB2 directory.

Sybase considerations

The following sections list the requirements for Sybase and provide information to help you install and configure Sybase for use with Tivoli products.

Requirements for Sybase

The following requirements apply to Sybase RIM hosts:

- Ensure that the SYBASE variable is set to the directory where the Sybase software is installed.
- Ensure that the DSQUERY variable is set the identifier that the client uses to extract information from the interfaces file.

Installing and configuring Sybase

Install the Sybase RDBMS. For installation details, refer to your Sybase documentation.

Complete the following steps to configure the RIM

1. Install the Sybase client software, which includes isql, on the RIM host.
2. If your RIM host is a Windows operating system, ensure that the PATH variable includes the directory where the Sybase dynamic link library (DLL) files are installed.
3. If your RIM host is a UNIX system, copy the interfaces file from the RDBMS server to the directory on the RIM host where the Sybase client software is installed.

   Note: If the RDBMS server is a Solaris system and your RIM host is not or vice versa, do not copy the interfaces file to your RIM host. The interfaces file for Solaris is not compatible with other operating systems. You must create a new interfaces file that is compatible. Consult your database administrator for assistance.
4. For Sybase 12 and 12.5, use the Tivoli **odadmin environ set** command to set environment variables. The following table lists the environment variables and their required settings.

Environment Variable	Value for Version 12.0	Value for Version 12.5
SYBASE	*client_install_path*	*client_install_path*
SYBASE_OCS	OCS-12_0	OCS-12_5
PATH	*client_install_path*/OCS-12_0/ bin:$PATH	*client_install_path*/OCS-12_5/ bin:$PATH
AIX: LIBPATH	*client_install_path*/OCS-12_0/ lib:$LIBPATH	*client_install_path*/OCS-12_5/ lib:$LIBPATH
HP-UX: SHLIB_PATH	*client_install_path*/OCS-12_0/ lib:$SHLIB_PATH	*client_install_path*/OCS-12_5/ lib:$SHLIB_PATH
Solaris: LD_LIBRARY_PATH	*client_install_path*/OCS-12_0/ lib:$LD_LIBRARY_PATH	*client_install_path*/OCS-12_5/ lib:$LD_LIBRARY_PATH

5. Verify connectivity using the Sybase isql client application.

Creating RIM objects

Generally, RIM objects are created during product installation. However, there might be times where you need to create a RIM object. The only way to create a RIM object is by using the **wcrtrim** command. For a complete list of options for the **wcrtrim** command, refer to the *Tivoli Management Framework Reference Manual*.

Moving RIM objects

Use the **wmvrim** command to move a RIM object from one supported managed node to another. When moving a RIM object, ensure that the new, hosting managed node is correctly preconfigured. The following is the general syntax for using the **wmvrim** command to create a RIM object:

wmvrim {**–h** *host_name* | **–o** *host_oid*} [**–H** *db_home*] [**–I** *instance_home*] *rim_name*

For a complete list of options for the **wmvrim** command, refer to the *Tivoli Management Framework Reference Manual*.

Changing the password for RIM objects

Use the **wsetrimpw** command to change the default or current password to the password used by the database, or instance, owner. The instance owner is the user that was specified during the creation of the database on the RDBMS server. The following is the general syntax for using the **wsetrimpw** command to create a RIM object:

wsetrimpw *rim_name old_password new_password*

For a complete list of options for the **wsetrimpw** command, refer to the *Tivoli Management Framework Reference Manual*.

Modifying RIM objects

Use the **wgetrim** and **wsetrim** commands to retrieve and set the attributes for a RIM object. The **wgetrim** command retrieves the current values set for the attributes of a RIM object, and the **wsetrim** command sets one or more attributes of a RIM object.

The following is the general syntax for using the **wgetrim** command to create a RIM object:

wgetrim *rim_name*

Note: The **wgetrim** command does not retrieve the application type (**–a** option) or the maximum number of connections (**–m** option) for a RIM object. To retrieve these attributes, use the **idlcall** command as documented within the **wgetrim** documentation.

The following is the general syntax for using the **wsetrim** command to create a RIM object:

wsetrim [**–n** *name*] [**–d** *database*] [**–u** *user*] [**–H** *db_home*] [**–s** *server_id*] [**–I** *instance_home*] [**–t** *instance_name*] [**–a** *application_label*] [**–m** *max_connections*] *rim_name*

For a complete list of options for the **wgetrim**, and **wsetrim** commands, refer to the *Tivoli Management Framework Reference Manual*.

Deleting RIM objects

The following is the general syntax for using the **wdel** command to create a RIM object:

wdel [**–I**] *label*...

For a complete list of options for the **wdel** command, refer to the *Tivoli Management Framework Reference Manual*.

Chapter 19. Creating InstallShield response files

You can use the InstallShield installations on the Tivoli Management Framework 1 of 2 CD to create custom silent installation for Windows and NetWare endpoints, NetWare gateways, and Tivoli Desktop for Windows. You can use your custom installation to install these resources without user interaction.

The high-level process for creating custom installation for InstallShield images is as follows:

- To record a silent InstallShield installation, run the **setup –r** command.
- To play back a silent InstallShield installation, run the **setup –s** command.

For an example of this procedure, refer to "Installing the Tivoli desktop on Windows operating systems using a response file" on page 66.

Recording the response file

To start recording the response file, perform the following steps:

1. Change to the directory containing the setup script.
2. Run the following command:

   ```
   setup -r -f1my_response_file.iss
   ```

   where *my_response_file* is any name you want to use. Remember that `-f1` uses the number 1 instead of the lowercase letter L.

   For example, to create a response file for endpoint installation, you can call it endpoint.iss and store it in the c:\tivoli\endpoint\ directory. In this case, enter:

   ```
   setup -r -f1c:\tivoli\endpoint\endpoint.iss
   ```

   This action creates the response file in the designated location.

   Note: If no location is specified by the `-f1` option, a setup.iss file is written to the %SystemRoot%\windir directory for Windows operating systems. For example on Windows NT, the directory might be c:\winnt; for Windows 98, it might be c:\windows.
3. Complete the setup windows, actually completing a normal installation. The response file records all the setting specified.

At this point, you can install the endpoint in interactive mode, using **setup.exe**, or in unattended (or silent) mode, by pointing to the endpoint.iss file during playback.

Note: There is also an **–f2** option that is used to designate the location of the InstallShield log file. The InstallShield log file is created during the installation process. If no location is specified, the InstallShield log file is written to the same location as the response file.

Playing back a response file

After you have created your response file, you can run the installation. You can also copy the files to a network drive so that users can map to this drive and run the installation in interactive or unattended mode (depending on your business practices). When running the installation in unattended mode, no messages are displayed. All messages are written to the log file. Review this log file to determine the results of the installation.

To move the installation image to a network drive, perform the following steps:

1. Copy the entire installation directory to a network drive.
2. Copy the response file to this network drive.
3. Create a shortcut to the **setup.exe** file.
4. Rename the shortcut to something easily understood. In this case, consider **endpoint.exe**
5. Modify the properties of the shortcut as follows (assume the new name):
   a. Right-click **endpoint.exe** and select Properties. The Properties window is displayed.
   b. In the Target field, add `-s -f1`*`response_file_path`*`\endpoint.iss` to the end of the command, where *response_file_path* is the full path to the response file.

      For example, the response file is stored in the tivoli\endpoint directory, the information in the Target field would be as follows:

      ```
      "c:\tivoli\endpoint\setup.exe" -s -f1c:\tivoli\endpoint\endpoint.iss
      ```
6. Instruct users to map to this location and select the **endpoint.exe** file to install the endpoint.

Note: With this design, users can run the command in interactive or unattended mode. To run in interactive mode, users select **endpoint.exe**. To run in unattended mode, users select **endpoint.exe**.

Alternatively, users can run the command from the command line. The general instructions to run a custom installation is to change to the directory containing the image and run the following command:

```
setup -s [-f1response_file] [-f2IS_log_file]
```

where:

–s
Specifies that the installation is to run in unattended mode.

–f1*response_file*
Specifies the file name and location of the customized response (.ISS) file. Do not include a space between **–f1** and *response_file*. If you do not use this option, the default setup.iss file is used.

–f2*IS_log_file*
Specifies the file name and location where the InstallShield log files will be written. Do not include a space between **–f2** and *IS_log_file*. If you do not use this option, the default setup.log file is created.

Chapter 20. Uninstalling a Tivoli environment

This chapter provides information on uninstalling Tivoli managed resources and Tivoli Enterprise product from your Tivoli environment.

This chapter contains the following information:

- An overview of the uninstallation process
- Determining what products and patches are installed
- Determining product tags
- Uninstalling products
- Uninstalling Tivoli Authentication Package
- Removing Tivoli Remote Execution Service
- Uninstalling Tivoli managed resources, which includes endpoints, gateways, managed nodes, and the Tivoli server

Overview of the uninstallation process

This section provides an overview of the steps for the following types of uninstallation:

- Removing a Tivoli product from a Tivoli region
- Removing a system from a Tivoli region
- Removing a Tivoli region

Deleting Tivoli files from a system does not completely uninstall a product or resource. Follow the complete procedures described in this chapter to ensure that the Tivoli object database and operating system files are updated.

Removing a Tivoli product from a Tivoli region

Before uninstalling a Tivoli product, refer to its documentation for complete instructions. The documentation provides information such as the following:

- Information about where the components of the product are installed

  Some products are installed only on the Tivoli server or on the Tivoli server and gateways. Others have components that are installed directly on individual endpoints or managed nodes.
- A list of components that must be uninstalled and the order in which to uninstall them
- Instructions for any manual cleanup that is required
- Any differences from the standard uninstallation process described in this chapter

Some Tivoli products can be uninstalled completely using the **wuninst** command. Other products require you to run other uninstallation scripts or to manually remove directories and files. Products that are installed directly on an endpoint or on a NetWare gateway often require manual cleanup.

Details about uninstalling an individual product component are located in "Uninstalling a product" on page 257.

Removing a system from a Tivoli region

To completely remove a system other than the Tivoli server from a Tivoli region, perform the following high-level steps:

1. If any Tivoli products have components that are installed on the system, uninstall each component as described in its documentation.
2. If the system is an endpoint, uninstall the endpoint and remove it from the Tivoli object database as described in "Deleting and uninstalling endpoints" on page 258.
3. If the system is a gateway, delete the gateway as described in "Uninstalling gateways" on page 263.
4. If the system is a managed node, uninstall it as described in "Uninstalling managed nodes" on page 264.

Removing a Tivoli region

To completely remove a Tivoli region, perform the following high-level steps:

1. Uninstall each Tivoli product as outlined in "Removing a Tivoli product from a Tivoli region" on page 255.
2. Uninstall all endpoints and remove them from the Tivoli object database as described in "Deleting and uninstalling endpoints" on page 258.
3. Delete all gateways as described in "Uninstalling gateways" on page 263.
4. Uninstall all managed nodes as described in "Uninstalling managed nodes" on page 264.
5. Uninstall the Tivoli server as described in "Uninstalling Tivoli servers" on page 265.

Determining what products and patches are installed

To determine what products and patches are installed in a Tivoli region, use one of the following methods:

- From the Tivoli desktop, click **Desktop → About**.
- From the command line, use the **wlsinst** command. For more information, see *Tivoli Management Framework Reference Manual*.

Determining product tags

When uninstalling a Tivoli product, you must know the product tag for each separately installable component of the product.

To determine the tag to use when uninstalling a product or product component, use the following command:

```
wuninst -list
```

For more information, see *Tivoli Management Framework Reference Manual*.

Determining where a product is installed

To determine the systems on which Tivoli products are installed, use the following command:

```
wlsinst -p -h
```

To determine the systems from which a product can be uninstalled, use the following command:

```
wuninst tag -list
```

where *tag* is a product tag displayed by **wuninst –list**.

For more information about the **wlsinst** and **wuninst** commands, see *Tivoli Management Framework Reference Manual*.

Uninstalling a product

Tivoli Management Framework provides a command line utility to remove Tivoli products from a specified machine or from the entire Tivoli region. The **wuninst** command is a wrapper script that invokes product-specific scripts for uninstalling its components.

See *Tivoli Management Framework Reference Manual* for detailed information about the **wuninst** command.

Note: You need to review the product documentation for details on the order in which to uninstall the components of a specific product. Additionally, the product documentation can contain specific post-uninstallation steps for specific components.

The **wuninst** command has the following syntax:

wuninst *tag node_name* [**–rmfiles**] [**–all**]

where:

tag
Specifies the registered product tag for the component to be uninstalled. To determine the tag for a specific component, refer to the product documentation or use the **wuninst –list** command.

node_name
Specifies the managed node to which this request is directed. This can be any managed node in the region. If you specify the Tivoli server, the actions take place on all managed nodes in your region.

–rmfiles
Deletes binaries, libraries, man pages, objects, methods, and potentially other files associated with the product.

–all
Removes the product from the entire Tivoli region.

After uninstalling a component, use the **wchkdb –ux** command to update the Tivoli object database.

For example, the following commands uninstall the Tivoli Software Installation Service depot and remove the associated file from the managed node oak, and then update the Tivoli object database:

```
wuninst SISDepot oak -rmfiles
wchkdb -ux
```

Note: You cannot use the **wuninst** command to uninstall products from NetWare gateways. This command runs a shell script on the machine, but NetWare systems do not have a shell interpreter. To uninstall a product from a NetWare machine, refer to the documentation for that product.

Uninstalling Tivoli Authentication Package

To uninstall Tivoli Authentication Package, perform the following steps:

1. Disable Tivoli Authentication Package by entering one of the following commands:
   - On an endpoint, enter the following command:
     ```
     wlcftap -d -r ""
     ```
   - On a Tivoli server or managed node, enter the following command:
     ```
     wsettap -d -r ""
     ```
2. Reboot the system. The reboot is necessary to unload Tivoli Authentication Package, which is loaded by the kernel and cannot be removed while the system is running.
3. After the system is booted, delete the following file:

   %SystemRoot%\system32\TivoliAP.dll

Removing Tivoli Remote Execution Service

To remove Tivoli Remote Execution Service, perform the following steps:

1. Stop the Tivoli Remote Execution Service service by entering the following command:
   ```
   net stop trip
   ```

   Note: Alternatively, you can stop this service from the Services window.
2. After the service is stopped, enter the following command to remove the service:
   ```
   trip -remove
   ```
   If the previous command fails, you will need to use the Registry Editor. When using the Registry Editor, exercise extreme caution. Errors can cause the system to become unusable.

   To uninstall Tivoli Remote Execution Service using the Registry Editor, perform the following steps:
   a. Open the Registry Editor by entering the following command:
      ```
      regedit
      ```
   b. Delete the following directories (hives) entries under the KEY_LOCAL_MACHINE directory:

      SYSTEM\CurrentControlSet\Services\trip
   c. Exit the Registry Editor.

Uninstalling resources

This section contains the instructions for uninstalling managed resources from your Tivoli environment.

Deleting and uninstalling endpoints

Deleting and uninstalling endpoints are different operations. Uninstalling an endpoint removes the Tivoli Management Framework files from the endpoint. Deleting an endpoint removes the endpoint information from the Tivoli object database. The following sections provide more information.

Deleting endpoints from the Tivoli object database

The **wdelep** command removes the specified endpoints from the Tivoli object database. The **wdelep** command with the **–d** option stops the endpoint service and removes the specified endpoints from the Tivoli object database.

The endpoints that the **wdelep** command is run against no longer appear on the desktop and no longer appear in reports. However, the endpoint software remains on these machines for future use.

To delete the endpoint software from the machine, you must uninstall it. Procedures for uninstalling endpoints are provided in the following section.

For additional information about the **wdelep** command, refer to the *Tivoli Management Framework Reference Manual*.

Uninstalling endpoints

Uninstalling an endpoint erases the endpoint software on the machine where it was installed. However, uninstalling the software from the machine does not delete its associated information from the Tivoli object database. To delete the endpoint from the object database, refer to "Deleting endpoints from the Tivoli object database."

The general procedure for uninstalling any endpoint involves the following:

- Stopping the endpoint service
- Erasing the endpoint directories and files on the client computer
- Removing startup statements from system configuration files on the client computer

Different platform types require different specific steps for uninstalling endpoints. The next sections describe how to uninstall UNIX and PC endpoints.

Uninstalling endpoints from UNIX platforms: The general procedure for removing endpoints is the same on all UNIX platforms, but the specific configuration files and directories vary. The next sections provide removal steps specific to different UNIX platforms.

Note: The directories specified in the following procedures assume the default locations.

Uninstalling endpoints from AIX systems: To uninstall an endpoint on an AIX machine, perform the following steps:

1. Stop the endpoint daemon using one of the following methods:
   - Enter the **/opt/Tivoli/lcf/dat/1/lcfd.sh stop** command.
   - Find the process identification (PID) of the daemon and enter the **kill** command against it.
2. Remove the endpoint installation directory and subdirectories. The default location is /opt/Tivoli/lcf.
3. Remove the endpoint environment directory, subdirectory, and files. The directory is /etc/Tivoli/lcf. The environment files are lcf_env.sh and lcf_env.csh.
4. Remove the endpoint startup entry in /etc/inittab using the following command:

   ```
   /usr/sbin/rmitab rctma1
   ```

5. Remove the /etc/rc.tma1 and /etc/inittab.before.tma1 files.
6. Remove the /etc/Tivoli/*/userlink.htm file.

Uninstalling endpoints from HP-UX systems: To uninstall an endpoint on an HP-UX machine, perform the following steps:

1. Stop the endpoint daemon using one of the following methods:
   - Enter the **/opt/Tivoli/lcf/dat/1/lcfd.sh stop** command.
   - Find the process identification (PID) of the daemon and enter the **kill** command against it.
2. Remove the endpoint installation directory and subdirectories. The default location is /opt/Tivoli/lcf.
3. Remove the endpoint environment directory, subdirectory, and files. The directory is /etc/Tivoli/lcf. The environment files are lcf_env.sh and lcf_env.csh.
4. Remove the /sbin/init.d/lcfd1.sh file, which is the endpoint startup script.
5. Remove the following symbolic links:

   ```
   /sbin/rc0.d/K100Tivoli_lcf1
   /sbin/rc1.d/K100Tivoli_lcf1
   /sbin/rc2.d/K100Tivoli_lcf1
   /sbin/rc3.d/S500Tivoli_lcf1
   ```
6. Remove the /etc/Tivoli/*/userlink.htm file.

Uninstalling Endpoints on Solaris systems: To uninstall an endpoint on a Solaris machine, perform the following steps:

1. Stop the endpoint daemon using one of the following methods:
   - Enter the **/opt/Tivoli/lcf/dat/1/lcfd.sh stop** command.
   - Find the process identification (PID) of the daemon and enter the **kill** command against it.
2. Remove the endpoint installation directory and subdirectories. The default location is /opt/Tivoli/lcf.
3. Remove the endpoint environment directory, subdirectory, and files. The directory is /etc/Tivoli/lcf. The environment files are lcf_env.sh and lcf_env.csh.
4. Remove the /etc/init.d/lcfd1.rc file, which is the endpoint startup script.
5. Remove the following symbolic links:

   ```
   /etc/rc0.d/K50Tivoli_lcf1
   /etc/rc1.d/K50Tivoli_lcf1
   /etc/rc2.d/K50Tivoli_lcf1
   /etc/rc3.d/S99Tivoli_lcf1
   ```
6. Remove the /etc/Tivoli/*/userlink.htm file.

Uninstalling endpoints from Linux platforms: The general procedure for removing endpoints is the same as removing endpoints from UNIX platforms. The specific configuration files and directories vary. The next sections provides removal steps specific to the different Linux platforms. The procedures describe how to use the command line to remove the endpoint.

Note: The directories specified in the following procedures assume you installed the endpoint in the default location.

Uninstalling endpoints from Linux for Intel systems: To uninstall an endpoint from a Linux for Intel system, perform the following steps:

1. Stop the endpoint service using the **lcfd.sh stop** command.
2. Remove the installation directory:
   ```
   rm -r /opt/Tivoli/lcf/1
   ```
3. Remove the endpoint environment directory:
   ```
   rm -r /etc/Tivoli/lcf/1
   ```
4. Remove the /etc/Tivoli/*/userlink.htm file.
5. Remove the following autostart files and symbolic links from the system, if they exist:

   /etc/rc.d/init.d/lcfd1.rc
   /etc/rc.d/init.d/lcfd1.sh
   /etc/rc.d/init.d/Tivoli_lcfd1
   /etc/rc.d/rc0.d/K10Tivoli_lcf1
   /etc/rc.d/rc1.d/K10Tivoli_lcf1
   /etc/rc.d/rc2.d/S99Tivoli_lcf1
   /etc/rc.d/rc3.d/S99Tivoli_lcf1
   /etc/rc.d/rc4.d/S99Tivoli_lcf1
   /etc/rc.d/rc5.d/S99Tivoli_lcf1
   /etc/rc.d/rc6.d/S99Tivoli_lcf1

Uninstalling endpoints from SuSE systems (Intel and zOS): To uninstall an endpoint from Intel and zOS SuSE system, perform the following steps:

1. Stop the endpoint service using the **lcfd.sh stop** command.
2. Remove the installation directory:
   ```
   rm -r /opt/Tivoli/lcf/1
   ```
3. Remove the endpoint environment directory:
   ```
   rm -r /etc/Tivoli/lcf/1
   ```
4. Remove the /etc/Tivoli/*/userlink.htm file.
5. Remove the following autostart files and symbolic links from the system, if they exist:

   /etc/rc.d/init.d/Tivoli_lcfd1
   /etc/rc.d/rc2.d/K10Tivoli_lcf1
   /etc/rc.d/rc2.d/S99Tivoli_lcf1
   /etc/rc.d/rc3.d/K10Tivoli_lcf1
   /etc/rc.d/rc3.d/S99Tivoli_lcf1
   /etc/rc.d/rc4.d/K10Tivoli_lcf1
   /etc/rc.d/rc4.d/S99Tivoli_lcf1
   /etc/rc.d/rc5.d/K10Tivoli_lcf1
   /etc/rc.d/rc.5.d/S99Tivoli_lcf1

Uninstalling endpoints from PC platforms: The general procedure for removing endpoints is the same on all PC platforms, but the specific configuration files and directories vary. The next sections provide removal steps specific to different PC platforms. The procedures describe how to use the command line to remove the endpoint. In some cases, you might have an alternate choice of removing the endpoint with an uninstall program.

Most PC platforms provide an option to create desktop icons when installing an endpoint. One of the icons is for uninstalling the endpoint. If there is an uninstall icon, use it instead of the command line procedure.

Note: The directories specified in the following procedures assume you installed the endpoint in the default location.

Uninstalling endpoints on Windows NT, Windows 2000, and Windows XP systems: The procedure for uninstalling an endpoint on Windows NT, Windows 2000, and Windows XP systems depends on which of the following methods was used to install it:

- If you installed the endpoint using InstallShield, uninstall it using the **uninst.bat** command in the %SystemDrive%\Program Files\Tivoli\lcf directory.
- If you installed the endpoint using Tivoli Software Installation Service or the **winstlcf** command, uninstall it by performing the following steps:
  1. Stop and remove the endpoint service by entering the following command on one line:

     ```
     %SystemDrive%\Program Files\Tivoli\lcf\bin\w32-ix86\mrt\lcfd -r "instance"
     ```

     where *instance* is the name of the endpoint service associated with the endpoint to be removed.
  2. Enter the following command:

     ```
     %SystemDrive%\Program Files\Tivoli\lcf\bin\w32-ix86\mrt\lcfep -s
     ```

     Note: Tivoli Management Framework does not support the **lcfep** process in the environments where Windows Terminal Servers, Remote Desktop, or Fast User Switching is installed. **lcfep** checks for the presence of these programs when it is started, and if any of them exist in the registry, the program immediately exits. **lcfep.exe** will not show in the list of running processes, and its Task Bar icon will not be displayed. This generally affects any ″server″ edition of Windows (for example, Windows 2000 Server, Windows 2003 Server, Windows 2008 Server) and newer versions of Windows for workstations, such as Windows XP and Windows Vista. Disabling these services does not allow **lcfep** to run; they must be uninstalled before **lcfep** can function.
  3. Remove the endpoint installation directory and subdirectories. The default directory is c:\Program Files\Tivoli\lcf\instance.
  4. Remove the endpoint environment files, subdirectory, and directory. The default directory is %SystemDrive%\Program Files\Tivoli\lcf\instance and the environment files are lcf_env.sh and lcf_env.cmd.
  5. Remove the %SystemRoot%\etc\tivoli\instance\c\userlink.htm file.

Uninstalling endpoints on Windows 98 systems: The procedure for uninstalling an endpoint on Windows 95 and Windows 98 systems depends on which of the following methods was used to install it:

- If the endpoint was installed using InstallShield, uninstall it using the **uninst.bat** file in the %SystemDrive%\Program Files\Tivoli\lcf directory.
- If the endpoint was installed using Tivoli Software Installation Service through the PC agent, uninstall it by performing the following steps:
  1. Stop and remove the endpoint process by entering the following command on one line:

     ```
     %SystemDrive%\Program Files\tivoli\lcf\bin\win95\mrt\
     lcfd -r "lcfd"
     ```
  2. From the same location, enter the following command:

     ```
     lcfep -s
     ```
  3. Remove the endpoint installation directory and subdirectories. The default location is %SystemDrive%\Program Files\Tivoli\lcf.

4. Remove the endpoint environment file, subdirectory, and directory. The directory is %SystemRoot%\Tivoli\lcf. The environment file is **lcf_env.bat**.
5. Edit the Windows Registry:
   a. Start the **regedit** program from the Run window.
   b. Expand the registry to the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion \RunServices key.
   c. Delete the lcfd entry.
6. Remove the %SystemDrive%\Etc\Tivoli\C\userlink.htm file.

Uninstalling endpoints on OS/2 systems: The procedure for uninstalling an endpoint on OS/2 systems depends on which of the following methods was used to install it:

- If the endpoint was installed using the OS/2 installer, remove it by entering the following command:
  ```
  %BootDrive%\Tivoli\lcf\uninstal.cmd
  ```
- If the endpoint was installed using Tivoli Software Installation Service, remove it by performing the following steps:
  1. Stop and remove the endpoint process by entering the following command on one line:
     ```
     %BootDrive%\Tivoli\lcf\bin\os2-ix86\mrt\wos2proc -alcfd.exe
     ```
  2. Remove the endpoint installation directory and subdirectories. The default location is %BootDrive%\Tivoli\lcf.
  3. Remove the endpoint environment files, subdirectory, and directory. The directory is %BootDrive%\os2\Tivoli\lcf. The environment files are **lcf_env.cmd** and **lcf_env.sh**.
  4. Remove the Tivoli endpoint icon from the **Startup** folder.
  5. Remove the %BootDrive%\Etc\Tivoli\C\userlink.htm file.

Uninstalling endpoints on NetWare systems: To remove an endpoint on a NetWare computer, perform the following steps:

1. Stop and remove the endpoint process by entering the following command at a NetWare server console:
   ```
   lcfstop
   ```
2. Remove the endpoint installation directory and subdirectories. The default location is SYS:\Tivoli\lcf.
3. Remove the endpoint environment directory, subdirectory, and file. The directory is SYS:\system\Tivoli\lcf.
4. Remove the endpoint startup files, lcf.ncf and lcf.sys. The files are in the SYS:\system directory. Do not remove the directory.
5. Edit the NetWare SYS:\system\autoexec.ncf file to remove the line that starts the lcf.ncf script.
6. Delete from the LCFRSRVD account the user rights to access directories on the NetWare server where the endpoint resides. If no other NetWare server in your Tivoli environment is using the LCFRSRVD account, delete it using the **nwadmin32** utility.

Uninstalling gateways

Depending on the operating system running on the gateway, there are different procedures for uninstalling.

Uninstalling UNIX and Windows gateways

The **wdelgate** command deletes a UNIX and Windows gateway from your Tivoli environment. After deleting a gateway, run the **wchknode** command to remove reference to the deleted gateway. For additional information about the **wdelgate** and **wchknode** commands, see *Tivoli Management Framework Reference Manual*.

Uninstalling NetWare gateways

To uninstall a NetWare gateway from your Tivoli environment, perform the following steps:

1. From the Tivoli server, enter the following command:

   ```
   wrmnode managed_node
   ```

   where *managed_node* is the host name of the NetWare server.
2. On the Tivoli server, enter the following command to update the object database:

   ```
   wchkdb -ux
   ```
3. On the NetWare server, enter the following command to stop the object dispatcher:

   ```
   oservend
   ```
4. On the NetWare server, remove the following directories:
   - SYS:tivoli
   - SYS:public\tivoli

   Note: These are the default directories. If you specified different directories, remove them.
5. Delete from the TMERSRVD account the user rights to access directories on the NetWare server where the gateway resides. If no other NetWare server in your Tivoli environment is using the TMERSRVD account, delete it using the **nwadmin32** utility.

Uninstalling managed nodes

The **wunstmn** command removes the managed node entry from the database and optionally removes the Tivoli Management Framework files from a managed node.

Note: You cannot use this procedure to uninstall a NetWare managed node. To uninstall a NetWare gateway, see "Uninstalling NetWare gateways."

To completely uninstall an OS/2, Windows, or UNIX managed node, perform the following steps:

1. Log in to the Tivoli server or a managed node not to be uninstalled as an administrator with the **senior** or **super** authority.
2. Run the **setup_env** command appropriate for your operating system or shell. For details, refer to "Setting Tivoli environment variables" on page 30.
3. Use the **wunstmn** command to uninstall the managed node. For example, to uninstall managed node marquet, enter the following command:

   ```
   $BINDIR/TAS/UNINST/wunstmn -A marquet
   ```

   For additional information about **wunstmn**, see *Tivoli Management Framework Reference Manual*.
4. Run the **wchkdb –ux** command to clean up all references to the deleted managed node. For additional information about **wchkdb**, see *Tivoli Management Framework Reference Manual*.

5. For Windows managed nodes, perform the following steps on the managed node:
   a. Uninstall the object dispatcher by entering the following command:
      ```
      oinstall -remove
      ```
      If the previous command fails, you must manually edit the registry. When editing the registry, exercise extreme caution. Errors can cause the system to become unusable.

      To manually uninstall the object dispatcher, perform the following steps:
      1) Open the registry by entering the following command:
         ```
         regedit
         ```
      2) Delete the following directories (hives) entries under the HKEY_LOCAL_MACHINE directory:

         SOFTWARE\Tivoli\Platform
         SOFTWARE\Tivoli\TivoliManagementFramework
         SYSTEM\CurrentControlSet\Services\oserv
      3) Exit the Registry Editor.
   b. Delete the following directories (assuming default installation location):

      \Tivoli
      %SystemRoot%\system32\drivers\etc\Tivoli
   c. From **Start**, select **Programs** → **Administrative Tools** → **User Manager**. The User Manager window is displayed.
   d. Delete the following accounts:

      Tivoli_Admin_Privileges
      tmersrvd
   e. Uninstall the Tivoli Remote Execution Service (TRIP) service as described in "Removing Tivoli Remote Execution Service" on page 258.
   f. Uninstall the Tivoli Authentication Package (TAP) as described in "Uninstalling Tivoli Authentication Package" on page 258.

Uninstalling Tivoli servers

The following sections contain the procedures from uninstalling a Tivoli server for Windows operating systems and UNIX operating systems.

Uninstalling a Tivoli server from a Windows operating system

To uninstall a Tivoli server from a Windows operating system, perform the following steps:

1. Log in to the Windows operating system as a user with root principal authority in the Tivoli region. This user must also be a member of the Administrators security group. In most cases, you can log in as Administrator.
2. Open an MS-DOS command window.
3. Set the Tivoli environment variables by entering the following command:
   ```
   %SystemRoot%\system32\drivers\etc\Tivoli\setup_env.cmd
   ```
4. Stop the object dispatcher by entering the following command:
   ```
   net stop oserv
   ```
5. Stop the Tivoli Remote Execution Service service by entering the following command:
   ```
   net stop trip
   ```

6. Uninstall the object dispatcher by entering the following command:

   ```
   oinstall -remove
   ```

   If the previous command fails, you must use the Registry Editor. When using the Registry Editor, exercise extreme caution. Errors can cause the system to become unusable.

   To uninstall the object dispatcher using the Registry Editor, perform the following steps:

   a. Open the Registry Editor by entering the following command:

      ```
      regedit
      ```

   b. Delete the following directories (hives) entries under the HKEY_LOCAL_MACHINE directory:

      SOFTWARE\Tivoli\Platform
      SOFTWARE\Tivoli\TivoliManagementFramework
      SYSTEM\CurrentControlSet\Services\oserv

   c. Exit the Registry Editor.

7. Uninstall the Tivoli Remote Execution Service service as described in “Removing Tivoli Remote Execution Service” on page 258.

8. Uninstall the Tivoli Authentication Package as described in “Uninstalling Tivoli Authentication Package” on page 258.

9. Delete the following directories (assuming default installation location):

   \Tivoli
   %SystemRoot%\system32\drivers\etc\Tivoli

10. From **Start**, select **Programs → Administrative Tools → User Manager**. The User Manager window is displayed.

11. Delete the following accounts:

    Tivoli_Admin_Privileges
    tmersrvd

12. Reboot the system. The reboot is necessary to unload Tivoli Authentication Package, which is loaded by the kernel and cannot be removed while the system is running.

13. After the system is booted, delete the following file:

    %SystemRoot%\system32\TivoliAP.dll

Uninstalling a Tivoli server from a UNIX operating system

To uninstall a Tivoli server for a UNIX operating system, perform the following steps:

1. Log in to the UNIX operating system as root.

2. Set the Tivoli environment variables by entering one of the following command:

   For sh:

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   For csh:

   ```
   . /etc/Tivoli/setup_env.csh
   ```

3. Stop the object dispatcher by entering the following command:

   ```
   odadmin shutdown all
   ```

4. Delete the following directories recursively (assuming default installation locations):

```
/usr/local/Tivoli
/usr/lib/X11/app_defaults/Tivoli
/var/spool/Tivoli
/etc/Tivoli
```

5. Edit or delete the files that were modified or added when the Tivoli server was installed. For details about which files were modified or added, see “Files modified” on page 386.

Part 5. Scenario and additional considerations

Chapter 21. Installation scenario

This chapter combines the techniques from this book to create a small Tivoli region that consists of six machines and two Tivoli products.

Note: The information in this chapter is intended as an overview of the installation process only. Not all of the products or platforms shown are currently supported.

This chapter describes the planned Tivoli region in detail and discusses installation issues and alternatives. This region is designed to encounter planning and installation pitfalls. For example, the region uses a UNIX Tivoli server but contains both UNIX and Windows NT managed nodes. This introduces additional steps to the installation process. These issues are discussed as they are encountered.

The chapter contains several examples, each describing how to create the same region using one of the following methods:

- The Tivoli Software Installation Service console
- Tivoli Software Installation Service commands and response files
- The Tivoli desktop
- Tivoli Management Framework commands

Some parts of the scenario are independent of which installation method is used. For example, the planned region includes a Windows 98 endpoint, which can only be installed using a platform-specific method.

An administrator can deploy a region using any combination of Tivoli Software Installation Service, the Tivoli desktop, and Tivoli commands. Each example in this scenario uses only one of the primary ways to deploy Tivoli Enterprise software, except where a platform-specific method is required. In an actual deployment, an administrator can combine the methods.

After installing the Tivoli server, Tivoli Enterprise software is deployed in two phases. The first phase creates the infrastructure of the region, which includes managed nodes, gateways, and endpoints. After the first phase, a backup is made. The second phase installs the Tivoli Enterprise applications that manage the IT environment. This two-phase approach is suggested when creating a region.

Planning the Tivoli region for the scenario

Creating a Tivoli environment requires planning. For this scenario, planning consists of defining the following information:

- The machines and operating systems to be used
- The managed resources for each machine
- The Tivoli Enterprise software for each machine
- The installation options for each resource and application
- The policy regions for the resources
- The Tivoli administrators for managing the environment

Additional planning is required to use Tivoli Enterprise applications. This planning includes defining profile manager and subscription hierarchies, creating profiles, distributing profiles to subscribers, and application-specific planning. These planning activities are outside of the scope of an installation scenario and are not described in this chapter.

The planned Tivoli region uses the following deployment conventions:

- An endpoint is installed on each machine in the Tivoli region.

  This is a recommended practice. Although not used in this Tivoli region, Tivoli Software Distribution, Version 4.0, can only distribute software packages to endpoints.
- The label for each endpoint is *hostname*-ep.

  This is a recommended practice. It is important that a managed node and endpoint installed on the same machine have different labels. It is easier to control the label of an endpoint than of a managed node.
- Tivoli Enterprise software is installed to a separate, local file system on each machine.

  This is a recommended practice. Using a local file system is important for performance reasons. Using separate file systems makes it easier to manage the allocation of space between the operating system, Tivoli Enterprise software, and other applications.
- All files are installed in the same directories independent of operating system. For Windows systems, the directories are on the c: drive.

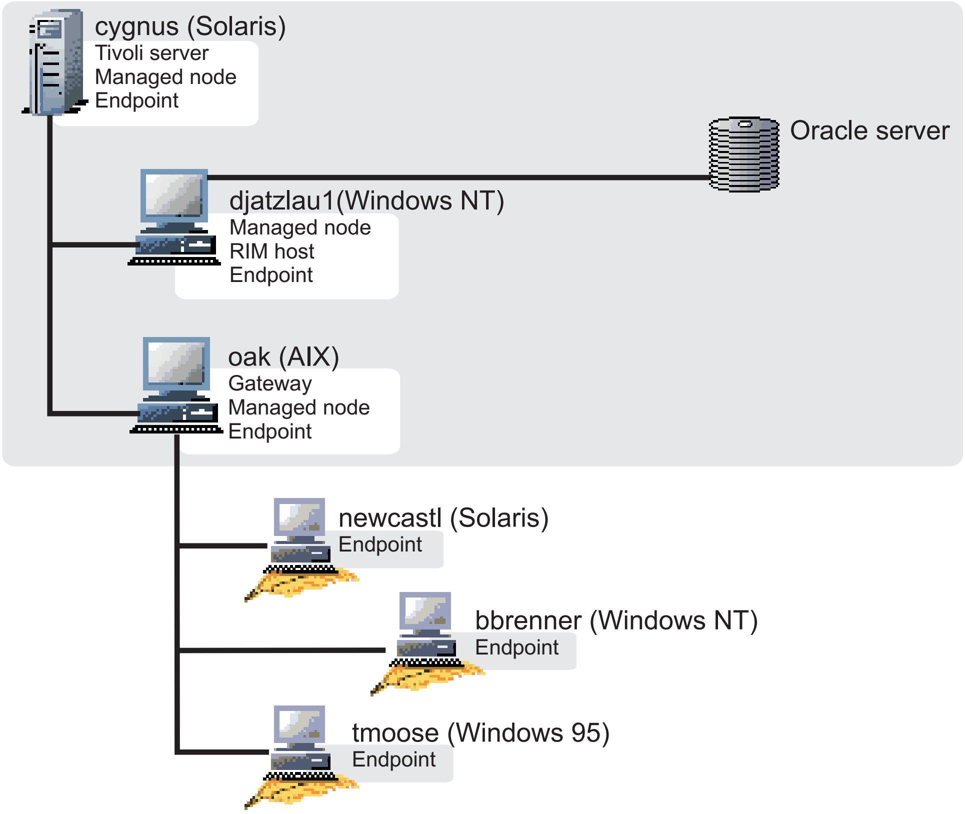
  This is a suggested practice. Using consistent directories simplifies the installation and maintenance processes.
- The default policy region, cygnus-region that is created during the installation of the Tivoli server, is used for all managed resources.

  In a real-world deployment, multiple policy regions are used to organize managed resources and application resources into logical groups.
- Endpoints are not assigned to a policy region.

  This is a suggested practice. Endpoints are created in the endpoint manager and do not need to be placed in a policy region.

Machines and managed resources

The machines in the Tivoli region include a set of server-class machines, which are located in a secure area, and a workstation in the office of each administrator. The following figure illustrates the machines in the region and the managed resources that will be installed on each machine.



The following table lists the machines in the planned Tivoli region, the operating system and interpreter type of each, and the managed resources to be installed on each. It also describes the role each machine will serve in the region.

Table 10. Configuration and role of each machine

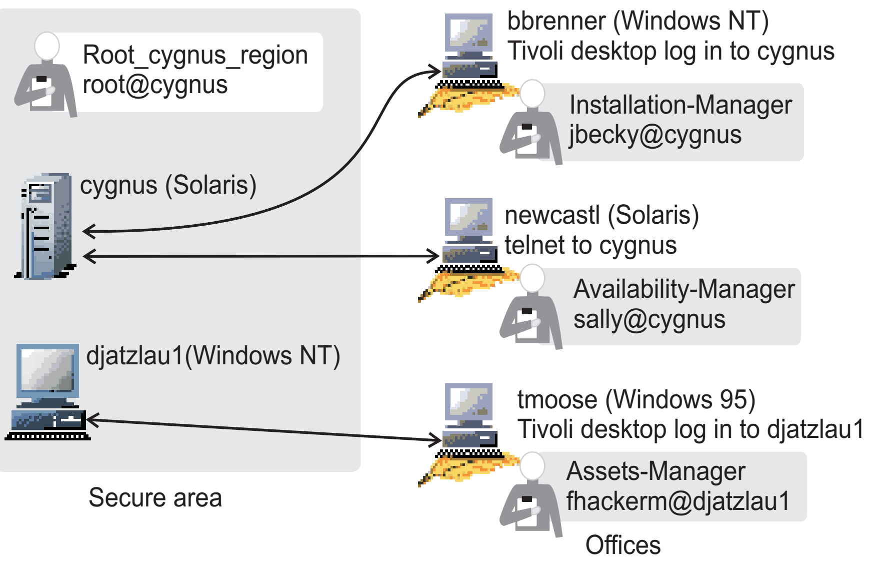
Host Name	Operating System (Interpreter Type)	Resource Type	Additional Comments
cygnus	Solaris (solaris2)	TME_server	• Hosts the endpoint manager • Mail server for the Tivoli region
		ManagedNode	• Remote host for the installation administrator • Remote host for the availability administrator • Tivoli Software Installation Service depot • Tivoli Software Installation Service client
		Endpoint	• Logs in to the gateway on oak

Table 10. Configuration and role of each machine (continued)

Host Name	Operating System (Interpreter Type)	Resource Type	Additional Comments
djatzlau1	Windows NT (w32-ix86)	ManagedNode	• Tivoli Remote Execution Service needs to be installed before installing the managed node • Windows NT repeater for installing other Windows NT machines • RIM host for Tivoli Inventory • Oracle client is already installed and connects to an Oracle server on a non-Tivoli machine • Remote host for the inventory administrator
		Endpoint	• Logs in to the gateway on oak
oak	AIX (aix4-r1)	ManagedNode	• Hosts the gateway
		Gateway	• Gateway for all endpoints
		Endpoint	• Logs in to the gateway on oak
newcastl	Solaris (solaris2)	Endpoint	• Logs in to the gateway on oak
bbrenner	Windows NT (w32-ix86)	Endpoint	• Logs in to the gateway on oak • Runs the Tivoli desktop for the installation administrator
tmoose	Windows 98 (win95)	Endpoint	• Logs in to the gateway on oak • Runs the Tivoli desktop for the inventory administrator • Installed using the InstallShield image

Administrators

In a real-world Tivoli environment, multiple administrators are defined to manage specific aspects of enterprise management. The Tivoli region in this chapter is managed by the Tivoli administrators shown in the following figure.



The following list describes each administrator and describes how each accesses the Tivoli commands and desktop. Although only one administrator performs actions in this scenario, the others are described to provide a context for the installation of Tivoli Desktop for Windows.

Root_cygnus-region

This is the root administrator, the default administrator account created during the installation of the Tivoli server. This administrator has complete access to all Tivoli region and resource roles and is primarily used to assign authority and policy regions to other Tivoli administrators and to perform management activities that span the authority of subordinate administrators.

This administrator accesses the Tivoli environment using the login root on the Tivoli server, a UNIX machine named cygnus. This administrator contains the root logins required for the Tivoli Software Installation Service depot (in this case, root@cygnus). This requirement is described in "Authorizing the Tivoli Software Installation Service depot" on page 81.

Installation-Manager

Becky is responsible for creating the infrastructure of managed nodes, gateways, and endpoints, and for deploying Tivoli Enterprise software.

She accesses the Tivoli environment using her login jbecky on the UNIX machine cygnus, which is the machine from which installations are performed. Because cygnus is in a secured area, she installs Tivoli Desktop for Windows on bbrenner, the Windows NT machine in her office, which will connect to cygnus. Alternatively, she can telnet to cygnus where she can use the command line interface or, using an X Window System emulator, run the Tivoli desktop or the Tivoli Software Installation Service console.

Because she performs the installations, her required Tivoli roles are listed in Table 11 on page 276.

Inventory-Manager

Fyvush is responsible for using Tivoli Inventory to track hardware and software assets on all machines in the Tivoli region.

He accesses the Tivoli environment using his login fhackerm on the Windows NT machine djatzlau1, which is the machine on which Tivoli Inventory runs. He creates and distributes Inventory profiles. Because djatzlau1 is in a secured area, he installs Tivoli Desktop for Windows on tmoose, the Windows 98 machine in his office, which will connect to djatzlau1.

Because no Tivoli Inventory actions are shown, no Tivoli roles are listed.

Availability-Manager

Sally is responsible for using Tivoli Distributed Monitoring to monitor system and application availability on all machines in the Tivoli region.

She accesses the Tivoli environment using her login sally on the UNIX machine cygnus, which is the machine on which Tivoli Distributed Monitoring runs. Because cygnus is in a secured area, she telnets to cygnus from newcastl, the UNIX machine in her office.

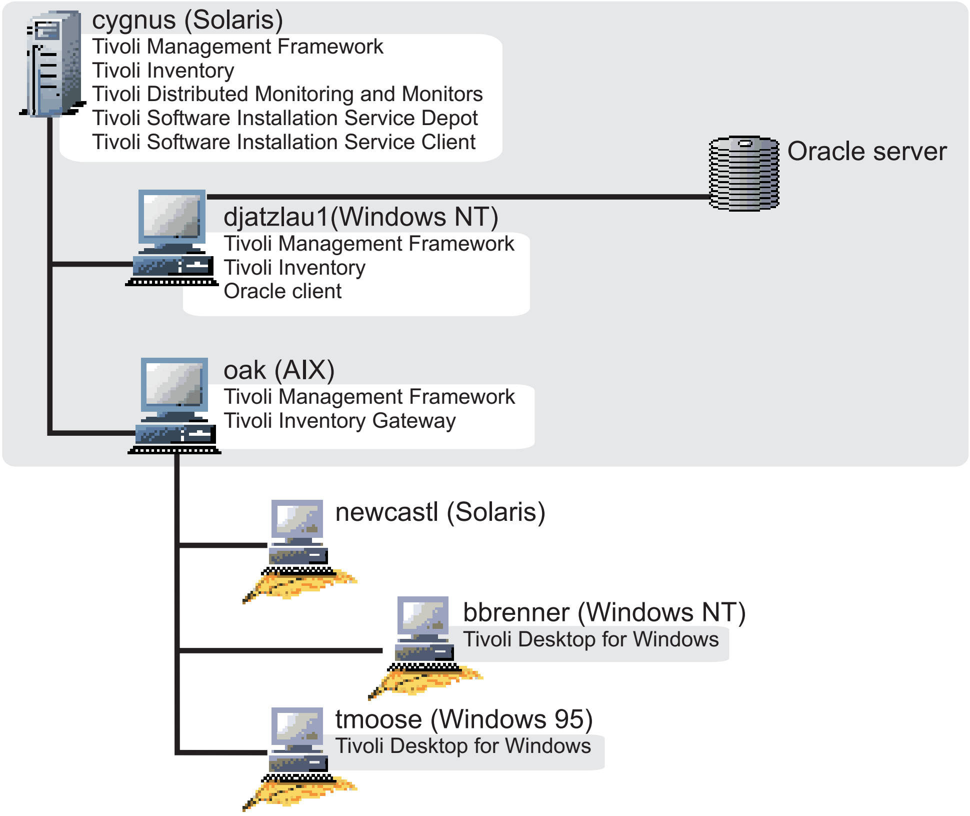
Because no Tivoli Distributed Monitoring actions are shown, no roles are listed.

Table 11. Administrator names, login, and roles

Administrator Name	Logins	Roles (Resource)
Root_cygnus-region	root@cygnus	All
Installation-Manager	jbecky@cygnus.dev.tivoli.com	backup (TMR) install_client (TMR) install_product (TMR) restore (TMR) senior (TMR) super (TMR) user (TMR)
Inventory-Manager	fhackerm@djatzlau1.dev.tivoli.com	Not relevant to this scenario.
Availability-Manager	sally@cygnus.dev.tivoli.com	Not relevant to this scenario.

Tivoli products

The following figure illustrates the Tivoli Enterprise software that will be installed to create the planned Tivoli region.


cygnus (Solaris)
Tivoli Management Framework
Tivoli Inventory
Tivoli Distributed Monitoring and Monitors
Tivoli Software Installation Service Depot
Tivoli Software Installation Service Client
Oracle server
djatzlau1(Windows NT)
Tivoli Management Framework
Tivoli Inventory
Oracle client
oak (AIX)
Tivoli Management Framework
Tivoli Inventory Gateway
newcastl (Solaris)
bbrenner (Windows NT)
Tivoli Desktop for Windows
tmoose (Windows 95)
Tivoli Desktop for Windows

The following table lists the Tivoli Enterprise applications that will be installed and describes the machine on which each is installed in terms of its role in the Tivoli region.

Table 12. Tivoli Enterprise applications

Application and Version	Goes on...
Tivoli Inventory, Version 3.6.2	Tivoli server (cygnus) RIM host (djatzlau1)
Tivoli Inventory, Version 3.6.2, Gateway	Gateway (oak)
• TME 10™ Distributed Monitoring 3.6 • Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2	Tivoli server (cygnus)
• TME 10 Distributed Monitoring NT Monitors 3.6 • Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, NT Monitors	Tivoli server (cygnus)
• TME 10 Distributed Monitoring Universal Monitors 3.6 • Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, Universal Monitors	Tivoli server (cygnus)
• TME 10 Distributed Monitoring Unix Monitors 3.6 • Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, UNIX Monitors	Tivoli server (cygnus)

The following table lists the Tivoli Management Framework and Tivoli Software Installation Service components and resources that comprise the infrastructure of the Tivoli environment and describes the machine on which each is installed in terms of its role in the Tivoli region.

Table 13. Tivoli Management Framework infrastructure

Application and Version	Goes on...
Tivoli Management Framework, Version 4.3.1	Tivoli server (cygnus) RIM host (djatzlau1) Gateway (oak)
Tivoli Desktop for Windows	Office workstations of Inventory-Manager (tmoose) and of Installation-Manager (bbrenner)
Tivoli Remote Execution Service	All Windows NT systems (djatzlau1 and bbrenner)
Endpoint	All systems
Tivoli Software Installation Service Depot, Version 3.7	Installation server (cygnus)
Tivoli Software Installation Service Client, Version 3.7	Installation server (cygnus)
Java for Tivoli, Version 4.1	Installation server (cygnus)
Tivoli Java Client Framework, Version 4.1	Installation server (cygnus)

The following sections contain planning information about each product to be installed in the Tivoli region. Planning information includes installation options that an administrator can specify during the installation. Depending on the installation method an administrator chooses, an installation option can be specified in one of the following ways:

- In fields in a graphical user interface (for example, **Endpoint Label**)
- As options to a command (for example, **–g oak_gateway**)
- As installation variables (for example, `IRDIR=/data`)
- In response to a prompt (for example, `Enter the password for user ID root on machine oak:`)
- Implicitly by virtual of location (for example, when using the Tivoli desktop, an administrator specifies the policy region in which to create a managed node by using the **Create** menu in that policy region)

Planning for the Tivoli server and managed nodes

This section contains planning information about the Tivoli server and manages nodes. It describes special considerations for the Windows NT managed node in this scenario, specifies the installation password, and lists the installation options for the Tivoli server and all managed nodes.

The following considerations apply to managed nodes in this scenario:

- On the UNIX machines in this scenario, the directory /data is a separate file system reserved for Tivoli Enterprise software, as required by the installation conventions of this scenario.

  On the Windows NT managed node in this scenario, the directory c:\data is not in a separate file system. However, in a real-life deployment, it is important to create a separate local partition for Tivoli Enterprise software.
- Because the Tivoli server (cygnus) is on Solaris system, the **wmailhost** command must be run locally on the Windows NT machine djatzlau1 after the managed node is installed. In this scenario, the command is run manually. If there are

many Windows NT managed nodes in the region, an administrator can create a task to run this command remotely on each machine.

- The Windows NT managed node in this scenario uses static IP addressing, which is necessary because the Tivoli server is on a UNIX machine. The Windows NT endpoints can use either DHCP or static IP addressing, because the endpoint manager can resolve changes in the IP address of an endpoint.
- The Windows NT managed node in this scenario does not require a Tivoli remote access account user name or password because this scenario does not access remote file systems. All files are installed locally. If an administrator later installs software or creates tasks that require access to remote resources, she must use the **wsettap** command to create the remote access account.
- The Windows NT machine must be rebooted after installing a managed node to enable Tivoli Authentication Package.

The following table lists the installation password for this scenario.

Table 14. Installation password

Information	Value
Installation password	manageIT

The following table contains the values for the installation options for the Tivoli server and managed nodes.

Table 15. Installation options for Tivoli Management Framework

Installation Option	Value
Binaries (BIN)	UNIX: /data/Tivoli/bin Windows NT: c:\data\Tivoli\bin
Libraries (LIB)	UNIX: /data/Tivoli/lib Windows NT: c:\data\Tivoli\lib
Message Catalogs (CAT)	UNIX: /data/Tivoli/cat Windows NT c:\data\Tivoli\cat
X11 Resource Files (APPD)	UNIX: /data/Tivoli/X11/app-defaults Windows NT c:\data\Tivoli\X11\app-defaults This directory does not apply to Windows operating systems. However, it is displayed in the Tivoli desktop installation windows. When using Tivoli Software Installation Service, it is not displayed.
Client/Server Database (DB/ALIDB)	UNIX: /data/Tivoli/db Windows NT c:\data\Tivoli\db
Man Pages (MAN)	UNIX: /data/Tivoli/man Windows NT c:\data\Tivoli\man
Policy Region	cygnus-region Not required for the Tivoli desktop.
Arrange for start of the Tivoli daemon at system (re)boot time (Autostart)	On
Configure remote start capability of the Tivoli daemon (SetPort)	On Required for Tivoli Desktop for Windows connections.

Table 15. Installation options for Tivoli Management Framework (continued)

Installation Option	Value
When installing, create “Specified Directories” if missing (CreatePaths)	On
Tivoli remote access account user †	None
Tivoli remote access account password †	None
Reboot †	Yes
† For Windows NT only.	

Planning for the gateway

The following table contains the values for the installation options for the gateway. This scenario uses the default values.

Table 16. Installation options for the gateway

Installation Option	Value
Gateway Name	oak-gateway
Managed Node Proxy	oak
Port	9494
Socket	Not applicable for this scenario.

Planning for endpoints

This section contains planning information about endpoints. It describes special considerations for the Windows NT endpoint in this scenario and lists the installation options for all endpoints.

Although the Tivoli server is a UNIX system, the Windows NT endpoints can use either DHCP or static IP addressing, because the endpoint manager can resolve changes in the IP address of an endpoint.

The Windows NT endpoint in this scenario does not require a Tivoli remote access account user or password because this scenario does not access remote file systems. All files are installed locally. If an administrator later decides to create tasks that require access to remote resources, she must use the **wlcftap** command to create the remote access account.

A Windows NT machine must be rebooted after installing an endpoint to enable Tivoli Authentication Package.

The following table contains the values for the installation options for the endpoints.

Table 17. Installation options for endpoints

Installation Option	Value
Endpoint binaries	UNIX: /data/Tivoli/lcf Windows NT: c:\data\Tivoli\lcf Windows 98: c:\data\Tivoli\lcf
Endpoint label	*hostname*-ep

Table 17. Installation options for endpoints (continued)

Installation Option	Value
Optional port	Not applicable for this scenario.
Additional options	Not applicable for this scenario.
Gateway name	oak-gateway
Add Endpoint Icon to Policy Region	None
Check for TMA login to Gateway	Yes
Seconds to wait for login	300
Tivoli remote access account user †	None
Tivoli remote access account password †	None
Reboot †	Yes
† For Windows NT only.	

Planning for Tivoli Remote Execution Service

The following table contains the values for the installation options for Tivoli Remote Execution Service. This service is only for Windows NT managed nodes and, when installing with Tivoli Software Installation Service, for Windows NT endpoints.

Table 18. Installation options for Tivoli Remote Execution Service

Installation Option	Value
Destination directory	c:\Tivoli\trip

Planning for Tivoli Software Installation Service

The following recommendations apply to the location of the install repository and client log directories:

- The install repository should be placed in a local file system and in a different file system than the one in which an administrator installs the Tivoli Enterprise software.

  For simplicity in this scenario, the file system is local but not separate.
- The client log directory must be in a location with adequate space.

The following tables contain the values for the installation options for the Tivoli Software Installation Service depot and client.

Table 19. Installation options for the depot

Installation Option	Value
Install repository directory	/data

Table 20. Installation options for the client

Installation Option	Value
SIS client log directory	/data/sisclient/logs

Note: This scenario does not install Tivoli Software Installation Service on any Windows NT system. Therefore, these values are not provided.

Planning for Tivoli Distributed Monitoring

To install Tivoli Distributed Monitoring, Version 3.6.2, an administrator installs Tivoli Distributed Monitoring, Version 3.6, and then applies an upgrade to Version 3.6.2. The same process applies to each monitor collection.

The following table contains the values for the installation options for Tivoli Distributed Monitoring. There are no installation options required for the upgrades or for the monitors.

Table 21. Installation options for Tivoli Distributed Monitoring

Installation Option	**Value**
Header Files (INC)	/data/Tivoli/include

Note: This scenario does not install Tivoli Distributed Monitoring on any Windows NT system. Therefore, these values are not provided.

For complete information about installing Tivoli Distributed Monitoring, refer to the product documentation.

Planning for Tivoli Inventory

An administrator can install Tivoli Inventory, Version 3.6.2 as a new installation. It is not necessary to install Version 3.6 and then upgrade to Version 3.6.2.

The following table contains the values for the installation options for Tivoli Inventory. No installation options are required for Tivoli Inventory Gateway.

Table 22. Installation options for Tivoli Inventory

Installation Option	**Value**
Database vendor	Oracle
RIM Host	djatzlau1
Database ID †	inventory
Database Home ‡	c:\oracle
Server ID ‡	grande.world
User Name †	tivoli
Instance Home ‡	(Applies only to DB2)
† The values required by Tivoli Inventory. ‡ Get these values from the database administrator.	

For complete information about installing Tivoli Inventory, refer to the product documentation.

Note: This scenario does not use the Scalable Collection Service, the Tivoli Management Platform MCollect service patch.

Planning for Tivoli Desktop for Windows

The following table contains the values for the installation options for Tivoli Desktop for Windows used in this scenario.

Table 23. Installation options for Tivoli Desktop for Windows

Installation Option	Value
Destination directory	c:\Tivoli\Desktop

Starting the deployment

Parts of this scenario are independent of whether the administrator chooses to use Tivoli Software Installation Service or Tivoli Management Framework. For example, the administrator must install the Tivoli server using a platform-specific method. This section covers the following common tasks:

- Installing the Tivoli server
- Installing Tivoli Remote Execution Service on the first Windows NT system in the Tivoli region
- Installing Tivoli Desktop for Windows on Windows NT systems
- Creating Tivoli administrators and policy regions
- Installing an endpoint on the Windows 98 machine

Installing the Tivoli server

The Installation-Manager installs the Tivoli server on cygnus using the procedure described in "Installing on a UNIX operating system" on page 26. She uses the installation values as listed in Table 15 on page 279.

In a Bourne shell window on cygnus, she enters the following commands to install the Tivoli server in this scenario without invoking the X11 installation windows:

```
mkdir /tmp/install_Tivoli

cd /tmp/install_Tivoli

DOGUI=no

export DOGUI

/cdrom/cdrom0/WPREINST.SH
. /wserver -c /cdrom/cdrom0 \
 BIN=/data/Tivoli/bin \
 LIB=/data/Tivoli/lib \
 ALIDB=/data/Tivoli/db \
 MAN=/data/Tivoli/man \
 APPD=/data/Tivoli/X11/app-defaults \
 CAT=/data/Tivoli/msg_cat \
 LK=1234567890XYZZY \
 IP=manageIT \
 AutoStart=1 \
 SetPort=1 \
 CreatePaths=1
```

After installing the Tivoli server, she backs up the Tivoli region using the procedure described in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Installing Tivoli Remote Execution Service

Because the Tivoli server runs Solaris, the Installation-Manager must manually install Tivoli Remote Execution Service on the first Windows NT managed node (djatzlau1).

Tivoli Software Installation Service uses this service when remotely installing managed nodes and endpoints on Windows machines. There must be at least one Windows NT or Windows 2000 managed node in the region before Tivoli Software Installation Service can install a managed node or endpoint to a Windows NT or Windows 2000 machine that does not already have Tivoli Remote Execution Service installed.

The Tivoli Management Framework installation methods use this service when installing managed nodes on Windows NT machines, but use a different method to install endpoints. This means that in the Tivoli Management Framework parts of this scenario, the Installation-Manager must manually install the first Windows NT endpoint, even if the machine has the service on it.

The Installation-Manager uses the procedure described in “Manually installing Tivoli Remote Execution Service” on page 341 to install this service.

Note: The Installation-Manager can install this service on djatzlau1 at any time prior to installing the first Windows NT managed node or endpoint.

Installing Tivoli Desktop for Windows

Becky and Fyvush manually install Tivoli Desktop for Windows on their respective workstations, bbrenner and tmoose. Each installs the software following the procedures in “Installing Tivoli Desktop for Windows on Windows operating systems” on page 65. This software can only be installed using the provided InstallShield image.

They can install this software at any time, even before the Tivoli server is created. However, they cannot use the Tivoli desktop until it can connect to a managed node. To use the Tivoli desktop, each follows the procedures in “Starting the Tivoli desktop” on page 69.

Fyvush accesses the Tivoli environment by running the Tivoli desktop on tmoose, the Windows 98 machine in his office. The Tivoli desktop connects to djatzlau1, which is the machine on which Inventory runs.

To use the Tivoli Management Framework installation methods, Becky can access the Tivoli environment by running the Tivoli desktop on bbrenner, the Windows NT machine in her office. The Tivoli desktop connects to cygnus, which is a managed node. However, to run Tivoli Software Installation Service, she must be logged in to cygnus, the system on which the Tivoli Software Installation Service client is installed. In this scenario, she uses an X Windows System emulator to telnet to cygnus from her workstation.

Creating Tivoli administrators

After installing the Tivoli server, Becky creates the Tivoli administrator account, **Installation-Manager**, that she will use to install Tivoli Enterprise software. She performs the following steps to create the **Installation-Manager** administrator:

1. Logs in to cygnus as root.
2. Sets the Tivoli variables. Because the shell for root@cygnus is the Bourne shell, she enters the following command:

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   For more information, refer to “Setting Tivoli environment variables” on page 30.
3. Enters the following command to create the administrator:

```
wcrtadmin -l jbecky@cygnus.dev.tivoli.com \
 -r global,user:super:install_client \
 -r global,install_product:backup:restore:senior \
 Installation-Manager
```

Because the Inventory-Manager and Availability-Manager administrators do not have tasks to perform in this scenario, they do not need to be created at this time. For details about the administrators in this scenario, refer to “Administrators” on page 274. For more information about creating Tivoli administrators, refer to the *Tivoli Management Framework User's Guide*.

Installing the Windows 98 endpoint

Neither Tivoli Software Installation Service nor Tivoli Management Framework installation mechanisms can install an endpoint on the Windows 98 machine (tmoose) in this scenario.

Becky can install the endpoint on tmoose at any time after creating the gateway. There are several ways she can install an endpoint on a Windows 98 machine, but she chooses to install it using InstallShield as described in “Installing endpoints using InstallShield” on page 215. Because the default installation location differs when using InstallShield to create endpoints, she uses the directory location and other installation options as listed in Table 17 on page 280.

Deployment using Tivoli Software Installation Service

Becky can deploy the Tivoli environment using any combination of the Tivoli Software Installation Service console and Tivoli Software Installation Service commands, but chooses to use a single method at a time.

She installs Tivoli Software Installation Service, then performs the deployment in two phases, as follows:

1. The first phase installs the Tivoli Management Framework infrastructure of managed nodes, gateways, and endpoints.

   Because this scenario has the Tivoli server on a UNIX system with Windows NT managed nodes and endpoints, the infrastructure phase requires two steps:

   a. Create a managed node on a Windows NT machine to make Tivoli Remote Execution Service available for the installation of the remaining managed nodes and endpoints on Windows NT machines.

   b. Create the remaining managed node, gateway, and endpoint infrastructure of the Tivoli region.

   If the Tivoli server in the scenario were a Windows NT or Windows 2000 system instead of a UNIX system, installing the infrastructure would be a single step.

   Alternatively, because this scenario has only one additional Windows NT machine, she could simplify the process by installing Tivoli Remote Execution Service manually on the other Windows NT machine. However, she uses Tivoli Software Installation Service to remotely install Tivoli Remote Execution Service to practice the process.

2. The second phase installs the Tivoli Enterprise applications that manage the IT environment.

Using an incremental installation (with a backup between each phase) is recommended when creating a new Tivoli region. If Becky encounters problems at any stage, the backup from the previous phase enables her to restore the region to a known working state without having to start over.

For future installation of managed resources or Tivoli Enterprise applications to the existing region, she can install them in a single phase.

Considerations for products requiring a RIM host

In this scenario, installing the infrastructure and applications separately prevents another problem. To install an application that requires a RIM host, such as Tivoli Inventory, a RIM object is required on a managed node. If the RIM object does not exist, it is created.

In this scenario the RIM host for Tivoli Inventory is on djatlzau1, which is not the Tivoli server.

Consider that Tivoli Software Installation Service installs products on the Tivoli server before it installs them on managed nodes, and that it treats the installation of a managed node like any other product installation. If Becky attempts to install everything in this scenario in one phase, the following events occur:

1. When Tivoli Software Installation Service is used to install Tivoli Inventory on the Tivoli server, it cannot create the RIM host on djatzlau1 because that machine is not yet a managed node. This causes the installation of Tivoli Inventory on the Tivoli server to fail.
2. What happens next depends on whether the administrator uses the **Install to Machines Maximizing Network Bandwidth** or **Install to Machines Independently** installation method.
   - If she uses **Install to Machines Maximizing Network Bandwidth**, the installation is attempted to the other machines even if an installation on the Tivoli server fails. Product dependencies ensure that Tivoli Management Framework is installed on djatzlau1 before Tivoli Inventory. Therefore the managed node is created.

     However, Tivoli Software Installation Service does not attempt to install Tivoli Inventory on djatzlau1 because Tivoli Inventory cannot be installed to a managed node before it is installed to the Tivoli server.

     In this situation, the infrastructure of the Tivoli region (managed nodes, gateway, and endpoints) is installed, but some or all of the Tivoli Enterprise applications are not installed.
   - If she uses **Install to Machines Independently**, Tivoli Software Installation Service does not install to any other machines after an installation fails on the Tivoli server. Therefore, installation on all machines stops because of the RIM problem with Tivoli Inventory. In this situation, some products are installed on the Tivoli server, but none on any other machines.
3. If she is installing from the console, after an installation failure, the installation worksheet is updated to show only the products that were not installed. (If she is installing with the **wsis** command, she must update the response file manually after an installation failure, regardless of which installation method she used.)

   Again, what happens next depends on whether the administrator used the **Install to Machines Maximizing Network Bandwidth** or **Install to Machines Independently** installation method in the initial attempt.
   - If she used **Install to Machines Maximizing Network Bandwidth**, simply restarting the installation should cause the rest of the products to install.

Tivoli Inventory installs on the Tivoli server now, because the managed node specified as its RIM host exists, allowing the RIM host object to be created.

- If she used **Install to Machines Independently**, restarting the installation requires her to modify the installation worksheet to prevent Tivoli Inventory from installing in the same phase as Tivoli Management Framework.

The same installation order dependencies exist when using Tivoli Management Framework. If the administrator specifies an installation option that attempts to create a RIM object on a managed node that does not exist, the installation fails. However, the administrator cannot simultaneously install multiple products and must plan for this type of dependency when performing installation using the Tivoli desktop or using Tivoli Management Framework commands.

Installing Tivoli Software Installation Service

In this scenario, the Tivoli Software Installation Service depot and client are installed on the Tivoli server. Because this is a new deployment, there are no performance considerations when using the Tivoli server as the installation server. In a production environment, an administrator can move the depot and client to other machines after the initial installation of the Tivoli region.

To install Tivoli Software Installation Service on the Tivoli server, Becky performs the following steps:

1. Opens an X11 window on cygnus using an X Windows emulator program on bbrenner, the Windows NT system in her office. She logs in as jbecky, which is a valid login for the Installation-Manager administrator.

   Because she is already using an X Windows emulator, xhost permissions are already set.
2. She sets the Tivoli variables. Because the shell for root@cygnus is the Bourne shell, she enters the following command:

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   For more information, refer to “Setting Tivoli environment variables” on page 30.
3. Inserts the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive of the Tivoli server. In this scenario, the CD mounts at /cdrom/cdrom0.
4. Installs the Java components required by Tivoli Software Installation Service by entering the following commands:

   ```
   winstall -c /cdrom/cdrom0/JAVA -i JRE cygnus
   winstall -c /cdrom/cdrom0/JAVA -i JCF cygnus
   ```
5. Installs the Tivoli Software Installation Service depot by entering the following command:

   ```
   winstall -c /cdrom/cdrom0/SIS -i SISDEPOT IRDIR=/data cygnus
   ```

   Note: The user root on the system where the depot is installed must be defined as a login to a Tivoli administrator. Because the Tivoli Software Installation Service depot is installed on the Tivoli server in this scenario, the default administrator (Root_cygnus-region) already has a login for root@cygnus.
6. Installs the Tivoli Software Installation Service client by entering the following command:

   ```
   winstall -c /cdrom/cdrom0/SIS -i SISCLNT \
    CLIENTLOGDIR=/data/sisclient/logs cygnus
   ```
7. Removes the CD.

8. Backs up the Tivoli region using the procedure described in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Deploying the scenario using the Tivoli Software Installation Service console

To create the sample Tivoli environment using the Tivoli Software Installation Service console, Becky performs the following high-level steps:

1. Starts the Tivoli Software Installation Service console.
2. Imports the installation images for all Tivoli Enterprise software.
3. Creates the Tivoli Management Framework infrastructure, which requires these high-level steps:
   a. Sets defaults for installation options for Tivoli Management Framework.
   b. Creates the first Windows NT managed node, which provides the Windows NT repeater required to connect to other Windows NT machines.
   c. Sets the default installation options for endpoints.
   d. Provides connection information for the remaining machines in the Tivoli region.
   e. Specifies which products to install on which machines.
   f. Performs the installation.
4. Installs the Tivoli Enterprise applications that will manage the IT environment, which requires these high-level steps:
   a. Sets the default installation options for the Tivoli Enterprise applications.
   b. Specifies which applications to install on which machines.
   c. Performs the installation.

These procedures are described in the following sections.

Starting the Tivoli Software Installation Service console

Becky is still logged in to cygnus as jbecky in the same X window she used to install Tivoli Software Installation Service. The Tivoli variables are still set. To start the Tivoli Software Installation Service console, she performs the following steps:

1. Enters the following command:
   ```
   wsisgui
   ```
2. In the Get Installation Password window that is displayed, she types the installation password for the Tivoli region, `manageIT`, and clicks **OK**.

The installation worksheet is displayed. She leaves it open for the next step.

Importing installation images using the Tivoli Software Installation Service console

Becky imports all the products into the install repository at one time. To reduce the space required for the install repository, she imports only the interpreter types required for the Tivoli management region.

After importing, she can use the installation images in the install repository for future installations using Tivoli Software Installation Service or Tivoli Management Framework. To import the required products, she performs the following actions:

1. In the installation worksheet, she selects **Worksheet → Select products**. The Select Products window is displayed, which lists all products that have been imported into the install repository.

Because Tivoli Software Installation Service has just been installed, the install repository contains installation images for only the endpoint and gateway products. These products are available by default and do not need to be imported.

2. Imports the installation images required for the Tivoli Management Framework by performing the following steps:
   a. Inserts the Tivoli Management Framework 1 of 2 CD. In this scenario, the CD is mounted at /cdrom/cdrom0.
   b. In the Select Products window, she clicks **Import Images**. The Locate images window is displayed.
   c. Navigates to the directory /cdrom/cdrom0, selects any file in the directory, and then clicks **OK**.
   d. In the Import Product window, she deselects each component except **Tivoli Management Framework**.
   e. In the **Tivoli Management Framework** component, deselect all interpreter types except **aix4-r1** and **w32-ix86**. She does not import the files for **solaris2**, because it is already installed on the Tivoli server, which is the only **solaris2** machine in the scenario.
   f. Clicks **Import** to import the installation images.
   g. When the Product Import Progress window shows that the product is successfully imported, she clicks **OK** to close the window.
   h. Removes the CD.

   The Tivoli Management Framework is now listed in the Select Products window.
3. Imports the installation images for the Tivoli Enterprise applications in the scenario. She repeats step 2 to import each of the components and interpreter types listed in the following table. She can import the products in any order.

Table 24. Components and interpreter types for Tivoli Enterprise software

Product CD	Components to Import	Interpreter Types
Tivoli Inventory, Version 3.6.2	Tivoli Inventory, Version 3.6.2	solaris2 w32-ix86
	Tivoli Inventory, Version 3.6.2, Gateway	aix4-r1
TME 10 Distributed Monitoring, Version 3.6	TME 10 Distributed Monitoring 3.6	solaris2
	TME 10 Distributed Monitoring Universal Monitors 3.6	generic
	TME 10 Distributed Monitoring NT Monitors 3.6	generic
	TME 10 Distributed Monitoring Unix Monitors 3.6	generic

Table 24. Components and interpreter types for Tivoli Enterprise software (continued)

Product CD	Components to Import	Interpreter Types
Tivoli Distributed Monitoring, Version 3.6.2	Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2	solaris2
	Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, Universal Monitors	generic
	Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, NT Monitors	generic
	Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, UNIX Monitors	generic

After she has imported the last product, the products are listed in the Select Products window.

4. Clicks **OK** to close the Select Products window. This returns her to the installation worksheet.

First phase—creating the Tivoli Management Framework infrastructure

To create the managed node, gateway, and endpoint infrastructure of the Tivoli region, Becky performs the following steps:

1. In the installation worksheet, she selects **Worksheet → Select products**.
2. Selects the Tivoli Management Framework, Tivoli Gateway, and Tivoli Endpoint products in the Select Products window and clicks **OK** to add them to the installation worksheet.
3. Sets the default installation options by performing the following steps:
   a. In the installation worksheet, she right-clicks the name of the Tivoli Management Framework product to open the Product details window.

      If necessary, she resizes the window to display all the installation options.

   b. For both interpreter types in the **Available Interpreter Types** list, she sets the default values as described in Table 15 on page 279.
   c. After she sets the defaults for all options for both interpreter types, she clicks **Set Defaults** to save the values.
4. Selects the machine to be the first Windows NT managed node, which becomes the CurrentNtRepeat object from which Tivoli Remote Execution Service is installed on other Windows machines, by performing the following steps:
   a. In the installation worksheet, she selects **Worksheet → Select machines**. The Select Machines window opens.
   b. Sets the **Policy Region** and **Machine Type** filters to **All**. Only the Tivoli server, cygnus, is listed.
   c. Clicks **Add Machine** to open the Add Machine window and adds a machine by performing the following steps:
      1) In the **Hostname** field, she types `djatzlau1`.
      2) Ensures that the **REXEC/Account** button is selected.
      3) In the **User ID** field, she types `Administrator`, the name of the administrator account for this machine.
      4) In the **Password** field, she types the password for the Administrator account on djatzlau1.

5) Clicks **Submit**. When a test connection has been made, the `Information retrieved` message appears in the status area.

6) Click **Add & Close** to add djatzlau1 to the Select Machines window.

d. In the Select Machines window, the newly added machine is already selected. She clicks **Add & Close** to add it to the installation worksheet.

5. Clicks the product name for Tivoli Management Framework to indicate that djatzlau1 will become a managed node.

   She observes that when she clicks Tivoli Management Framework, the cell for Tivoli Gateway on djatzlau1 changes from dark gray to light gray, indicating that dependencies have been met, specifically, that a gateway can only be created on a managed node. However, in this scenario, djatzlau1 does not host a gateway, so she does not select that product.

   She does not need to override the default installation options for any products in this scenario.

6. Starts the installation to create the first Windows NT managed node by performing the following steps:

   a. In the installation worksheet, she selects **Worksheet → Install**. The Select Installation Mechanism window is displayed.

   b. Selects **Install to Machines Maximizing Network Bandwidth** and clicks **OK**.

      She watches the progress of the installation in the Installation Progress window.

   c. When the installation is complete, she clicks **OK** to close the Installation Progress window. If errors are encountered, she clicks **View logs** to examine the logs for the machines that reported failure.

   The machine djatzlau1 is now a managed node and is designated as the CurrentNtRepeat object.

7. Adds the rest of the machines to the installation worksheet by performing the following steps:

   a. Defines the connection information for the machines oak, newcastl, and bbrenner using the same procedure as for djatzlau1 (step 4).

      When she clicks **Submit** to add the Windows NT machine bbrenner, a prompt asks whether she wants to install Tivoli Remote Execution Service on the machine. She clicks **Yes**.

   b. On the Select Machines window, she ensures that the **Policy Region** and **Machine Type** filters are set to **All**.

   c. Selects all of the machines, including the Tivoli server cygnus, then clicks **Add & Close**.

   If necessary, she resizes the installation worksheet to display all of the machines.

8. Makes a gateway available for endpoints by clicking the cells on the installation worksheet to select Tivoli Management Framework and Tivoli Gateway on oak.

   Note: Because there is no gateway in the Tivoli region at this time, the **Gateway Name** field contains **Broadcast to Gateways**. Because she does not want the endpoints to broadcast initial log in requests, Becky selects the gateway product on oak before selecting the endpoint on any machine. This makes oak-gateway the default gateway for all endpoints installed in this region.

She does not have to perform the installation that creates the managed node and gateway on oak. Selecting the machine on the installation worksheet is sufficient. When the installation is performed with the **Install to Machines Maximizing Network Bandwidth** installation mechanism, Tivoli Software Installation Service controls the installation sequence to ensure that the managed node and gateway are installed on oak before an endpoint is installed on any machine.

9. Sets defaults for the installation options for endpoints to the values described in “Planning for endpoints” on page 280. She follows the same procedure she used earlier in step 3.

   She sets the installation options for the interpreter types solaris2, aix4-r1, and w32-ix86. The install repository contains installation images for all endpoint interpreter types, but she sets only those for interpreter types she will have in the Tivoli region.

10. To select the endpoint for each machine, she clicks **Tivoli Endpoint** in the list of products in the installation worksheet.

    If this scenario had required additional managed nodes and gateways, she would select them now.

11. Starts the installation. Tivoli Software Installation Service sequences the installations to ensure that the gateway on oak is available before installing any endpoints. She performs the following steps to start the installation:

    a. In the installation worksheet, she selects **Worksheet → Install**.

    b. In the Select Installation Mechanism window that is displayed, she selects **Install to Machines Maximizing Network Bandwidth** and clicks **OK**.

    c. To view the progress of any machine during the installation, she selects the button next to the machine name in the list at the top of the Installation Progress window.

    d. When the installation is complete, she closes the Installation Progress window by clicking **OK**. If errors are encountered, she clicks **View logs** to examine the logs for the machines that reported failure.

12. After the gateway is created, she can manually install the endpoint on the Windows 98 machine tmoose at her convenience.

    There are several ways to install the endpoint on a Windows 98 machine. She chooses to use InstallShield, as described in “Installing endpoints using InstallShield” on page 215. She knows that the InstallShield installation has different defaults for the installation options, so she is careful to specify the installation options listed in Table 17 on page 280.

13. Because **djatzlau1** is a Windows NT managed node, Becky needs to configure **djatzlau1** to send e-mail alerts and other messages through the UNIX mail server on **cygnus**. She can do this at any time after installing the managed node on **djatzlau1**.

    To configure **djatzlau1** to send e-mail through the UNIX mail server on **cygnus**, she performs the following steps:

    a. Logs in to **djatzlau1** as **Administrator**.

    b. Opens a DOS window.

    c. Sets the Tivoli environment variables and starts the bash shell by entering the following commands:

       ```
       %systemroot%\system32\drivers\etc\tivoli\setup_env.cmd
       bash
       ```

    d. Specifies the mail server to be used by managed node **djatzlau1** by entering the following command:

       ```
       wmailhost cygnus
       ```

14. Backs up the Tivoli region as described in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

The infrastructure of the Tivoli region is now complete. She leaves the Tivoli Software Installation Service console open to use in the next phase.

Final phase—installing the Tivoli Enterprise applications

The final phase installs the Tivoli Enterprise applications required in this scenario, Tivoli Inventory and Tivoli Distributed Monitoring.

The installation worksheet from the previous phase displays all of the machines in the Tivoli region and shows the installation status of each of the infrastructure products on each machine. To install the Tivoli Enterprise applications, Becky performs the following steps:

1. In the installation worksheet, she selects **Worksheet → Select products**.
2. Selects all the Tivoli Enterprise applications in the Select Products window and clicks **OK** to add them to the installation worksheet.

   If necessary, she resizes the installation worksheet to display all of the products and machines.
3. Sets defaults for the installation options for each remaining product. In this scenario, the only products with options are Tivoli Distributed Monitoring, Version 3.6 (refer to Table 21 on page 282) and Tivoli Inventory, Version 3.6.2 (refer to Table 22 on page 282).
4. Selects which Tivoli Enterprise applications to install on which machines. To install Tivoli Distributed Monitoring and Tivoli Inventory according to the plan for this scenario, she selects cells in the installation worksheet as follows:
   a. All products must be installed on the Tivoli server. For the machine cygnus, she selects the cells for the following products:
      - Tivoli Inventory, Version 3.6.2

        She observes that when she clicks Tivoli Inventory 3.6.2 on cygnus, the cell for that product on djatzlau1 changes from dark gray to light gray. This indicates that all dependencies have been met, in particular, the dependency that this product must be installed on the Tivoli server before it can be installed on any managed nodes. The products are installed in the correct order.

        Only djatzlau1 changes because she did not import the installation images for the interpreter type of any other machines. To view representative error messages describing why the cells are not selectable, she clicks the cells for bbrenner and oak.
      - TME 10 Distributed Monitoring 3.6

        As she selects TME 10 Distributed Monitoring 3.6 on the Tivoli server, the cell on that machine for Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2 becomes selectable, as do the cells for the three version 3.6 monitor collections.
      - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2
      - TME 10 Distributed Monitoring Universal Monitors 3.6

        As she selects each 3.6 monitor collection, the corresponding 3.6.2 upgrade becomes selectable.
      - TME 10 Distributed Monitoring NT Monitors 3.6
      - TME 10 Distributed Monitoring Unix Monitors 3.6
      - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, Universal Monitors

   - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, NT Monitors
   - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, UNIX Monitors

   b. Selects the cell for Tivoli Inventory, Version 3.6.2 on djatlzau1.

   c. Selects the cell for Tivoli Inventory, Version 3.6.2, Gateway on oak.

      This provides the gateway with the methods it needs to communicate with endpoints managed by Tivoli Inventory.

      Tivoli Distributed Monitoring does not have a separate gateway product.
5. Starts the installation.
   a. In the installation worksheet, she selects **Worksheet → Install**. The Select Installation Mechanism window is displayed.
   b. Selects **Install to Machines Maximizing Network Bandwidth** and clicks **OK**.
   c. To view the progress of any machine, she selects the button next to the machine name in the list at the top of the Installation Progress window.
   d. When the installation is complete, she closes the Installation Progress window by clicking **OK**. If errors are encountered, she clicks **View logs** to examines the logs for the machines that reported failure.
6. Performs any post-installation configuration required by the applications just installed. For details, she refers to the installation instructions in the product documentation for Tivoli Distributed Monitoring and Tivoli Inventory.
7. After the products are installed successfully, she backs up the Tivoli region as described in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

The sample Tivoli region is now installed.

Deploying the scenario using Tivoli Software Installation Service commands

To create the sample Tivoli environment using the Tivoli Software Installation Service commands, Becky performs the following high-level steps:

1. Imports the installation images for the Tivoli Enterprise applications.
2. Performs the first phase to install the infrastructure, which consists of these high-level steps:
   a. Exports a response file template for installing the first Windows NT managed node.
   b. Modifies the template as required by the scenario.
   c. Performs the installation.
   d. Exports a response file template to install the rest of the infrastructure.
   e. Modifies the template as required by the scenario.
   f. Performs the installation.
3. Performs the final phase to install the Tivoli Enterprise applications:
   a. Exports a response file template for installing the Tivoli Enterprise applications.
   b. Modifies the template as required by the scenario.
   c. Performs the installation.
4. Optionally, she sets the default installation options for all Tivoli Enterprise components.

These procedures are described in the following sections.

Importing installation images using the wimport command

Becky is still logged in to cygnus as jbecky in the same X window she used to install Tivoli Software Installation Service. The Tivoli variables are still set.

Becky imports all the products into the install repository at one time. To reduce the space required for the install repository, she imports only the interpreter types required for the Tivoli management region.

After importing, she can use the installation images in the install repository for future installations using Tivoli Software Installation Service or Tivoli Management Framework. To import the required products, she performs the following actions:

1. Mounts the Tivoli Management Framework 1 of 2 CD and enters the following command to import Tivoli Management Framework, Version 4.3.1, which is used to create managed nodes:

   ```
   wimport -c /cdrom/cdrom0 -i TMF aix4-r1 w32-ix86
   ```

   The solaris2 installation image is not required because the Tivoli server is the only machine of that type in this scenario.
2. Mounts the Tivoli Inventory 3.6.2 CD and enters the following command to import Tivoli Inventory, Version 3.6.2 and Tivoli Inventory, Version 3.6.2, Gateway:

   ```
   wimport -c /cdrom/cdrom0 \
    -i 362_INV w32-ix86 solaris2 \
    -i 362_GW aix4-r1
   ```
3. Mounts the TME 10 Distributed Monitoring 3.6 CD and enters the following command to import TME 10 Distributed Monitoring 3.6:

   ```
   wimport -c /cdrom/cdrom0 \
    -i SENT36 solaris2 \
    -i UNVM36 \
    -i UNXM36 \
    -i NTMON36
   ```

   She specifies the interpreter type (solaris2) for the Tivoli Distributed Monitoring base component (SENT36) to import only the installation image for that operating system. However, the interpreter type can be omitted for the monitor collections because those products contain only the generic interpreter type, which is imported by default.
4. Mounts the Tivoli Distributed Monitoring 3.6.2 CD and enters the following command to import the upgrade images for Tivoli Distributed Monitoring 3.6.2:

   ```
   wimport -c /cdrom/cdrom0 \
    -i DMN362MR solaris2 \
    -i UNVMN362 generic \
    -i UNXMN362 generic \
    -i NTMON362 generic
   ```
5. Determines the exact name and number of the products in the install repository. She uses this information to create response files. She enters the following command to write the list of products to the file /tmp/wimport-l.output:

   ```
   wimport -l > /tmp/wimport-l.output
   ```

The endpoint and gateway products are always present in the install repository and do not need to be imported.

First phase—creating the Tivoli region infrastructure

To create the infrastructure of the Tivoli region, Becky performs the following steps:

1. Creates and uses a response file to install the first Windows NT managed node:
   a. To create a response file template, she enters the following command:
      ```
      wsis -x /tmp/first_nt.rsp -p TMF-client-3.7 djatzlau1 w32-ix86
      ```
      The resulting response file template can be viewed in Figure 5 on page 301.

      Note: Although she can specify each product using the product number displayed by **wimport –l**, she uses the unique product ID instead. This makes the response file more robust because the unique product ID does not change, while the product number can change when additional products are imported into the install repository.
   b. Modifies the response file for the planned Tivoli region by making the following changes:
      - Adds the Administrator user ID in the [machine] section.
      - Sets the values for the installation options as described in Table 15 on page 279.
      - Verifies that the `InstallAlgorithm` entry has the appropriate value. She uses `mdist` (she can also type the name of the corresponding installation mechanism in the Tivoli Software Installation Service console, **Install to Machines Maximizing Network Bandwidth**).
      - Modifies the file to make it easier to understand by rearranging sections, adding comments, changing alias names, and deleting one of the [byProduct] or [byNode] sections.

      The resulting response file can be viewed in Figure 6 on page 301. For additional information about modifying a response file, refer to “Scenario: Modifying an exported response file” on page 162.
   c. Imports the response file to install the Windows NT managed node by entering the following command:
      ```
      wsis -i /tmp/first_nt.rsp
      ```
      Because the response file does not contain passwords, she will be prompted for the installation password for the Tivoli region and the administrator password for djatzlau1 when she uses the response file.

      Because the response file specified `Reboot=yes`, the machine is rebooted after installing the managed node. This activates Tivoli Authentication Package.
   d. If she encounters errors, she uses the Tivoli Software Installation Service console to examine the logs for the machines that reported failure.
2. Creates and uses a response file to install the remaining managed resources of the Tivoli region by performing the following steps:
   a. Creates a response file template by entering the following command:
      ```
      wsis -P -x /tmp/infrastructure.rsp \
      -p TMF-client-3.7 oak aix4-r1 \
      -p GATEWAY-Tivoli_Gateway oak aix4-r1 \
      -p TMA-Tivoli_Endpoint cygnus \
      -p TMA-Tivoli_Endpoint djatzlau1 \
      -p TMA-Tivoli_Endpoint oak aix4-r1 \
      -p TMA-Tivoli_Endpoint newcastl solaris2 \
      -p TMA-Tivoli_Endpoint bbrenner w32-ix86
      ```
      She does not specify the interpreter type for the machines cygnus and djatzlau1 because Tivoli Software Installation Service can determine the interpreter type of managed nodes.

To avoid typing in the installation password and the password for each machine, she uses the **–P** options to include the passwords in the exported response file.

The resulting response file template can be viewed in Figure 7 on page 302.

b. Because the response file contains unencrypted passwords, she sets the file permissions to prevent unauthorized access by entering the following command:

```
chmod 600 /tmp/infrastructure.rsp
```

c. Modifies the response file for the installation, making the following changes:

- Adds the user ID and password to each [machine] section.
- Changes `aliasname`*N* to meaningful terms throughout the file. This makes the file easier to read and makes it easier to combine [alias] sections.
- Because this scenario uses the same installation options for all machines of a similar type, she combines the [alias] sections for similar machines. For example, by making specific changes to one [alias] section, she can use it to install an endpoint on all three UNIX machines. Similarly, she creates another [alias] section to install a endpoint on both Windows NT machines.

  Note: When combining [alias] sections for similar installations, she is careful to combine only those with identical values or whose values can be made generic using variables.

  For example, she makes the endpoint [alias] sections apply to multiple machines by changing the machine-specific value of the `@EndpointLabel@` entry to `@HostName@-ep`.
- Sets the log in gateway for each endpoint [alias] section to `@GatewayName@=oak-gateway`.
- Sets the values for other installation options as appropriate to the scenario. She uses the managed node values described in Table 15 on page 279 and the endpoint values described in Table 17 on page 280. The gateway values do not need to be changed specifically for this scenario, but she makes the gateway label generic so she can reuse this response file in the future.
- Verifies that the `InstallAlgorithm` entry has the appropriate value. For this scenario, she uses `mdist`.
- She modifies the file to make it easier to understand, by rearranging sections, adding comments, changing alias names, and deleting one of the [byProduct] or [byNode] sections.

The resulting response file can be viewed in Figure 8 on page 304. For additional information about modifying a response file, refer to "Scenario: Modifying an exported response file" on page 162.

d. Imports the response file to install the endpoints, gateway, and managed node by entering the following command:

```
wsis -i /tmp/infrastructure.rsp
```

e. If she encounters errors, she uses the Tivoli Software Installation Service console to examine the logs for the machines that reported failure.

f. Because djatzlau1 is a Windows NT managed node, she needs to configure djatzlau1 to send e-mail alerts and other messages through the UNIX mail server on cygnus. She can do this at any time after installing the managed node on djatzlau1.

    To configure djatzlau1 to send e-mail through the UNIX mail server on cygnus, she performs the following steps:

    1) Logs in to djatzlau1 as Administrator.
    2) Opens a DOS window.
    3) Sets the Tivoli variables and starts the bash shell by entering the following commands:

       ```
       %systemroot%\system32\drivers\etc\tivoli\setup_env.cmd
       bash
       ```

    4) Specifies the mail server to be used by managed node djatzlau1 by entering the following command:

       ```
       wmailhost cygnus
       ```

g. Backs up the Tivoli region as described in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Because the gateway is now created, she can manually install the endpoint on tmoose at her convenience, as described in “Installing the Windows 98 endpoint” on page 285.

Final phase—installing the Tivoli products

To create and use a response file to install the Tivoli products, Becky performs the following steps:

1. Creates a response file template by entering the following command:

   ```
   wsis -x /tmp/install_applications.rsp \
    -p Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2 cygnus \
    -p Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2 djatzlau1 \
    -p InventoryGW-3.6.2-Tivoli_Inventory,_Version_3.6.2,_Gateway oak \
    -p Sentry2.0.2-3.6-TME_10_Distributed_Monitoring_3.6 cygnus \
    -p 3.6.2-DMN-MR-Tivoli_Distributed_Monitoring_Upgrade,_Version_3.6_3.6.1_\
       to_3.6.2 cygnus \
    -p NTMonitors-3.6-TME_10_Distributed_Monitoring_NT_Monitors_3.6 cygnus \
    -p NTMonitors_362-Tivoli_Distributed_Monitoring_Upgrade,_Version_3.6_3.6.1_to_\
       3.6.2,_NT_Monitors cygnus \
    -p UniversalMonitors-3.6-TME_10_Distributed_Monitoring_Universal_Monitors_3.6 \
       cygnus \
    -p UniversalMonitors_362-Tivoli_Distributed_Monitoring_Upgrade,_Version_3.6_\
       3.6.1_to_3.6.2,_Universal_Monitors cygnus \
    -p UnixMonitors-3.6-TME_10_Distributed_Monitoring_Unix_Monitors_3.6 cygnus \
    -p UnixMonitors_362-Tivoli_Distributed_Monitoring_Upgrade,_Version_3.6_3.6.1_\
       to_3.6.2,_UNIX_Monitors cygnus
   ```

   She does not specify the interpreter type of any of the machines because they all have a Tivoli connection at this point.

   The resulting response file template can be viewed in Figure 9 on page 305.

2. Modifies the template as required for the installation. Her changes include the following:

   - Combines the two [alias] sections for Tivoli Inventory can be combined because they have identical installation options.
   - Sets installation options for the applications. She uses the Tivoli Distributed Monitoring installation options listed in Table 21 on page 282 and the Tivoli Inventory installation options listed in Table 22 on page 282.
   - Verifies that the `InstallAlgorithm` entry is set to `mdist`.
   - Modifies the file to make it easier to understand, by rearranging sections, adding comments, changing alias names, and deleting one of the [byProduct] or [byNode] sections.

   The resulting response file can be viewed in Figure 10 on page 307.

3. Imports the response file to install the applications by entering the following command:

   ```
   wsis -i /tmp/install_applications.rsp
   ```

   She will be prompted for the installation password when she uses the response file, because she did not export passwords. Passwords are not required to connect to the machines because they use the existing Tivoli connection.
4. If she encounters errors, she uses the Tivoli Software Installation Service console to examine the logs for the machines that reported failure.
5. Performs any post-installation configuration required by the applications just installed. For details, she refers to the installation instructions in the product documentation for Tivoli Distributed Monitoring and Tivoli Inventory.
6. Backs up the Tivoli region as described in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

The sample Tivoli region is now installed.

Setting defaults for installation options for later use

The install repository now contains the installation images for all of the Tivoli Enterprise software installed in the Tivoli region. However, the product defaults have not been saved in the install repository. Later, if Becky wants to add new machines to the Tivoli region, she will have to look up the defaults in the deployment plan and set them manually at that time.

Instead, she chooses to set the default installation options for each product using information in the response files she just created and used. There are various approaches to this task. She performs the following steps:

1. She refers to /tmp/wsisdefaults-l.output. This file, which she created earlier, lists the name and number of each product in the install repository.
2. Creates one file that contains all of the response files. There are many ways to do this. She uses the following command:

   ```
   cat /tmp/first_nt.rsp /tmp/infrastructure.rsp \
   /tmp/install_applications.rsp > /tmp/product.defaults
   ```
3. Using her favorite text editor, she modifies the product.defaults file as follows:
   - Removes all sections except [alias] sections.
   - Removes the [alias] section for each product that does not have installation options. For example, she deletes the [alias] section named `InventoryGW_362new`.
   - Converts the alias names to interpreter types. She uses the output of the **wsisdefaults –l** command to determine which interpreter types are imported for each product. In the following examples, the text that is changed is shown in **bold**.

     For example, the product.defaults file contains these lines:

     ```
     # ======== Gateway Definitions ==========
     [alias Gateway GATEWAY-Tivoli_Gateway]
     @GATE_NAME@=@HostName@-gateway
     @GATE_PORT@=9494
     Overwrite=
     ```

     The output of **wsisdefaults** shows that the gateway product has a single interpreter type, generic. Therefore, change the alias name (Gateway) to the interpreter type (generic), as follows:

```
# ======== Gateway Definitions ==========
[alias generic GATEWAY-Tivoli_Gateway]
@GATE_NAME@=@HostName@-gateway
@GATE_PORT@=9494
Overwrite=
```

Similarly, the Tivoli Inventory product has the following entry (comment lines have been removed for this example, but are shown in the completed file):

```
# =========== Inventory Definitions ============
[alias Inventory_new362 Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2]
@RDBMS_Vendor@=Oracle
@RDBMS_Host@=djatzlau1
@RDBMS_DB_Name@=inventory
@RDBMS_DB_Home@=c:\oracle
@RDBMS_DB_Param_one@=grande.world
@RDBMS_DB_UserName@=tivoli
@RDBMS_DB_Param_two@=
Overwrite=
```

The **wsisdefaults** command shows that this product has installation options for the interpreter types solaris2 and w32-ix86. In the Tivoli region in this scenario, both of the interpreter types have identical installation option values. Therefore she changes that section as follows:

```
# =========== Inventory Definitions ============
[alias solaris2,w32-ix86 Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2]
@RDBMS_Vendor@=Oracle
@RDBMS_Host@=djatzlau1
@RDBMS_DB_Name@=inventory
@RDBMS_DB_Home@=c:\oracle
@RDBMS_DB_Param_one@=grande.world
@RDBMS_DB_UserName@=tivoli
@RDBMS_DB_Param_two@=
Overwrite=
```

The completed file is shown in "Product defaults file" on page 308.

4. Imports the product defaults file into the install repository by entering the following command:

   ```
   wsisdefaults -i /tmp/product.defaults
   ```

The default values of the installation options for each product imported in this scenario are now saved in the install repository. If she exports response file templates or uses the console to plan and perform an installation, the new default values will be used.

Sample files for the command line scenario

The following sections contain the files used in this scenario:

- "Response file to install the first Windows NT machine—as exported"
- "Response file to install the first Windows NT machine—modified" on page 301
- "Infrastructure response file—as exported" on page 302
- "Infrastructure response file—modified" on page 304
- "Tivoli products response file—as exported" on page 305
- "Tivoli products response file—modified" on page 307
- "Product defaults file" on page 308

Response file to install the first Windows NT machine—as exported

The following file is generated by the **wsis** command as a response file template to install first Windows NT managed node in the scenario. This is described in step

1a.

Figure 5. The /tmp/first_nt.rsp file generated by the wsis command

```
[machine djatzlau1]
access=rexec
userid=
autoInstallTrip=yes
promptForPassword=yes
interp=w32-ix86

[machine djatzlau1]
access=rexec
userid=
autoInstallTrip=yes
promptForPassword=yes
interp=w32-ix86

[byProduct]
aliasname1=djatzlau1

[alias aliasname1 TMF-client-3.7]
CAT=c:/usr/local/Tivoli/msg_cat
LIB=c:/usr/local/Tivoli/lib
BIN=c:/usr/local/Tivoli/bin
DB=c:/data/Tivoli/db
MAN=c:/usr/local/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
tapUser=
TapPassword=
Reboot=No
Overwrite=

[byNode]
djatzlau1=aliasname1

[globals]
InstallAlgorithm=mdist
```

Response file to install the first Windows NT machine—modified

The following file is the result of modifying the file in Figure 5 to create the first Windows NT managed node in the scenario. The instructions for modifying the file are in step 1b.

Figure 6. The /tmp/first_nt.rsp file after editing

```
# ======== Global Information ==========
[globals]
InstallAlgorithm=mdist

# ======== What Goes Where ==========
[byNode]
djatzlau1=NT_managed_node

# ======== Define Machines ==========
[machine djatzlau1]
access=rexec
userid=Administrator
autoInstallTrip=yes
promptForPassword=yes
interp=w32-ix86
```

```
# ======== Managed Nodes Options ==========
[alias NT_managed_node TMF-client-3.7]
CAT=c:/data/Tivoli/msg_cat
LIB=c:/data/Tivoli/lib
BIN=c:/data/Tivoli/bin
DB=c:/data/Tivoli/db
MAN=c:/data/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
tapUser=
TapPassword=
Reboot=Yes
Overwrite=
```

Infrastructure response file—as exported

The following file is generated by the **wsis** command as a response file template to create the Tivoli region infrastructure for the scenario. This is described in step 2a.

Figure 7. The /tmp/infrastructure.rsp file generated by the wsis command

```
[machine cygnus]
access=tivoli
interp=solaris2

[byProduct]
aliasname1=oak
aliasname2=oak
aliasname3=cygnus
aliasname4=djatzlau1
aliasname5=oak
aliasname6=newcastl
aliasname7=bbrenner

[globals]
InstallPassword=manageIT
InstallAlgorithm=mdist

[machine oak]
access=rexec
userid=
password=
promptForPassword=no
interp=aix4-r1

[alias aliasname2 GATEWAY-Tivoli_Gateway]
@GATE_NAME@=oak-gateway
@GATE_PORT@=9494
Overwrite=

[machine newcastl]
access=rexec
userid=
password=
promptForPassword=no
interp=solaris2

[byNode]
oak=aliasname1 aliasname2 aliasname5
cygnus=aliasname3
djatzlau1=aliasname4
newcastl=aliasname6
bbrenner=aliasname7
```

```
[alias aliasname3 TMA-Tivoli_Endpoint]
TMABIN=/opt/Tivoli/lcf
@EndpointLabel@=cygnus
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=Broadcast to Gateways
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

[alias aliasname4 TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=djatzlau1
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=Broadcast to Gateways
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

[alias aliasname5 TMA-Tivoli_Endpoint]
TMABIN=/opt/Tivoli/lcf
@EndpointLabel@=oak
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=Broadcast to Gateways
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

[alias aliasname6 TMA-Tivoli_Endpoint]
TMABIN=/opt/Tivoli/lcf
@EndpointLabel@=newcastl
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=Broadcast to Gateways
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

[alias aliasname7 TMA-Tivoli_Endpoint]
TMABIN=c:/Tivoli/lcf
@EndpointLabel@=bbrenner
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=Broadcast to Gateways
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
tapUser=
TapPassword=
Reboot=No
Overwrite=

[machine djatzlau1]
access=tivoli
interp=w32-ix86

[alias aliasname1 TMF-client-3.7]
CAT=/usr/local/Tivoli/msg_cat
LIB=/usr/local/Tivoli/lib
BIN=/usr/local/Tivoli/bin
```

```
DB=/data/Tivoli/db
MAN=/usr/local/Tivoli/man
APPD=/usr/lib/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
Overwrite=

[machine bbrenner]
access=rexec
userid=
password=
autoInstallTrip=yes
promptForPassword=no
interp=w32-ix86
```

Infrastructure response file—modified

The following file is the result of modifying the file in Figure 7 on page 302 to perform the infrastructure phase of the installation in the scenario. Changes are described in step 2c.

Figure 8. The /tmp/infrastructure.rsp file after editing

```
# ======== Global Information ==========
[globals]
InstallPassword=manageIT
InstallAlgorithm=mdist

# ======== What Goes Where ==========
[byNode]
oak=UNIX_managed_node Gateway UNIX_endpoint
cygnus=UNIX_endpoint
djatzlau1=NT_endpoint
newcastl=UNIX_endpoint
bbrenner=NT_endpoint

# ======== Define Machines ==========
[machine cygnus]
access=tivoli
interp=solaris2

[machine djatzlau1]
access=tivoli
interp=w32-ix86

[machine bbrenner]
access=rexec
userid=Administrator
password=pa$$w0rd
autoInstallTrip=yes
promptForPassword=no
interp=w32-ix86

[machine oak]
access=rexec
userid=root
password=pa$$w0rd
promptForPassword=no
interp=aix4-r1

[machine newcastl]
access=rexec
userid=root
password=pa$$w0rd
```

```
promptForPassword=no
interp=solaris2

# ======== Gateway Definitions =========
[alias Gateway GATEWAY-Tivoli_Gateway]
@GATE_NAME@=@HostName@-gateway
@GATE_PORT@=9494
Overwrite=

# ======== Endpoint Definitions =========
[alias NT_endpoint TMA-Tivoli_Endpoint]
TMABIN=c:/data/Tivoli/lcf
@EndpointLabel@=@HostName@-ep
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=oak-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=
tapUser=
TapPassword=
Reboot=Yes
Overwrite=

[alias UNIX_endpoint TMA-Tivoli_Endpoint]
TMABIN=/data/Tivoli/lcf
@EndpointLabel@=@HostName@-ep
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=oak-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

# ======== Managed Node Definitions =========
[alias UNIX_managed_node TMF-client-3.7]
CAT=/data/Tivoli/msg_cat
LIB=/data/Tivoli/lib
BIN=/data/Tivoli/bin
DB=/data/Tivoli/db
MAN=/data/Tivoli/man
APPD=/data/Tivoli/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
Overwrite=
```

Tivoli products response file—as exported

The following file is generated by **wsis** as a response file template for installing the Tivoli products in the final phase of the installation. This is described in step 1.

Note: The [alias] statement for products with long names is shown here on two lines. In the actual file, the [alias] statement must be on a single line.

The list of products to install on cygnus in the [byNode] section is shown on several lines. In the actual file, the entry for each machine in the [byNode] section must be on a single line.

Figure 9. The /tmp/install_applications.rsp file generated by the wsis command

```
[alias aliasname8
UniversalMonitors-3.6-TME_10_Distributed_Monitoring_Universal_Monitors_3.6]

[alias aliasname5
3.6.2-DMN-MR-Tivoli_Distributed_Monitoring_Upgrade,_Version_3.6_3.6.1_to_3.6.2]

[alias aliasname3
InventoryGW-3.6.2-Tivoli_Inventory,_Version_3.6.2,_Gateway]

[machine cygnus]
access=tivoli
interp=solaris2

[byProduct]
aliasname1=cygnus
aliasname2=djatzlau1
aliasname3=oak
aliasname4=cygnus
aliasname5=cygnus
aliasname6=cygnus
aliasname7=cygnus
aliasname8=cygnus
aliasname9=cygnus
aliasname10=cygnus
aliasname11=cygnus

[globals]
InstallAlgorithm=prereq

[alias aliasname6 NTMonitors-3.6-TME_10_Distributed_Monitoring_NT_Monitors_3.6]

[machine oak]
access=tivoli
interp=aix4-r1

[alias aliasname9 UniversalMonitors_362-Tivoli_Distributed_Monitoring_Upgrade,_
Version_3.6_3.6.1_to_3.6.2,_Universal_Monitors]

[alias aliasname4 Sentry2.0.2-3.6-TME_10_Distributed_Monitoring_3.6]
INC=/usr/local/Tivoli/include
Overwrite=

[alias aliasname7 NTMonitors_362-Tivoli_Distributed_Monitoring_Upgrade,_Version_
3.6_3.6.1_to_3.6.2,_NT_Monitors]

[byNode]
cygnus=aliasname1 aliasname4 aliasname5 aliasname6 aliasname7 aliasname8
       aliasname9 aliasname10 aliasname11
djatzlau1=aliasname2
oak=aliasname3

[alias aliasname10 UnixMonitors-3.6-TME_10_Distributed_Monitoring_Unix_Monitors_3.6]

[alias aliasname1 Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2]
@RDBMS_Vendor@=Sybase
@RDBMS_Host@=ALI_host
@RDBMS_DB_Name@=
@RDBMS_DB_Home@=
@RDBMS_DB_Param_one@=
@RDBMS_DB_UserName@=tivoli
@RDBMS_DB_Param_two@=
Overwrite=

[alias aliasname2 Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2]
@RDBMS_Vendor@=Sybase
@RDBMS_Host@=ALI_host
@RDBMS_DB_Name@=
@RDBMS_DB_Home@=
@RDBMS_DB_Param_one@=
@RDBMS_DB_UserName@=tivoli
@RDBMS_DB_Param_two@=
Overwrite=

[machine djatzlau1]
access=tivoli
interp=w32-ix86

[alias aliasname11 UnixMonitors_362-Tivoli_Distributed_Monitoring_Upgrade,_
Version_3.6_3.6.1_to_3.6.2,_UNIX_Monitors]
```

Tivoli products response file—modified

The following file is used to install the Tivoli products in the final phase of the installation process. It is the result of modifying the file in Figure 9 on page 305 as described in step 2.

Figure 10. The /tmp/install_applications.rsp file after editing

```
# ======== Global Information ==========
[globals]
InstallAlgorithm=mdist

# ======== What Goes Where ==========
[byNode]
cygnus=Inventory_new362 DistMon_36 DistMon_362upgrade NT_Monitor_36
       NT_Monitor_362upgrade Uninversal_Monitor_36 Universal_Monitor_362upgrade
       UNIX_Monitor_36 UNIX_Monitor_362upgrade
djatzlau1=Inventory_new362
oak=InventoryGW_362new

# ======== Define Machines ==========
[machine cygnus]
access=tivoli
interp=solaris2

[machine oak]
access=tivoli
interp=aix4-r1

[machine djatzlau1]
access=tivoli
interp=w32-ix86

# =========== Distributed Monitoring Definitions ============
[alias DistMon_36 Sentry2.0.2-3.6-TME_10_Distributed_Monitoring_3.6]
INC=/data/Tivoli/include
Overwrite=

[alias DistMon_362upgrade 3.6.2-DMN-MR-Tivoli_Distributed_Monitoring_Upgrade,_
Version_3.6_3.6.1_to_3.6.2]

[alias NT_Monitor_36 NTMonitors-3.6-TME_10_Distributed_Monitoring_NT_Monitors_
3.6]

[alias NT_Monitor_362upgrade NTMonitors_362-Tivoli_Distributed_Monitoring_Up
grade,_Version_3.6_3.6.1_to_3.6.2,_NT_Monitors]

[alias UNIX_Monitor_36 UnixMonitors-3.6-TME_10_Distributed_Monitoring_Unix_Mon
itors_3.6]

[alias UNIX_Monitor_362upgrade UnixMonitors_362-Tivoli_Distributed_Monitoring_Up
grade,_Version_3.6_3.6.1_to_3.6.2,_UNIX_Monitors]
```

```
[alias Uninversal_Monitor_36 UniversalMonitors-3.6-TME_10_Distributed_Monitoring
_Universal_Monitors_3.6]

[alias Universal_Monitor_362upgrade UniversalMonitors_362-Tivoli_Distributed_Mon
itoring_Upgrade,_Version_3.6_3.6.1_to_3.6.2,_Universal_Monitors]

# =========== Inventory Definitions ============
[alias Inventory_new362 Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2]
@RDBMS_Vendor@=Oracle
# This is the RIM host
@RDBMS_Host@=djatzlau1
# "inventory" is the standard database name, but check with database
# administrator
@RDBMS_DB_Name@=inventory
# Get database home directory from database administrator
@RDBMS_DB_Home@=c:\oracle
# Param_one is server ID, which you get from database administrator
@RDBMS_DB_Param_one@=grande.world
# "tivoli" is the standard user name, but check with database administrator
@RDBMS_DB_UserName@=tivoli
# Param_two does not apply to Oracle
@RDBMS_DB_Param_two@=
Overwrite=

[alias InventoryGW_362new InventoryGW-3.6.2-Tivoli_Inventory,_Version_
3.6.2,_Gateway]
```

Product defaults file

The following file sets the installation options for all Tivoli Enterprise software used in this scenario. This file is created following the procedures in “Setting defaults for installation options for later use” on page 299.

Figure 11. The /tmp/product.defaults file after editing

```
# ==== Distributed Monitoring Definitions =====
[alias solaris2 Sentry2.0.2-3.6-TME_10_Distributed_Monitoring_3.6]
INC=/data/Tivoli/include
Overwrite=

# =========== Inventory Definitions ============
[alias w32-ix86,solaris2 Inventory-3.6.2-Tivoli_Inventory,_Version_3.6.2]
@RDBMS_Vendor@=Oracle
# This is the RIM host
@RDBMS_Host@=djatzlau1
#"inventor" is the standard database name, but check with
# database administrator
@RDBMS_DB_Name@=inventory
# Get database home directory from database administrator
@RDBMS_DB_Home@=c:\oracle
# Param_one is server ID, which you get from database
# administrator
@RDBMS_DB_Param_one@=grande.world
# "tivoli" is the standard user name, but check with
# database administrator
@RDBMS_DB_UserName@=tivoli
# Param_Two does not apply to Oracle
@RDBMS_DB_Param_two@=
Overwrite=

# ======== Gateway Definitions ==========
[alias generic GATEWAY-Tivoli_Gateway]
@GATE_NAME@=@HostName@-gateway
@GATE_PORT@=9494
Overwrite=
# ======== Endpoint Definitions ==========
[alias w32-ix86 TMA-Tivoli_Endpoint]
TMABIN=c:/data/Tivoli/lcf
@EndpointLabel@=@HostName@-ep
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=oak-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=
tapUser=
TapPassword=
Reboot=Yes
Overwrite=

[alias solaris2,aix4-r1 TMA-Tivoli_Endpoint]
TMABIN=/data/Tivoli/lcf
@EndpointLabel@=@HostName@-ep
@EndpointPort@=9495
@EndpointStartupOpts@=
@GatewayName@=oak-gateway
@PolicyRegionName@=None
@CheckLogin@=On
@EndpointStartupTimeout@=300
Overwrite=

# ======== Managed Node Definitions ==========
[alias aix4-r1 TMF-client-3.7]
CAT=/data/Tivoli/msg_cat
LIB=/data/Tivoli/lib
BIN=/data/Tivoli/bin
DB=/data/Tivoli/db
MAN=/data/Tivoli/man
APPD=/data/Tivoli/X11/app-defaults
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
Overwrite=

[alias w32-ix86 TMF-client-3.7]
CAT=c:/data/Tivoli/msg_cat
LIB=c:/data/Tivoli/lib
BIN=c:/data/Tivoli/bin
DB=c:/data/Tivoli/db
MAN=c:/data/Tivoli/man
@CreatePaths@=On
@AutoStart@=On
@SetPort@=On
PR_NAME=cygnus-region
tapUser=
TapPassword=
Reboot=Yes
Overwrite=
```

Deploying the scenario using the Tivoli desktop

To create the sample Tivoli environment from the Tivoli desktop, Becky performs the following procedures:

1. Creates the infrastructure (managed nodes and gateways).

   Note: Endpoints cannot be created from the Tivoli desktop. She creates the endpoints on all the machines in this Tivoli region using the **winstlcf** command or the provided InstallShield images.

2. Installs the Tivoli Enterprise software.

After she installs the Tivoli Enterprise software, she backs up the Tivoli region at regular intervals. By having a backup, she can return this region to a known working state if she encounters installation problems. If she needs additional information about backing up a region, she refers to the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Creating the infrastructure from the Tivoli desktop

To install the managed nodes and gateways from the Tivoli desktop, Becky performs the following steps:

1. Logs in to cygnus as jbecky, which is a valid login for the Installation-Manager administrator.
2. Sets the Tivoli variables. Because the shell for root@cygnus is the Bourne shell, she enters the following command:

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   If she needs additional information, she refers to “Setting Tivoli environment variables” on page 30.
3. Starts the Tivoli desktop. Because cygnus is a Solaris system, she enters the following command:

   ```
   tivoli
   ```
4. Inserts the Tivoli Management Framework 1 of 2 CD into the CD-ROM drive.

   Note: Because cygnus is a Solaris system, she does not need to mount the CD-ROM drive. If she were using another UNIX system, she might have to mount the CD-ROM drive using a command specific to that operating system.
5. From the Tivoli desktop, double-clicks the cygnus-region icon to display the Policy Region: cygnus-region window.
6. Selects **Create → ManagedNode** to display the Client Install window.
7. Creates managed nodes on djatzlau1 and oak by performing the following steps:
   a. Clicks **Add Clients** to display the Add Clients window.
   b. In the **Add Client** field, types `djatzlau1`.
   c. Deselects the **Use Default Access Method** check box.
   d. Selects the **Account** radio button, and types `Administrator` and its account password in the provided fields.
   e. Clicks **Add**. Only the **Add Client** field is cleared. The remainder of the window stays the same.
   f. In the **Add Client** field, types `oak`.
   g. Changes `Administrator` (beside the **Account** radio button) to `root`.
   h. Clicks **Add & Close** to return to the Client Install window, which now contains djatzlau1 and oak in the **Install These Clients** text area.
   i. Clicks **Select Media**, provides the path to the CD-ROM drive, and clicks **Select Media & Close** to display the Install Options window.
   j. In this window, types the installation options from Table 15 on page 279 and clicks **Set** to return to the Client Install window.
   k. Clicks **Install & Close** to display the Client Install window.

l. After this window lists which installation actions will occur, clicks **Continue Install** to display the Tivoli Remote Access Account on NT window.

m. In this window, ensures that the **None** radio button is selected and clicks **Continue** to return to the Client Install window.

   Note: If this scenario required access to remote Windows NT resources (such as file shares), she would select the appropriate radio button and specify the Tivoli remote access account user name and password.

n. When the installation is almost complete, the Confirm Reboot of Clients window is displayed and lists djatzlau1 as needing to be rebooted. Clicks **OK** to reboot djatzlau1.

8. When the installation is complete, she clicks **Close** to return to the Policy Region: cygnus-region window. The managed nodes are installed.
9. Closes the Policy Region: cygnus-region window.
10. Removes the CD from the CD-ROM drive.
11. Creates the gateway on managed node oak using the following steps:
    a. From the Tivoli desktop, she right-clicks the EndpointManager icon and selects **Create Gateway** to display the Create Gateway window.
    b. Provides the installation options from Table 16 on page 280 and clicks **Create & Close** to create the gateway and return to the Tivoli desktop.
12. Endpoints cannot be created from the Tivoli desktop.
    - For djatzlau1 and tmoose, she uses the InstallShield images following the procedure in “Installing the Tivoli desktop on Windows operating systems from CD” on page 65.
    - For cygnus, oak, newcastl, and bbrenner, she uses the **winstlcf** command.
13. Closes the Tivoli desktop.
14. Because **djatzlau1** is a Windows NT managed node, Becky needs to configure **djatzlau1** to send e-mail alerts and other messages through the UNIX mail server on **cygnus**. She can do this at any time after installing the managed node on **djatzlau1**.

    To configure **djatzlau1** to send e-mail through the UNIX mail server on **cygnus**, she performs the following steps:
    a. Logs in to **djatzlau1** as **Administrator**.
    b. Opens a DOS window.
    c. Sets the Tivoli environment variables and starts the bash shell by entering the following commands:
    ```
    %systemroot%\system32\drivers\etc\tivoli\setup_env.cmd
    bash
    ```
    d. Specifies the mail server to be used by managed node **djatzlau1** by entering the following command:
    ```
    wmailhost cygnus
    ```
15. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Installing Tivoli products from the Tivoli desktop

This section provides an example of installing Tivoli Desktop for Windows, Tivoli Inventory, and Tivoli Distributed Monitoring for the sample Tivoli region.

Notes:

1. The first Windows NT managed node (djatzlau1) must be installed from the Tivoli desktop.
2. The first Windows NT endpoint (djatzlau1) must be installed using the InstallShield image.
3. The Windows 98 endpoint (tmoose) must be installed using the InstallShield image.

Installing Tivoli Desktop for Windows

Tivoli Desktop for Windows can only be installed using the provided InstallShield image. Tivoli Desktop for Windows needs to be installed on bbrenner and tmoose. To install, follow the procedure in “Installing Tivoli Desktop for Windows on Windows operating systems” on page 65.

Installing Tivoli Inventory and Tivoli Distributed Monitoring

To install Tivoli Inventory and Tivoli Distributed Monitoring in the sample Tivoli region, perform the following steps:

1. Logs in to cygnus as jbecky, which is a valid login for the Installation-Manager administrator.
2. Sets the Tivoli variables. Because the shell for root@cygnus is the Bourne shell, she enters the following command:

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   If she needs additional information, she refers to “Setting Tivoli environment variables” on page 30.
3. Starts the Tivoli desktop. Because cygnus is a Solaris system, she enters the following command:

   ```
   tivoli
   ```
4. Installs Tivoli Inventory on cygnus and djatzlau1 by performing the following steps:
   a. Inserts the Tivoli Inventory, Version 3.6.2 CD into the CD-ROM drive.
   b. From the **Desktop** menu, selects **Install → Install Product** to display the Install Product window.
   c. In the **Select Product to Install** list, selects **Tivoli Inventory, Version 3.6.2** to display the Install Options window.
   d. Uses the values for the installation options from Table 22 on page 282 and clicks **Set** to return to the Install Product window.
   e. Ensures that **cygnus** and **djatzlau1** are the only machines listed in the **Clients to Install On** list.
   f. Clicks **Install & Close** to display the Product Install window, which provides a list of the operations as they occur.
   g. Clicks **Continue Install** to start the installation.
   h. When the installation is complete, clicks **Close** to return to the Install Product window.

   After installing Tivoli Inventory, she performs any post installation procedures described in the Tivoli Inventory, Version 3.6.2 product documentation.
5. Installs Tivoli Inventory Gateway on oak by performing the following steps:
   a. In the **Select Product to Install** list, selects **Tivoli Inventory, Version 3.6.2, Gateway**.

      Note: There are no installation options for the Tivoli Inventory Gateway.

   b. Ensures that **oak** is the only machine listed in the **Clients to Install On** list.
   c. Clicks **Install & Close** to display the Product Install window, which provides a list of the operations as they occur.
   d. Clicks **Continue Install** to start the installation.
   e. When the installation is complete, clicks **Close** to return to the Install Product window.
   f. Clicks **Close** to close the Install Product window
   g. Removes the CD from the CD-ROM drive.

   After installing Tivoli Inventory Gateway, she performs any post installation procedures described in the Tivoli Inventory, Version 3.6.2 product documentation.

6. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.
7. Installs Tivoli Distributed Monitoring on cygnus by performing the following steps:
   a. Inserts the TME 10 Distributed Monitoring, Version 3.6 CD into the CD-ROM drive.
   b. From the **Desktop** menu, selects **Install → Install Product** to display the Install Product window.
   c. In the **Select Product to Install** list, selects **TME 10 Distributed Monitoring, Version 3.6** to display the Install Options window.
   d. Uses the values for the installation options from Table 23 on page 283 and clicks **Set** to return to the Install Product window.
   e. Ensures that **cygnus** is the only machine listed in the **Clients to Install On** list.
   f. Clicks **Install & Close** to display the Product Install window, which provides a list of the operations as they occur.
   g. Clicks **Continue Install** to start the installation.
   h. When the installation is complete, clicks **Close** to return to the Install Product window.

   After installing Tivoli Distributed Monitoring, she performs any post installation procedures described in the TME 10 Distributed Monitoring, Version 3.6 product documentation.

8. Installs the monitor collections for Tivoli Distributed Monitoring on cygnus by performing the following steps:
   a. In the **Select Product to Install** list, selects **TME 10 Distributed Monitoring Universal Monitors 3.6**.

      Note: There are no installation options for the monitor collections.
   b. Ensures that **cygnus** is the only machine listed in the **Clients to Install On** list.
   c. Clicks **Install & Close** to display the Product Install window, which provides a list of the operations as they occur.
   d. Clicks **Continue Install** to start the installation.
   e. When the installation is complete, clicks **Close** to return to the Install Product window.
   f. Repeats steps 8a through 8e but replaces:
      - TME 10 Distributed Monitoring Universal Monitors 3.6 with TME 10 Distributed Monitoring NT Monitors 3.6 on the first repetition.

- TME 10 Distributed Monitoring NT Monitors 3.6 with TME 10 Distributed Monitoring Unix Monitors 3.6 on the second repetition.

g. After installing the three monitor collects, clicks **Close** to close the Install Product window.

h. Removes the CD from the CD-ROM drive.

After upgrading Tivoli Distributed Monitoring, she performs any post installation procedures described in the Tivoli Distributed Monitoring, Version 3.6.2 product documentation.

9. Upgrades Tivoli Distributed Monitoring and the monitor collections on cygnus by performing the following steps:

   a. Inserts the Tivoli Distributed Monitoring, Version 3.6.2 CD into the CD-ROM drive.

   b. From the **Desktop** menu, selects **Install → Install Product** to display the Install Product window.

   c. In the **Select Patch to Install** list, selects **Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2**.

      Note: There are no installation options for the upgrades.

   d. Ensures that **cygnus** is the only machine listed in the **Clients to Install On** list.

   e. Clicks **Install & Close** to display the Patch Install window, which provides a list of the operations as they occur.

   f. Clicks **Continue Install** to start the installation.

   g. When the installation is complete, clicks **Close** to return to the Install Patch window.

   h. Repeats steps 9c through 9g, but replaces:

      - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2 with Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, Universal Monitors on the first repetition
      - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, Universal Monitors with Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, NT Monitors on the second repetition.
      - Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, NT Monitors with Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2 UNIX Monitors on the third repetition.

   i. After upgrading the base product and the three monitor collects, clicks **Close** to close the Install Product window.

   j. Removes the CD from the CD-ROM drive.

10. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Deploying the scenario using Tivoli commands

To create the sample Tivoli environment using Tivoli commands, Becky performs the following procedures:

1. Creates Tivoli Management Framework managed resources (managed nodes, gateway, endpoints).
2. Installs and upgrades the Tivoli Enterprise applications.

Creating managed resources using Tivoli commands

To create the sample Tivoli environment using Tivoli commands, Becky performs the following steps:

1. Logs in to cygnus as jbecky, which is a valid login for the Installation-Manager administrator.
2. Sets the Tivoli variables. Because the shell for root@cygnus is the Bourne shell, she enters the following command.

   ```
   . /etc/Tivoli/setup_env.sh
   ```

   If she needs additional information, she refers to "Setting Tivoli environment variables" on page 30.
3. On djatzlau1, locally installs Tivoli Remote Execution Service following the procedure in "Manually installing Tivoli Remote Execution Service" on page 341.
4. On cygnus, inserts the Tivoli Management Framework 1 of 2 CD into the CD-ROM drive.
5. Uses the **wclient** command to create the managed node on djatzlau1. She enters the following command:

   ```
   wclient -I -c /cdrom/cdrom0 -p cygnus-region \
    CAT=c:/data/Tivoli/msg_cat LIB=c:/data/Tivoli/lib \
    BIN=c:/data/Tivoli/bin DB=c:/data/Tivoli/db \
    @CreatePath@=1 @AutoStart@=1 @SetPort=1 djatzlau1
   ```

   When prompted, she enters the installation password, `manageIT`.
6. Uses the **wclient** command to create the managed node on oak. She enters the following command:

   ```
   wclient -I -c /cdrom/cdrom0 -p cygnus-region \
    CAT=/data/Tivoli/msg_cat LIB=/data/Tivoli/lib \
    BIN=/data/Tivoli/bin DB=/data/Tivoli/db \
    MAN=/data/Tivoli/man \
    APPD=/data/Tivoli/X11/app-defaults @CreatePath@=1 \
    @AutoStart@=1 @SetPort=1 oak
   ```

   When prompted, she enters the installation password, `manageIT`.
7. Removes the CD from the CD-ROM drive.
8. Uses the following **wcrtgate** command to create a gateway on managed node oak. She enters the following command:

   ```
   wcrtgate -n oak-gateway -h oak
   ```
9. On djatzlau1, she uses the InstallShield images to create the endpoint. She uses the procedure from "Installing endpoints using InstallShield" on page 215.

   Because djatzlau1 is also a Windows NT managed node and she is sitting at this machine, she configures djatzlau1 to send e-mail alerts and other messages through the UNIX mail server on cygnus. She performs the following steps:

   a. Logs in to djatzlau1 as Administrator.

   b. Opens a DOS window.

   c. Sets the Tivoli variables and starts the bash shell by entering the following commands:

   ```
   %systemroot%\system32\drivers\etc\tivoli\setup_env.cmd
   bash
   ```

   d. Specifies the mail server to be used by managed node djatzlau1 by entering the following command:

   ```
   wmailhost cygnus
   ```

10. On tmoose, she uses the InstallShield images to create the endpoint. She uses the procedure from "Installing endpoints using InstallShield" on page 215.
11. Uses the **winstlcf** command to create endpoint on cygnus, oak, newcastl, and bbrenner. She enters the following commands:

    ```
    winstlcf -d /data/Tivoli/lcf -g oak-gateway \
     -n cygnus-ep cygnus
    winstlcf -d /data/Tivoli/lcf -g oak-gateway \
     -n oak-ep oak
    winstlcf -d /data/Tivoli/lcf -g oak-gateway \
     -n newcastl-ep newcastl
    winstlcf -d /data/Tivoli/lcf -g oak-gateway \
     -N djatzlau1 -n bbrenner-ep bbrenner
    ```

    Note: When she runs these commands, she is prompted for the system user account and password. She enters the user ID and password for the root user.
12. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Installing Tivoli products using Tivoli commands

This section provides an example of installing the Tivoli Inventory and Tivoli Distributed Monitoring to the sample Tivoli region.

Installing Tivoli Inventory

To install Tivoli Inventory in the sample Tivoli environment, Becky performs the following steps:

1. On cygnus, enters the following command to install Tivoli Inventory, Version 3.6.2 on cygnus and djatzlau1:

   ```
   winstall -c /cdrom/cdrom0 @RDBMS_Vendor@=Oracle \
    @RDMBS_Host@=djatzlau1 @RDBMS_DB_Name@=inventory \
    @RDBMS_DB_Home@=c:\oracle \
    @RDBMS_DB_Param_one@=grande.world \
    @RDBMS_UserName@=tivoli -i 362_INV cygnus djatzlau1
   ```
2. Enters the following command to install Tivoli Inventory Gateway, Version 3.6.2 on oak:

   ```
   install -c /cdrom/cdrom0 -i 362_GW oak
   ```
3. Performs any post installation procedures described in the Tivoli Inventory, Version 3.6.2 product documentation.
4. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

Installing Tivoli Distributed Monitoring

To install and upgrade Tivoli Distributed Monitoring in the sample Tivoli region, the Installation-Manager installs the following products and patches on the cygnus:

- TME 10 Distributed Monitoring 3.6
- TME 10 Distributed Monitoring Universal Monitors 3.6
- TME 10 Distributed Monitoring NT Monitors 3.6
- TME 10 Distributed Monitoring Unix Monitors 3.6
- Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2
- Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, Universal Monitors
- Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2, NT Monitors
- Tivoli Distributed Monitoring Upgrade, Version 3.6/3.6.1 to 3.6.2 UNIX Monitors

To install and upgrade Tivoli Distributed Monitoring on cygnus, the Installation-Manager performs the following steps:

1. On cygnus, inserts the TME 10 Distributed Monitoring, Version 3.6 CD into the CD-ROM.
2. From the command line, enters the following commands:
   ```
   winstall -c /cdrom/cdrom0 -i SENT36 INC=/data/Tivoli/include cygnus
   winstall -c /cdrom/cdrom0 -i UNVM36 cygnus
   winstall -c /cdrom/cdrom0 -i NTMON36 cygnus
   winstall -c /cdrom/cdrom0 -i UNXM36 cygnus
   ```
3. Removes the CD from the CD-ROM drive
4. Performs any post installation procedures described in the TME 10 Distributed Monitoring, Version 3.6 product documentation.
5. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.
6. Inserts the Tivoli Distributed Monitoring, Version 3.6.2 CD into the CD-ROM drive.
7. From the command line, enters the following commands:
   ```
   wpatch -c /cdrom/cdrom0 -i DMN362MR -n
   wpatch -c /cdrom/cdrom0 -i UNVMN362 -n
   wpatch -c /cdrom/cdrom0 -i NTMON362 -n
   wpatch -c /cdrom/cdrom0 -i UNXMN362 -n
   ```

   Note: Becky does not specify which machines to install the upgrade images on. The **–n** option ensures that the upgrade is installed to each machine in the Tivoli region where the base component is already installed.
8. Removes the CD from the CD-ROM drive.
9. Performs any post installation procedures described in the Tivoli Distributed Monitoring, Version 3.6.2 product documentation.
10. Backs up the Tivoli region following the procedure in the *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

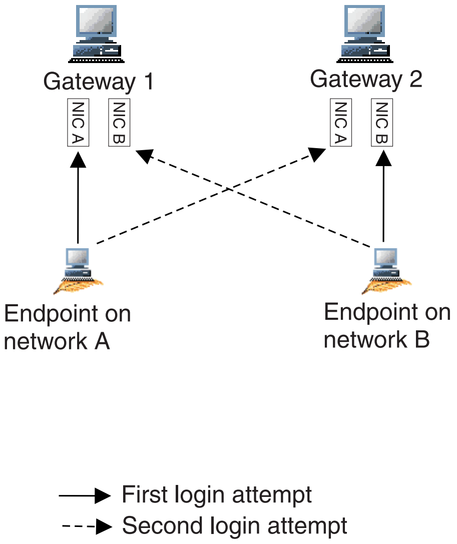
Chapter 22. Configuring multi-NIC gateways and endpoints

Tivoli Management Framework enables you to control which network interface card (NIC) an endpoint connects to on a multi-NIC gateway. This feature is especially useful when you want an endpoint to log in to a NIC on one gateway, and if that fails, to log in to a NIC on a different gateway rather than attempting another NIC on the first gateway.

For example, consider a deployment with two networks: network A and network B, and two gateways: Gateway 1 and Gateway 2. Each gateway has two NICs: one for network A and one for network B. The NICs are configured as follows:

- **Gateway 1**
  - NIC A: IP address 10.10.10.1 and hostname nwa–gw1
  - NIC B: IP address 20.20.20.1 and hostname nwb–gw1
- **Gateway 2**
  - NIC A: IP address 10.10.10.2 and hostname nwa–gw2
  - NIC B: IP address 20.20.20.2 and hostname nwb–gw2

This scenario is shown in the following illustration.



During normal logins, an endpoint from network A should first attempt to log in to Gateway 1 using the NIC for network A. If this login attempt fails, the endpoint should attempt to log in to Gateway 2 using the NIC for network A. Conversely, an endpoint in network B should first attempt to log in to Gateway 2 using the NIC for network B, and if that fails, attempt to log in to Gateway 1 using the NIC for network B.

To implement this scenario in your deployment, you must configure your deployment as follows:

- Modify the **select_gateway_policy** script.
- Configure gateways to not send the gateway aliases list to endpoints.
- Configure endpoints to use the appropriate local IP interface.

- Disable broadcast logins for endpoints.

When you configure your deployment in this way, endpoints will first try to log in to the appropriate NIC on the assigned gateway. If they are unable to reach the assigned gateway, they try to connect to the next gateway listed in their login interfaces list, which is the appropriate NIC on the other gateway. Because broadcast logins are disabled, the endpoints will not go into isolation mode.

The following sections describe how to configure these settings.

Modifying select_gateway_policy

The **select_gateway_policy** script determines the set of gateways that are allowed to manage the endpoint. To control which NIC card an endpoint contacts on the gateway, you must modify this script to do the following:

- Remove existing gateways from the login interfaces list of the endpoint.
- Add no intercepting gateways to the login interfaces list.
- Add only the host names or IP addresses that identify interfaces on gateways that are to be used by the endpoint.

For more information about the **select_gateway_policy** script, see the chapter about endpoint policy scripts in the *Tivoli Management Framework Reference Manual*.

The following sample **select_gateway_policy** script is configured to enable the deployment with two networks (network A and network B) and two gateways (Gateway 1 and Gateway 2) as described earlier in this section.

```
#!/bin/sh

FOUND=FALSE

# Subnet info
SUBNET_NWA=10
SUBNET_NWB=20

# Gateway oids
GW1_OID="1234567890.2.556"
GW2_OID="1234567890.5.556"

# Gateway hostnames for network A interfaces
GW1_NWA_HN="nwa-gw1"
GW2_NWA_HN="nwa-gw2"

# Gateway hostnames for network B interfaces
GW1_NWB_HN="nwb-gw1"
GW2_NWB_HN="nwb-gw2"


# Endpoint IP from select_gateway_policy args
EP_IP=$5

# Get subnet of endpoint
SUBNET='echo $EP_IP | cut -d"." –f3'

# Check and return list of primary and failover gateways
# in gwoid|hostname|mode format
if [ $SUBNET -eq $SUBNET_NWB ]
then
    FOUND=TRUE
    echo "$GW1_OID|$GW1_NWB_HN|0 $GW2_OID|$GW2_NWB_HN|0 3"
fi
```

```
if [ $SUBNET -eq $SUBNET_NWA ]
then
    FOUND=TRUE
    echo "$GW2_OID|$GW2_NWA_HN|0 $GW1_OID|$GW1_NWA_HN|0 3"
fi

if [ "$FOUND" = "TRUE" ]
then
    exit 0
else
    exit 1
fi
```

Configuring the gateway

By default, a gateway creates an aliases list that contains all of the gateway's configured interfaces and sends it to an endpoint during a normal login. If the endpoint cannot log in to the gateway using the primary IP address, the endpoint attempts to log in to the aliases on the list. If you want to an endpoint to log in to certain aliases only, you must configure the gateway so that it does not send the aliases list.

To prevent the gateway from sending its aliases list to endpoints during normal logins, use the **wgateway update_gwalias** command. When you set this option to **0**, the gateway does not send its aliases to an endpoint during a normal endpoint login. The endpoint attempts to log in to its assigned gateway using only one IP address. If the endpoint cannot log in to the gateway using this IP address, the endpoint attempts to log in to the gateways in its login interfaces list.

Configuring endpoints to log in to specific NICs

To configure endpoints to log in to a specific NIC on each gateway, you must specify the correct interface for the endpoint to use. In the example deployment in this section, you would configure endpoints in network A to use the 10.10.10.*n* interface only, and endpoints in network B to use the 20.20.20.*n* interface only. You set this using the **lcfd local_ip_interface** command.

Moreover, you must disable broadcast logins for endpoints in both network A and network B. You configure this using the **lcfd bcast_disable** command.

Chapter 23. Troubleshooting

Use this chapter to resolve problems you have installing Tivoli software or using Tivoli Software Installation Service (SIS), the Tivoli desktop, or the command line. This chapter contains the following sections:

- "Problems using Tivoli Software Installation Service"

  This section helps you debug problems that might occur when using or installing Tivoli software with SIS.
- "Installation problems using Tivoli Management Framework" on page 326

  This section helps you debug problems that might occur when installing Tivoli software using the Tivoli desktop or the command line.
- "Problems installing NetWare gateways" on page 327

  The section helps you debug problems that might occur when installing the NetWare gateway.
- "Problems logging in to the Tivoli desktop" on page 328

  This section helps you debug problems that might occur when using the Tivoli desktop to connect to a Tivoli server or managed node.
- "Troubleshooting a RIM installation" on page 328

  This section helps you test RDBMS Interface Module (RIM) connectivity and provides a template to use when requesting support for a RIM problem.
- "Troubleshooting a RIM configuration" on page 329

  This section helps you run the RIM database agent manually to troubleshoot RIM connection problems after you set the Tivoli environment variables.

Problems using Tivoli Software Installation Service

I cannot start the Tivoli Software Installation Service console. What is wrong?

- If you have just shut down the Tivoli Software Installation Service depot, wait a few seconds before trying to use the Tivoli Software Installation Service client. You cannot reconnect to the Tivoli Software Installation Service depot until its shutdown process is complete.
- If you are trying to start the Tivoli Software Installation Service console from the Tivoli desktop, ensure that the Tivoli Software Installation Service client is installed on the system where the Tivoli desktop is running. You cannot start the Tivoli Software Installation Service console on a remote machine.

  If you are using Tivoli Desktop for Windows, the Tivoli Software Installation Service client must be installed on the system where the Tivoli desktop is installed, not the managed node to which the desktop is connected.
- If you are trying to start the Tivoli Software Installation Service console on a UNIX system, you must enable X Window System access to that machine. This is necessary even when the Tivoli Software Installation Service console is on the same machine as the X Window System display. For example, if your system name is fido, enter the following command:

  ```
  xhost +fido
  ```

- Examine the file /tmp/sis_client.out (UNIX) or $DBDIR\tmp\sis_client.out (Windows) on the machine where the Tivoli Software Installation Service client is running. This file records any problems that occur when starting the client. When you use the **wsisgui** command, the file is updated only if there is an error. When you start the Tivoli Software Installation Service console, the file is updated regardless of whether problems were encountered.
- Examine the file /tmp/sissvr.log (UNIX) or $DBDIR\tmp\sissvr.log (Windows) on the machine where the Tivoli Software Installation Service depot is running. This file records errors starting the Tivoli Software Installation Service depot.
- Examine the client logs for clues. The client log directory is listed in the `client_data_dir` entry in the file $DBDIR/sisclnt.ini on the machine where the Tivoli Software Installation Service client is installed. Using any HTML browser, open the file sis-*date-time*.html in the most recent subdirectory (iu-*date-time*) of the client logs.

  The following log entry indicates that the Tivoli Software Installation Service client could not connect to the Tivoli Software Installation Service depot:

  ```
  Opening SIS log at: Sat Apr 01 18:18:36 CST 2000
  Connecting to SIS Depot
  ServerProxy: trying to use SisDepot named oak
  ServerProxy: server named "oak" is not responding...
  com.tivoli.framework.runtime.ExObjAdapter,
  minor code = 23, completion status = No
  SIS failed
  ```

  A common cause is that the Tivoli administrator using the Tivoli Software Installation Service client does not have a login entry for user root on the machine where the Tivoli Software Installation Service depot is installed. To correct this problem, define the additional login name for the Tivoli administrator. For example, if the Tivoli Software Installation Service depot is on oak.dev.tivoli.com, add the login root@oak.dev.tivoli.com.

The Tivoli Software Installation Service console does not list all the machines in my Tivoli region or all the products that are installed on a specific machine. What can I do?

- If someone else used another Tivoli Software Installation Service console or the SIS commands to install products, create policy regions, or define machines while your Tivoli Software Installation Service console is open, use the **Refresh** button on either the Select Machines or Select Products windows to refresh your Tivoli Software Installation Service console.
- If it appears that machines are missing from the Select Machines window, ensure that the **Policy Region** and **Machine Type** filters are set correctly. Remember that by default endpoints are not listed when you select **All** in the **Machine Type** filter.

How do I find out where my Tivoli Software Installation Service depot is?

To list the machines where the Tivoli Software Installation Service depot is installed, run any Tivoli Software Installation Service command with the **–D** option. For example:

```
wsis -D
```

To determine the Tivoli Software Installation Service depot to which the Tivoli Software Installation Service console is connected, view the Tivoli Software Installation Service main window. The name of the host running the Tivoli Software Installation Service depot is listed at the top of installation worksheet in this window.

Installing from a response file or selecting a cell in an installation worksheet, I get the following message:

```
The node node is locked by another user user@depotname
```

This message indicates that the node is locked by another process. A probable cause is that another administrator has an installation worksheet on which one or more products are selected for this machine or is currently performing an installation to this machine.

I am running Tivoli Software Installation Service from the command line using the wsis command. It says I have a Tivoli region lock and an IR lock. How do I remove them?

Ensure that the Tivoli Software Installation Service console is not running. If it is, exit and the locks will be removed. Otherwise, perform the following steps to remove the locks manually:

1. Remove the Tivoli region lock:

   ```
   wregister -u SIS
   ```

2. Remove the install repository lock:

   ```
   rm $IRDIR/TMR/Defaults/ir.lck
   ```

   Where $IRDIR represents the name of the directory containing the install repository. To locate the install repository, refer to "Finding the install repository" on page 108.

I am trying to create a managed node or endpoint. The policy region I want to put this resource into does not appear in the Policy Region selection list. What should I do?

Make sure that the policy region in which you want add the resource has the associated resource associated with it. Tivoli Software Installation Service displays only those policy regions that can accept the managed resource you want to create.

When I added a machine to Tivoli Software Installation Service, I used the wrong connection type, user ID, or password. How do I change it?

Using the Tivoli Software Installation Service console, go to the Select Machines window and select the machine. Click **Remove** to remove the definition and then click **Add Machine** to redefine the machine.

Using the command line, create a response file with a [machine] section that contains the correct information for the machine. Use the **wsis** command with the **–i** option to import that response file into Tivoli Software Installation Service.

How do I modify a product index (.IND) file?

Note: Do not modify product index files unless instructed to do so by your support provider.

If you change an index file, you must cause Tivoli Software Installation Service to update the product.sav file before these changes take effect.

To cause Tivoli Software Installation Service to update the product.sav file, complete these steps:

1. Stop the Tivoli Software Installation Service depot.
2. Remove the product.sav file in the image directory.

3. Remove the product.sav file in product-specific subdirectory of the local defaults directory, TMR/*region_number*/ProductDefaults, if it exists.
4. Remove the miniprod.sav file from the TMR/Defaults directory if the install repository is used in non-shared mode. Remove miniprod.sav from the TMR/*region_number* directory.
5. Restart the Tivoli Software Installation Service depot.

Tivoli Software Installation Service detects that the product.sav file does not exist and rebuilds the file by scanning the Framework, Products, and Patches subdirectories.

Installation problems using Tivoli Management Framework

Why can I not remotely install using rexec or rsh?

Generally, the service is not correctly configured. To verify whether the service is correctly configured issue one of the following commands:

For rexec

```
rexec -l root -p password target_system ls
```

For rsh

```
rsh -l root target_system ls
```

If you do not receive a listing of files in this home directory, you need to configure the service. For information, contact your operating system documentation or contact your operating system vendor. If you receive the directory listing, contact your support provider.

What happens when the login shell of the root user is not sh, ksh, or bash?

The login shell of the root user cannot be the C shell. To successfully install Tivoli software, the login shell for the root user must be sh, ksh, or bash. If you use Tivoli Software Installation Service, the prerequisite checks determine whether the login shell of the root user is correct and provides appropriate error messages when it is not correct.

How do I remove a managed node from my installation?

Use the **wuninst** command to remove applications installed on the managed node. Then run the **wunstmn** command to remove the managed node. For details, refer to Chapter 20, “Uninstalling a Tivoli environment,” on page 255.

I am trying to create a managed node from the Tivoli desktop but the Create menu does not show the ManagedNode resource, what should I do?

Add the ManagedNode resource to the list of available resources for the policy region.

Why can I not use a fully qualified domain name to install an endpoint or managed node to a Windows system?

Windows operating systems must be in the same domain or in a domain that has a trust relationship allowing for remote access.

When I try to launch the Tivoli desktop, I receive the following error: `The ordinal` *`nnnn`* `could not be located in the dynamic link library MFC42.DLL`. How do I resolve this error?

Tivoli desktop requires version 6.*x.x* of the MFC42.DLL file. However, other applications might require version 4.*x.x* of the MFC42.DLL file. You cannot simultaneously run an application that uses version 6.*x.x* of the MFC42.DLL file and an application that uses version 4.*x.x*, because different versions of this DLL file cannot be loaded into memory at the same time. To resolve this issue, install version 4.*x.x* of the MFC42.DLL file in the working directory of the application that uses it and run applications that use different versions of MFC42.DLL separately. You must exit out of all the running applications that use one version of the MFC42.DLL file before launching an application that uses a different version of the file.

Problems installing NetWare gateways

The object dispatcher (oserv) on a NetWare gateway did not start after installation. What do I do?

If you did not perform the installation from a system running Novell Requester, the object dispatcher is not automatically started. To start it manually after installation, enter the following commands:

```
oservend
SYS:Tivoli\bin\nwr-ix86\bin\oserv1st -s installation_password
```

where *installation_password* is the installation password if one was defined.

I ran the oserv1st command to manually start the object dispatcher (oserv) after installation and it did not start. What do I do?

If you specified an incorrect installation password, you need to run the **oserv1st** command using the correct installation password.

For all other situations, you need to uninstall the managed node and reinstall it using the following steps:

1. From the command line of the Tivoli server, enter the following command:

   ```
   wrmnode hostname
   ```

   where *hostname* is the host name of the NetWare system.
2. Enter the following command:

   ```
   odadmin odlist rm_od dispatcher_id
   ```

   where *dispatcher_id* is the ID of the object dispatcher running on the NetWare gateway.
3. Clean up the object database on the Tivoli server by entering the following command:

   ```
   wchkdb -ux
   ```
4. If the binaries are already installed, you do not need to reinstall them, but you do need to delete the contents of the $DBDIR directory on the NetWare system before reinstalling the NetWare gateway.
5. Reinstall the NetWare gateway following the procedure in “Registering the NetWare managed node” on page 353.

How do I restart the object dispatcher (oserv) on a NetWare gateway?

If you stopped the object dispatcher and need to manually restart it, enter the following command:

```
SYS:Tivoli\bin\nwr-ix86\bin\oservrun
```

How do I create a NetWare account?

Generally the accounts required by Tivoli Management Framework are created during installation. If you need to manually create the tmersrvd or lcfrsrvd accounts on NetWare systems, use the provided **addadmin** utility.

Do not use the NetWare **NWADMIN** utility to create these accounts. For complete details on creating these accounts, refer to "NetWare accounts" on page 351.

Problems logging in to the Tivoli desktop

- Check your host name, user account, and password for spelling or case. These fields are case sensitive.
- Ensure that TCP/IP is working and able to connect to your host. Use telnet to verify that you are able to connect to and log in to your host using the same user account and password.
- Ensure that the oserv is running on the managed node.
- Ensure that Remote Connections are allowed to the managed node. Refer to "Enabling remote connections" on page 32 for more information.
- Ensure that your login account is a valid Tivoli administrator:
  1. From a working Tivoli desktop, open the Administrators window.
  2. Verify that your login account is one of the listed administrators.
- Ensure that your login account has a valid, non-blank, group property as follows:
  1. From a working desktop, open the Administrators window.
  2. Right-click on the administrator icon.
  3. Select **Edit Properties** to open the Administrator Properties window. Ensure that the **Group Name** field contains a valid, non-blank, value.
- Ensure you are connecting to a managed node whose label and host name are the same.

Troubleshooting a RIM installation

You can use the **wrimtest** command to troubleshoot RIM connection problems. To correctly test RIM connectivity, you need to know the name of the RIM object. You can determine the names of RIM objects by running the **wlookup** command with the **–ar RIM** options.

After obtaining the RIM object name, run the command as follows:

```
wrimtest -l rim_object_name
```

where *rim_object_name* is the name of the RIM object.

To exit the **wrimtest** utility, enter the `x` command option.

For additional information about the **wlookup** and **wrimtest** commands, refer to the *Tivoli Management Framework Reference Manual*.

If you need to contact your Tivoli service provide for problems concerning RIM, be prepared to provide the following information:

- The following RIM host and database client questions:
  - What database vendor are you using—DB2, Oracle, Sybase, Microsoft SQL Server, or Informix?
  - What release of the database client are you using?
  - What operating system are you using to run the database client?
  - What release of the operating system are you using to run the database client?

  - Are the RIM host and the Tivoli server on the same machine?
- The following database server questions:
  - What release of the database server are you using?
  - What operating system are you using to run the database server?
  - What release of the operating system are you using to run the database server?
- The following Tivoli Management Framework questions:
  - What version of Tivoli Management Framework are you using (including maintenance releases)?
  - Are the RIM host and the Tivoli server at the same release level?
  - What patches, if any, are installed on the Tivoli server and RIM host?
- The following general questions:
  - What were you doing when the error occurred?
  - Can you establish connectivity using the native database language?
  - Can you select data from your database using the native database language?
  - Can you run **wrimtest** command with the **–l** *rim_object_name* option with the RIM object you are working with successfully? Be prepared to provide this output.
  - Do you have any recommendations as to the cause or solution to this problem?

Troubleshooting a RIM configuration

You can run the RIM database agent manually to troubleshoot RIM connection problems using the **RIM_***DB***_Agent** command located in $BINDIR/bin. It is in $PATH after you set the Tivoli environment variables. To run the RIM database agent manually, do the following:

1. Set the shared library path. Note the following considerations:
   - If your RIM host is on a Windows NT system, make sure that the PATH system environment variable includes the path to the database vendor DLL files.
   - If your RIM host is on a UNIX system, you need to set the shared library path only if you are using DB2 or Informix.
   - For DB2, the shared library path must include $INSTHOME/sqllib/lib.
   - For Informix, the shared library path must include $INFORMIXDIR/lib:$INFORMIXDIR/lib/cli:$INFORMIXDIR/lib/esql.

     The following table lists the name of the environment variable to use as the shared library path for each operating system type.

Operating System	Variable
AIX	LIBPATH
HP-UX	SHLIB_PATH
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH
Windows	PATH

2. Run the **wgetrim** command to obtain the options for the RIM object. Use the values of these options in the next step.

3. Run **RIM_*DB*_Agent** using the values of the RIM object options that you obtained in step 2 on page 329. The syntax of **RIM_*DB*_Agent** is as follows:

   RIM_*DB*_Agent –d *database* **–u** *user* [**–p** *password*] **–H** *rdbms_home* **–s** *server_id* [**–I** *instance_home*]

   where:

   DB Specifies the database type. Options are `DB2`, `Informix`, `MS_SQL`, `Oracle`, and `Sybase`.

   –d *database*
   Specifies the database ID.

   –u *user*
   Specifies the RDBMS user name.

   –p *password*
   Specifies the RDBMS user password.

   –H *rdbms_home*
   Specifies the database home directory. That is, the location in which the database client is installed on the RIM host.

   –s *server_id*
   Specifies the server ID.

   –I *instance_home*
   Specifies the instance home directory. (DB2 only)

If the connection is successful, the **RIM_*DB*_Agent** command displays the message `Connection Successful.` If the connection fails, the appropriate database errors are displayed.

Chapter 24. Reading index files

Caution

Do not edit an index file unless explicitly instructed to by your support provider.

Use this chapter to find how how to read the index (.IND) files shipped on Tivoli product CDs.

Note: This chapter uses the term *product* to refer to both products and patches.

General syntax

The general syntax for lines in a product index file is as follows:

```
tag:keyword:value_list
```

where:

tag
Indicates the registered product tag. It is the same for all lines in the index file.

keyword
Indicates the type of information. The keywords are as follows:

- **description**
- **revision**
- **patch_for**
- **id**
- **fp**
- **gui**
- **patch_id**
- **depends**
- **lcf_allow**

value_list
Indicates the values in a colon-separated list. The interpretation of these values depends on the keyword on that line.

Syntax for the description line

The description line describes the index file. There is only one description line in an index file.

The syntax for the description line is as follows:

```
tag:description:package_description:license_tag
```

Explanation of description syntax

package_description
Indicates the free-form description of the installation image. This

description is displayed when installing the product using the Tivoli desktop or Tivoli Software Installation Service.

license_tag
Indicates a value used internally by the installation mechanisms.

Examples of description lines

1. The following is an example of the description line in the 42_INV_F.IND file:

   ```
   InventoryServer:description:Tivoli Inventory Server, Version 4.2: \
      TME 10 Inventory
   ```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryServer`
   - The description for this component is `Tivoli Inventory Server, Version 4.2`
   - The product license key for this component is `TME 10 Inventory`

2. The following is an example of the description line in the 42_INV.IND file:

   ```
   InventoryServer_4.2:description:TIvoli Inventory Server, Version 4.2: \
      InventoryServer_4.2
   ```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryServer_42`
   - The description for this component is `Tivoli Inventory Server, Version 4.2`
   - The product license key for this component is `InventoryServer_4.2`

Syntax for the revision line

The revision line indicates the version of the product. There can be only one revision line in an index file. This line is required only for product installation images.

The syntax for the revision line is as follows:

```
tag:revision:version
```

Explanation of revision syntax

version The version and release numbers of the product.

Examples of revision syntax

1. The following is an example of the revision line in the 42_INV_F.IND file:

   ```
   InventoryServer:revision:4.2
   ```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryServer`
   - The version and release of this component is `4.2`

2. The following is an example of the revision line in the 42_GW_FR.IND file:

   ```
   InventoryGateway:revision:4.2
   ```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryGateway`
   - The version and release of this component is `4.2`

Syntax for the patch_for line

The patch_for line indicates a dependency for a current Tivoli Enterprise product. There can be only one patch_for line in an index file. This line is required only for patch installation images.

The syntax for the patch_for line is as follows:

```
tag:patch_for:package_tag::
```

Explanation of patch_for syntax

package_tag
Indicates the Tivoli Enterprise product that must be installed before installing this patch image. This item prevents an installation of a patch unless a required product is already installed.

Note: There are two colons (::) after the *package_tag* option.

Examples of patch_for lines

1. The following is an example of the patch_for line in the 42_INV.IND file:
   ```
   InventoryServer_4.2:patch_for:InventoryServer
   ```
   In this example, the following is known:
   - The registered product tag for this component is `InventoryServer_42`
   - This patch is for a product with a product tag of `InventoryServer`
2. The following is an example of the patch_for line in the 42_GW.IND file:
   ```
   InventoryGateway_4.2:patch_for:InventoryGateway
   ```
   In this example, the following is known:
   - The registered product tag for this component is `InventoryGateway_42`
   - This patch is for a product with a product tag of `InventoryGateway`

Syntax for the id Lines

Each id line defines an identifier. There can be multiple id lines in a single index file. Identifiers are used by other entries in the index file. Identifiers can be either variable identifiers or package identifiers.

The syntax for the id line is as follows:

```
tag:id:name:description:valid_for:subdir:value:env:chain_id
```

Explanation of id syntax

name Indicates the name of the identifier. This name is used when the identifier is referenced by an fp or gui line. This is the value used when installing the product from the command line.

description
Indicates the label of the identifier when installing the product using the Tivoli desktop or Tivoli Software Installation Service.

valid_for
Indicates for which type of installation the identifier is valid. The value for this field is either blank or one of the following:

product
Indicates that the item is valid for product installations.

patch Indicates that the item is valid for patch installations.

server Indicates that the item is valid for installation on the Tivoli server.

client Indicates that the item is valid for installations on managed nodes.

both Indicates that the item is valid for installations on both the Tivoli server and managed nodes.

If no value is specified, the item is valid for all installations.

subdir Indicates the subdirectory where the contents of a file package are written on the file system.

value Indicates the values that are assigned to names. This option can set a default value for the name by specifying `default=` or a set of values by specifying `choice=`.

env Indicates variables that are passed to a before script or after script that is used during this installation. When passing multiple variables, each ends with a semicolon (;).

chain_id
Identifies the name of another file package that will be distributed immediately after this file package is distributed.

Examples of id lines

1. The following is an example of an id line in the 42_INV_F.IND file:

```
InventoryServer:id:CAT:Message Catalogs:both:: \
  default=/usr/local/Tivoli/msg_cat:ThisDir=@CAT@;ThisHost=@HostName@; \
  ThisPkg=CAT;:
```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryServer`
   - The name of the identifier, as used in commands, is `CAT`
   - The label of the identifier, as shown in a graphical interface, is `Message Catalogs`
   - The associated files are for installation to both the Tivoli server and managed nodes
   - There is no subdirectory required for writing the contents to the file system
   - The file system location for the message is `/usr/local/Tivoli/msg_cat`
   - There are three variables passed to the scripts: `ThisDir` as `@CAT@`, `ThisHost` as `@HostName@`, and `ThisPkg` as `CAT`

2. The following is another example of an id line in the 42_INV_F.IND file:

```
InventoryServer:id:@RDBMS_Vendor@:RIM Host:product:: \
   choices=DB2,Sybase,Oracle,MS_SQL,Informix::
```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryServer`
   - The name of the identifier, as used in commands, is `@RDBMS_Vendor@`
   - The label of the identifier, as shown in a graphical interface, is `RIM Host`
   - The association is for a product installation
   - There is no subdirectory required for writing the contents to the file system
   - The available choices for this keyword as `DB2`, `Sybase`, `Oracle`, `MS_SQL`, or `Informix`
   - There no variables passed to the scripts

3. The following is another example of an id line in the 42_INV_F.IND file:

```
InventoryServer:id:@INV_DATA_HOST@:Data Handler Host:product:: \
   default=TMR_Server::
```

In this example, the following is known:

- The registered product tag for this component is `InventoryServer`
- The name of the identifier, as used in commands, is `@INV_DATA_HOST@`
- The label of the identifier, as shown in a graphical interface, is `Data Handler Host`
- The association is for a product installation
- There is no subdirectory required for writing the contents to the file system
- The product association is `TMR_Server`
- There no variables passed to the scripts

Syntax for the fp lines

Each fp line describes a specific file package associated with the installation image. There can be multiple fp lines in a single index file.

The syntax for the fp line is as follows:

```
tag:fp:package_name:interpreter_type::size:number
```

Explanation of fp syntax

package_name
Indicates the name of the package of which the file package is a part. Examples of package names are BIN, LIB, DB, MAN, INC, and so forth.

interpreter_type
Indicates which interpreter type is associated with the file package. For file packages that are not specific to a particular interpreter type, `generic` is used.

size
Indicates the size of the file package in kilobytes. This number is used to determine whether enough disk space is available to install this file package.

number
Indicates the file package number. This option points to the FILE*number*.PKT file on the product CD.

Note: There are two colons (::) between *interpreter_type* and *size*.

Examples of fp lines

1. The following is an example of a fp line in the 42_INV_F.IND file:

   ```
   InventoryServer:fp:CAT:generic::53:1
   ```

   In this example, the following is known:

   - The registered product tag for this component is `InventoryServer`
   - The identifier associated with the file package is `CAT`
   - This file package is installed on all interpreter types, or `generic`
   - The file package size in kilobytes is `53`
   - The file package number is `1`
2. The following is another example of a series of fp lines in the 42_INV_F.IND file:

```
InventoryServer:fp:LIB:solaris2::32:2
InventoryServer:fp:LIB:hpux10::48:3
InventoryServer:fp:LIB:aix4-r1::63:4
InventoryServer:fp:LIB:win32-ix86::0:5
InventoryServer:fp:LIB:linux-ix86::30:6
InventoryServer:fp:LIB:linux-s390::47:7
```

In this example, the following is known:

- The registered product tag for this component is `InventoryServer`
- The identifier associated with the file packages are for library files, or `LIB`
- There are separate file packages for each of the supported interpreter types: `solaris2` (Solaris), `hpux10` (HP-UX), `aix4-r1` (AIX), `win32-ix86` (Windows), `linux-ix86` (Linux for Intel), and `linux-s390` (Linux for z/OS®)
- The size of each associated file package
- The package number of each file package

Syntax for the gui lines

Each gui line specifies a graphical user interface (GUI) item that is presented as part of the installation windows. There can be multiple gui lines in a single index file.

The syntax for the gui line is as follows:

tag:gui:*item_type*:*variable_name*:*valid_for*:*default_value*

Explanation of gui syntax

item_type
Indicates the type of item to be displayed in an installation window. Values of the same type are grouped together when displayed. The value is one of the following:

G General

L Location

S Switch

variable_name
Indicates the name of a variable that is associated with the GUI item. The variable is defined by a corresponding id line in the index file.

valid_for
Indicates for which type of installation this GUI item is valid. The value is either not specified or one of the following:

server Indicates that the item is valid for installation on the Tivoli server.

client Indicates that the item is valid for installations on managed nodes.

product
Indicates that the item is valid for product installations.

patch Indicates that the item is valid for patch installations.

If no value is specified, the item is valid for all installations.

default_value
Indicates the default value for the GUI item. For switches, the value is either `0` or `1`, where `1` indicates that the switch is set.

Examples of gui lines

1. The following is an example of a gui line in the 42_INV_F.IND file:

   ```
   InventoryServer:gui:G:INV_DATAHOST:product:
   ```

   In this example, the following is known:
   - The registered product tag for this component is `InventoryServer`
   - The item type is general, or `G`
   - The variable for the item is `INV_DATAHOST`
   - The association is for a product installation
2. The following is another example of a gui line in the 42_INV_F.IND file:

   ```
   InventoryServer:gui:G:RDBMS_Vendor:product:
   ```

   In this example, the following is known:
   - The registered product tag for this component is `InventoryServer`
   - The item type is general
   - The variable for the item is `RDBMS_Vendor`
   - The association is for a product installation

Syntax for the patch_id lines

Each patch_id line specifies other products that are included in this product image. There can be multiple patch_id lines in a single index file. Because installation images can overlap, it is common for a newer installation image to contain previous images.

The syntax for the patch_id line is as follows:

```
tag:patch_id:product_tag
```

Explanation of patch_id syntax

product_tag
Indicates the registered product tag for a product that is contained in this installation image.

Examples of patch_id lines

1. The following is an example of a patch_id line in the 42_INV.IND file:

   ```
   InventoryServer_42:patch_for:InventoryServer
   ```

   In this example, the following is known:
   - The registered product tag for this component is `InventoryServer_42`
   - The registered product for the patch is `InventoryServer`
2. The following is an example of a patch_id line in the 42_GW.IND file:

   ```
   InventoryGateway_42:patch_for:InventoryGateway
   ```

   In this example, the following is known:
   - The registered product tag for this component is `InventoryGateway_42`
   - The registered product for the patch is `InvnetoryGateway`

Syntax for the depends lines

Each depends line indicates a product dependency. There can be multiple depends lines in a single index file. If a product cannot be installed unless one or more products is already installed, the depends lines are used to enforce these dependencies.

The syntax for the depends line is as follows:

```
tag:depends:product_tag
```

Explanation of depends syntax

product_tag
Indicates the registered product tag for a product that must be installed before this installation image can be installed.

Examples of depends lines

1. The following is an example of a series depends lines in the 42_INV_F.IND file:
   ```
   InventoryServer:depends:TMF_4.1
   InventoryServer:depends:TMF_DDC_4.1
   InventoryServer:depends:JCF41
   InventoryServer:depends:JRE130
   ```
   In this example, the following is known:
   - The registered product tag for this component is `InventoryServer`
   - The product prerequisites for the installation are `TMF_4.1`, `TMF_DDC_4.1`, `JCF41`, and `JRE130`
2. The following is an example of a patch_id line in the 42_GW_FR.IND file:
   ```
   InventoryGateway:depends:TMF_4.1
   ```
   In this example, the following is known:
   - The registered product tag for this component is `InventoryGateway`
   - The product prerequisite for the installation is `TMF_4.1`

Syntax for the lcf_allow line

The lcf_allow line indicates that the product can be installed on endpoints. There is only one lcf_allow line in an index file.

The syntax for the lcf_allow line is as follows:

```
tag:lcf_allow
```

Note: No syntax explanation or examples are provided for this line.

Chapter 25. Operating system considerations

This chapter contains information specific to the following operating systems when used in a Tivoli environment:

- Windows operating systems
- OS/2 operating systems
- NetWare operating systems
- OS/400 operating systems

Using Windows systems

This section contains information specific to using Windows operating systems in a Tivoli environment. The following terms and concepts are relevant to the information contained in this section:

Tivoli Remote Execution Service
Provides the ability to remotely connect to a Windows operating system and run commands. This service is used during installation, interconnection, and remote startup. It is installed using InstallShield or remotely by another Windows operating system that is already running this service. This service is equivalent to the UNIX **rexec** or **rsh** utility.

Note: This service is also known as Tivoli Remote Installation Protocol (TRIP).

Tivoli Authentication Package
Provides the ability to run methods in the context of a given user on a Windows operating system. It is similar to the UNIX **su** command or **setuid** function. It is installed when the Tivoli server, managed node, or endpoint is installed and is loaded by the Local Security Authority (LSA) subsystem on Windows operating systems. The system has to be rebooted for Tivoli Authentication Package to be started.

Tivoli Remote Access Account
Provides the ability to access remote Windows file systems such as mapped drives on another system. The account and password are set on a per system basis, although they can be set to a domain account and password. This function resides on top of Tivoli Authentication Package.

An example of using Tivoli remote access account would be a file package after script that copies the installation log to a remote share. The account could be set to DOMAIN\tivuser and this account would allow write access to this remote share.

This section contains the following information:

- Accessing remote file systems
- Using Tivoli Remote Execution Service
- Accounts and user login maps
- Accounts created during installation
- Considerations for domain controllers

Accessing remote systems

To manage remote resources, Tivoli Enterprise software must be able to access remote systems and run processes on these systems. In a Windows environment, the Tivoli remote access account provides this ability.

Tivoli Authentication Package

Tivoli Authentication Package with the Tivoli remote access account set to a valid user ID and password enables Tivoli Enterprise products to access remote systems in the context of the Tivoli remote access account. Tivoli Authentication Package is created for managed nodes or endpoints on Windows operating systems. Tivoli Authentication Package is configured by using the **wsettap** command (on managed nodes) or by using the **wlcftap** command (on endpoints).

The **wsettap** and **wlcftap** commands set the properties of Tivoli Authentication Package in a dynamically linked library, TivoliAP.dll, on managed nodes and endpoints respectively. Tivoli Authentication Package also enables Windows operating systems to run **setuid** methods; that is, to run a method in the context of a user associated with the method.

Note: To run a method in the security context of the object dispatcher or endpoint service on Windows operating systems that are based on the Windows NT kernel (for example Windows 2000, Windows XP, and Windows Server 2003), run the method under the user name tmesec. When you run a method under this user name, the object dispatcher or endpoint service uses its own token handle for impersonation.

The TivoliAP.dll optionally uses the Tivoli remote access account, which specifies the user name and password that Tivoli Enterprise products use to access remote systems. This Tivoli remote access account is created during installation and is registered with the LSA on the managed node or endpoint.

When activating Tivoli Authentication Package for the first time or when changing its properties, you must reboot the system for changes to take effect. For managed nodes, reboot after installing the managed node. For endpoints, reboot after installing the endpoint. If the managed node and endpoint are on the same system, they share Tivoli Authentication Package and you need to reboot after installing the first of the two resources.

In a Windows 2000 domain environment that uses Active Directory, Tivoli Authentication Package uses the Tivoli remote access account to connect to Active Directory rather than using an anonymous login. To use this feature, you must set the Tivoli remote access account to a valid domain-qualified user ID and password.

For detailed information about the **wsettap** and **wlcftap** commands, refer to *Tivoli Management Framework Reference Manual*.

Tivoli Remote Execution Service

The Tivoli remote access account is a Windows system account that is specified locally on the machine and is used when a Tivoli process must access a remote resource for that machine. By default, no Tivoli process requires that the Tivoli remote access account be defined. However, you might need remote access to enable a task to run a domain command or to have an after script write a file to a remote share.

If you are going to use remote access for managed nodes or endpoints, you should grant only the limited user rights policies required for that machine. This account should *not* be a member of the Administrator group and should *not* be granted the **Act as part of the operating system** user rights policy. Using an account with limited rights for the Tivoli remote access account is prudent because all methods and tasks that run on the system where the Tivoli remote access account is set have only the access that the Windows account grants to remote resources.

If a password changes, you must update the Tivoli remote access account on all machines where the account is set. For endpoints, the lcfd service will fail to restart. For managed nodes, the access to the remote resources will fail, but the oserv service will start.

An example would be a company that creates a domain account MASTER\tivuser and grants this account write access to a remote share called \\SERVER1\tivfiles. MASTER\tivuser has only write access, but a member of the Administrator group would have full access to the remote share. You should set the password for MASTER\tivuser to not expire, and only the Tivoli administrator responsible for installation should know this password.

For more information about creating Windows system accounts, refer to your operating system documentation.

Using Tivoli Remote Execution Service

Tivoli Remote Execution Service provides support for remote operations on Windows managed nodes from either Tivoli servers or other managed nodes. This service enables you to perform the following functions:

- Remotely install managed nodes on Windows operating systems
- Using Tivoli Software Installation Service, remotely install endpoints on Windows operating systems
- Remotely connect a Windows Tivoli server to another Windows or UNIX Tivoli server
- Run the **odadmin** command with the **start** option to start Windows managed nodes from another managed node

After installing Tivoli Remote Execution Service, you can start and stop this service from the command line using the short name trip or start and stop this service from the Windows Services window.

To start Tivoli Remote Execution Service from the command line, use the following command:

```
net start trip
```

To stop Tivoli Remote Execution Service from the command line, use the following command:

```
net stop trip
```

Note: The **start** option of the **odadmin** command requires that Tivoli Remote Execution Service run on port 512. If another service uses port 512, Tivoli Remote Execution Service cannot operate.

Manually installing Tivoli Remote Execution Service

If your Tivoli server is a Windows machine, you do not need to install Tivoli Remote Execution Service on any system that will be a managed node.

If your Tivoli server is not a Windows machine, you need to manually install Tivoli Remote Execution Service on at least one system before it can become a managed node.

Use the following procedure to install this service manually:

1. Insert the Tivoli Management Framework 2 of 2 CD into the CD-ROM drive.
2. From the **Start** menu, select **Run** to display the Run window.
3. Type `x:\trip\setup` in the **Open** field, where *x* is the CD-ROM drive.
4. Click **OK** to start the installation process. The Welcome window is displayed.
5. Click **Next**. The Choose Destination Location window is displayed.
6. Click **Next** to accept the default installation directory.

   Note: Do not specify a different directory.
7. Follow the instructions on the remaining windows to complete the installation.

Windows repeaters

The first Windows managed node in the Tivoli region that has Tivoli Remote Execution Service installed becomes the current Windows repeater. (The Windows repeater used by Tivoli Remote Execution Service is distinct from a Tivoli repeater.) This repeater distributes Tivoli Remote Execution Service when a Windows managed node is created. Although you can have multiple repeaters in a region, a single repeater can support an unlimited number of Windows managed nodes.

Note: Windows repeaters can distribute across subnets, but not to untrusted domains.

After Tivoli Remote Execution Service is installed, the installation process uses this service to communicate with that system. After installing the managed node, do not delete the repeater. You will need it when you add new Windows managed nodes to the Tivoli region.

Note: The current repeater is used by Tivoli Software Installation Service to install endpoints on Windows operating systems.

Deploying Windows repeaters: To support multiple Windows managed nodes, a repeater must meet the following criteria:

- Be able to connect to other Windows operating systems in the trusted domain
- Be able to write to other Windows operating systems in the trusted domain

Before designating a Windows repeater, determine which, if any, Windows operating system in your proposed Tivoli region can be this repeater.

If your Tivoli server is a UNIX system or if your Windows Tivoli server is not configured to connect and write to all other Windows operating systems, you need to install Tivoli Remote Execution Service manually on one or more systems.

Designating a Windows repeater: After Tivoli Remote Execution Service is installed, you can designate that machine as the current Windows repeater. If the designated repeater becomes unavailable or cannot connect to or write to some Windows systems, you can designate another system as the current repeater.

Use the commands in the following procedure to designate a new Windows repeater:

1. Find the current repeater:

```
wlookup CurrentNtRepeat
```

Note: The CurrentNtRepeat object is the repeater for both Windows machines.

2. Find other available repeaters:

```
wlookup -ar NtRepeat
```

3. Remove the old repeater designation:

```
wregister -u CurrentNtRepeat
```

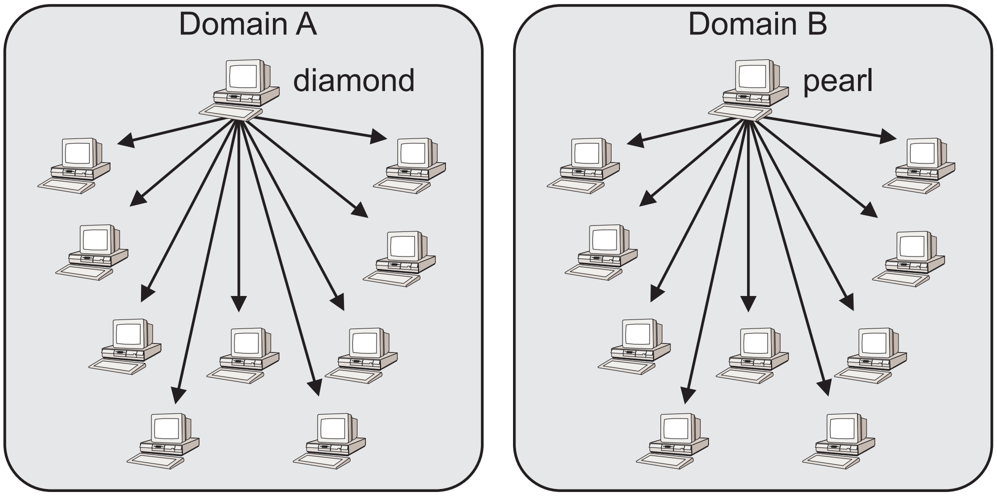
4. Designate the new repeater:

```
wregister CurrentNtRepeat object_reference
```

where *object_reference* is one of the available repeaters.

For detailed information about **wlookup** and **wregister**, refer to *Tivoli Management Framework Reference Manual*.

Repeater scenario: In the following scenario, the Tivoli server is a UNIX system, and the Tivoli region contains two untrusted domains. All Windows machines in each domain can connect and write to each other, but none can connect across the domain boundaries. This situation is shown in the following figure.



To efficiently install Tivoli Management Framework in this situation, one system in the domain must become a Windows repeater. The Tivoli administrator could perform the following procedure:

1. Install Tivoli Remote Execution Service on diamond in Domain A. For details, refer to “Manually installing Tivoli Remote Execution Service” on page 341.
2. Install a managed node on diamond.

   Note: Because Tivoli Remote Execution Service is already installed, diamond is registered as the current repeater.
3. Install managed nodes on the remaining systems in Domain A.

   Note: The installation process uses diamond to remotely install Tivoli Remote Execution Service to these systems.
4. Install Tivoli Remote Execution Service on pearl in Domain B. For details, refer to “Manually installing Tivoli Remote Execution Service” on page 341.
5. Install a managed node on pearl.
6. Change the designated repeater to pearl in Domain B. The following example assumes that the object identifier for the repeater object for pearl is 1266616029.3.11:

```
wregister -u CurrentNtRepeat
wregister CurrentNtRepeat 1266616029.3.11
```

For complete details, refer to "Designating a Windows repeater" on page 342.

7. Install managed nodes on the remaining systems in Domain B.

If the repeater cannot connect to or cannot write to any Windows machine during the Tivoli Management Framework installation, you must install Tivoli Remote Execution Service manually. For details, refer to "Manually installing Tivoli Remote Execution Service" on page 341.

Accounts and user login maps

Because Tivoli Enterprise software spans a heterogeneous environment, you can map a special ID, referred to as a *user login map*, to an operating system-specific user account. On Windows operating systems, the user login map can contain a reference to w32-ix86, which is the identification within Tivoli Enterprise software for Windows operating systems.

The user login map root_user is a preconfigured user login map that resolves on Windows operating systems to BuiltinNTAdministrator. This map is used for various Windows processes. When a Tivoli service needs to resolve a method that is to run as root_user, it runs the method as the user assigned to root_user. Therefore, the root_user user login map must map correctly to a local or domain user account.

Notes:

1. $root_user must be a member of the Administrators group and the Tivoli_Admin_Privileges group.
2. Some Tivoli methods run as root rather than $root_user. On UNIX and Linux systems, these methods default to the root user. On Windows systems, these methods default to the built-in administrator account. If the built-in administrator account has a restriction such as an expired password, the method will fail.

To create or modify user login maps, use the **widmap** command. For example, if you have a user named fhackerm and want to add him to the **root_user** user login map, enter the following command:

```
widmap add_entry root_user fhackerm w32-ix86
```

If you have Windows systems with different administrator accounts, you can use the **widmap** command to map these accounts to the built-in administrator account instead of an account specified by name. You do this by mapping root_user to BuiltinNTAdministrator as shown in the following example:

```
widmap rm_entry root_user w32-ix86
widmap add_entry root_user w32-ix86 BuiltinNTAdministrator
```

Note: The BuiltinNTAdministrator account is not an actual Windows account name.

You can also use a domain account in one of the following ways:

- If the *machine*\Administrator account is not renamed, do not modify the root_user user login map unless you want to run the Tivoli Enterprise privileged programs as another local or domain account.
- If the *machine*\Administrator account is renamed or the design of the Tivoli region dictates using a domain account for privileged accounts, ensure that *machine*\Administrator renaming is consistent on all managed nodes and

endpoints or a local Administrator account is created on each managed node or endpoint. You can name the account with a name other than Administrator, but it must be consistent on all managed nodes and endpoints. The root_user user login map must be updated to reflect the new name.

Note: The $root_group map or another group ID must exist for an administrator. This user login map is not used when a process starts. However, it is important that $root_group map has a group listed for w32-ix86. This map does not need to be a privileged group.

For information about the **widmap** command, refer to the *Tivoli Management Framework Reference Manual*.

Accounts created during installation

When you install Tivoli Management Framework, the tmersrvd user account and the Tivoli_Admin_Privileges group account are created. These accounts are created locally in the Security Accounts Manager (SAM) database on the Windows system and are configured the same for managed nodes and endpoints.

The tmersrvd user account

The tmersrvd account is an unprivileged account. A password is randomly generated at installation. The account can be disabled without affecting Tivoli Management Framework. Many Tivoli methods run in the context of tmersrvd. You can change the password.

You can refer to the tmersrvd account configuration for minimum requirements when creating a custom account. Refer to the Local Security Policy object in Adminstrative Tools window of the Windows Control panel for additional information about defining security policies for custom accounts.

The tmersrvd account needs the following user rights:

Bypass Traverse Checking

The tmersrvd account does not get assigned this user right directly. When a Windows system is installed, this right is assigned to the special group Everyone. This right allows a user to traverse a directory tree even if the user has no other rights to access this directory. If security policies in your enterprise disallow Bypass Traverse Checking, add this right to the tmersrvd account.

Note: For non-U.S. versions of a Windows system, the Everyone group account is referred to by its local language equivalent.

Log on Locally

This right is assigned to the tmersrvd account during installation of Tivoli Management Framework.

The Tivoli_Admin_Privileges group account

The Tivoli_Admin_Privileges group account is assigned by default to the built-in administrator or $root_user map unless the Tivoli server is on a Windows system. In this case, the account used to install the Tivoli server is assigned to this group.

The Tivoli_Admin_Privileges account needs the following advanced user rights:

- Act as Part of the Operating System
- Increase Quotas
- Replace a process level token

The Act as Part of the Operating System right is required when running the **wsettap** and **wlcftap** commands without options. These commands communicate with the LSA to retrieve the current configuration of Tivoli Authentication Package. Other operations of the **wsettap** and **wlcftap** commands communicate with the registry and not the LSA, so they do not require special rights, except they can only be run by a member of the Administrators group.

The Increase Quotas and Replace a process level token rights are required to start a process as a different user. Examples of these processes are the **run_task** and **sentry_engine** methods.

Note: If you change the value of $root_user, you need to ensure that this account is a member of the Tivoli_Admin_Privileges group. If the account is not part of this group, you will receive the following error:

```
tap_call_init failed, error 38
```

Considerations for domain controllers

This section discusses issues particular to primary and backup domain controllers.

Authentication to the primary domain controller

When using Tivoli Authentication Package, it requests domain user authentication from the Primary Domain Controller (PDC) and bypasses any local backup domain controllers (BDCs). This can flood the primary domain controller with authentication requests if the domain account is used for the $root_user user login map or for applications, such as Tivoli Distributed Monitoring, that can run a large number of processes in a short span of time.

Many Windows environments use several domains to manage the environment. Commonly, a master domain with resource domains that are two-way trusted with the master domain are used. If requirements demand that you use a domain account for your Tivoli environment and the Windows domains, you could create an account in each domain with the same names. For example, you could create the tivuser account in multiple domains:

- MASTER\tivuser
- US\tivuser
- GERMANY\tivuser
- JAPAN\tivuser

The $root_user map would map w32-ix86 to tivuser. When a managed node or endpoint runs a privileged process, Tivoli Authentication Package ensures that the map references tivuser. It first looks in the SAM database. If Tivoli Authentication Package does not find the account there, it queries the primary domain controller for the system. Therefore, a system in the JAPAN domain will only be authenticated to the JAPAN domain controller rather than the MASTER domain controller.

Using the same model in the JAPAN domain, assume a given task must run as MASTER\tivuser. You would specify MASTER\tivuser in the user ID (UID) field of the task or create a user login map that resolves to this account. If MASTER is not part of the specified account, the same system in the JAPAN domain gets the system ID (SID) for the JAPAN\tivuser account rather than the MASTER\tivuser account.

Accounts created on the domain controllers

If using both primary and backup domain controllers within a Tivoli environment, it is recommended that you install the managed node or endpoint on the primary domain controller first and then synchronize the backup domain controllers to allow the newly created accounts to propagate. If an installation is attempted first on a backup domain controller, the installation fails because the accounts have not been updated from the primary domain controller. Wait 15 minutes for the domain servers to resynchronize and attempt the installation again.

For account management, primary and backup domain controller accounts are considered a domain account. When Tivoli Management Framework runs on either a primary domain controller or backup domain controller, the authentication will still take place on the local SAM database with no impact to the network or other domain controllers. Additionally, Tivoli Enterprise software does not force partial or full synchronization with a domain.

Using OS/2 systems

This section contains information specific to installing preliminary security software prior to installing Tivoli Management Framework on an IBM OS/2 operating system. This information applies to managed nodes and gateways.

Security support is installed independently at the OS/2 machine by performing the following procedures:

- Enabling Security Enabling Services (SES)
- Installing system fixes and upgrades
- Upgrading the rexecd utility
- Ensuring reverse name resolution
- Installing Tivoli Management Platform Security
- Making security modifications
- Adding new users
- Blocking unauthorized TCP/IP access
- Configuring TCP/IP to support the OS/2 gateway

Because OS/2 is a single user system, you need to install SES to add UNIX-like security. After installing the Tivoli management platform security for OS/2 function, all installations that affect controlled files (such as config.sys, secure.sys, or the Tivoli directories) must be done from the root user, the default administrator. All parts of the security enabling process are performed on the OS/2 machine except mutual verification of name resolution, which must be performed on the OS/2 machine and Tivoli server.

Enabling Security Enabling Services

Before you can install necessary OS/2 system fixes (fixpaks), you must enable SES. To enable SES, perform the following steps:

1. From the OS/2 desktop, double-click the OS/2 System icon. The OS/2 System window is displayed.
2. Double-click **System Setup**. The System Setup window is displayed.
3. Double-click **Install/Remove**. The Install/Remove window is displayed.
4. Double-click **Selective Install**. The Selective Install window is displayed.
5. Click **Next**.

6. Click **Next**.
7. Select the **Optional System Components** check box.
8. Click the **More** button beside this option.
9. Select the **Security** check box.
10. Click **OK**.
11. Click **Next** and proceed with the installation.
12. Reboot your system.

Enablement services can be downloaded from IBM Web site. Contact your IBM service provider for information about locating and accessing the appropriate Web site. From the Web site, download the following files:

security.bbs
An upack2-format file of the code with the installation utility included.

warpses.txt
The file that explains how to install security.bbs.

Installing system fixes and upgrades

After you have enabled security, you must install system fixes and upgrades.

Note: For Warp 4.0, the minimum fixpak is XR_M007. For Warp Server Advanced, the minimum fixpak is XR_W037.

If you use Remote Software Upgrade (RSU) technology, you can access these fixpaks from the same Web site. If you want, you can install RSU technology if you do not already have it.

Contact your IBM service provider for information about locating and accessing the appropriate Web site.

Upgrading the rexecd utility

If you are using TCP/IP, Version 4.0 or 4.1, you need to upgrade the **rexecd** utility to make this utility compatible with Tivoli Remote Execution Service on Windows operating systems and **rexecd** on UNIX operating systems.

Upgrading rexecd for TCP/IP Version 4.0

If you have TCP/IP Version 4.0 installed, perform the following steps:

1. Download both rexecd fix IC19029 and the associated README file from IC19029 for TCP/IP, Version 4.0.
2. Follow the instructions in the README file to ensure that you add the **–q** option on the invocation of **rexecd**:
   a. Issue the **tcpcfg** command from an OS/2 command line.
   b. When the Settings window is displayed, click the **Autostart** tab.
   c. Modify the **rexecd** settings to add the **–q** option.

Upgrading rexecd for TCP/IP Version 4.1

If you are using TCP/IP, Version 4.1, perform the following steps:

1. Download both rexecd fix IC19029 and the associated README file from IC19029 for TCP/IP, Version 4.0.
2. Follow the instructions in the README file to ensure that you add the **–q** option on the invocation of **rexecd**.
   a. Double-click the TCP/IP Configuration icon.

b. When the Settings window appears, click the **Autostart** tab.

c. Modify the **rexecd** settings to add the **–q** option.

Verifying the upgrades

When these upgrades are complete, run the **syslevel** command to verify that you have the correct corrective service delivery (CSD) levels installed. The output of the **syslevel** command shows CSD level XR_M007 (for Warp 4.0) or XR_W037 (for all other versions of the operating system).

Ensuring reverse name resolution

You need to ensure that the Tivoli server and the gateway have reverse name resolution. To do this, you need to use the **nslookup** command to ensure that the Tivoli server has the correct name for the OS/2 machine and that the OS/2 machine has the correct name for the Tivoli server:

1. On the Tivoli server, enter the following command and record the host name and host ID that it displays:
   ```
   nslookup gateway
   ```
2. On the OS/2 machine, enter the following command and record the host name and host ID that it displays:
   ```
   nslookup Tivoli_server
   ```

The results are the host name and IP address of the queried system. If you get unexpected results, your Domain Name System (DNS) resolution is not operational.

You can also see the host name and IP address for the OS/2 machine by running the **hostname** and **hostid** commands, respectively. If you get unexpected results, check the hosts file on the Tivoli server to see whether these entries are in it. Ensure that the names in the hosts file match, including case.

Installing Tivoli Management Platform Security

Installable Security Subsystems operate in an environment provided by a component of the OS/2 operating system called Security Enabling Services (SES). The architecture of SES allows only one Installable Security Subsystem at a time to be active on an OS/2 machine. Because of this, you must use the Installable Security Subsystem supplied with Tivoli Management Framework.

Before you can install a gateway on an OS/2 system, you must install the Tivoli management platform security for OS/2 function. This OS/2 function is packaged on the Tivoli Management Framework 1 of 2 CD. To install the security application, perform the following steps:

1. With the Tivoli Management Framework CD in the CD-ROM drive, change to the \OS2SEC subdirectory.
2. Enter `install`.
3. The information window is displayed. After you read this information, click **Continue**.
4. When the Install window is displayed, click **OK**.
5. An **Install Progress** bar and a message window appear. When installation is complete, the Installation and Maintenance window prompts you to reboot because of changes to your config.sys file. Click **OK**.
6. Reboot your system. Log on with the default user ID `root` and the default password `root`.

7. After reboot, when the Tivoli management platform security for OS/2 screen appears, press **Ctrl+Alt+Del** to change your password. Enter your new password twice in the Change User Password window and click **OK**.

Making security modifications

Because the Tivoli management platform security for OS/2 function blocks all known opportunities for unintended modification of your OS/2 system, you might need to disable parts of the security system until after you install the OS/2 gateway. Until installation is complete, you might want to edit your config.sys file to change the trusted path statement to read:

```
set trustedpath=no
set backgroundbitmap=x:\os2\security\ses\tivoli2.bmp
```

where *x* is the boot drive.

This allows you to press the **Enter** key to open the security log window instead of **Ctrl+Alt+Del**. It also allows you take a system dump to diskette using **Ctrl+Alt+NumLock**.

Adding new users

After the Tivoli management platform security for OS/2 function is installed, you can add new users. root is your default administrator. You can define additional administrators or users as needed.

Note: You do not need to add new users for Tivoli functions.

To add new users, perform the following steps:

1. From the OS/2 desktop, double-click the Tivoli Management Platform (TMP) Security icon. The TMP Security window is opened.
2. From this window, double-click the Local Work Station icon to open its properties window.
3. Click the **User** tab and the click the **Create user** button. This opens a Create a User Account window.
4. Type the user ID and a description of the user you are adding. A description must be entered or the record cannot be created.
5. Select a user type, either user or administrator. To define an administrator, select **Administrator**.
6. Allow the user to log on by selecting **Logon-Allowed**.
7. Enter a password for the user. Enter it again for verification.
8. Select the **do not lock** check box or set a time value as the **Inactivity Time-Out** period.
9. When this information is complete, click **OK**.
10. Close the windows you no longer need.

Blocking unauthorized TCP/IP access

Although the Tivoli management platform security for OS/2 function is robust enough to block access from users entering the system using the **rexecd** utility, it cannot block access from TCP/IP file transfer utilities. This includes:

ftp Prior to TCP/IP for OS/2, Version 4.1, user ID and password security for file transfer protocol (FTP) were controlled within FTP using the trusers file. The Tivoli management platform security for OS/2 function cannot control FTP access.

telnet TCP/IP for OS/2, Version 4.1, provides improved security support. All releases of TCP/IP for OS/2 prior to Version 4.1 include the telnet password in the config.sys file. You control telnet access using this password.

rsh Remote shell (RSH) controls access using the rhosts file.

The system administrator is responsible for managing these applications.

Configuring TCP/IP to support the OS/2 gateway

You need to configure your TCP/IP system to automatically start the inetd, rexecd, sendmail, and portmap services. The rexecd service processes binary data and might cause your system to beep and your installation could fail. This can be disabled in the following ways:

- Starting the rexecd service in a foreground session (minimized) and passing it the **–q** option.
- Starting the rexecd service under the super daemon (inetd) and adding the following line to the config.sys file:

  ```
  set quietmode=yes
  ```

Using NetWare systems

This section contains information specific to installing endpoints and gateways on NetWare systems. A NetWare gateway running on a NetWare server can manage endpoint connections in TCP/IP or IPX/SPX protocol.

NetWare considerations

The endpoint authenticates methods on a NetWare server by using the directory services. When you install a gateway or endpoint on a NetWare server, you must specify the organizational context in which the installation application creates the tmersrvd account (gateway). These accounts are very limited (similar to the nobody account used by the Tivoli object dispatcher). These accounts are used to authorize unprivileged methods. If a method is unprivileged, it cannot write, remove, copy, move, or affect any system function.

NetWare accounts

Tivoli Management Framework requires special accounts to manage NetWare gateways and endpoints. The following accounts are needed when you use Novell Directory Service (NDS):

- For endpoints, Admin (for root) and lcfrsrvd, (for nobody)
- For gateways, Admin (for root) and tmersrvd (for nobody)

If you use bindery emulation instead of NDS, the supervisor account is needed instead of the Admin account.

These accounts are created during the installation when it is done using Novell Requester. In bindery, the accounts are created on the local machine. In NDS, you choose whether to have a single lcfrsrvd or tmersrvd account in the NDS tree where the machine is located or to have one in each NDS subcontext.

When you install a NetWare gateway or endpoint using the InstallShield image, you are prompted to provide the NDS context in which the NetWare server works. This will be the context in which the account will be created or, if already present, updated with the information on the new gateway or endpoint. If you want to have a unique user in the NDS tree, you will have to provide the name of the higher level context. If you want to have a user in the subcontext, you have to specify the name of the subcontext where the machine is located.

For example, you have an NDS tree called `IBM` and two NetWare systems named Server1 and Server2. `Server1` is located in NDS context `USA` and `Server2` is located in subcontext `Texas.USA`. You want to install an endpoint on Server2. An endpoint was already installed on Server1 and its installation created the lcfrsrvd account in the context `USA`. In this situation, you can update this account or create a new account in the subcontext (`Texas.USA`). In the first case, you need to specify `USA` as the context. Otherwise, you need to specify `Texas.USA` as the context.

If the Windows system does not have Novell Requester, the accounts are not created. To create the accounts, use the **addadmin** utility provided by Tivoli Management Framework.

Note: Never use the Novell **NWADMIN** utility to create the accounts.

To create a Tivoli unprivileged account for use with Tivoli Management Framework, perform the following steps:

1. Load the appropriate library:
   - For gateways, enter the following command:
     ```
     load SYS:tivoli\bin\nwr-ix86\bin\libnds5 NDS_context
     ```
   - For endpoints, enter the following command:
     ```
     load SYS:tivoli\lcf\bin\nw4\mrt\lcfutil5 NDS_context
     ```
2. Load the **ADDADMIN.NLM** module:
   - For gateways, enter the following command:
     ```
     load SYS:tivoli\bin\nwr-ix86\tmf\lcf\addadmin
     ```
   - For endpoints, enter the following command:
     ```
     load SYS:tivoli\lcf\bin\mrt\addadmin
     ```
3. Use the **addadmin** command to create the accounts in your context.

With this utility, you can create the account in the subcontext or update the one in the tree. When logging in from a subcontext, log in to the tree as `.Admin.`*`context`*, where *context* is the context of the Admin account. When creating or updating the one in the tree, log in as Admin.

When prompted to specify the account (tmersrvd or lcfrsrvd) and the context where you want to create the account is in a subcontext, use dotted notation. For example, to create the tmersrvd account in subcontext agodina, type `.tmersrvd.agodina` as the account name.

Installing NetWare gateways

You can install the gateway on a NetWare system from any supported Windows system that can access the NetWare server. You must log in to the NetWare system as Admin (NDS) or supervisor (bindery).

Notes:

1. You must install the gateway on the SYS volume.

2. If you are logged in to the tree where the NetWare system is but this tree is not set as primary for the Windows system, the appropriate user accounts cannot be created or updated correctly. The installation gives you an error message saying that it was unable to create the appropriate accounts.

Note:

Installing a gateway on a NetWare system consists of the following steps:

1. Install the NetWare binaries for the gateway.
2. Register the NetWare managed node.
3. Create the gateway.

Installing the NetWare binaries

To create a NetWare gateway, the first step is to run **/PC/NWGW/SETUP.EXE** command from the Tivoli Management Framework 1 of 2 CD. A welcome window is displayed stating that you are creating the NetWare gateway. Follow the instruction displayed in the windows. When prompted for the name, specify the host name of the Tivoli server to which the gateway will connect. When prompted, specify the Tivoli server installation password, if previously defined.

Note: Because this is an InstallShield program, you can use the procedure, Chapter 19, “Creating InstallShield response files,” on page 253, to create a response file to install the image in unattended mode.

When you install the NetWare binaries, the installation adds the following files to the NDS context:

- oserv1st.ncf
- oservrun.ncf

After running **setup.exe** command, the object dispatcher is started.

Note: If you installed the NetWare gateway from a system not running Novell Requester to map the NetWare system, you need to run the following command from the NetWare server console to start the object dispatcher the first time:

```
SYS:tivoli\bin\nwr-ix86\oserv1st -s Install_password
```

The **–s** option specifies the installation password, if defined.

Registering the NetWare managed node

Registering the NetWare managed node consists of the following steps:

1. Starting the object dispatcher
2. Running the registration script

Starting and stopping the object dispatcher: To restart the object dispatcher, run the following command:

```
oservrun
```

To stop the object dispatcher, run the following command:

```
oservend
```

Running the registration script: After starting the object dispatcher on the NetWare system, you need to run the **nw_TMF_Install.sh** script from the Tivoli server to register the managed node with the Tivoli name registry.

```
bash nw_TMF_Install.sh host_name
```

where *host_name* is the TCP/IP host name of the NetWare system.

Creating the NetWare gateway

After registering the NetWare managed node, you can create the gateway using any of the available installation mechanisms. For detailed information, refer to "Creating gateways" on page 207.

Installing endpoints on NetWare systems

You can install the endpoint on a NetWare system from any supported Windows system that can access the NetWare server. For NDS, log in as Admin. For bindery, log in as supervisor.

Notes:

1. You must install the endpoint on the SYS volume.
2. If you are logged in to the tree where the NetWare system is but this tree is not set as primary for the Windows system, the appropriate user accounts cannot be created or updated correctly. The installation gives you an error message saying that it was unable to create the appropriate accounts.

The endpoint must be installed on the SYS volume. The method cache, however, can reside on any volume by setting the **cache_loc** option on the **lcfd** command. (Refer to the **lcfd** command in *Tivoli Management Framework Reference Manual* for the configuration options.) You can override default settings using the Advanced Configuration window of the installation process.

To remotely install to a NetWare system, the Windows system must be running Novell Requester. If remotely installed, you must manually start the endpoint from the local machine. At the NetWare console, enter **lcf** to start the endpoint. The **lcf** command run the LCF.NCF file, which was created during the remote installation. The LCF.NCF file contains the following:

- Load statement for **LCFUTIL5.NLM** and **LCFD.NLM**
- Any configuration changes specified on the Advanced Configuration window

Note: The LCFRUNBASE001 entry and the **–C** option in the LCF.NCF file contain the path to the endpoint run directory. This directory is created during installation and is the working directory for the endpoint.

Endpoints in Novell Directory Services (NDS)

An endpoint on a NetWare server can run in NDS with bindery either on or off. If bindery is set, no adjustments to the endpoint are required after installation. If the server is in bindery is off or if you have switched from bindery modes after installing the endpoint, the following procedure is needed to run the endpoint:

1. If you have an endpoint running, stop the endpoint daemon with the following command:

   ```
   lcfstop
   ```

2. Choose the context in which you want the lcfrsrvd account to reside.
3. If you installed an endpoint in bindery, remove the previously installed lcfrsrvd account.
4. Load the **addadmin** utility with the following commands:

   ```
   LOAD SYS:tivoli\lcf\bin\mrt\nw4\lcfutil5 NDS_context
   LOAD SYS:tivoli\lcf\bin\mrt\addadmin NDS_context
   ```

Use the **addadmin** utility to create the lcfrsrvd account in the context that you choose. For detailed instructions, refer to “NetWare accounts” on page 351.

5. Edit the SYS:system\lcf.sys file.
   a. Add the following line:
      ```
      NWDS_CONTEXT=your_context
      ```
   b. Remove the following line:
      ```
      BINDERY_EMULATION=yes
      ```
6. Enter `lcf` at the console to start the endpoint.

Using OS/400 systems

Tivoli Management Framework, Version 4.3.1, includes support for endpoints on OS/400 systems. You can install the endpoint in any OS/400 system running a supported version of OS/400. Refer to *Tivoli Management Framework Release Notes* for a list of supported operating systems.

Configuring TCP/IP

Before you install the endpoint, you must configure TCP/IP on your OS/400 system. The following is a brief example on how to configure TCP/IP for an OS/400 that is connected in an Ethernet network:

1. Sign on to the OS/400 system using any 5250 emulator or Client Access.
2. Get the line description resource name. Run the following command from the OS/400 command line:
   ```
   DSPHDWRSC *CMN
   ```
   The following is an example of the output of this command:
   ```
   CC01 2617  Operational Combined function IOP
   LIN03 2617  Operational LAN Adapter CMN02
   2617  Operational Ethernet Port
   ```
3. Create a line using the resource name from the previous step. Execute the **CRTLINETH** command as follows:
   ```
   CRTLINETH LIND(TCPLIND) RSRCNAME(CMN02)
   ```
4. Vary on the line, execute the **WRKCFGSTS *LIN** command from the command line, and then select option 1 in the TCPLIND line description to vary this on.
5. Add a TCP/IP interface. Run the following command to add the TCP/IP interface:
   ```
   ADDTCPIFC INTNETADR('xxx.xxx.xxx.xxx') LIND(TCPLIND) SUBNETMASK('255.255.255.0')
   ```
   where '*xxx.xxx.xxx.xxx*' is the TCP/IP address of this OS/400 endpoint.
6. Configure the router for TCP/IP. Run the following command to configure the router:
   ```
   ADDTCPRTE RTEDEST(*DFTROUTE) SUBNETMASK(*NONE) NEXTHOP('yyy.yyy.yyy.yyy')
   ```
   where *yyy.yyy.yyy.yyy* is the TCP/IP address of the network gateway or router through which your OS/400 system is connected.
7. Ensure that the domain name, host name, and name server are set using the **CFGTCP** command and selecting following options:
   ```
   12. Change local domain and host names
   ```
8. Start TCP/IP jobs by executing the **STRTCP** command.

You may need to refer to other IBM books for specific information about a topic. For OS/400 information, you can access the OS/400 Online Publication Web site.

Installing OS/400 endpoints

Installing the endpoint creates the QTMELCF library. When you configure the endpoint for an OS/400 system, three user profiles are created on the endpoint. These profiles are shipped without passwords and are used by the endpoint. The following table describes the generated profiles.

User Profile	Description	Authority needed
QTIVOLI	QLCFD job runs under QTIVOLI. This user profile is required to start the endpoint and owns all OS/400 unique Tivoli objects. This profile must have *UPDATE authority to all the directories in the path to the configuration files.	*JOBCTL *SAVSYS *SPLCTL
QTIVROOT	Used when the management server requires root authority.	*ALLOBJ *AUDIT *IOSYSCFG *JOBCTL *SAVSYS *SECADM *SERVICE *SPLCTL
QTIVUSER	Equivalent to the UNIX nobody user ID.	No special authority

Note: If any of these profiles are deleted or corrupted, you must delete and reinstall the OS/400 endpoint to recreate the user profiles and the objects that belong to them.

Installing internationalization support

You can install the OS/400 endpoint (1TMELCF) to use multiple languages. Although Tivoli software is translated into the following languages, the OS/400 system can be operating in other languages:
- Chinese, Simplified (Code 2989)
- Chinese, Traditional (Code 2987)
- English, uppercase and lowercase (Code 2924)
- English, uppercase DBCS (Code 2938)
- French (Code 2928)
- German (Code 2929)
- Italian (Code 2932)
- Japanese (Code 2962)
- Korean (Code 2986)
- Portuguese, Brazilian (Code 2980)
- Spanish (Code 2931)

The default installation language is English. If you install the OS/400 endpoint code without specifying a language, English (2924) is installed.

When you install the OS/400 endpoint using the installation script (**w4inslcf.pl**), you specify the languages using the **–N** option. For example, to install French (2928) and German (2929), the command line is as follows:

```
w4inslcf.pl -g gateway[+port]-N '2928 2929' endpoint
```

where:

gateway[+*port*]
: The host name or IP address and optionally the port number for the gateway to which the OS/400 endpoint will log in.

endpoint
: The host name or IP address of the OS/400 endpoint.

If you are running the OS/400 system in a primary language other than those supported by Tivoli Management Framework, you must add the appropriate Tivoli language library to your library list to access help information while using the control language (CL) commands. The library format is `QTME`*`xxxx`* (where *xxxx* is the 4-digit language code). For example, the French language library is QTME2928.

Starting and stopping OS/400 endpoints

Tivoli provides the following CL commands to start and stop the endpoint daemon:

STRTMEEPT

Starts the OS/400 endpoint daemon. When installing an OS/400 endpoint, Tivoli Software Installation Service or the **w4instlcf.pl** script starts the endpoint daemon job by executing the **STRTMEEPT** command. You must use this command to manually start the endpoint daemon job. To start the endpoint at initial program load (IPL), you can add the **STRTMEEPT** command in the initialization program (QSYS/QSTRUP).

This command can take the **OPTIONS** option to enable you to specify endpoint startup options. Each entry is a key-value pair. These key-value pairs can be any of those defined for starting endpoints using the **lcfd –D** command. These key-value pairs also correspond to the **w4inslcf.pl –L** command.

For example, if the OS/400 endpoint contains multiple network interface cards (NICs), use the **OPTIONS** option to set the **local_ip_interface** option to indicate which to use. Run the following command on the OS/400 command line to start the OS/400 endpoint using the network interface card whose IP address is 146.84.39.126:

```
STRTMEEP LGNINTRFC(IBMTMP1 8752) EPTNAME('queso') \
  PORT(18752) + THRESHOLD(3) \
  OPTIONS('local_ip_interface=146.84.39.126')
```

ENDTMEEPT

Stops the OS/400 endpoint daemon, performs clean up operations, and disconnects from the gateway.

These commands are native OS/400 commands that you can execute from any OS/400 command line after the endpoint code is installed on the OS/400. You can also launch both commands from a remote machine. Both commands are shipped with authority set to `PUBLIC(*EXCLUDE)`. You must grant privileges to any users who will be using these commands. Additionally, you must authorize these users to access the following associated command processing programs.

Library	Command Processing Program	Authority needed
QTMELCF	QLCFSTAR	*USE
QTMELCF	QLCFEND	*USE

Starting the endpoint after boot

To automatically start the endpoint after an IPL operation, you must modify the **QSYS/QSTRUP** program. Perform the following steps:

1. Retrieve CL source into a member of a source file by executing the following command:

   ```
   RTVCLSRC PGM(QSYS/QSTRUP) SRCFILE(QGPL/QCLSRC)
   ```

2. Start the source entry utility:

   ```
   STRSEU SRCFILE(QGPL/QCLSRC) SRCMBR(QSTRUP) TYPE(CLP)
   ```

3. Add the following lines as the last job to start:

   ```
   QSYS/STRTMEEPT
   MONMSG MSGID(CPF0000)
   ```

4. Save the old program if desired.
5. Compile the program, thus replacing the old **QSYS** program:

   ```
   CRTCLPGM PGM(QSYS/QSTRUP) SRCFILE(QGPL/QCLSRC)
   CHGOBJOWN OBJ(QSYS/QSTRUP) OBJTYPE(*PGM) NEWOWN(QSYS)
   ```

Removing the endpoint from OS/400 systems

To remove the endpoint code from an OS/400 machine, perform the following steps:

1. Stop all Tivoli jobs or any other jobs that were started by Tivoli Enterprise products and are running on the endpoint, and make sure that no job has any locks to any language libraries.

   Notes:

   a. Run the **ENDTMEEPT** command to stop the endpoint process.

   b. Run the **WRKACTJOB** command to find other jobs that are running under any of the Tivoli user accounts. To stop a job, use the **ENDJOB** command or option **4**.

      Endpoint processes can include the QLCFD daemon process as well as any job started by Tivoli Enterprise products. For example, you must stop the monitoring engine and any monitor processes running on the endpoint.

   c. Run the **WRKOBJLCK** command to determine who added any of the Tivoli language libraries to the job library list. For example to determine who added English, enter:

      ```
      WRKOBJLCK OBJ(QSYS/QTME2924) OBJTYPE(*LIB)
      ```

2. Delete the licensed program by entering the following command:

   ```
   DLTLICPGM LICPGM(1TMELCF)
   ```

3. Run the following **WRKOBJOWN** commands on each Tivoli user account to display all objects owned by the specified user:

   ```
   WRKOBJOWN USRPRF(QTIVUSER)
   WRKOBJOWN USRPRF(QTIVROOT)
   WRKOBJOWN USRPRF(QTIVOLI)
   ```

   The **WRKOBJOWN** command lists all objects that belong to a user. To prevent any listed object from being deleted, you *must* change the ownership of that object by selecting option **9** for the object from the **Work with Objects by Owner** display. Repeat this step for each object that you want to save.

4. Delete the user account names by specifying the ***DLT** option, which deletes all objects owned by a user. The **DLTUSRPRF** commands must be entered in the following order:

   ```
   DLTUSRPRF USRPRF(QTIVUSER) OWNOBJOPT(*DLT)
   DLTUSRPRF USRPRF(QTIVROOT) OWNOBJOPT(*DLT)
   DLTUSRPRF USRPRF(QTIVOLI) OWNOBJOPT(*DLT)
   ```

Note: If you delete QTIVOLI account before deleting other user accounts, some objects might not be deleted and you might have problems reinstalling the endpoint.

Appendix A. Installation Commands

This appendix lists, in alphabetical order, the Tivoli commands related to installing Tivoli products using either Tivoli Management Framework or Tivoli Software Installation Service (SIS). It contains reference information about the SIS commands.

It is often necessary or convenient to perform a Tivoli operation using the command line interface (CLI) rather than the Tivoli desktop. For example, any of the following reasons may make it preferable to use the CLI:

- You do not have access to a Tivoli desktop.
- You want to group several Tivoli commands in a shell script or batch file.
- You prefer to run a command from within a script that performs multiple operations.

Command Syntax

The reference pages in this appendix use the following special characters to define the command syntax:

[ ] Identifies optional options. Options not enclosed in brackets are required.

... Indicates that you can specify multiple values for the previous option.

| Indicates mutually exclusive information. You can use either the option to the left of the separator or the option to the right of the separator. You cannot use both options in a single use of the command.

{ } Delimits a set of mutually exclusive options when one of the options is required. If the options are optional, they are enclosed in brackets ([ ]).

The following is an example of command syntax:

wchkdb [**–o** *outfile*] [**–u**] [**–x**] {**–f** *infile* | **–i** | *object* ...]

In this example, the **–f** *infile*, **–i**, and *object* options are mutually exclusive. The braces ({ }) indicate that one of these options is required. If you choose to specify the *object* option, you can optionally specify more than one object name or ID.

Tivoli Management Framework Commands

Tivoli Management Framework provides a command line interface for installing and uninstalling Tivoli products and managed resources. This includes the following commands.

Command	Purpose
wclient	Installs a managed node.
wcrtgate	Creates an endpoint gateway.
wcrtrim	Creates a RIM object.
wdelep	Deletes an endpoint.
wdelgate	Deletes an endpoint gateway.
winstall	Installs a Tivoli product.

Command	Purpose
w4inslcf.pl	Installs an endpoint on an AS/400 machine.
winstlcf	Installs an endpoint on a UNIX or Windows machine.
wpatch	Installs a Tivoli patch.
wrmnode	Uninstalls a managed node.
wserver	Installs a Tivoli server on UNIX machines.
wstandalone	Installs a managed node or endpoint without using rsh or rexec.
wuninst	Uninstalls Tivoli applications from a specified node or from the entire Tivoli region.
wunstmn	Removes Tivoli Management Framework files from a managed node.

These commands are documented in the *Tivoli Management Framework Reference Manual*.

Tivoli Software Installation Service Commands

The Tivoli Software Installation Service client provides a command line interface that enables you to perform many of the same operations that you can do using the Tivoli Software Installation Service console. The following table lists the commands provided by the Tivoli Software Installation Service client.

Command	Purpose
wimport	Adds installation images to and removes installation images from the Tivoli Software Installation Service depot, and displays products available in the Tivoli Software Installation Service depot.
wsis	Creates Tivoli Software Installation Service response files, checks the syntax of a response file, and installs Tivoli products, patches, and clients.
wsisdefaults	Displays and modifies values of installation options for products in the Tivoli Software Installation Service depot.
wsisgui	Starts the Tivoli Software Installation Service console.
wsisprefs	Displays and sets the SIS preferences.
wsisprereq	Enables, disables, imports, exports, resequences, or removes SIS installation prerequisites.

wimport

Adds installation images to and removes installation images from the Tivoli Software Installation Service depot, and displays products available in the Tivoli Software Installation Service depot.

Syntax

wimport –?

wimport –D

wimport –c *source_dir* [**–i** *index_file* [*interp*]] [**–p**] [**–d** *depot*]

wimport –l [**–d** *depot*]

wimport –r *product* [*interp*...]... [**–p**] [**–d** *depot*]

Description

The **wimport** command imports products to or removes them from the Tivoli Software Installation Service depot. **wimport** also list the products in the Tivoli Software Installation Service depot.

If you import, remove, and list products with a single **wimport** command, the actions are performed in this order: all remove requests are processed, all import requests are processed, and then the list is displayed.

Note: Tivoli Software Installation Service treats product and patch installation images identically, and refers to both types of installation images as products.

Authorization

Requires Tivoli role **user** plus one of **super**, **senior**, **install_client**, or **install_product**.

Options

–? Displays a usage statement for this command.

–c *source_dir*
Identifies the path to the directory that contains the installation images to be imported. This can be any directory, including a CD-ROM drive mounted from a remote system. You can also import from another install repository by specifying the directory containing the install repository.

–c is required when importing products with **–i**.

Specifying **–c** and **–p** together lists the products in the installation source directory *source_dir*.

–d *depot*
Identifies the Tivoli Software Installation Service depot into which to import or from which to remove products. Specify the name of the managed node on which the Tivoli Software Installation Service depot is installed.

You can omit this option if only one Tivoli Software Installation Service depot exists in the Tivoli region. If you omit **–d** and more than one Tivoli

Software Installation Service depot exists in the region, the command prompts for which Tivoli Software Installation Service depot to use. If you do not respond within the client timeout period, the command connects to the last depot used by this client. If no previous connection has been made, the command exits.

–D Lists the available Tivoli Software Installation Service depots. If you specify **–D** with other options, the syntax of the other options is checked, but they are not processed.

–i *index_file* [*interp* ...]

Identifies a product to be added (imported) to the Tivoli Software Installation Service depot. If you specify one or more *interp* options, only the files for those interpreter types (plus the **generic** interpreter type) are imported from *index_file*. Otherwise, files for all interpreter types contained in *index_file* are imported.

index_file is the name of an index file (.IND) in the source directory specified by **–c** *source_dir*. You may omit the .IND file extension from the name of the index file. To import from multiple index files, repeat the **–i** option for each index file. If you omit **–i**, all index files in the source directory are imported.

The interpreter type **generic** is always imported.

This option may be specified alone or in combination with **–l** or **–r**. When used in combination, the actions are performed in this order: all remove requests are processed, all import requests are processed, and then the list is displayed.

–l Lists the products in the Tivoli Software Installation Service depot.

This option may be specified alone or in combination with **–i** or **–r**. When used in combination, the actions are performed in this order: all remove requests are processed, all import requests are processed, and then the list is displayed.

–p Displays (previews) the actions that will be taken by **wimport** with the given options, without importing or removing.

–r *product* [*interp* ...]

Identifies a product to be removed from the Tivoli Software Installation Service depot. To remove multiple products, repeat the **–r** *product* option for each product.

You can specify *product* using either the unique product ID (which is created using *tag-revision-description*) or the number of the product in the Tivoli Software Installation Service depot. To determine the unique product ID and product number, use the **–l** option. Be sure to follow the conventions of your shell or command processor if the unique product ID contains characters that need to be escaped or quoted.

If you specify one or more *interp* options, only those files are removed for *product*. Otherwise, files for all interpreter types supported by *product* are removed. The **generic** interpreter type is removed only after removing all other interpreter types.

This option may be specified alone or in combination with **–i** or **–l**. When used in combination, the actions are performed in this order: all remove requests are processed, all import requests are processed, and then the list is displayed.

Examples

1. The following example lists the contents of the Tivoli Software Installation Service depot if a single Tivoli Software Installation Service depot exists in this Tivoli region:

   ```
   wimport -l
   ```

   This command returns output similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus

   ==================================
   Products found in Depot:

   --------------Product 1--------------
   Product Name: Admin-3.6.2-Tivoli_User_Administration_3.6.2
   Interp Name: generic
   Interp Name: solaris2
   Interp Name: aix4-r1
   Interp Name: w32-ix86

   --------------Product 2--------------
   Product Name: GATEWAY-Tivoli_Gateway
   Interp Name: generic

   --------------Product 3--------------
   Product Name: TMA-Tivoli_Endpoint
   Interp Name: aix4-r1
   Interp Name: hpux10
   Interp Name: nw3
   Interp Name: nw4
   Interp Name: os2
   Interp Name: os2-ix86
   Interp Name: solaris2
   Interp Name: w32-ix86
   Interp Name: win95
   Interp Name: nt
   Interp Name: windows
   Interp Name: netware
   Interp Name: os400
   ```

2. The following example lists the contents of the Tivoli Software Installation Service depot on managed node kiwi:

   ```
   wimport -l -d kiwi
   ```

3. The following example imports all interpreter types of the products Tivoli Management Framework (index file TMF.IND), Tivoli Application Development Environment (index file ADE.IND), and Tivoli Application Extension Facility (index file AEF.IND) from the CD in drive E:

   ```
   wimport -i TMF -i ADE -i AEF -c e:\
   ```

4. The following example imports the Windows 32-bit, and Solaris interpreter types of the Tivoli Software Distribution product from the index file COURIER.IND on the CD in drive E:

   ```
   wimport -i COURIER w32-ix86 solaris2 -c e:\
   ```

5. The following example displays which of the specified interpreter types are available to be imported. This command does not import the products, it merely shows you what is available.

   ```
   wimport -i COURIER w32-ix86 solaris2 -c -e:\ -p
   ```

   The **generic** interpreter type is also listed.

6. To completely remove the product `Inventory-3.6-TME_10_Inventory,_Version_3.6`, enter the following command:

   ```
   wimport -r Inventory-3.6-TME_10_Inventory,_Version_3.6
   ```

If the **–l** option of the **wimport** lists this as product number 5 in the Tivoli Software Installation Service depot, the following command is equivalent:

```
wimport -r 5
```

7. To remove installation images for interpreter types w32-ix86 and solaris2 for product `Inventory-3.6-TME_10_Inventory,_Version_3.6`, enter the following command:

   ```
   wimport -r Inventory-3.6-TME_10_Inventory,_Version_3.6 w32-ix86 solaris2
   ```

   If the **–l** option of the **wimport** lists this as product number 5 in the Tivoli Software Installation Service depot, the following command is equivalent:

   ```
   wimport -r 5 w32-ix86 solaris2
   ```

   If w32-ix86 and solaris2 are the only interpreter types for that product in the install repository, these commands also remove the **generic** interpreter type, which completely removes the product.

8. To import all products in the source directory /dev/cdrom/8 into the Tivoli Software Installation Service depot on managed node kiwi, enter the following command:

   ```
   wimport -d kiwi -c /dev/cdrom/8
   ```

See Also

wsis

wsis

Creates Tivoli Software Installation Service response files, checks the syntax of a response file, and installs Tivoli products, patches, and clients.

Syntax

wsis –?

wsis –D

wsis –c *response_file* [**–d** *depot*] [**–r**]

wsis –i *response_file* [**–d** *depot*] [**–o**] [**–r**]

wsis –x *response_file* **–p** *product hostname* [*interp*] [**–p** *product hostname* [*interp*]]... [**–d** *depot*] [**–P**] [**–r**]

Description

The **wsis** command provides a command line interface to the Tivoli Software Installation Service operations. Use it to perform the following tasks:

- Create (export) a response file.
- Check the syntax of a response file.
- Install Tivoli clients, products, and patches.
- Refresh the Tivoli Software Installation Service depot from the Tivoli server.
- Add machines to the Tivoli Software Installation Service depot.

You cannot combine the check (**–c**), export (**–x**), and install (**–i**) options.

Authorization

Tivoli role **user** plus one of **super**, **senior**, **install_client**, or **install_product**.

Options

–? Displays a usage statement for this command.

–c *response_file*
Checks the syntax of the response file and attempts to contact the specified machines. Does not perform an installation. The *response_file* option specifies the complete path of the response file.

Notes:

- If the response file contains [machine] sections that define new machines, checking the syntax of the response file makes a test connection to each machine and, if the test connection is successful, adds the machine.
- If the response file lists Windows machines on which Tivoli Remote Execution Service is not installed, checking the syntax of the response file will install Tivoli Remote Execution Service. If the [machine] section contains the entry `autoInstallTrip=no`, a prompt will ask whether you want to install Tivoli Remote Execution Service. If the entry is omitted or specifies `autoInstallTrip=yes`, Tivoli Remote Execution Service will be installed without a prompt.

–d *depot*

Identifies the Tivoli Software Installation Service depot to use. Specify the name of the machine on which the Tivoli Software Installation Service depot is installed.

You can omit this option if only one Tivoli Software Installation Service depot exists in the Tivoli region. If you omit **–d** and more than one Tivoli Software Installation Service depot exists in the region, the command prompts for which Tivoli Software Installation Service depot to use. If you do not respond within the client timeout period, the command connects to the last depot used by this client. If no previous connection has been made, the command exits.

–D
Lists the available Tivoli Software Installation Service depots. If you specify **–D** with other options, the syntax of the other options is checked, but they are not processed.

–i *response_file*

Performs an installation using the information in a response file. The *response_file* option specifies the complete path of the response file.

–o
Forces (overrides) the installation of a product when it has been subsumed by or is equivalent to another product that is either already installed on the machine or will be installed by the response file.

–p *product hostname* [*interp*]

Specifies a product, the name of a managed node, and optionally the interpreter type for which a response file is exported. To create a response file containing multiple products or the same product on multiple machines or interpreter types, repeat the **–p** option.

You can specify *product* using either the unique product ID (which is created using *tag-revision-description*) or the number of the product in the Tivoli Software Installation Service depot. To determine the unique product ID and product number, use the **–l** option. Be sure to follow the conventions of your shell or command processor if the unique product ID contains characters that need to be escaped or quoted.

The *hostname* option specifies the machine for which information is to be exported. The **wsis** command searches first for a Tivoli node with this name. If one is not found, *hostname* must be the domain name system (DNS) name of a machine. If you specify a machine by its DNS name, you must also specify *interp*. If you specify *interp* for a *hostname* that is already known to Tivoli Software Installation Service, it is ignored.

–P
Causes the Tivoli installation password and the login passwords for machines to be included in the response file. By default, passwords are not exported.

Note: The passwords are displayed as clear text in the response file. Ensure that directory and file permissions allow only authorized users to access this information.

–r
Causes the product and machine list in the Tivoli Software Installation Service depot to be refreshed from the Tivoli server. You must use **–r** in combination with **–c**, **–i** or **–x**. When used with any of these options, the Tivoli Software Installation Service depot is refreshed *before* the response file is created, imported or exported.

–x *response_file*

Exports a response file. The *response_file* option specifies the complete path of the response file that is created. If *response_file* already exists, it is overwritten.

Examples

1. To refresh the Tivoli Software Installation Service depot and then check the response file fw361_antares_kmcbride_foxy.rsp without performing an installation, enter the following command:

   ```
   wsis -r -c fw361_antares_kmcbride_foxy.rsp
   ```

2. The following command performs an installation of products and machines specified in the response file gw_ganamede.rsp:

   ```
   wsis -i gw_ganamede.rsp
   ```

3. The following command, using response file Force_DMUnivMon_Install.rsp, causes the Tivoli Software Installation Service depot to be refreshed and forces the installation of any product that is subsumed by another product either already installed or listed in the response file for installation on this machine:

   ```
   wsis -r -o -i Force_DMUnivMon_Install.rsp
   ```

4. To generate a response file called ntep.rsp that will be used to install product number 4 on managed node galileo, enter the following command:

   ```
   wsis -x ntep.rsp -p 4 galileo
   ```

   Because galileo is already a managed node, you do not need to specify its interpreter type.

   Alternately, you can specify the product by its unique product ID. If product number 4 has the unique product ID of `ADE-3.6-TME_10_ADE,_Version_3.6____3.6_-_build_08_10_`, the following command has the same result:

   ```
   wsis -x NTEP.RSP -p \
   "ADE-3.6-TME_10_ADE,_Version_3.6____3.6_-_build_08_10_" \
   galileo
   ```

   The unique product ID is enclosed in double quotation marks (″) because it contains a special character, the comma (,).

5. To create a response file called /tmp/aix.rsp that lets you easily install managed nodes on the AIX systems cuervo and sauza and on the Windows system herradura, enter the following command:

   ```
   wsis -x /tmp/aix.rsp -p 2 cuervo aix4-r1 \
     -p 2 sauza aix4-r1 -p 2 herradura w32-ix86
   ```

   Because systems cuervo, sauza, and herradura are not currently managed nodes, you must specify the interpreter type for each.

See Also

wimport

wsisdefaults

Displays and modifies values of installation options for products in the Tivoli Software Installation Service depot.

Syntax

wsisdefaults –?

wsisdefaults –D

wsisdefaults –i *file_name* [**–d** *depot*]

wsisdefaults –l [**–d** *depot*]

wsisdefaults –x *file_name* [*product* ...] [**–d** *depot*]

Description

Use the **wsisdefaults** command to view or modify the default values of the installation options of one or more products in the Tivoli Software Installation Service depot. This command provides a function similar to that provided by the Product details window of the Tivoli Software Installation Service console.

You can combine the import, export, and list functions of **wsisdefaults**. The operations are performed in this order: all export operations are performed, all import operations are performed, and then the list operation is performed.

To modify defaults, you export a product defaults file for one or more products. This file contains the default values of each installation option for each interpreter type currently imported into the install repository. Change this file using any text editor. Import the modified product defaults file to change the values. The format of the product defaults file is described in “Syntax of the product defaults file” on page 122.

Each interpreter type supported by a product has a separate set of default values. You can use **wsisdefaults** to set the same defaults for multiple interpreter types.

If you define an option more than once for the same interpreter type in the product defaults file, the last reference is used. If you omit an option, it is unchanged.

Authorization

Tivoli role **user** plus one of **super**, **senior**, **install_client**, or **install_product**.

Options

–? Displays a usage statement for this command.

–d *depot*
Identifies the Tivoli Software Installation Service depot to use. Specify the name of the machine on which the Tivoli Software Installation Service depot is installed.

You can omit this option if only one Tivoli Software Installation Service depot exists in the Tivoli region. If you omit **–d** and more than one Tivoli Software Installation Service depot exists in the region, the command prompts for which Tivoli Software Installation Service depot to use. If you do not respond within the client timeout period, the command connects to the last depot used by this client. If no previous connection has been made, the command exits.

–D Lists the available Tivoli Software Installation Service depots. If you specify **–D** with other options, the syntax of the other options is checked, but they are not processed.

–i *file_name*

Sets defaults for product installation options by importing the product defaults file specified by *file_name*. You can specify *file_name* as either a relative or full path name.

–l Displays an alphabetic list showing each product in the Tivoli Software Installation Service depot that has installation options and the current values of the installation options for each product.

–x *file_name* [*product*]

Creates a template for a product defaults file by exporting information in the Tivoli Software Installation Service depot.

You can specify *file_name* as either a relative or full path name. If *file_name* already exists, it is overwritten.

product specifies one or more products whose defaults are written to the product defaults file. The products must already exist in the Tivoli Software Installation Service depot. If you omit the *product* option, the generated file contains all products in the Tivoli Software Installation Service depot that have modifiable installation options.

You can specify *product* using either the unique product ID (which is created using *tag-revision-description*) or the number of the product in the Tivoli Software Installation Service depot. To determine the unique product ID and product number, use the **–l** option. Be sure to follow the conventions of your shell or command processor if the unique product ID contains characters that need to be escaped or quoted.

Examples

1. To list each product in the Tivoli Software Installation Service depot with its default installation options, enter the following command:

   ```
   wsisdefaults -l
   ```

   The output of this command contains an alphabetic list of products and defaults, similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   Reading defaults from cygnus

   --------------Product 1--------------
   Product Name: GATEWAY-Tivoli_Gateway

        Interp Name: generic
   @GATE_NAME@ = @HostName@-gateway
   @GATE_PORT@ = 9494

   --------------Product 4--------------
   Product Name: TMA-Tivoli_Endpoint

        Interp Name: aix4-r1
   TMABIN = /opt/Tivoli/lcf
   @EndpointLabel@ = @HostName@
   @EndpointPort@ = 9495
   ```

```
@EndpointStartupOpts@ =
@GatewayName@ = Broadcast to Gateways
@PolicyRegionName@ = None
@CheckLogin@ = On
@EndpointStartupTimeout@ = 300

     Interp Name: hpux10
TMABIN = /opt/Tivoli/lcf
@EndpointLabel@ = @HostName@
@EndpointPort@ = 9495
@EndpointStartupOpts@ =
@GatewayName@ = Broadcast to Gateways
@PolicyRegionName@ = None
@CheckLogin@ = On
@EndpointStartupTimeout@ = 300

... <additional interpreter types deleted> ...
--------------Product 5--------------
Product Name: TMF-client-3.7

     Interp Name: solaris2
CAT = /usr/local/Tivoli/msg_cat
LIB = /usr/local/Tivoli/lib
BIN = /usr/local/Tivoli/bin
DB = /data/Tivoli3.7
MAN = /usr/local/Tivoli/man
APPD = /usr/lib/X11/app-defaults
@CreatePaths@ = On
@AutoStart@ = On
@SetPort@ = On
PR_NAME = NoonTide-Region
```

The output lists only products in the install repository that have installation options. Products without installation options are not listed.

2. To create a product defaults file called moredefaults (in the current directory) that contains the current default values for installation options for the products 1 and 4 in the Tivoli Software Installation Service depot, enter the following command:

   ```
   wsisdefaults -x moredefaults 1 4
   ```

   You can also type the unique product ID of either product. For example, if the output of **–l** option of the **wsisdefaults** list product 4 with the product name `TMA-Tivoli_Endpoint`, the following command has equivalent results:

   ```
   wsisdefaults -x moredefaults 1 TMA-Tivoli_Endpoint
   ```

3. To set the defaults for installation options using the information in the product defaults file /tmp/mydefaults, enter the following command:

   ```
   wsisdefaults -i /tmp/mydefaults
   ```

4. To set product installation option defaults using the information in the file c:\Temp\NewDefaults, and then to list the defaults for all products in the install repository, enter the following command:

   ```
   wsisdefaults -l -i c:\Temp\NewDefaults
   ```

wsisgui

Starts the Tivoli Software Installation Service console.

Syntax

wsisgui [–?]

Description

The **wsisgui** command starts the Tivoli Software Installation Service console.

You can also start the Tivoli Software Installation Service console from the Tivoli desktop. On the **Desktop** menu, select **Install** and then select **SIS Client**.

On UNIX systems, perform the following actions to allow the console to open on the X Window System display:

- Set the DISPLAY environment variable to the X Window System display on which to display the Tivoli Software Installation Service console.

  For example, to open the Tivoli Software Installation Service console on the display named cygnus:0.0, a Bourne or Korn shell user would enter the following commands:

  ```
  DISPLAY=cygnus:0.0
  export DISPLAY
  ```

- Enable remote connections to the X Window System.

  This step is necessary even if the Tivoli Software Installation Service console runs on the same machine as the X Window System display. For example, if the Tivoli Software Installation Service client is installed on cygnus, enter the following command to enable the Tivoli Software Installation Service console to start on the display:

  ```
  xhost +cygnus
  ```

Authorization

Tivoli role **user** plus one of **super**, **senior**, **install_client**, or **install_product**.

Options

–? Displays a usage statement for this command.

Examples

The following command starts the console:

```
wsisgui
```

wsisprefs

Displays and sets the Tivoli Software Installation Service preferences.

Syntax

wsisprefs –?

wsisprefs –D

wsisprefs –l [**–v**] [**–d** *depot*]

wsisprefs –r [**–d** *depot*]

wsisprefs –s *preference value* [**–s** *preference value*]... [**–d** *depot*]

Description

The **wsisprefs** command displays and sets the Tivoli Software Installation Service depot and Tivoli Software Installation Service client preferences. This command performs the same function as the SIS Preferences window of the Tivoli Software Installation Service console. For additional information about specific preferences, refer to "Tivoli Software Installation Service depot preferences" on page 105 and "Tivoli Software Installation Service client preferences" on page 107.

Authorization

Tivoli role **user** plus one of **super**, **senior**, **install_client**, or **install_product**.

Options

–? Displays a usage statement for this command.

–d *depot*
Specifies the Tivoli Software Installation Service depot whose preferences you want to display or modify. Specify the name of the machine on which the Tivoli Software Installation Service depot is installed.

You can omit this option if only one Tivoli Software Installation Service depot exists in the Tivoli region. If you omit **–d** and more than one Tivoli Software Installation Service depot exists in the region, the command prompts for which Tivoli Software Installation Service depot to use. If you do not respond within the client timeout period, the command connects to the last depot used by this client. If no previous connection has been made, the command exits.

–D Lists the available SIS depots. If you specify **–D** with other options, the syntax of the other options is checked, but they are not processed.

–l Lists the name, label, and value of each preference. Use with **–v** to obtain detailed information about each preference.

If you specify **–l** with either **–r** or **–s**, the restore or set is performed, and then the preferences are listed.

–r Restores the preferences to their default values. Does not change the value of preferences that set directory names, such as `Install repository location` and `Client log directory`.

–s *preference value*

Sets one or more preferences to the specified values. You can specify *preference* by either its label or the number displayed in the output of the **–l** option of the **wsisprefs** command.

–v Provides a detailed (verbose) listing of the preferences. Additional information includes the default and the possible values. This option must be used with **–l**.

Examples

1. In a Tivoli region with one Tivoli Software Installation Service depot, the following example lists the values of the preferences:

```
wsisprefs -l
```

This command returns output similar to the following:

```
Connecting to SIS Depot
Successfully connected to SIS Depot cygnus
SIS Depot Preferences for cygnus

#  Name                                         Label                     Value
-- -------------------------------------------- ------------------------- -----
1  Client connections to SIS depot              client_connections        5
2  Dispatch threads                             dispatch_threads          16
3  Client packet push size                      client_pkt_push_size      40000
4  Client configure retries                     client_cfg_retries        8
5  Client configure sleep (msecs)               client_cfg_sleep          15000
6  REXEC port                                   rexec_port                512
7  RSH port                                     rsh_port                  514
8  Remote connection timeout (msecs)            remote_connection_timeout 45000
9  Read-only install repository                 read_only_ir              false
10 Install repository location                  depot_location      /data/IR/IR
11 TRIP drive                                   trip_drive                C
12 Archive SIS depot logs after (days, 0=never) log_archive_days          5
13 Delete SIS depot logs after (days, 0=never)  log_purge_days            0

Client Preferences

#  Name                  Label                 Value
-- --------------------- --------------------- ----------------------
14 Client log directory  client_data_dir       /data/clientlogs
15 Client prompt timeout client_prompt_timeout 20
```

2. The following example lists the values of the preferences for the Tivoli Software Installation Service depot on managed node cygnus:

```
wsisprefs -l -d cygnus
```

3. In a Tivoli region with one Tivoli Software Installation Service depot, the following command specifies the D drive for the `TRIP drive` preference:

```
wsisprefs -s trip_drive D
```

Note: The drive letter is specified without a colon.

If the `TRIP drive` preference is listed as number 11 in the output of the **–l** option of the **wsisprefs** command, the following command performs the same function:

```
wsisprefs -s 11 D
```

4. The following command provides a detailed list of the preferences and their default and possible values:

```
wsisprefs -lv
```

In a Tivoli region with a single Tivoli Software Installation Service depot, this command returns output similar to the following:

```
Connecting to SIS Depot
Successfully connected to SIS Depot cygnus
SIS Depot Preferences for cygnus

#  Label                     Value       Default Min  Max
-- ------------------------- ----------- ------- ---- ----------
1  client_connections        5           5       1    10
2  dispatch_threads          16          16      1    100
3  client_pkt_push_size      40000       40000   1    2147483647
4  client_cfg_retries        8           8       1    1024
5  client_cfg_sleep          15000       15000   500  2147483647
6  rexec_port                512         512     1    65535
7  rsh_port                  514         514     1    1024
8  remote_connection_timeout 45000       45000   1    2147483647
9  read_only_ir              false       false   true false
10 depot_location            /data/IR/IR
11 trip_drive                C           c       a    z
12 log_archive_days          5           5       0    365
13 log_purge_days            0           0       0    365

Client Preferences

#  Label                 Value                  Default Min Max
-- --------------------- ---------------------- ------- --- ---
14 client_data_dir       /data/clientlogs
15 client_prompt_timeout 20                     20      1   100
```

5. The following command sets the Dispatch threads in the Tivoli Software Installation Service depot on managed node kiwi to 15:

   ```
   wsisprefs -s dispatch_threads 15 -d kiwi
   ```

   If the Dispatch threads preference is listed as number 2 in the output of the **–l** option of the **wsisprefs** command, the following command performs the same function:

   ```
   wsisprefs -s 2 15 -d kiwi
   ```

6. The following command sets the Dispatch threads preference in the Tivoli Software Installation Service depot on managed node kiwi to 15 and the Client log directory preference on the current client to /tmp/clientlogs, and then lists all the preferences:

   ```
   wsisprefs -s dispatch_threads 15 \
   -s client_data_dir /tmp/clientlogs -d kiwi -l
   ```

wsisprereq

Enables, disables, imports, exports, resequences, or removes Tivoli Software Installation Service installation prerequisites.

Syntax

wsisprereq –?

wsisprereq –D

wsisprereq –l [*type*] [**–v**] [**–d** *depot*]

wsisprereq –L [**–d** *depot*]

wsisprereq –i *file_name type* [**–d** *depot*]

wsisprereq –x*directory* [[*type* [*file_name*]] | [*type file_name*]...] [**–d** *depot*]

wsisprereq –e [**yes** | **no**] *type number* [**–d** *depot*]

wsisprereq –s *sequence type number* [**–d** *depot*]

wsisprereq –r *type number* [**–d** *depot*]

Description

Use the **wsisprereq** command to do the following:

- List the types of installation for which prerequisites can be defined.
- List the names of prerequisites.
- Specify whether a specific prerequisite should be run.
- Create or remove a prerequisite.
- View or change the order in which prerequisites are checked.
- View or modify the details of prerequisites.

Tivoli Software Installation Service provides prerequisites that check for common causes of installation failure and enables you to create prerequisites to test conditions specific to your Tivoli environment.

You can enable, disable, or resequence Tivoli-defined prerequisites. You cannot create, remove, modify, or export Tivoli-defined prerequisites.

You can create, remove, modify, export, enable, disable, or resequence user-defined prerequisites.

Import (**–i**), export (**–x**), list (**–l**), and remove (**–r**) can be combined. They are processed in this order: remove, import, export, list.

This command uses a prerequisite definition file that specifies the name, sequence number, failure string, and shell script for one or more prerequisites. It also specifies whether the prerequisite is enabled to run. It is recommended that you create this file by exporting the existing user-defined prerequisites and modifying the file in the following ways:

- To add a prerequisite, add information describing the prerequisite to the file.

- To remove a prerequisite, remove the information about that prerequisite from the file.
- To modify a prerequisite, change the information in the file.

The format of the prerequisite definition file is described in "Format of the prerequisite definition file" on page 191.

The following types of installations are supported:

Client These prerequisites are run before installing Tivoli Management Framework to create a managed node (client).

Product
These prerequisites are run before installing a Tivoli product on a managed node. These prerequisites are not checked when installing Tivoli products to endpoints.

Patch These prerequisites are run before applying a patch to a Tivoli product on a managed node. These prerequisites are not checked when applying patches to Tivoli products installed on endpoints.

Endpoint
These prerequisites are run before installing an endpoint on a managed node or new machine. These prerequisites are not checked when installing Tivoli Enterprise software to existing endpoints.

Note: Special prerequisites are run before installing an endpoint on an OS/400 machine. You cannot enable or disable these prerequisites or create user-defined prerequisites for OS/400 endpoints.

The case of *type* is not considered. That is, you can specify the endpoint prerequisite type as `endpoint`, `Endpoint`, `ENDPOINT`, and so forth.

Authorization

Tivoli role **user** plus one of **super**, **senior**, **install_client**, or **install_product**.

Options

–? Displays a usage statement for this command.

–e [yes | no] *type number*
Specifies whether to enable (`yes`) or disable (`no`) checking of the prerequisite of the specified type and number. If you do not specify either `yes` or `no`, the prerequisite is enabled.

–d *depot*
Identifies the Tivoli Software Installation Service depot to use. Specify the name of the machine on which the Tivoli Software Installation Service depot is installed.

You can omit this option if only one Tivoli Software Installation Service depot exists in the Tivoli region. If you omit **–d** and more than one Tivoli Software Installation Service depot exists in the region, the command prompts for which Tivoli Software Installation Service depot to use. If you do not respond within the client timeout period, the command connects to the last depot used by this client. If no previous connection has been made, the command exits.

–D Lists the available Tivoli Software Installation Service depots. If you specify **–D** with other options, the syntax of the other options is checked, but they are not processed.

–i *file_name type*
Uses data from the prerequisite definition file specified by *file_name* to modify, add, or remove the prerequisites of the specified *type*.

It is recommended that you create the imported file by modifying a file exported using the **–x** option.

–l [*type*]
Lists the name, number, and sequence number of the specified prerequisites, and indicates whether it is user-defined.

If you omit *type*, **–l** lists each prerequisite of all types. If you specify a prerequisite *type*, **–l** lists each prerequisite for the specified installation type.

–L Lists the types of prerequisites in the Tivoli Software Installation Service depot. Use these values when specifying *type*.

–r *type number*
Removes the prerequisite with the specified type and number. You can remove only user-defined prerequisites, not Tivoli-defined ones.

–s *sequence type number*
Changes the order in which prerequisite checks are run. *sequence* specifies the order in which the prerequisite runs. *number* is the number of the prerequisite, displayed by **wsisprereq –l**.

Prerequisites are run in numerical order, beginning with sequence number 0. You can skip or duplicate sequence numbers. If two prerequisites have the same sequence number, the order in which they run relative to one another is undefined.

–x *directory* [[*type* [*file_name*]] | [*type file_name*]...]
Exports information about user-defined prerequisites of the specified installation types. Tivoli-defined prerequisites are not exported.

To export all user-defined prerequisites, use **–x** *directory*, where *directory* is the name of an existing directory to receive the exported files. One file is created for each type of prerequisite. The files are named *type*, for example, endpoint or client.

To export user-defined prerequisites of a specific type, use **–x** *directory type* or **–x** *directory type file_name*. When exporting prerequisites for a single type of installation, you can omit *file_name*; the resulting file is named *type*, where *type* is the prerequisite type, such as endpoint or client. When exporting prerequisites for a more than one type of installation, you must specify *file_name* for each.

Comments at the beginning of the exported file describe the format and content of the file.

–v Displays additional information about the specified prerequisites, including the failure string and the shell script that is run to test the prerequisite. Use this option with **–l**.

Examples

1. To list all prerequisites that are run for endpoint installations, enter the following command:

```
wsisprereq -l endpoint
```

Output from this command is similar to the following:

```
Connecting to SIS Depot
Successfully connected to SIS Depot cygnus
---------- Prerequisites for endpoint ----------

Number Name                           Enabled User-defined Sequence
------ ------------------------------ ------- ------------ --------
1      Gateway is alive               true    false        0
2      Need to install worldname.exe  true    false        1
3      /dev/null permissions          true    false        2
4      Connection Test                true    false        3
5      Two way comm with TMR(ping)    true    false        4
6      Two way comm with Gateway(ping) true   false        5
7      US keyboard exists             true    false        6
8      Look for file in /tmp          true    true         12
```

The number in front of the prerequisite name (for example, 1) is the prerequisite number. Use this number to specify a prerequisite for the **wsisprereq** command. Contrast this with the sequence number for prerequisite 1 (in this example, 0), which indicates the order in which the prerequisite is tested.

Prerequisite number 8, `Look for file in /tmp`, is user-defined.

2. To display detailed information about all prerequisites that are run for patch installations, including the failure string that is recorded in the logs and the script that is run to test each prerequisite, enter the following command:

   ```
   wsisprereq -l patch -v
   ```

   Output from this command is similar to the following:

   ```
   Connecting to SIS Depot
   Successfully connected to SIS Depot cygnus
   ---------- Prerequisites for patch ----------

   Number Name                 Enabled User-defined Sequence
   ------ -------------------- ------- ------------ --------
   1      Connection to target true    false        0

   Prerequisite failure string: ISV:0112 Error establishing
   connection to remote host.


   # echo a zero to test if we are talking to the node
   #
   echo "0"
   ```

   This example shows one Tivoli-defined prerequisite that runs before installing a patch. No user-defined prerequisites exist for patch installations in this install repository.

3. To make the user-defined endpoint prerequisite number 8 run at sequence 0, enter the following command:

   ```
   wsisprereq -s 0 endpoint 8
   ```

4. To disable prerequisite checking for product prerequisite number 3, enter the following command:

   ```
   wsisprereq -e no product 3
   ```

5. To enable prerequisite checking for product prerequisite number 3, use one of the following commands:

   ```
   wsisprereq -e product 3
   wsisprereq -e yes product 3
   ```

6. To export the user-defined prerequisites for all installation types and create the exported files in the directory /tmp/sisprereqs, enter the following command:

```
wsisprereq -x /tmp/sisprereqs
```

The following files are created in that directory:

client
endpoint
product
patch

7. To export the user-defined prerequisites for installation types **client** and **endpoint** into the files /tmp/client.prereqs and /tmp/ep.prereqs, respectively, enter the following command:

   ```
   wsisprereq -x /tmp client client.prereqs endpoint ep.prereqs
   ```

8. To add a user-defined prerequisite to run when installing endpoints, enter the following procedure:

   a. Create and test the prerequisite script. For this example, the following script will run to test the prerequisite:

      ```
      #!/bin/sh
      FILE=@TMP_DIR@/testfile
      if [ -f $FILE ]
      then
        echo "0"
      else
        echo "Cannot find file $FILE on @HostName@."
      fi
      ```

      This script tests for the existence of a file, testfile, in the temporary directory appropriate to the platform of the target machine (@TMP_DIR@), on the machine to which software is to be installed (@HostName@). If the file exists, the script writes the number 0 to standard output to indicate that the prerequisite succeeded. If the file does not exist on the target machine, the script writes an error message that is recorded in the Tivoli Software Installation Service depot logs and stops the installation.

   b. Export the existing user-defined prerequisites for an endpoint installation. For example, to export into the file /tmp/ep.prereqs, enter the following command:

      ```
      wsisprereq -x /tmp endpoint ep.prereqs
      ```

      The exported file contains comments explaining the format of the file and the definition of each existing user-defined prerequisite.

   c. Add the new prerequisite to the exported file, following the format described in “Format of the prerequisite definition file” on page 191. For example, to create an enabled prerequisite named **/tmp/testfile exists** that runs at sequence 12, and issues the message **Can’t find /tmp/testfile** if the script fails, add the following lines:

      ```
      [Look for file in /tmp]
      sequence=12
      enabled=true
      failStr=Can't find /tmp/testfile
      description=#!/bin/sh
      description=FILE=@TMP_DIR@/testfile
      description=if [ -f $FILE ]; then
      description=  echo "0"
      description=else
      description=  echo "Cannot find file $FILE on @HostName@."
      description=fi
      ```

   d. Import the prerequisite definition file. For example, the following command imports prerequisites from the file **/tmp/ep.prereqs**:

      ```
      wsisprereq -i /tmp/ep.prereqs endpoint
      ```

e. Verify the prerequisite. For example, the following command lists the resulting prerequisites:

```
wsisprereq -l endpoint
```

The following sample output shows that the new prerequisite was added as prerequisite number 8:

```
Connecting to SIS Depot
Successfully connected to SIS Depot cygnus
---------- Prerequisites for endpoint ----------

Number Name                            Enabled User-defined Sequence
------ ------------------------------- ------- ------------ --------
1      Gateway is alive                true    false        0
2      Need to install worldname.exe   true    false        1
3      /dev/null permissions           true    false        2
4      Connection Test                 true    false        3
5      Two way comm with TMR(ping)     true    false        4
6      Two way comm with Gateway(ping) true    false        5
7      US keyboard exists              true    false        6
8      Look for file in /tmp           true    true         12
```

9. To remove the user-defined endpoint prerequisite number 8, enter the following command:

```
wsisprereq -r endpoint 8
```

Appendix B. Directory structure and system variables

This section describes the directory structures and system variables created during the installation of Tivoli Management Framework.

UNIX Tivoli servers and managed nodes

When you install Tivoli Management Framework on UNIX operating systems, many new files and directories are created, some existing files are modified, and several system variables are defined.

Note: When you install Tivoli Management Framework on a UNIX operating system, the /tmp/.tivoli directory is created. This directory contains files that are required by the object dispatcher process. You should not delete this directory or any of its contents unless explicitly directed to by your Tivoli support provider. You should also ensure that regularly scheduled disk clean-up jobs (cron or Tivoli jobs) do not remove this directory or its contents.

To use a different directory, you must set an environment variable in both the object dispatcher and the shell. After installing Tivoli Management Framework, perform the following steps to set the necessary environment variables:

1. Create a directory. This directory must have at least public read and write permissions. However, define full permissions and set the sticky bit to ensure that users cannot modify files that they do not own.
2. Set the environment variable in the object dispatcher:
   a. Enter the following command:
      ```
      odadmin environ get > envfile
      ```
   b. Add the following line to the `envfile` file and save it:
      ```
      TIVOLI_COMM_DIR=new_directory_name
      ```
   c. Enter the following command:
      ```
      odadmin environ set < envfile
      ```
3. Edit the Tivoli-provided setup_env.csh, setup_env.sh, and oserv.rc files in the /etc/Tivoli directory to set the TIVOLI_COMM_DIR variable.
4. For HP-UX and Solaris systems, add the following line to the file that starts the object dispatcher:
   ```
   TIVOLI_COMM_DIR=new_directory_name
   ```
   Insert the line near where the other environment variables are set, in a location that runs before the object dispatcher is started. The following list contains the file that needs to be changed on each operating system:
   - For HP-UX operating systems: /sbin/init.d/Tivoli
   - For Solaris operating environments: /etc/rc3.d/S99Tivoli
5. Shut down the object dispatcher by entering the following command:
   ```
   odadmin shutdown all
   ```
6. Restart the object dispatcher on the Tivoli server by entering the following command:
   ```
   odadmin reexec 1
   ```

7. Restart the object dispatcher on the managed nodes by entering the following command:

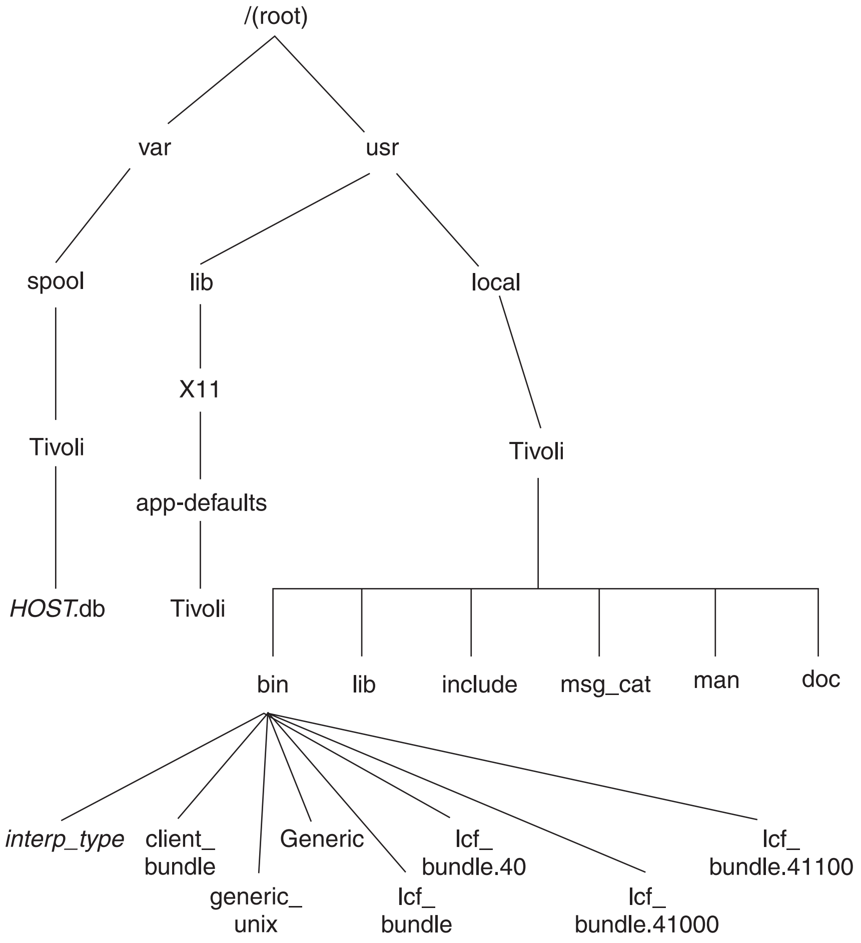
```
odadmin reexec clients
```

Directory Structure

The Tivoli Management Framework installation process installs the following files on UNIX Tivoli servers or managed nodes:

- Libraries
- Binaries
- Command reference pages (manual pages)
- X11 resource files
- Message catalogs
- Databases (server and client)
- Environment setup files

You can install these directories in the root directory (/) or in an installation directory that you create. These files are installed in the directories shown in the following figure.



/var/spool/Tivoli/*host*.db
Contains the database files for the Tivoli servers or managed nodes. These files should always be installed locally on each host.

/usr/lib/X11/app-defaults
Contains the X11 resource files.

/usr/local/Tivoli/bin
Contains the following Tivoli Enterprise binary directories:

client_bundle
Contains files necessary for bootstrap client installation. This directory is installed on the Tivoli server only.

generic
Contains Web pages and language code sets.

generic_unix
Contains generic binaries for all supported platforms.

lcf_bundle
Contains the endpoint binaries required for each supported platform running Tivoli Enterprise software released prior to Tivoli Management Framework, Version 3.6.

lcf_bundle.40
Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 3.7.*x*.

lcf_bundle.41000
Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 4.1.

lcf_bundle.41100
Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 4.3.1.

interp_type
Indicates the interpreter type. A separate subdirectory is created for each interpreter type that you have in your Tivoli region. The contents of these directories depend on the Tivoli Enterprise products that you have installed.

/usr/local/Tivoli/lib
Contains a subdirectory for each interpreter type that you have installed.

interp_type
These subdirectories contain the operating system-specific Tivoli libraries.

/usr/local/Tivoli/include
Contains the header files used by Tivoli Application Development Environment (ADE).

/usr/local/Tivoli/msg_cat
Contains the Tivoli message catalogs.

/usr/local/Tivoli/man
Contains the UNIX manual pages. These files can be installed on a file server accessible by the Tivoli server and its clients.

/usr/local/Tivoli/doc
Contains the most recent Tivoli documentation.

The Tivoli environment setup files are created in the /etc/Tivoli directory. This directory contains the following subdirectories:

bin
: Contains the Perl binaries required for many Tivoli operations.

lib
: Contains the Perl language library required for many Tivoli operations.

tll.conf
: Contains configuration information for the Task Library Language.

The following files are created during installation in the /etc/Tivoli directory:

oserv.rc
: Contains scripts to start or stop the object dispatcher.

setup_env.sh
: Sets up the Tivoli system variables if you are using a Bourne-type shell. Run this script before you start the Tivoli desktop.

setup_env.csh
: Sets up the Tivoli system variables if you are using C shell. Run this script before you start the Tivoli desktop.

Files modified

The following system startup files are added or modified during the installation of a UNIX managed node or Tivoli server.

Operating system	Files modified
AIX	/etc/inittab /etc/services /etc/inetd.conf
HP-UX	/sbin/rc0.d/K100Tivoli /sbin/rc1.d/K100Tivoli /sbin/rc2.d/K100Tivoli /sbin/rc3.d/S500Tivoli
Solaris	/etc/init.d/Tivoli /etc/rc3.d/S99Tivoli /etc/rc2.d/K50Tivoli /etc/rc0.d/K50Tivoli /etc/rc1.d/K50Tivoli /etc/services /etc/inetd.conf
SuSE Linux	/etc/rc.d/rc2.d/S99Tivoli /etc/rc.d/rc3.d/S99Tivoli /etc/rc.d/rc4.d/S99Tivoli /etc/rc.d/rc5.d/S99Tivoli /etc/rc.d/rc2.d/K10Tivoli /etc/rc.d/rc3.d/K10Tivoli /etc/rc.d/rc4.d/K10Tivoli /etc/rc.d/rc5.d/K10Tivoli
Other types of Linux	/etc/rc.d/rc2.d/S99Tivoli /etc/rc.d/rc3.d/S99Tivoli /etc/rc.d/rc4.d/S99Tivoli /etc/rc.d/rc5.d/S99Tivoli /etc/rc.d/rc0.d/K10Tivoli /etc/rc.d/rc1.d/K10Tivoli /etc/rc.d/rc6.d/K10Tivoli

System variables

The following system variables are set when you run either of the setup scripts contained in the /etc/Tivoli directory. If you have previously set system variables, the Tivoli setup program overrides these variables: BINDIR, DBDIR, INTERP, LIBDIR, o_dispatch, EtcTivoli, TMR, and WLOCALHOST. For LD_LIBRARY_PATH, LIBPATH, SHLIB_PATH, MANPATH, NLSPATH, and PATH, the Tivoli settings are added to your existing variable.

System variable	Default setting
BINDIR	/usr/local/Tivoli/bin/*interp_type*
DBDIR	/var/spool/Tivoli/*host*.db
INTERP	*interp_type* Refer to the *Tivoli Management Framework Release Notes* for a complete list of interpreter types.
LD_LIBRARY_PATH LIBPATH (AIX only) SHLIB_PATH (HP-UX only)	"$LIBDIR":/usr/lib:/usr/ucblib
LIBDIR	/usr/local/Tivoli/lib/*interp_type*
MANPATH	/usr/local/Tivoli/man/*interp_type*:$MANPATH
NLSPATH	/usr/local/Tivoli/msg_cat/%L/%N.cat
PATH	"$BINDIR/bin:$BINDIR/ADE:$BINDIR/AEF :$BINDIR/kerberos:$PATH"
XKEYSYMDB	/usr/lib/X11/XKeysymDB
XUSERFILESEARCHPATH	/usr/lib/X11/app-defaults/%L/%N.cat
o_dispatch	94
TISDIR	$BINDIR/../generic
TMR	*region_number*
WLOCALHOST	Host name to listen on

Windows Tivoli servers and managed nodes

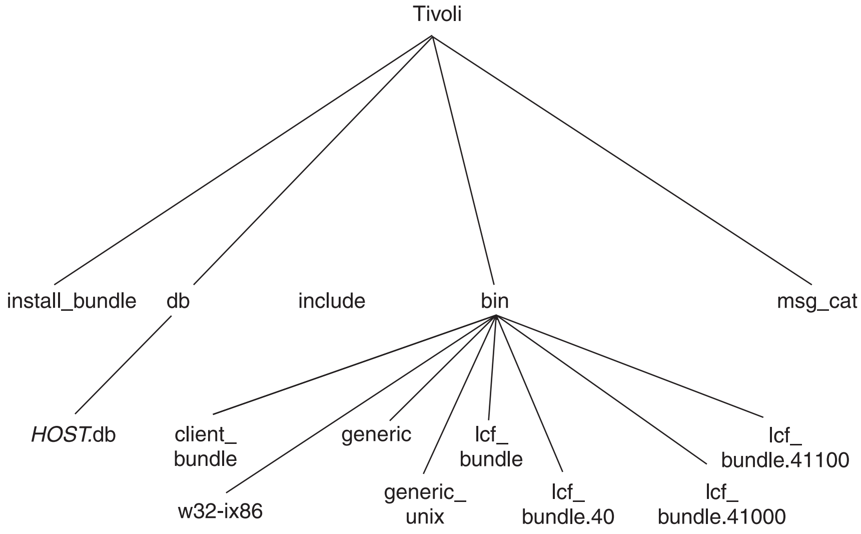
The following sections detail the directory structure created and the system variables defined when you install Tivoli Management Framework on a Microsoft Windows Tivoli server or managed node.

Directory structure

When you install Tivoli servers and managed nodes on Windows operating systems, the installation process creates several directories that contain the following files:

- Libraries
- Binaries
- Header files
- Message catalogs
- Databases

These files are installed in the directories shown in the following figure.



\Program Files\Tivoli\db
: Contains the database files for the Tivoli server and managed nodes. These files must be installed on a local Windows NT File System (NTFS).

\Program Files\Tivoli\include
: Contains the header files used by Tivoli Application Development Environment (ADE).

\Program Files\Tivoli\bin
: Contains the following Tivoli binary directories:

  client_bundle
  : Contains files necessary for bootstrap managed node installation. This directory is installed on the Tivoli server only.

  w32-ix86
  : Contains the Tivoli binaries used by Windows Tivoli servers and managed nodes. The content of this directory depends on the Tivoli Enterprise products that you have installed.

  generic
  : Contains Web pages and language code sets.

  generic_unix
  : Contains generic binaries for all supported platforms.

  lcf_bundle
  : Contains the endpoint binaries required for each supported platform running Tivoli Enterprise software released prior to Tivoli Management Framework, Version 3.6.

  lcf_bundle.40
  : Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 3.7.*x*.

  lcf_bundle.41000
  : Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 4.1.

lcf_bundle.41100
: Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 4.3.1.

\Program Files\Tivoli\lib
: Contains the static Tivoli libraries required for Tivoli Application Development Environment (ADE). Shared libraries for Windows operating systems are located in the \Tivoli\bin directory.

\Program Files\Tivoli\msg_cat
: Contains the Tivoli message catalogs.

The Tivoli environment setup files are created in the %SystemRoot%\system32\drivers\etc\Tivoli directory. This directory contains the following subdirectories:

tll.conf
: Contains configuration information for the Task Library Language.

The following files are created in the %SystemRoot%\system32\drivers\etc\Tivoli directory during installation:

setup_env.sh
: Sets up the Tivoli system variables if you are using a Bourne-type shell. Run this script before you start the Tivoli desktop.

setup_env.bat
: Sets up the Tivoli system variables if you are using the DOS shell. Run this script before you start the Tivoli desktop.

Registry variables

When you install Tivoli Management Framework on Windows operating systems, the appropriate system variables and directory paths are added to the Windows registry.

The HKEY_LOCAL_MACHINE\SOFTWARE\Tivoli\Platform key contains the value of localhost. This value identifies the preferred alias for the local host and is set using the **wlocalhost** command. The localhost key will not exist until the **wlocalhost** command is run.

System variables

The following system variables are set when you run either of the setup scripts contained in the %SystemRoot%\system32\drivers\etc\Tivoli directory. If you have previously set these system variables, Tivoli Management Framework overrides the variables except for PATH. For PATH, the settings are added to the existing variable.

For systems running Windows 2000 and beyond, the default object dispatcher setting for the TMP and TEMP variables, %DBDIR%\tmp, is overwritten with the system values for these variables. This is because, by default, these variables are included in the system environment set, which is not the case for Windows NT systems.

The system settings for TMP and TEMP are unrelated to the values the object dispatcher uses for these variables. The object dispatcher uses %DBDIR%\tmp by default, unless it is overridden by a value for the TMP variable placed in the object dispatcher environment and specified using the **odadmin environ** command, as in the following example:

1. Enter the following:
   ```
   odadmin environ get > env
   ```
2. Add the following statement to env:
   ```
   TMP=temporary_directory
   ```
   where *temporary_directory* is the fully qualified path to the desired temporary directory.
3. Enter the following:
   ```
   odadmin environ set < env
   ```
4. Recycle the object dispatcher.

System variable	Default setting
BINDIR	`%SystemDrive%\Program Files\Tivoli\bin\w32-ix86`
DBDIR	`%SystemDrive%\Program Files\Tivoli\db\`*`host`*`.db`
INTERP	`w32-ix86`
NLSPATH	`%SystemDrive%\Program Files\Tivoli\msg_cat\%L\%N.cat`
o_dispatch	`94`
PATH	`"%BINDIR%\bin;%BINDIR%\tools; %BINDIR%\ADE;%BINDIR%\AEF; %PATH%"`
PERLLIB	`%BINDIR%\tools\lib\perl`
TEMP	`%DBDIR%\tmp`
TISDIR	`%BINDIR%\..\generic`
TMP	`%DBDIR%\tmp`
TMR	*region_number*
WLOCALHOST	Host name to listen on

RIM host

You can set a timer on the RIM agent to throw an exception and exit if the RIM agent does not receive communication from the client application for a specified timeout period. By default, the timer is not active and the RIM agent waits indefinitely for the next command from the client application. This timer can be activated and configured by using the **odadmin** environment variable **RIM_IOM_TIMEOUT**. The value of this variable is the maximum amount of time in hours that the agent waits to receive a command from the client.

The variable **RIM_IOM_TIMEOUT** can be set when you install a RIM host. If you have previously set this system variable, Tivoli Management Framework does not override the variable.

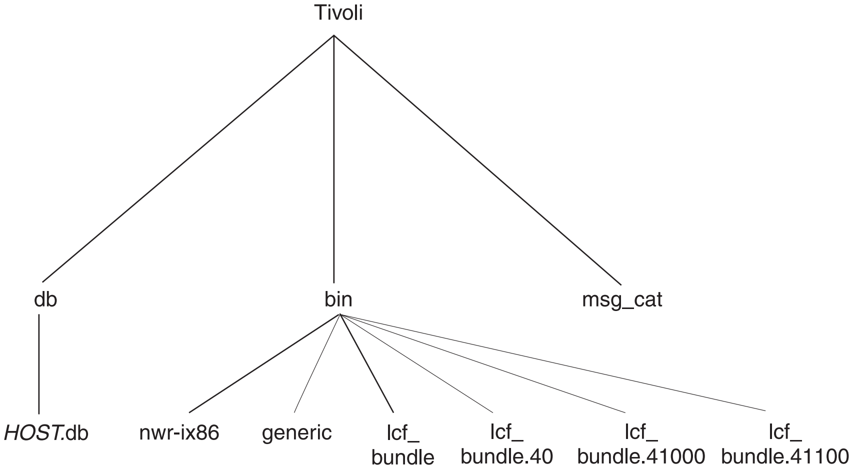
System variable	Value	Is the timer active?	Agent behavior
RIM_IOM_TIMEOUT	Not set	No	The agent waits indefinitely for the next command.
	≤ 0	No	The agent waits indefinitely for the next command.
	≥ 1	Yes	The agent waits up to the number of hours specified for the next command. The maximum number of hours that the agent waits is 48. If the number of hours specified is greater than 48, the agent waits up to 48 hours.

NetWare gateways directory structure

When you install a gateway on a Novell NetWare operating system, the installation process creates several directories that contain the following files:

- Binaries
- Message catalogs
- Databases

These files are installed in the directories shown in the following figure.



\Tivoli\db

Contains the database files for the managed nodes.

\Tivoli\bin

Contains the following Tivoli binary directories:

nwr-ix86

Contains the Tivoli binaries used by NetWare managed nodes. The content of this directory depends on the Tivoli Enterprise products that you have installed.

generic

Contains Web pages and language code sets.

lcf_bundle

Contains the endpoint binaries required for each supported platform running Tivoli Enterprise software released prior to Tivoli Management Framework, Version 3.6. These binaries support product compatibility with Tivoli Management Framework, Version 4.3.1.

lcf_bundle.40

Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 3.7.*x*.

lcf_bundle.41000

Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 4.1.

lcf_bundle.41100

Contains the endpoint binaries required for each supported platform running Tivoli Management Framework, Version 4.3.1.

\Tivoli\msg_cat

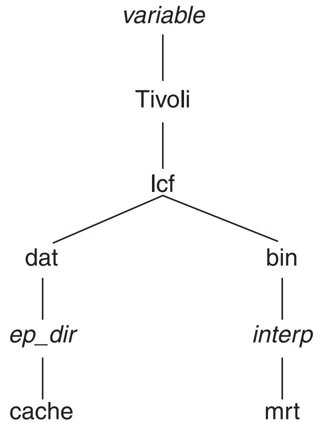
Contains the Tivoli message catalogs.

Endpoints

The following sections detail the directory structure created and the system variables defined when you install a Tivoli endpoint on supported UNIX or Windows operating systems.

Default directory structure

The default directory structure created when you install a Tivoli endpoint is shown in the following figure.



variable

Drive, directory, or volume, which varies depending on the platform where the endpoint is installed:

UNIX operating systems

/opt

Windows operating systems

%SystemDrive%\Program Files

OS/2 operating systems
%BootDrive%

Tivoli\lcf\dat\ep_dir
Endpoint working directory, which contains the following directories and files, among others:

cache Contains the binaries and files used by the endpoint to execute Tivoli Management Framework operations. These binaries and files are downloaded from the assigned gateway as needed by the endpoint.

last.cfg
Contains the latest configuration information used to start the endpoint.

lcf.dat Contains login information.

lcfd.log
Contains log messages generated by the endpoint.

lcfd.sh Contains the script file for starting the endpoint. (UNIX only)

lcfd.st Contains the status of the endpoint.

Tivoli\lcf\bin\interp\mrt
Contains the platform-specific binaries and files needed to start, stop, and manage the endpoint.

Files modified

The following system startup files are added or modified during the installation of a UNIX endpoint.

Operating system	Files modified
AIX	/etc/rc.tma1 /etc/inittab.before.tma1
HP-UX	/sbin/init.d/lcf1.sh /sbin/rc0.d/K100Tivoli_lcf1 /sbin/rc1.d/K100Tivoli_lcf1 /sbin/rc2.d/K100Tivoli_lcf1 /sbin/rc3.d/S500Tivoli_lcf1
Solaris	/etc/init.d/lcf1.rc /etc/rc0.d/K50Tivoli_lcf1 /etc/rc1.d/K50Tivoli_lcf1 /etc/rc2.d/K50Tivoli_lcf1 /etc/rc3.d/S99Tivoli_lcf1
SuSE Linux	/etc/rc.d/rc2.d/S99Tivoli_lcf1 /etc/rc.d/rc3.d/S99Tivoli_lcf1 /etc/rc.d/rc4.d/S99Tivoli_lcf1 /etc/rc.d/rc5.d/S99Tivoli_lcf1 /etc/rc.d/rc2.d/K10Tivoli_lcf1 /etc/rc.d/rc3.d/K10Tivoli_lcf1 /etc/rc.d/rc4.d/K10Tivoli_lcf1 /etc/rc.d/rc5.d/K10Tivoli_lcf1

Operating system	Files modified
Other types of Linux	/etc/rc.d/rc2.d/S99Tivoli_lcf1 /etc/rc.d/rc3.d/S99Tivoli_lcf1 /etc/rc.d/rc4.d/S99Tivoli_lcf1 /etc/rc.d/rc5.d/S99Tivoli_lcf1 /etc/rc.d/rc0.d/K10Tivoli_lcf1 /etc/rc.d/rc1.d/K10Tivoli_lcf1 /etc/rc.d/rc6.d/K10Tivoli_lcf1

System variables

The following system variables can be set after you install an endpoint.

System variable	Default setting
LCFROOT	Endpoint working directory
LCF_DATDIR	`$LCFROOT/dat/`*`ep_dir`*
TISDIR	`$LCFROOT/dat/`*`ep_dir`*
INTERP	*interp_type*

To use these system variables, you must run the environment setup scripts for the endpoint. For details on running these scripts, refer to "Setting system variables for endpoints."

Setting system variables for endpoints

For UNIX operating systems, the installation process creates the following setup scripts:

- /etc/Tivoli/lcf/*ep_dir*/lcf_env.csh
- /etc/Tivoli/lcf/*ep_dir*/lcf_env.sh

You can change your login initialization procedure to use the appropriate setup file so that the necessary environment variables and search paths are set to allow you to start the endpoint.

For example, you might add the following to your initialization procedure:

For C shell (csh):

```
if ( -f /etc/Tivoli/lcf/1/lcf_env.csh ) then
   source /etc/Tivoli/lcf/1/lcf_env.csh
endif
```

For Bourne or Korn shell (sh or ksh):

```
if [ -f /etc/Tivoli/lcf/1/lcf_env.sh ]; then
   . /etc/Tivoli/lcf/1/lcf_env.sh
fi
```

For Windows operating systems, except Windows 98, the installation process creates the following setup scripts:

- %SystemRoot%\Tivoli\lcf\ep_dir\lcf_env.cmd
- %SystemRoot%\Tivoli\lcf\ep_dir\lcf_env.sh

For Windows 98 operating systems, the installation process creates the following setup script:

- %SystemRoot%\Tivoli\lcf\ep_dir\lcf_env.bat

Run one of these scripts to initialize the Tivoli environment variables for the endpoint.

Tivoli Desktop for Windows installations

The following sections detail the directory structure created and the system variables defined when you install Tivoli Desktop for Windows. This product can be installed on supported Windows and OS/2 operating systems.

Note: You do not need to install Tivoli Desktop for Windows on UNIX operating systems.

Directory Structure

The default directory structure for Tivoli Desktop for Windows on supported Windows operating systems is the \Program Files\Tivoli\Desktop directory. The default directory structure for Tivoli Desktop for Windows on supported OS/2 operating systems is the \tivoli\desktop directory.

Registry contents

Tivoli Desktop for Windows stores three values in the Windows registry. The following table contains the values that are stored under the HKEY_CURRENT_USER\Software\Tivoli\Desktop\4.1.1 registry key.

Value name	Data type	Description
host	REG_SZ	Specifies the name of the host system.
port	REG_SZ	Specifies the port number used to connect to the host.
user	REG_SZ	Specifies the name of the user to connect to the host.

These registry values are documented for your information. You can modify all of these settings from the Tivoli command line, although you should exercise caution when doing so. You should not need to directly modify these registry values.

Note: You cannot view these values while you are running an OS/2 operating system.

Appendix C. X Window Resources

Tivoli Management Framework provides a set of X Window resources that you can use to customize your UNIX Tivoli environment. This appendix provides information about Tivoli Enterprise and X Window resources.

Tivoli-specific X Window Resources

The following table contains the Tivoli-specific X Window resources that you can use to customize Tivoli products.

Resource	Description
DmTextField	Text
DmText	Page
DmReadOnlyText	Read Only Page
DmSwitch	Switch
DmGroup	Group
Collection	Collection
DmMessage	Message
DmSome	Choice/List (Show = SOME)
OptionMenu	Choice/List (Show = ONE)
DmToggleGroup	Choice/List (Show = ALL)
ReadOnlyList	Read Only Choice/List (Show = SOME)
SingleChoice	Choice (Show = SOME)
MultipleChoice	List (Show = SOME)

The following example changes the background color of all user-specified text fields to blue:

```
Tivoli*DmTextField*background: blue
```

The following example changes the foreground color of all user-specified switches to yellow:

```
Tivoli*DmSwitch*foreground: yellow
```

X Window Resources Supported by Tivoli Products

You can use the following general X Window resources to customize Tivoli products:

–bd Specifies border color

–bg Specifies background color

–display
Specifies display name

–fg Specifies foreground color

–font Specifies font

–xrm Specifies X Window database resource

–geometry
Specifies geometry of Tivoli desktop

–iconic
Opens the Tivoli desktop as an icon

The following example opens the Tivoli desktop with a gray background, white foreground, on display device ayers-rock:0.0:

```
tivoli -bg grey90 -fg white -display ayers-rock:0.0
```

Note: The ***Help.sensitive:False** X Window database resource setting disables the help windows in your Tivoli environment.

Appendix D. 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

AIX, AS/400, DB2, IBM, MVS, OS/2, OS/400, S/390, Tivoli, Tivoli Enterprise, Tivoli Enterprise Console, TME, TME 10, WebSphere, WIN-OS/2, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see http://www.intel.com/sites/corporate/tradmarx.htm.

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.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Third Party License Terms and Conditions, Notices and Information

The license agreement for this product refers you to details concerning terms and conditions applicable to third party software code included in this product, and for certain notices and other information IBM must provide to you under its license to certain software code. The relevant terms and conditions, notices and other information are provided or referenced below. Please note that any non-English version of the licenses below is unofficial and is provided to you for your convenience only. The English version of the licenses below, provided as part of the English version of this file, is the official version.

Notwithstanding the terms and conditions of any other agreement you may have with IBM or any of its related or affiliated entities (collectively "IBM"), the third party software code identified below are "Excluded Components" and are subject to the following terms and conditions:

(a) the Excluded Components are provided on an "AS IS" basis;

(b) IBM DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS WITH RESPECT TO THE EXCLUDED COMPONENTS, INCLUDING, BUT NOT LIMITED TO, THE WARRANTY OF NON-INFRINGEMENT OR INTERFERENCE AND THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE;

(c) IBM will not be liable to you or indemnify you for any claims related to the Excluded Components; and

(d) IBM will not be liable for any direct, indirect, incidental, special, exemplary, punitive or consequential damages with respect to the Excluded Components.

Excluded Components:

- InstallShield 6.22 Open Install
- XML4J Java XML Parser
- OpenSSL
- Compress
- MD5
- ID-generating algorithm created using the Network Working Group Internet Draft ″UUIDS and GUIDS″

InstallShield 6.22 Open Install

The Program is third party software code identified as an "Excluded Component" and is subject to the terms and conditions as documented in the introduction to this appendix.

XML4J Java XML Parser

The Program includes software developed by Sun Microsystems, Inc. for Project JXTA. The portions of the Program which are based on software developed by Sun Microsystems, Inc. for Project JXTA are Copyright (c) 2001 Sun Microsystems, Inc. All rights reserved. IBM obtained this software under the terms and conditions of the following license from Sun Microsystems, Inc.

The Sun Project JXTA Software License (Based on the Apache Software License Version 1.1)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Sun Microsystems, Inc. for Project JXTA". Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

OpenSSL

OpenSSL Code

The Program includes software developed by The OpenSSL Project (http://www.openssl.org). The portions of the Program which are based on software developed by The Open SSL Project are Copyright (c) 1998-2001. The Open SSL Project. All rights reserved. IBM obtained the Open SSL software under the terms and conditions of the following license from The Open SSL Project:

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc.code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: “This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)” The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: “This product includes software written by Tim Hudson (tjh@cryptsoft.com)”

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Compress

This Program includes ″Compress″ code, which was licensed under the following terms and conditions.

Copyright (c) 1992, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ″AS IS″’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 1996 FreeBSD Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY FreeBSD Inc. AND CONTRIBUTORS ``AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL [your name] OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

$FreeBSD: /repoman/r/ncvs/src/usr.bin/compress/zopen.h,v 1.4.2.1 2002/07/16 00:52:08 tjr Exp $ /

Notice for the MD5 Message-Digest Algorithm

MD5 Message-Digest Algorithm. The Program includes software developed by RSA Data Security, Inc.. The portions of the Program which are based on software developed by RSA Data Security, Inc. are Copyright (c) 1991-2 RSA Data Security, Inc.. All rights reserved. IBM obtained the MD5 Message-Digest Algorithm under the terms and conditions of the following license from RSA Data Security, Inc.:

License to copy and use this software is granted provided that it is identified as the ″RSA Data Security, Inc. MD5 Message-Digest Algorithm″ in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as ″derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm″ in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided ″as is″ without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

Notice for the Network Working Group Internet Draft ″UUIDS AND GUIDS,″ Which was Used to Create an ID-Generating Algorithm

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an ″AS IS″ basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Glossary

A

authorization roles. A role assigned to Tivoli administrators to enable them to perform their assigned systems management tasks. A role can be granted over the entire Tivoli region or over a specific set of resources, such as those contained in a policy region. Examples of authorization roles include **super**, **senior**, **admin**, and **user**.

B

bulletin board. The mechanism by which the Tivoli Management Framework and Tivoli applications communicate with Tivoli administrators. The bulletin board collects notices in notice groups. Administrators can access the bulletin board from the Tivoli desktop. The bulletin board is an audit trail for important operations that the administrators perform.

C

Configuration Change Management System (CCMS). In a Tivoli environment, a data store of profiles that contain confugration data that is used by system management applications to make configuration changes on groups of systems. See also profile manager.

collection. A container that provides a single view of related resources.

configuration repository. The relational database that contains information collected or generated by inventory or software distribution operations.

D

default policy. A set of resource property values assigned to a resource when the resource is created. Contrast with validation policy.

Distribution Status console. An MDist 2 interface provided by Tivoli Management Framework that enables administrators to monitor and control distributions across a network. See also MDist 2.

downcall. A method call from the gateway to an endpoint. Contrast with upcall.

E

endpoint. In a Tivoli environment, the computer system that is the ultimate target for most Tivoli operations. Contrast with managed node.

endpoint list. In a Tivoli environment, A list of all endpoints in a Tivoli region with their assigned gateways. See endpoint manager.

endpoint manager. In a Tivoli environment, a service that runs on the Tivoli server, assigns endpoints to gateways, and maintains the endpoint list.

endpoint method. In a Tivoli environment, a method that runs on an endpoint as the result of a request from another managed resource. Results of the method are forwarded to the gateway and then to the calling managed resource.

F

fanout. In communications, the process of creating copies of a distribution to be delivered locally or sent over the network.

G

gateway. Software that provides services between endpoints and the rest of the Tivoli environment.

I

instance. A single occurrence of a resource.

J

job. In a Tivoli environment, a resource consisting of a task and its preconfigured execution parameters. Among other things, the execution parameters specify the set of hosts on which the job is to run.

L

lcfd. The Tivoli service that is used by an endpoint to communicate with a gateway. Contrast with oserv.

M

managed node. In a Tivoli environment, a computer system where Tivoli Management Framework is installed. Constrast with endpoint.

MDist. A multiplexed distribution service provided by Tivoli Management Framework that enables efficient transfer of data to multiple targets. Another distribution service, MDist 2, provides additional management features. See also MDist 2.

MDist 2. A multiplexed distribution service provided by Tivoli Management Framework that enables efficient transfer of data to multiple targets. Administrators can monitor and control a distribution throughout its life cycle. Another multiplexed distribution service, MDist, lacks these management features. See also Distribution Status console.

member. The contents of a collection. See also collection.

multiplexed distribution. The mechanism used by Tivoli Enterprise applications to transfer data to multiple targets. Tivoli Management Framework provides two multiplexed distribution services, MDist and MDist 2. See also MDist and MDist 2.

N

name registry. See Tivoli name registry.

notice. In a Tivoli environment, a message generated by a systems management operation that contains information about an event or the status of an application. Notices are stored in notice groups. See also bulletin board.

notice groups. In a Tivoli environment, an application- or operation-specific container that stores and displays notices pertaining to specific Tivoli functions. The Tivoli bulletin board is comprised of notice groups. A Tivoli administrator can subscribe to one or more notice groups; the a bulletin board contains only the notices that reside in a notice group to which the administrator is subscribed. See also bulletin board and notice.

NT repeater. The first Windows managed node where Tivoli Remote Execution Service is installed. Using fanout, the NT repeater distributes Tivoli Remote Execution Service to all other Windows managed nodes during the installation process.

O

object path. In a Tivoli environment, an absolute or relative path to a Tivoli object, similar to a path in a file system.

object reference. In a Tivoli environment, the object identifier (OID) that is given to an object during its creation.

object request broker (ORB). In object-oriented programming, software that serves as an intermediary by transparently enabling objects to exchange requests and responses.

oserv. The Tivoli service that is used as the object request broker (ORB). This service runs on each Tivoli server and managed node. Contrast with lcfd.

P

policy. A set of rules that are applied to managed resources. A specific rule in a policy is referred to as a policy method.

policy region. A group of managed resources that share one or more common policies and which model the management and organizational structure of a network computing environment. Administrators use policy regions to group similar resources, to define access to the resources, and to associate rules for governing the resources.

policy subregion. In a Tivoli environment, a policy region created or residing in another policy region. When a policy subregion is created, it initially used the resource and policy properties of the parent policy region. A Tivoli administrator can later change or customize these properties to reflect the specific needs and differences of the subregion.

profile. In a Tivoli environment, a container for application-specific information about a particular type of resource. A Tivoli application specifies the template for its profiles, which includes information about the resources that can be managed by that Tivoli application.

profile manager. In a Tivoli environment, a container for profiles that links the profiles to a set of resources, called subscribers. Tivoli administrators use profile managers to organize and distribute profiles. A profile manager can operate in dataless mode or database mode. created in the context of a policy region and is a managed resource in that policy region. See also Configuration Change Management System (CCMS).

proxy endpoint. In a Tivoli environment, a representation of an entity (such as a network device or a host) that functions as a subscriber for profile distribution. The proxy endpoint is created on a managed node, which performs the proxy role during profile distribution. Multiple proxy endpoints can be created on the same managed node.

pull. An operation that initiates an action by requesting it of a resource.

push. An operation that sends information to other resources.

Q

query. In a Tivoli environment, a combination of statements that are used to search the configuration repository for systems that meet certain criteria. The query object is created within a query library. See also query library.

query facility. In a Tivoli environment, a facility that enables the use of SQL functions to access information in an RDBMS Interface Module (RIM) repository.

query library. In a Tivoli environment, a facility that provides a way to create and manage Tivoli queries. See also query.

R

RDBMS. See relational database management system (RDBMS).

RDBMS Interface Module (RIM). In Tivoli Management Framework, the module in the distributed object database that contains information about the installation of the relational database management system (RDBMS).

region. See Tivoli region.

registered name. In a Tivoli environment, the name by which a particular resource is registered with the name registry when the resource is created.

relational database management system (RDBMS). A collection of hardware and software that organizes and provides access to a relational database.

repeater. In a Tivoli environment, a managed node that is configured for multiplexed distribution. A repeater receives a single copy of data and distributes it to the next tier of clients.

resource. A hardware, software, or data entity that is managed by Tivoli software products.

resource type. In a Tivoli environment, one of the properties of a managed resource. Resource types are defined in the default policy for a policy region.

RIM. See RDBMS Interface Module (RIM).

RIM repository. In a Tivoli environment, the relational database that contains information that is collected or generated by Tivoli software products.Examples of a RIM repository include the configuration repository and the event database.

root administrator. In a Tivoli environment, the account for the initial Tivoli administrator that is created during the installation of Tivoli Management Framework. This account is the equivalent of the root user on UNIX operating systems and a member of the administrator group on Microsoft Windows systems.

S

subscriber. In a Tivoli environment, a resource that is subscribed to a profile manager.

subscription. In a Tivoli environment, the process of identifying the subscribers to which profiles are distributed.

subscription list. In a Tivoli environment, a list that identifies the subscribers to a profile manager. A profile manager can be included in a subscription list to subscribe several resources simultaneously rather than adding each resource individually.

T

TAP. See Tivoli Authentication Package.

task. In a Tivoli environment, the definition of an action that must be routinely performed on various targets throughout the network. A task defines the executable files to be run when the task is initiated, the authorization role required to execute the task, and the user or group name under which the task runs.

task library. In a Tivoli environment, a container in which a Tivoli administrator can create and store tasks and jobs.

Tivoli administrator. In a Tivoli environment, a system administrator that is identified by system account maps who is authorized to perform systems management tasks and manage policy regions in one or more networks.

Tivoli Application Development Environment (ADE). A toolkit that contains the complete application programming interface (API) for Tivoli Management Framework. This toolkit enables customers and Tivoli Partners to develop their own applications for a Tivoli environment.

Tivoli Application Extension Facility (AEF). A toolkit that enables customers to extend the capabilities of Tivoli applications. For example, they can add fields to a dialog, create custom attributes and methods for application resources, or create custom icons and bitmaps.

Tivoli Authentication Package. A dynamically linked library (DLL) installed by Tivoli Management Framework, that is capable of creating Windows security tokens for a different user context. These tokens can be used for accessing network resources or creating processes in a different user context.

Tivoli client. A client of a Tivoli server. See Tivoli management region client (managed node) and Tivoli server.

Tivoli desktop. In a Tivoli environment, the desktop that system administrators use to manage their network computing environments.

Tivoli Event Integration Facility (EIF). A toolkit that provides a simple application programming interface (API) to enable customers and Tivoli Partners to develop new event adapters that can forward events. A customer can also translate events from third-party or in-house applications.

Tivoli environment. The Tivoli applications, based upon Tivoli Management Framework, that are installed at a specific customer location and that address network computing management issues across many platforms.

Tivoli Management Framework. The base software required to run many Tivoli software applications. This software infrastructure enables the integration of systems management applications from Tivoli and the Tivoli Partners. Tivoli Management Framework includes the following components:

- Object request broker (oserv service)
- Distributed object database
- Basic administration functions
- Basic application services
- Basic desktop services, such as the graphical user interface (GUI)

Tivoli client (managed node). In a Tivoli environment, any computer system—except the Tivoli server—on which Tivoli Management Framework is installed. The object dispatcher (or oserv service) runs on each client, and each client maintains a local object database. See Tivoli server.

Tivoli name registry. In a Tivoli environment, the table that map names of managed resources to resource identifiers (and the corresponding information) with a Tivoli region.

Tivoli region. In a Tivoli environment, a Tivoli server and the set of clients it serves. An organization can have more than one region. A Tivoli region addresses the physical connectivity of resources whereas a policy region addresses the local organization of resources.

Tivoli Remote Execution Service. The service that enables a Tivoli environment to perform remote operations on machines. These operations include remotely installing clients, connecting Tivoli management regions, and starting the object request broker from a remote machine.

Tivoli server. In a Tivoli environment, the server for a specific Tivoli region that holds or references the complete set of Tivoli software, including the full object database.

transaction. A specific set of input data that triggers a specific process or job.

U

upcall. A method invocation from an endpoint to the gateway. Contrast with downcall.

user login map. A mapping that associates a single user login name with user accounts on various operating systems. User login maps enable Tivoli administrators to log in to the Tivoli environment and perform operations within the Tivoli environment with a single user login name, independent of system accounts used on the various operating systems.

V

validation policy. In a Tivoli environment, the policy that ensures that all resources in a policy region comply with the established policy for the policy region. A validation policy prevents Tivoli administrators from creating or modifying resources that do not conform to the policy of the policy region in which the resouces were created. Contrast with default policy.

virtual user. A user ID (UID) mapping set up in Tivoli Management Framework. A single UID can be mapped to different actual users on different types of architectures. For example, the virtual user $root_user can be mapped to root on UNIX operating systems and **Administrator** on Windows operating systems. See user login map.

Index

Part Number: CT2JTIE

Printed in USA

(1P) P/N: CT2JTIE

GC32-0804-02