IBM Tivoli Monitoring
Version 6.2.3 Fix Pack 1

Installation and Setup Guide

GC32-9407-05

IBM Tivoli Monitoring

Version 6.2.3 Fix Pack 1

Installation and Setup Guide

GC32-9407-05

Note

Before using this information and the product it supports, read the information in Appendix L, “Notices,” on page 879.

This edition applies to version 6, release 2, modification 3 of IBM Tivoli Monitoring (product number 5724-C04) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Part 2. Planning your IBM Tivoli Monitoring deployment 31

Part 6. Integrating event management systems 641

Figures

Tables

Part 1. Introduction

The single chapter in this section, Chapter 1, “Overview of IBM Tivoli Monitoring,” on page 3, describes the architecture of the IBM® Tivoli® Monitoring products and provides information to help you plan your deployment and prepare to install, upgrade, or configure the product's base components.

Tivoli Monitoring products use a set of service components (known collectively as Tivoli Management Services) that are shared by a number of other product suites, including IBM Tivoli OMEGAMON® XE monitoring products, IBM Tivoli Composite Application Manager products, System Automation for z/OS®, Web Access for Information Management, and others. The information in this section is also relevant to these products.

Chapter 1. Overview of IBM Tivoli Monitoring

IBM Tivoli Monitoring products monitor the performance and availability of distributed operating systems and applications. These products are based on a set of common service components, referred to collectively as Tivoli Management Services. Tivoli Management Services components provide security, data transfer and storage, notification mechanisms, user interface presentation, and communication services in an agent-server-client architecture (Figure 1 on page 4). These services are shared by a number of other products, including IBM Tivoli OMEGAMON XE mainframe monitoring products and IBM Tivoli Composite Application Manager products, as well as other IBM Tivoli Monitoring products such as Tivoli Monitoring for Applications, Tivoli Monitoring for Business Integration, Tivoli Monitoring for Cluster Managers, Tivoli Monitoring for Databases, Tivoli Monitoring for Energy Management, Tivoli Monitoring for Messaging and Collaboration, Tivoli Monitoring for Microsoft .NET, Tivoli Monitoring for Microsoft Applications, Tivoli Monitoring for Transaction Performance, Tivoli Monitoring for Virtual Servers, and Tivoli Monitoring for Web Infrastructure.

This book contains information on deploying, installing, and configuring the common services components and the monitoring agents that comprise the base IBM Tivoli Monitoring product (Table 1) on distributed systems. If you purchased a product other than IBM Tivoli Monitoring that uses Tivoli Management Services, use this book to install and configure the common components. Do not install the base agents unless you have also licensed IBM Tivoli Monitoring. If you purchased additional IBM Tivoli Monitoring products, see their documentation for agent-specific installation and configuration information.

Table 1. IBM Tivoli Monitoring base monitoring agents

IBM Tivoli Monitoring 5.x Endpoint
Linux OS
UNIX Logs
UNIX OS
Windows OS
i5/OS®
IBM Tivoli Universal Agent
Warehouse Proxy
Summarization and Pruning
IBM Tivoli Performance Analyzer

IBM Tivoli Monitoring V6.2 products (including V6.2.3) are enabled for use with IBM Tivoli License Manager. Tivoli Management Services components and Tivoli Monitoring agents provide inventory signature files and usage definitions that allow License Manager to report installed products and product usage by computer. License Manager support is an optional capability that requires License Manager version 2.2 or later.

Components of the monitoring architecture

This section describes the architecture of the IBM Tivoli Monitoring products and provides information to help you plan your deployment and prepare to install or upgrade the base components of the product.

Tivoli Monitoring products use a set of service components (known collectively as Tivoli Management Services) that are shared by a number of other product suites, including IBM Tivoli OMEGAMON XE monitoring products, IBM Tivoli Composite Application Manager products, System Automation for z/OS, Web Access for Information Management, and others. The information in this section is also relevant to these products.

Tivoli Monitoring products, and other products that share Tivoli Management Services, participate in a server-client-agent architecture. Monitoring agents for various operating systems, subsystems, databases, and applications (known collectively as Tivoli Enterprise Monitoring Agents) collect data and send it to a Tivoli Enterprise Monitoring Server. Data is accessed from the monitoring server by Tivoli Enterprise Portal clients. A Tivoli Enterprise Portal Server provides presentation and communication services for these clients. Several optional components such as an historical data warehouse extend the functionality of the framework. Figure 1 shows the configuration of an IBM Tivoli Monitoring environment.

Before deciding where to deploy the components of the Tivoli Monitoring product in your environment, you should understand the components of the product, the roles that they play, and what affects the load on these components.

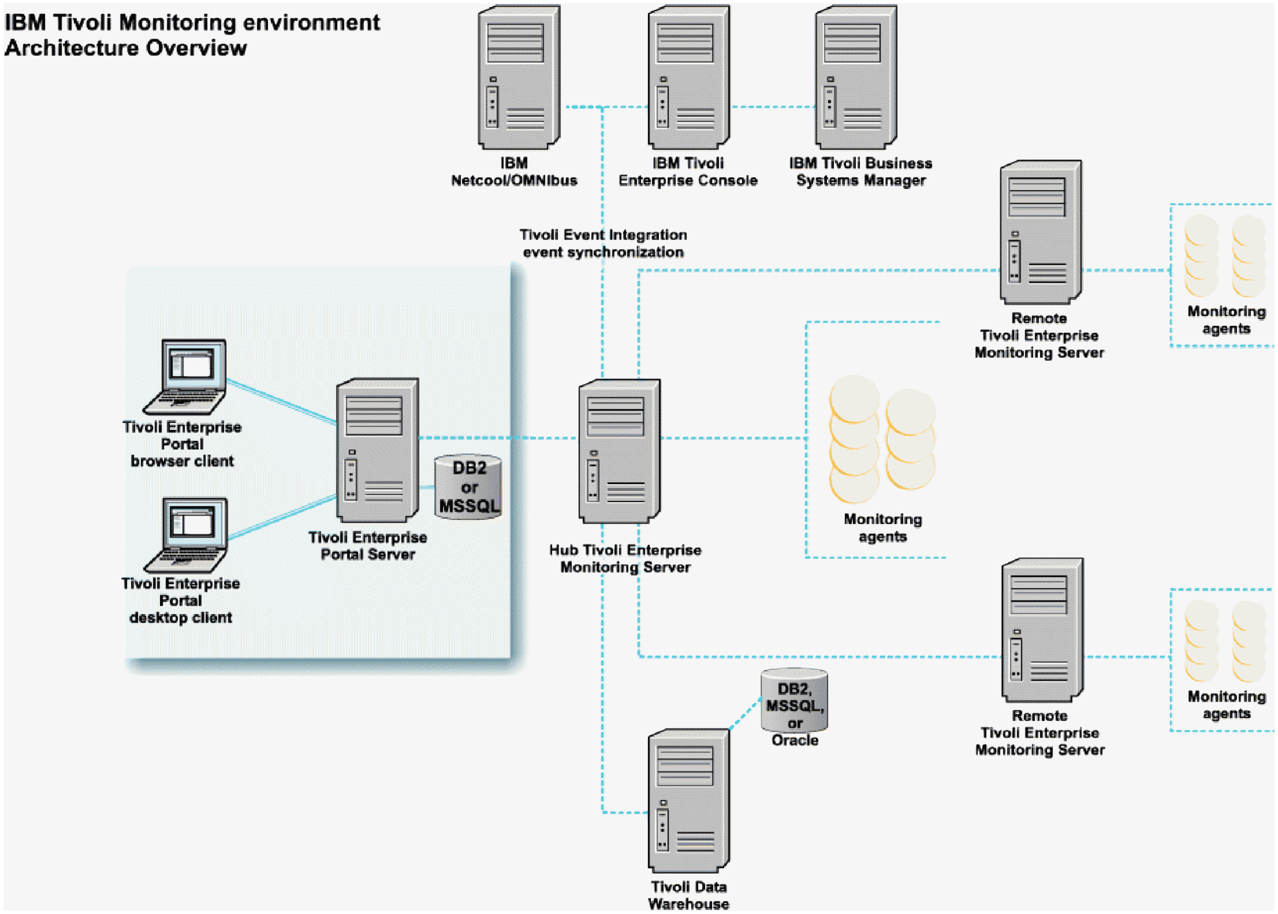


Figure 1. IBM Tivoli Monitoring environment

A typical IBM Tivoli Monitoring environment comprises the following components:

- One or more Tivoli Enterprise Monitoring Servers, which act as a collection and control point for alerts received from the agents, and collect their performance and availability data. The monitoring server also manages the connection status of the agents. One server in each environment must be designated as the *hub*.
- A Tivoli Enterprise Portal Server, which provides the core presentation layer for retrieval, manipulation, analysis, and pre-formatting of data. The portal server retrieves data from the hub monitoring server in response to user actions at the portal client, and sends the data back to the portal client for presentation. The portal server also provides presentation information to the portal client so that it can render the user interface views suitably.

- One or more Tivoli Enterprise Portal clients, with a Java-based user interface for viewing and monitoring your enterprise. Tivoli Enterprise Portal offers two modes of operation: desktop and browser.
- Tivoli Enterprise Monitoring Agents, installed on the systems or subsystems you want to monitor. These agents collect data from monitored, or managed, systems and distribute this information either to a monitoring server or to an SNMP Event Collector such as IBM Tivoli Netcool/OMNIbus.
- z/OS only: Tivoli Management Services:Engine (TMS:Engine) provides common functions, such as communications, multithreaded runtime services, diagnosis (dumps), and logging (RKLVLOG), for the Tivoli Enterprise Monitoring Server, monitoring agents, and OMEGAMON components of OMEGAMON XE products running on z/OS.
- An Eclipse Help Server for presenting help for the portal and all monitoring agents for which support has been installed.

An installation optionally includes the following components:

- Tivoli Data Warehouse for storing historical data collected from agents in your environment. The data warehouse is located on an IBM DB2 for Linux, UNIX, and Windows, DB2 on z/OS, Oracle, or Microsoft SQL database. To store data in this database, you must install the Warehouse Proxy Agent. To perform aggregation and pruning functions on the data, you must also install the Summarization and Pruning Agent.
- Event synchronization component, the Event Integration Facility, that sends updates to situation events that have been forwarded to a Tivoli Enterprise Console® event server or a Netcool/OMNIbus ObjectServer back to the monitoring server.
- Tivoli Performance Analyzer for predictive capability with Tivoli Monitoring so you can monitor resource consumption trends, anticipate future performance issues, and avoid or resolve problems more quickly.

The following sections describe each of these components in more detail.

Tivoli Enterprise Monitoring Server

The Tivoli Enterprise Monitoring Server (referred to as the *monitoring server*) is the first component to install to begin building the Tivoli Management Services foundation. The monitoring server is the key component on which all other architectural components directly depend.

The monitoring server is the collection and control point for performance and availability data and alerts received from monitoring agents. It is also responsible for tracking the online or offline status of monitoring agents.

Because of the number of functions the monitoring server performs, large-scale environments usually include a number of monitoring servers to distribute the load. One of the monitoring servers is designated the hub monitoring server, and the remaining servers are termed remote monitoring servers. Each remote monitoring server must be located on its own computer and have a unique monitoring server name (node), but the architectures of various remote monitoring servers might differ from each other and from the hub monitoring server. In other words, a remote monitoring server running on UNIX can report to a hub monitoring server running on Windows.

The portal server communicates with the hub, which in turn controls the remote servers, as well as any monitoring agents that might be connected to it directly.

The monitoring server storage repository is a proprietary database format (referred to as the Enterprise Information Base, or EIB). The hub holds the master copy of the EIB, while the remote servers maintain a subset of the EIB relevant to them, which is synchronized with the hub.

Tivoli Enterprise Portal

Tivoli Enterprise Portal is the interface to your monitoring products. The Tivoli Enterprise Portal consists of the Tivoli Enterprise Portal Server and one or more clients.

The Tivoli Enterprise Portal Server (referred to as the *portal server*) manages data access through user workspace consoles (the portal clients). The portal server connects to a hub monitoring server; it retrieves data from the hub in response to user actions at a portal client, and sends the data back to the portal client for presentation. The portal server also provides presentation information to the portal client so that it can render the user interface views suitably.

The portal server uses a DB2 for Linux, UNIX, and Windows or Microsoft SQL database to store various artifacts related to presentation at the portal client.

The portal client provides access to the Tivoli Enterprise Portal. There are two kinds of portal client:

- Browser client interface (automatically installed with Tivoli Enterprise Portal Server): The browser client can be run using Microsoft Internet Explorer or Mozilla Firefox; it connects to a Web server running in the Tivoli Enterprise Portal Server. Running the browser client is supported only on Windows and Linux computers.
- Desktop client interface: A Java-based graphical user interface on a Windows or Linux workstation. After the desktop client is installed and configured, you can use it to start Tivoli Enterprise Portal in desktop mode. You can also download and run the desktop client using Java Web Start, as discussed in "Java Web Start clients" on page 303.

See "Configuring clients, browsers, and JREs" on page 296 for a discussion of the comparative advantages of each type of portal client.

Tivoli Enterprise Monitoring Agents

Monitoring agents are data collectors. Agents monitor systems, subsystems, or applications, collect data, and pass the data to Tivoli Enterprise Portal through the monitoring server. The agents pass commands from the user to the system, subsystem, or application. An agent interacts with a single system or application and, in most cases, is located on the same computer where the system or application is running.

There are two types of monitoring agents:

- Operating system (OS) agents that monitor the availability and performance of the computers in your monitoring environment. An example of an OS agent is the Monitoring Agent for Windows OS, which monitors Windows XP, Windows 2000, and Windows 2003 operating systems.

  As of version 6.2.1, a special type of operating system agent, the agentless monitor, is available. It enables a remote node to monitor the health of nonessential desktop operating systems via a standard monitoring API such as SNMP and thus is also called a remote OS agent.

  In addition, as of Tivoli Monitoring V6.2.2, there is another class of operating-system agents, the System Monitor Agent. These lighter-weight agents (they require a much smaller footprint than full-function Tivoli Monitoring OS agents) are configured locally to the agent node. This configuration enables them to be deployed autonomously (in other words, without the support of a Tivoli Enterprise Monitoring Server): they send SNMP event information directly to an SNMP Event Collector such as IBM Tivoli Netcool/OMNIbus. The System Monitor Agents are meant as a replacement for the OMNIbus System Service Monitor agents.
- Other agents (referred to as *application agents* or *non-OS agents*) that monitor the availability and performance of systems, subsystems, and applications. An example of a non-OS agent is IBM Tivoli Monitoring for Microsoft Exchange, which monitors the Microsoft Exchange Server.

IBM Tivoli Monitoring also provides a customizable agent called the Tivoli Universal Agent. You can use this agent to monitor most types of data that you can collect in your environment. For example, you can use it to monitor the status of your company's Web site to ensure it is available.

The Performance Analyzer Warehouse Agent adds predictive capability to Tivoli Monitoring so you can monitor resource consumption trends, anticipate future performance issues, and avoid or resolve problems more quickly. For more information about the Performance Analyzer Warehouse Agent, see the *IBM Tivoli Performance Analyzer Installation Guide*.

You can also create your own IBM Tivoli Monitoring agent via the Agent Builder, a set of tools for creating agents and adding value to existing agents. Using the Agent Builder, you can quickly create, modify, and test an agent to collect and analyze data about the state and performance of different resources, such as disks, memory, CPU, or applications. The builder creates a data provider that allows you to monitor three types of data:

Availability
Process and service availability and functionality tests

Windows event log
Specific information from the Windows Event Log

External data sources
Data from external sources such as Windows Management Instrumentation (WMI), Performance Monitor (PerfMon), Simple Network Management Protocol Version 1 (SNMP V1), external scripts, and log files

With the Agent Builder's customizable Graphical User Interface installer, you can create agent-installation packages for easy agent distribution. A key feature of the installer is its ability to package and distribute extensions to existing agents. This lets you develop new situations, queries, and workspaces for an existing IBM Tivoli Monitoring V6.x agent. For complete information about the Agent Builder, see the *IBM Tivoli Monitoring: Agent Builder User's Guide*. Note that the Agent Builder is provided on its own distribution CD for IBM Tivoli Monitoring version 6.2.1.

In most cases the recommended choice for customized agents is the Agent Builder.

The IBM Tivoli Monitoring Infrastructure DVD (in addition to providing the Tivoli Enterprise Monitoring Server and its application support, the Tivoli Enterprise Portal Server and its application support, and the Tivoli Enterprise Portal desktop and browser clients together with their application support) also contains the Tivoli Data Warehouse agents: the Warehouse Proxy Agent and the Summarization and Pruning Agent, and the Tivoli Performance Analyzer. This DVD is platform-specific (Windows, Linux, or UNIX). Use the Agents DVD to install the monitoring agents in the following list (as well as application support for agentless monitoring). Note that this DVD, however, is platform-nonspecific; that is, it applies to Windows, Linux, and UNIX environments.

i5/OS
Windows OS
Linux OS
UNIX OS
UNIX Logs
IBM Tivoli Universal Agent

See “Selecting the correct support media” on page 268 for more information.

Note: For z/OS customers, IBM also provides a family of OMEGAMON Monitoring Agents that monitor both the z/OS operating system (as well as its key subsystems: VTAM®, CICS®, IMS™, DB2®, and storage subsystems) and the z/VM® operating system (as well as any Linux guests running under it). A complete suite of OMEGAMON product documentation is provided in the IBM Tivoli Monitoring information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/.

Tivoli Data Warehouse

With Tivoli Data Warehouse, you can analyze historical trends from monitoring agents. The Tivoli Data Warehouse uses a DB2 for Linux, UNIX, and Windows, DB2 on z/OS, Oracle, or Microsoft SQL Server database to store historical data collected across your environment. You can generate warehouse reports for short-term or long-term data through the Tivoli Enterprise Portal. Warehouse reports provide information about the availability and performance of your monitoring environment over a period of time. You can also use third-party warehouse reporting software, such as Crystal Reports or Brio, to generate reports.

Three specialized agents interact with the Tivoli Data Warehouse:

- The Warehouse Proxy Agent receives data collected by monitoring agents and moves it to the Tivoli Data Warehouse database.
- The Summarization and Pruning Agent provides the ability to customize the length of time for which to save data (pruning) and how often to aggregate granular data (summarization) in the Tivoli Data Warehouse database.
- The Tivoli Performance Analyzer extends the capability of Tivoli Monitoring by analyzing and enriching the data that is collected by its monitoring agents and by providing reports about the performance and capacity of your systems.

The Warehouse Proxy Agent, the Summarization and Pruning Agent, and the Tivoli Performance Analyzer, and support for those agents, are installed from the Infrastructure DVD.

See Part 5, “Setting up data warehousing,” on page 463 for information about setting up data warehousing.

Event synchronization component

The event synchronization component, the Event Integration Facility or EIF, sends updates to situation events that have been forwarded to a Tivoli Enterprise Console event server or a Netcool/OMNIbus ObjectServer back to the monitoring server. Figure 2 on page 9 shows the flow of situation events from the monitoring server to the event server as EIF events and the flow of updates to the situation events back to the monitoring server. The Situation Event Console, the Common Event Console, and the Tivoli Enterprise Console event views are synchronized with the updated status of the events.

If you are monitoring event data from a supported event management system in the Tivoli Enterprise Console event view or the Common Event Console view, you can filter out forwarded events. See the *IBM Tivoli Monitoring: Administrator's Guide*.

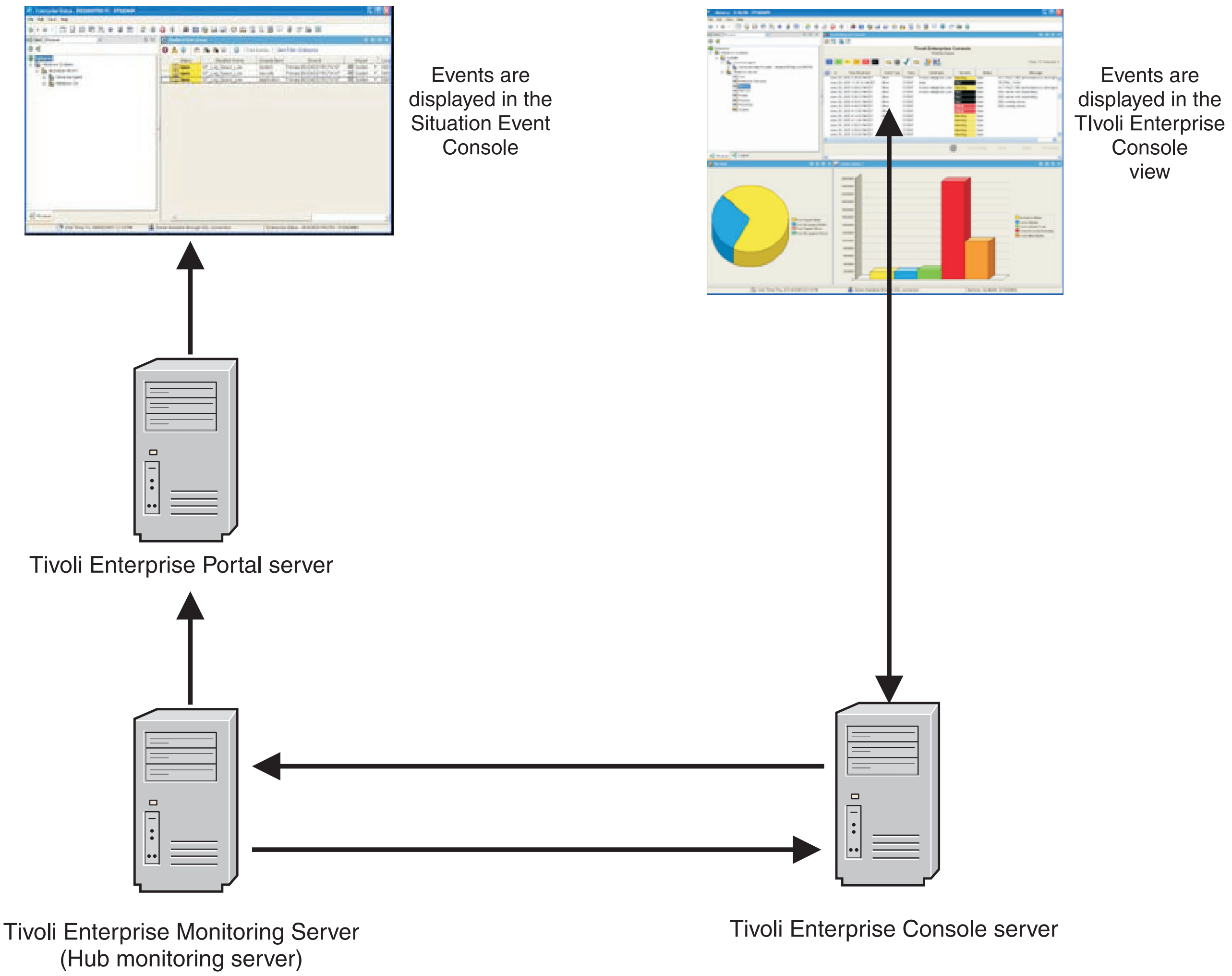

Events are displayed in the Situation Event Console
Events are displayed in the TIvoli Enterprise Console view
Tivoli Enterprise Portal server
Tivoli Enterprise Monitoring Server (Hub monitoring server)
Tivoli Enterprise Console server

Figure 2. Event synchronization overview

For information about the various configurations of monitoring servers and event servers that you can have in your environment, see Part 6, “Integrating event management systems,” on page 641.

Tivoli Enterprise Portal Server extended services

Tivoli Enterprise Portal Server extended services (TEPS/e) is an embedded, co-located extension of the Tivoli Enterprise Portal Server that provides J2EE-based Application Server integration facilities. TEPS/e supports a federated user repository such as those based on the Lightweight Directory Access Protocol (LDAP). For more information, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Tivoli Performance Analyzer

Tivoli Performance Analyzer adds predictive capability to Tivoli Monitoring so you can monitor resource consumption trends, anticipate future performance issues, and avoid or resolve problems more quickly. For example, you can use Tivoli Performance Analyzer to predict application bottlenecks and create alerts for potential service threats.

Tivoli Performance Analyzer helps IT managers to answer the following questions so they can optimize IT capacity:

- When will my application fail to meet service levels?
- How will application performance change if I modify the infrastructure?

- What is the best hardware solution to meet my performance and cost goals?
- Where are my under-utilized servers and networks?
- Which servers and network components really need an upgrade?
- Which application will experience the next performance issue? When?

Providing accurate IT forecasts and appropriate IT capacity to meet business needs are two major goals in the capacity management area. You can use the following key performance indicators (KPIs) to measure the critical success factors:

- Total value of unplanned or unused capacity expenditures
- Percent of capacity forecasts that were accurate
- Number of inaccurate business forecast inputs provided
- Number of incidents related to capacity or performance issues

Capacity management

Capacity management is a forward looking process that aims to align system capacity to the current and future demand in a cost-efficient manner. Within capacity management, performance analysis is a key activity. It requires tools and methods that predict how IT resources will behave in the near and mid term so you can avoid incidents and problems rather than have to react to them.

Tivoli Performance Analyzer extends the capability of Tivoli Monitoring by analyzing and enriching the data that is collected by its monitoring agents and by providing reports about the performance and capacity of your systems. Tivoli Performance Analyzer performs the following functions for individual IT components:

- Gathers and stores IT components such as CPU, disk, and memory utilization in a central data repository
- Provides a predictive analysis component that indicates trends in IT component utilization
- Retains the analyzed data in a central repository for reporting purposes and for input to the component management process

New in release 6.2

The following sections describe changes in this release that affect installation or configuration. For a complete list of new product features, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Note: Some of these features were introduced in fix packs or as special solutions posted on the IBM Tivoli Integrated Service Management Library Web site.

The following items are new in IBM Tivoli Monitoring V6.2:

- “Changes to installation media” on page 12
- “Changes to runtime prerequisites and platform support” on page 13
- “Changes to the Tivoli Data Warehouse” on page 13
- “Authentication using LDAP” on page 13
- “Help for the Tivoli Enterprise Portal presented by the Eclipse Help Server” on page 14
- “Event forwarding and synchronization for Tivoli Enterprise Console and Netcool/OMNIbus” on page 14
- “Support for /3GB boot option on 32-bit Windows” on page 14
- “Common Event Console view for events from multiple event servers” on page 14
- “Remote installation of application support files using Manage Tivoli Enterprise Monitoring Services on Linux” on page 14
- “Flexible scheduling of Summarization and Pruning Agent” on page 14
- “Validation of monitoring server protocols and standby configuration” on page 15

- “Support for License Manager” on page 15
- “Support for UNIX agents in Solaris local zones” on page 15
- “New managed system groups offer advanced features over managed system lists” on page 15

The following items are new in IBM Tivoli Monitoring V6.2 Fix Pack 1:

- “Base DVD split for Fix Pack 1” on page 15
- “Support for Sun Java Runtime Environment” on page 15
- “Support for the browser client on Linux” on page 16
- “Support for single sign-on for launch to and from other Tivoli applications” on page 16

The following items are new in Tivoli Monitoring V6.2.1:

- “Reconfigured product media” on page 16
- “New IBM Tivoli Monitoring High-Availability Guide provides resiliency information and instructions” on page 17
- “IPv6 communications protocol now fully supported” on page 17
- “RedHat Enterprise Linux 2.1 no longer supported on Intel platforms” on page 17
- “Asynchronous remote agent deployment and group deployment now supported” on page 17
- “Linux/UNIX users: 64-bit Tivoli Enterprise Portal Server now supported” on page 18
- “Support for 64-bit DB2 for Linux, UNIX, and Windows” on page 18
- “Tivoli Data Warehouse now supports DB2 on z/OS” on page 18
- “New schema publication tool simplifies generation of SQL statements needed to create the Tivoli Data Warehouse” on page 18
- “Tivoli Data Warehouse support for Solaris environments” on page 18
- “Agentless monitoring of distributed operating systems now supported” on page 18
- “The tacmd createNode command need no longer be executed on the monitoring server node” on page 18
- “Support for multiple remote Tivoli Enterprise Monitoring Servers on one Linux or UNIX computer” on page 18

The following items are new in Tivoli Monitoring V6.2.2:

- “Contents of the ***Deployment Guide*** merged” on page 19
- “Additional online user information while the Windows installer is running” on page 19
- “Embedded Java Runtime Environment now supported for Windows sites” on page 21
- “New installation process for language packs” on page 21
- “New System Monitor Agents provide autonomous-only monitoring of your operating system” on page 21
- “Derby now supported as a portal server database” on page 22
- “More memory required to install and run the portal server” on page 22
- “Common agent environment variables listed” on page 22
- “New Tivoli Enterprise Services User Interface Extensions” on page 22
- “Improved user control of agent or server restart after reconfiguration” on page 23
- “Dynamic affinity affects agent coexistence with prior releases” on page 23
- “New installation parameters protect your customized configuration settings” on page 24
- “Higher versions of the Firefox browser supported for Windows customers” on page 24
- “Simplified operating system selection for Linux and UNIX systems” on page 24
- “New installation option allows you to retain your customized seeding files” on page 24
- “Automatic installation of application support for Linux/UNIX monitoring servers” on page 24

- “New silent-response files simplify agent installations and updates” on page 24
- “Remote-deployment support extended to non-agent bundles” on page 25
- “Event integration of IBM Tivoli Monitoring with both IBM Tivoli Business Service Manager and Netcool/OMNIbus now supported” on page 25
- “OMEGAMON data warehouse migration tool no longer provided” on page 25

The following items are new in Tivoli Monitoring V6.2.2 fix pack 1:

- “Native 64-bit operating system agents available for 64-bit Windows environments” on page 25
- “Event forwarding by autonomous agents” on page 25
- “Support for DB2 Database for Linux, UNIX, and Windows version 9.7” on page 25

The following items are new in Tivoli Monitoring V6.2.2 fix pack 2:

- “64-bit System Monitor Agent now supported for Windows environments” on page 26
- “Autonomous operation of the Warehouse Proxy Agent and the Summarization and Pruning Agent” on page 26
- “32-bit DB2 for Linux, UNIX, and Windows no longer required for the Tivoli Enterprise Portal Server” on page 26
- “Further enhancements to the autostart scripts” on page 26
- “Procedure for taking a snapshot of your Tivoli Enterprise Portal Server configuration settings now documented” on page 26
- “Procedure for populating the data warehouse's ManagedSystem table now documented” on page 26

The following items are new in Tivoli Monitoring V6.2.3:

- “New 64-bit Warehouse Proxy Agent simplifies Tivoli Data Warehouse setup for Windows sites” on page 26
- “New prerequisite checking for IBM Tivoli Monitoring Agents” on page 26
- “Self-describing monitoring agents” on page 27
- “New Tivoli Monitoring Startup Center” on page 27
- “Tivoli Performance Analyzer integrated as a base component of IBM Tivoli Monitoring” on page 27
- “DB2 agent and Oracle agent integrated as optional components of IBM Tivoli Monitoring” on page 27
- “Changes to event integration with Netcool/OMNIbus” on page 28
- “Dynamic affinity affects agent coexistence with earlier releases” on page 28
- “The IBM HTTP Server is the default web server used for communication between the portal client and server” on page 28
- “The Portal client can communicate with the portal server by using only the HTTP or HTTPS protocol” on page 28
- “New performance-tuning information” on page 28

Changes to installation media

IBM Tivoli Monitoring distributed components and base agents are now supplied on two DVDs:

- *IBM Tivoli Monitoring V6.2.3 Base DVD*. This DVD contains the IBM Tivoli Monitoring base product components, the operating system agents for both Windows and UNIX, as well as application support for the operating system agents and a selected set of IBM Tivoli Monitoring 6.x based distributed monitoring agents.
- *IBM Tivoli Monitoring Tools DVD*. This DVD contains the IBM Tivoli Monitoring 5.1.2 Migration Toolkit, the Distributed Monitoring Upgrade Toolkit, the Agent Builder, and the Tivoli Event Integration event synchronization component.

Application support for additional V6.x-based distributed monitoring agents is provided on three DVDs:

- *IBM Tivoli Monitoring V6.2: Agent Support for Tivoli Enterprise Portal Server on AIX®*
- *IBM Tivoli Monitoring V6.2: Agent Support for Tivoli Enterprise Management Server on HP-UX on Itanium*
- *IBM Tivoli Monitoring V6.2: Agent Support for Tivoli Enterprise Portal Server for DB2 9.1*

CDs are still available by request. See “Installing and enabling application support” on page 266 for information on the products for which support is available on the Base DVD and the three Agent Support DVDs. The **IBM Tivoli Monitoring V6.2.0 Quick Start CD** (C00YPML) contains a guide to help you get started with IBM Tivoli Monitoring.

Language support for the agents on the Base DVD is provided on the following DVDs:

- *IBM Tivoli Monitoring V6.2 Language Support 1 of 3*. This DVD contains the national language versions of the help and presentation files for the following languages: French, German, Italian, Portuguese Brazilian, Spanish.
- *IBM Tivoli Monitoring V6.2 Language Support 2 of 3*. This DVD contains the language support for the following languages: Japanese, Korean, Simplified Chinese, Traditional Chinese.
- *IBM Tivoli Monitoring V6.2 Language Support 3 of 3*. This DVD contains the language support for the following languages: Czech, Hungarian, Polish, Russian

Language support for the Tools DVD is provided on the following DVDs:

- *IBM Tivoli Monitoring V6.2 DM Upgrade Toolkit Language Support*
- *IBM Tivoli Monitoring V6.2 ITM 5.1.2 Migration Toolkit Language Support*
- *IBM Tivoli Monitoring V6.2 Agent Builder Toolkit Language Support*

Note: IBM reminds sites to ensure all their product media are at the current release before beginning the installation process. Installation of back-level agents and other IBM Tivoli Monitoring components alongside a current Tivoli Enterprise Monitoring Server or Tivoli Enterprise Portal Server can introduce errors.

Changes to runtime prerequisites and platform support

Runtime prerequisites have changed in several environments, especially AIX. Support for some platforms has been added, and support for others dropped. Carefully review the information in “Hardware and software requirements” on page 138, especially the table footnotes, for V6.2 requirements and support.

Changes to the Tivoli Data Warehouse

For a list of changes to the Tivoli Data Warehouse, see “New in Version 6.2.3” on page 465. If you are upgrading from IBM Tivoli Monitoring V6.1, and you use the Tivoli Data Warehouse, you must migrate warehoused data. See “Upgrading the warehouse” on page 169.

Authentication using LDAP

Support has been added to enable external authentication of Tivoli Enterprise Portal users with standards-based Lightweight Directory Access Protocol (LDAP) to shared registries. The hub monitoring server can now be configured to validate user IDs and passwords using either the local system registry or a central LDAP authentication and authorization system. This enhancement permits the sharing of user authentication information among products. For more information, see “Security options” on page 136 and the *IBM Tivoli Monitoring: Administrator's Guide*. See also “Support for single sign-on for launch to and from other Tivoli applications” on page 16.

Note: The **sysadmin** administrative account initially defined for the Tivoli Enterprise Portal Server does not need to be added to LDAP. However, you can use the **sysadmin** ID to create another administration account known only to LDAP; you can then configure the portal server with LDAP and, from that point forward, use this alternate administrative account instead of **sysadmin**.

Help for the Tivoli Enterprise Portal presented by the Eclipse Help Server

The online Help for the Tivoli Enterprise Portal is now presented using the Eclipse Help Server. The Help Server is installed with the Tivoli Enterprise Portal Server and can be started and stopped only by starting and stopping the portal server.

Event forwarding and synchronization for Tivoli Enterprise Console and Netcool/OMNIbus

You can now forward events generated by Tivoli Enterprise Portal based agents to both Tivoli Enterprise Console and Netcool/OMNIbus event servers. Event forwarding is implemented using the Tivoli Event Integration Event Facility (EIF). Reverse synchronization for forwarded events (changes at the event server returned to the originating monitoring server) is provided by an event synchronization component (SitUpdateForwarder). If you want to forward events to either Tivoli Enterprise Console or Netcool/OMNIbus, you must enable event forwarding on the hub Tivoli Enterprise Monitoring Server and configure a default target event server. For more information, see 641 and Part 6, “Integrating event management systems,” on page 641.

Support for /3GB boot option on 32-bit Windows

Typically, Windows 32-bit supports only 2 GB of virtual memory. The /3GB boot option, available on Windows 2000 Advanced Server and later, allows you to allocate 3 GB of virtual memory. The Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server now support this option. For more details on this option and supported Windows versions, see http://technet.microsoft.com/en-us/library/e834e9c7-708c-43bf-b877-e14ae443ecbf.aspx.

Common Event Console view for events from multiple event servers

The Common Event Console enables you to view and manage events from the Tivoli Enterprise Monitoring Server in the same way as the situation event console. In addition, however, it incorporates events from the Tivoli Enterprise Console Event Server and the Tivoli Netcool/OMNIbus ObjectServer if your managed environment is configured for those servers.

Event data is retrieved from an event system by a common event connector, which also sends user-initiated actions to be run in that event system. To have the events from a specific event system displayed in the Common Event Console, you must configure a connector for that event system. During configuration of the Tivoli Enterprise Portal Server you can edit the default connector for the Tivoli Enterprise Monitoring Server and you can configure additional connectors for other event systems.

Remote installation of application support files using Manage Tivoli Enterprise Monitoring Services on Linux

In previous versions, the Manage Tivoli Enterprise Monitoring Services utility on Windows could be used to send application catalog and attribute files via FTP to nonlocal monitoring servers. The utility can also be used to transfer the .sql files required for application support on a nonlocal hub to a nonlocal monitoring server. In V6.2, these functions have been extended to Linux.

Flexible scheduling of Summarization and Pruning Agent

The Summarization and Pruning Agent has been enhanced to allow for flexible scheduling and to have the warehouse and warehouse aggregation logs trimmed automatically after a specified number of days, months, or years.

Validation of monitoring server protocols and standby configuration

In previous versions, there was no validation of hub and remote monitoring server protocols and standby configuration. In V6.2, the installer validates configuration values such as the local host name of address, the port number, and the communications protocols for hub, remote, and standby monitoring servers. You may see warnings if you enter incorrect values during configuration.

Support for License Manager

IBM Tivoli Monitoring V6.2 products are enabled for use with IBM Tivoli License Manager. Base components and Monitoring agents provide inventory signature files and usage definitions that allow License Manager to report installed products and product usage by computer. License Manager support is an optional capability that requires License Manager version 2.2 or later.

Support for UNIX agents in Solaris local zones

On Solaris, the UNIX OS monitoring agent running in a local zone and remote deployment to a local zone are now supported. See “Installing into Solaris zones” on page 134 for more information.

New managed system groups offer advanced features over managed system lists

Managed system groups are an expansion of the managed system lists known in prior releases. A managed system group comprises a named, heterogeneous list of similar or dissimilar managed systems for the distribution of historical collections, situations, and policies, and for assignment to queries and items in custom Navigator views. If a managed system group is updated (usually when a constituent managed system is added or deleted), then all the historical collections, situations, and policies that use that group are redistributed to all managed systems in the group. Managed system groups are created, modified, or deleted either by the Tivoli Enterprise Portal's Object Group editor or via the **tacmd** CLI command with the **createsystemlist**, **editsystemlist**, or **deletesystemlist** keywords.

New in release 6.2 fix pack 1

The following items are new in IBM Tivoli Monitoring V6.2 fix pack 1:

Base DVD split for Fix Pack 1

For Fix Pack 1, the Base DVD has been split into two DVDs, one for Windows and one for UNIX and Linux:

- **IBM Tivoli Monitoring V6.2 FP1 Windows Base DVD**. This DVD contains the IBM Tivoli Monitoring base product components, the operating system agents for Windows, as well as application support for the operating system agents and a selected set of IBM Tivoli Monitoring 6.x based distributed monitoring agents.
- **IBM Tivoli Monitoring V6.2 FP1 UNIX/Linux Base DVD**. This DVD contains the IBM Tivoli Monitoring base product components, the operating system agents for UNIX and Linux,, as well as application support for the operating system agents and a selected set of IBM Tivoli Monitoring 6.x based distributed monitoring agents.

Support for Sun Java Runtime Environment

The Tivoli Enterprise Portal client included with V6.2 FP1 and subsequent releases has been enhanced to support the Sun Java runtime environment, versions 1.5.0_xx through 1.6.0_xx. All client deployment modes are supported (desktop, browser, and Java Web Start). Both Internet Explorer and Mozilla Firefox browsers are supported using the Sun JRE and the IBM JRE. However, neither the IBM nor Sun JRE 1.6 is supported for Firefox on Linux.

Installer support for the Sun JRE is not available with FP1. The packaging, installation, and servicing of the Sun JRE is not provided by IBM. The Sun JRE must already be installed on the machines where the portal client will run, and in most cases some manual configuration is required to enable this feature. See "Configuring clients, browsers, and JREs" on page 296.

Notes:

1. *Support for the Sun JRE in FP1 is a Tivoli Enterprise Portal client feature only;* installation and use of the IBM JRE is still required for the Tivoli Enterprise Portal Server and other IBM Tivoli Monitoring components.
2. When running the browser client with Sun JRE 1.6.0_10 or higher, you may need to clear your browser's cache to ensure the updated applet.html is loaded.

Support for the browser client on Linux

Fix Pack 1 introduces support for the browser client on Linux computers running Mozilla Firefox. See "Browser clients" on page 297 for more information.

Support for single sign-on for launch to and from other Tivoli applications

Fix Pack 1 introduces support for a single sign-on capability between Tivoli applications. You can now launch from the Tivoli Enterprise Portal to other Tivoli web-based and web-enabled applications, and from those applications into the portal without re-entering your log-on credentials. This single sign-on solution uses a central LDAP-based user registry to authenticate sign-on credentials. For more information, see "Security options" on page 136 and the *IBM Tivoli Monitoring: Administrator's Guide*.

New in release 6.2.1

The following items are new in IBM Tivoli Monitoring V6.2.1:

- The 64-bit operating environments are now supported in the following IBM Tivoli Monitoring components for AIX versions 5.3 and 6.1:
  - Tivoli Enterprise Monitoring Server
  - Tivoli Enterprise Portal Server
  - Tivoli Enterprise Portal
  - Warehouse Proxy
  - Summarization and Pruning Agent
- The 64-bit operating environments are now supported in the following IBM Tivoli Monitoring components for SuSE Linux Enterprise Server versions 9 and 10 on zSeries®:
  - Tivoli Enterprise Monitoring Server
  - Tivoli Enterprise Portal Server
  - Tivoli Enterprise Portal
  - Warehouse Proxy
  - Summarization and Pruning Agent

Reconfigured product media

The IBM Tivoli Monitoring product media have changed again for version 6.2.1. There are now two Base DVDs:

- The *Infrastructure DVD* contains:
  - The Tivoli Enterprise Monitoring Server and its application support
  - The Tivoli Enterprise Portal Server and its application support
  - The Tivoli Enterprise Portal desktop and browser clients and their application support
  - The two Tivoli Data Warehouse agents: the Warehouse Proxy Agent and the Summarization and Pruning Agent

  This DVD is provided in platform-specific versions: one each for Windows, Linux, and UNIX.
- The *Agents DVD* contains these Base monitoring agents:
  - IBM Tivoli Monitoring 5.x Endpoint
  - Windows OS
  - Linux OS
  - UNIX OS

- UNIX Logs
- IBM Tivoli Universal Agent

This DVD is platform-nonspecific: the same DVD applies to Windows, Linux, and UNIX.

New IBM Tivoli Monitoring High-Availability Guide provides resiliency information and instructions

All information about IBM Tivoli Monitoring's resiliency features, is now covered in a separate book, the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*. These features include:

- your ability to assign Hot Standby monitoring servers (that is, backup, or secondary, monitoring servers).
- the clustering of Tivoli Monitoring components using standard clustering software such as the High-Availability Cluster Multiprocessing software provided for pSeries® AIX environments, IBM Tivoli's System Automation—Multiplatform, and Microsoft Cluster Server.
- agent autonomous mode, which ensures event information is not lost when communications are lost between an agent and its monitoring server. Agent autonomy comprises the following two running modes:

  Managed Mode

  These agents come in either of the following two flavors: a *fully connected agent* to the Tivoli Enterprise Monitoring Server. While in this mode, the agent behaves as agents traditionally do. But a *partially connected agent* runs in a semi-autonomous mode; that is, it is disconnected for some time from the monitoring server.

  Unmanaged Mode

  Such agents are fully autonomous and need not be connected to a monitoring server at all.

The information provided includes instructions for implementing these resiliency features and scenarios that lay out how and when they can best be used.

For z/OS customers, information about your options for configuring a Hot Standby hub monitoring server on z/OS (sometimes called a Movable Hub) is now provided in the *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS* guide; this manual also includes implementation instructions.

IPv6 communications protocol now fully supported

IP version 6 (IPv6) communication can now be enabled between any two IBM Tivoli Monitoring components, such as the portal server and the hub monitoring server, a remote monitoring server and the hub, or an agent and a hub. Both components need to be at version 6.2.*x* with the exception of agents, which can remain at version 6.1.

Note: Upgrading your Tivoli Enterprise Monitoring Server to IPv6 overwrites any customizations you may have made and written to the monitoring server's ms.ini file.

RedHat Enterprise Linux 2.1 no longer supported on Intel platforms

IBM Tivoli Monitoring support for 32-bit RedHat Enterprise Linux 2.1 on Intel servers has been dropped.

SELinux now supported when executing IBM Tivoli Monitoring: You can now run IBM Tivoli Monitoring with security-enhanced Linux (SELinux) active (although you still cannot install or configure IBM Tivoli Monitoring with SELinux active). See usage note 7 on page 149 for Table 19 on page 145.

Asynchronous remote agent deployment and group deployment now supported

Agents can now be remotely deployed asynchronously as well as synchronously. With asynchronous deployment, you can request that a second remote agent be deployed, even if the previously deployed agent has not initialized fully.

IBM Tivoli Monitoring also allows you to group agents for bulk remote deployment. The deployment commands have been modified to accept two new grouping parameters: the –g parameter lets you specify a deployment group and the –b parameter lets you specify a bundle group. For detailed information, see "Bulk agent deployment" on page 336; also see the *IBM Tivoli Monitoring: Command Reference*.

This enhancement also provides support for the deployment, upgrading, and configuration of Netcool/OMNIbus System Service Monitor (SSM) agents.

Linux/UNIX users: 64-bit Tivoli Enterprise Portal Server now supported

You can now install and configure a 64-bit portal server on Linux or UNIX. In addition, a procedure has been added that enables you to convert your 32-bit portal server to 64 bit; see "Upgrading a 32-bit portal server to 64 bit" on page 252.

Support for 64-bit DB2 for Linux, UNIX, and Windows

IBM DB2 Database for Linux, UNIX, and Windows V9.5 is supported in 64-bit mode.

Separate DB2 for Linux, UNIX, and Windows licensing no longer required: Since IBM Tivoli Monitoring now includes an edition of DB2 for Linux, UNIX, and Windows that does not require an activation CD or a separate product registration, the separate DB2 licensing step is no longer required. The DB2 license file, db2ese_o.lic.txt, has been removed from the distribution media.

Tivoli Data Warehouse now supports DB2 on z/OS

You can now create your Tivoli Data Warehouse repository using DB2 running on z/OS. While the Warehouse Proxy Agent still runs only on Windows, Linux, or UNIX, the data warehouse itself is stored in DB2 on z/OS databases. Data communication is supported using either an ODBC or a JDBC connection.

Instructions for setting up your Tivoli Data Warehouse environment to run with a DB2 on z/OS repository are provided in Chapter 21, "Tivoli Data Warehouse solution using DB2 on z/OS," on page 519.

New schema publication tool simplifies generation of SQL statements needed to create the Tivoli Data Warehouse

With the new schema publication tool, you can now generate the SQL statements needed to create the database objects (data warehouse tables, indexes, functions, views, and ID table inserts) required for initial setup of the Tivoli Data Warehouse. See Chapter 19, "Schema Publication Tool," on page 483.

Tivoli Data Warehouse support for Solaris environments

The Warehouse Proxy Agent now supports Solaris versions 9 and 10, along with support for Solaris zones.

Agentless monitoring of distributed operating systems now supported

IBM Tivoli Monitoring version 6.2.1 supports agentless monitoring of the distributed operating systems (Windows, Linux, UNIX). The agentless monitoring software identifies and notifies you of common problems with the operating system that it monitors, and includes monitoring, data-gathering, and event-management capabilities for Windows, Linux, AIX, HP-UX, and Solaris.

"Agentless monitoring versus monitoring agents" on page 66 introduces the concept of agentless monitoring and compares its advantages to the those of standard OS monitoring agents.

The tacmd createNode command need no longer be executed on the monitoring server node

The command was changed in this release to execute through the remote-deployment infrastructure on the Tivoli Enterprise Monitoring Server instead of executing locally. Thus you need no longer execute the **tacmd createNode** command on the machine running the monitoring server.

Support for multiple remote Tivoli Enterprise Monitoring Servers on one Linux or UNIX computer

You can now install and run multiple remote monitoring servers on a single Linux and UNIX node or LPAR; see "Linux or UNIX: Installing a remote monitoring server" on page 225. Note, however, that you are still restricted to a single *hub* monitoring server per computer or LPAR.

This enhancement does not apply to Windows nodes.

New in release 6.2.2

Contents of the *Deployment Guide* merged

The contents of the version 6.2 *IBM Tivoli Monitoring: Deployment Guide* have been merged into this guide and the *Deployment Guide* eliminated.

- A new section, Part 2, “Planning your IBM Tivoli Monitoring deployment,” on page 31, has been created to contain the *Deployment Guide's* three deployment chapters:
  - Chapter 2, “Pre-deployment phase,” on page 33
  - Chapter 3, “Deployment phase,” on page 83
  - Chapter 5, “Post-deployment phase,” on page 117

  These chapters replace the single deployment chapter (Chapter 2, *Planning your deployment*) that this guide used to contain, as they are more current and more complete.
- The *Deployment Guide's* tuning chapter, Chapter 17, “Performance tuning,” on page 421, has been added to the end of Part 4, “Postinstallation configuration and customization,” on page 361.
- The *Deployment Guide's* resources appendix, Appendix J, “Additional resources,” on page 871, has been placed with the rest of the appendixes in this guide, after Appendix I, “Documentation library,” on page 867.

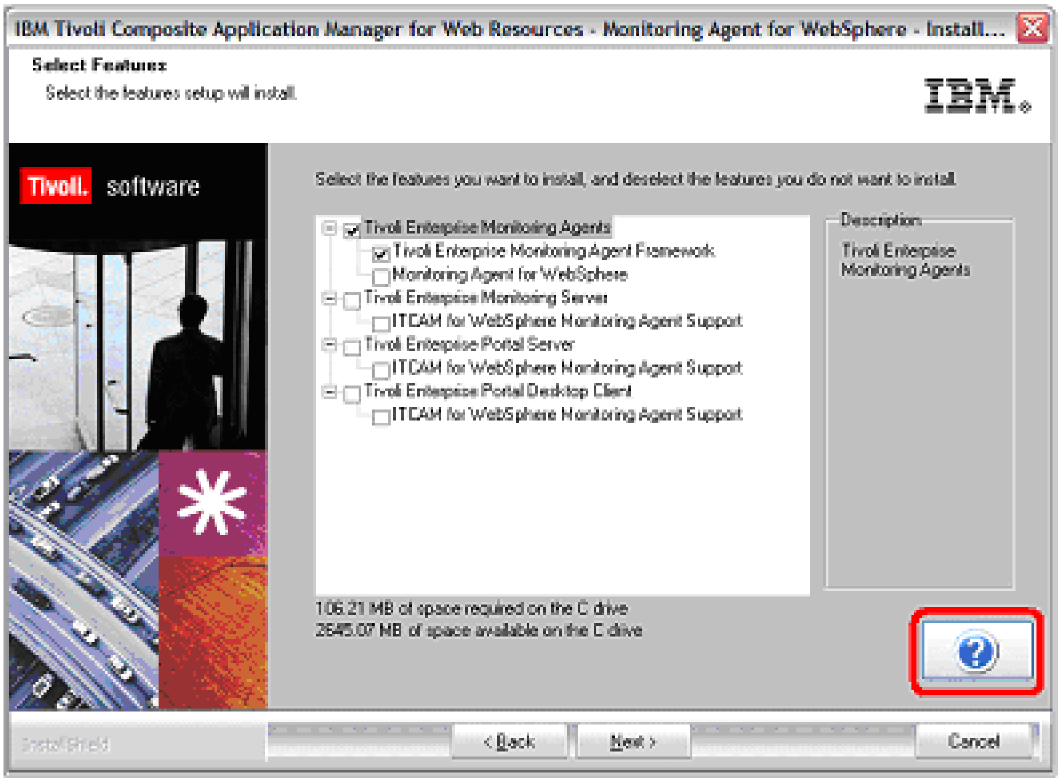
Readers should find that this additional material provides vastly more-detailed information about deploying and tuning their site's IBM Tivoli Monitoring environment.

Certain information unique to the former Chapter 2, *Planning your deployment*, has been moved to other, relevant locations in this guide.

- The Tivoli Data Warehouse planning information, “Planning considerations for the Tivoli Data Warehouse” on page 469, has been moved to the data warehousing section, in Chapter 18, “Tivoli Data Warehouse solutions,” on page 465.
- The explanatory information about the agentless monitors, “Agentless monitoring versus monitoring agents” on page 66, has been moved to the “Agent deployments” on page 58 discussion.
- Likewise, the event-integration scenarios for Tivoli Enterprise Console and Netcool/OMNIbus sites have been moved near the tops of “Event integration with Tivoli Enterprise Console” on page 644 and Chapter 26, “Setting up event forwarding to Netcool/OMNIbus,” on page 675.

Additional online user information while the Windows installer is running

Many of the installation tool's panels have been enhanced with a help facility that provides users with additional information about the panel's purpose and parameters; see Figure 3 on page 20 and Figure 4 on page 20.



IBM Tivoli Composite Application Manager for Web Resources - Monitoring Agent for WebSphere - Install...

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

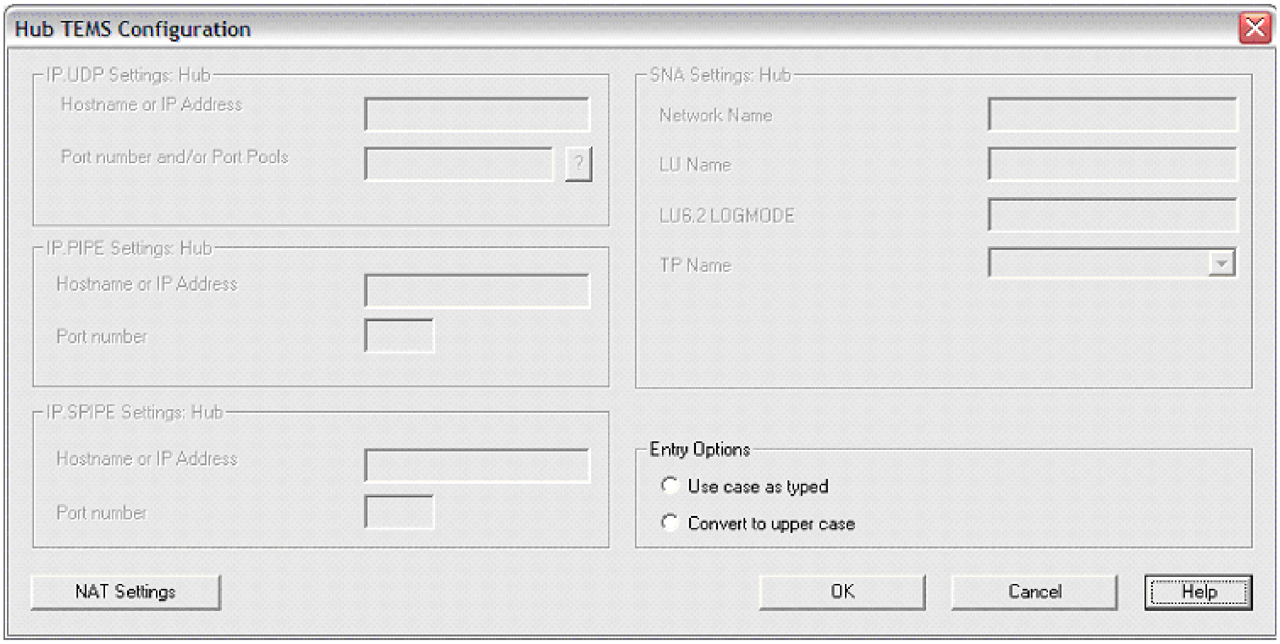
- ☑ Tivoli Enterprise Monitoring Agents
  - ☑ Tivoli Enterprise Monitoring Agent Framework
  - □ Monitoring Agent for WebSphere
- □ Tivoli Enterprise Monitoring Server
  - □ ITCAM for WebSphere Monitoring Agent Support
- □ Tivoli Enterprise Portal Server
  - □ ITCAM for WebSphere Monitoring Agent Support
- □ Tivoli Enterprise Portal Desktop Client
  - □ ITCAM for WebSphere Monitoring Agent Support

Description

Tivoli Enterprise Monitoring Agents

106.21 MB of space required on the C drive
2645.07 MB of space available on the C drive

Figure 3. Help button on the IBM Tivoli Monitoring installer's Select Features panel



Hub TEMS Configuration

IP.UDP Settings: Hub

Hostname or IP Address

Port number and/or Port Pools

IP.PIPE Settings: Hub

Hostname or IP Address

Port number

IP.SPIPE Settings: Hub

Hostname or IP Address

Port number

SNA Settings: Hub

Network Name

LU Name

LU6.2 LOGMODE

TP Name

Entry Options

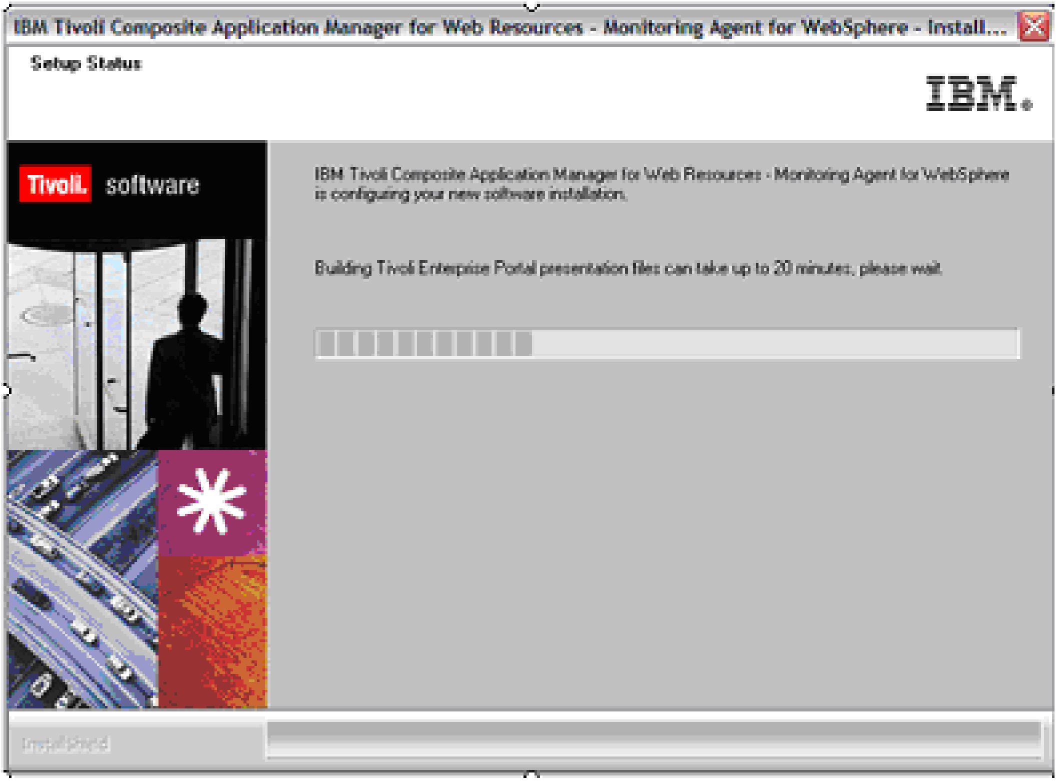
○ Use case as typed

○ Convert to upper case

Figure 4. Help button on the installer's Hub TEMS Configuration panel

In addition, progress bars have been added to many long-running processes (such as software installation and the installation of language support) so users can see visually how these processes are progressing;

for an example, see Figure 5.



IBM Tivoli Composite Application Manager for Web Resources - Monitoring Agent for WebSphere - Install...

Setup Status

IBM Tivoli Composite Application Manager for Web Resources - Monitoring Agent for WebSphere is configuring your new software installation.

Building Tivoli Enterprise Portal presentation files can take up to 20 minutes, please wait.

Figure 5. Status bar for the installer's software-installation phase

Embedded Java Runtime Environment now supported for Windows sites

The IBM Tivoli Monitoring installer no longer modifies the system JRE. It now installs its own local copy of Java, one that is private to Tivoli Monitoring. This change applies to all Java-dependent Tivoli Monitoring components, including those at a pre-6.2.2 level. IBM Tivoli Monitoring supports IBM Java SDK 5.0 SR9 or higher.

Users of Sun Java 1.6 can improve the performance of the Tivoli Enterprise Portal browser client on Windows: A new plug-in architecture was introduced and established as the default plug-in for users of Sun Java versions 1.6.0_10 or higher on Windows systems. This support enables an applet-caching feature called **legacy lifecycle** that, coupled with use of the Sun 1.6 JRE, significantly increases performance of the Tivoli Enterprise Portal browser client. Performance measurements using this configuration show that, with legacy lifecycle, performance of the browser client is virtually identical to that of the desktop and Java Web Start deployment modes.

Configuration changes are required to take advantage of this enhancement; see “Support for Java 6 or higher with browser clients on Windows” on page 302.

New installation process for language packs

IBM Tivoli Monitoring Version 6.2.2 includes a new installer for the language packs; thus the instructions in “Installing language packs” on page 292 have all been replaced.

New System Monitor Agents provide autonomous-only monitoring of your operating system

In addition to the traditional operating system agents (the knt, klz, and kux agents that monitor the Windows, Linux, and UNIX operating environments), IBM Tivoli Monitoring now includes System Monitor Agents for customers who want fully autonomous monitoring of their operating systems in environments where disk space or image transmission bandwidth is in short supply. Specifically, IBM Netcool® System Service Monitor customers may find these agents appropriate as an upgrade path. These new OS agents, which offer a significant reduction in bundle size and installed footprint, provide SNMP event information directly to an SNMP Event Collector such as IBM Tivoli Netcool/OMNIbus and are intended as replacements for the OMNIbus System Service Monitor agents.

No other agents should be installed alongside these new System Monitor Agents, nor should they be installed on nodes where there is already a Tivoli Monitoring agent in operation. Agents created with version 6.2.2 (or subsequent) of the Agent Builder tool are the only exception to this rule.

To support these new System Monitor Agents, there is a new silent installation process; see "Installing the System Monitor Agent on Windows systems" on page 350 and "Installing the System Monitor Agent on Linux or UNIX systems" on page 354. There is also a new uninstallation process for the autonomous OS agents; see "Uninstalling the Windows System Monitor Agent" on page 353 and "Uninstalling the Linux or UNIX System Monitor Agent" on page 358.

"Background information about agent autonomy" on page 65 introduces the concept of agents that operate autonomously (that is, without a connection to a Tivoli Enterprise Monitoring Server). For additional information, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Derby now supported as a portal server database

Version 6.2.2 includes an embedded version of the Apache Derby database server for default use as the Tivoli Enterprise Portal Server database. (Note, however, that you can still use either DB2 Database for Linux, UNIX, and Windows or Microsoft SQL Server as your portal server database if you choose and have already installed them.) IBM Tivoli Monitoring implements Derby as an embedded database with its portal server; in other words, the database is installed when installing the portal server, and it runs within the portal server's Java virtual machine.

The embedded Derby database is intended for use only by small to medium-size businesses. It is not intended for large IBM Tivoli Monitoring environments or for sites running a large number of Tivoli Enterprise Portal clients. Derby has been tested with up to 20 portal clients, and it does require higher CPU and memory usage than DB2 for Linux, UNIX, and Windows.

The Tivoli Enterprise Portal Server uses the implementation of Derby that comes with eWAS; hence it is supported on all platforms that support eWAS.

Note: If you transition from one supported database system to another for the Tivoli Enterprise Portal Server, your existing portal server data is not copied from the first system to the new one.

More memory required to install and run the portal server: As a result of support for both an embedded database and the eWAS server, the Tivoli Enterprise Portal Server's memory requirements have increased substantially. The portal server now requires 650 MB if your site does not use the embedded database, 1 GB if it does.

Common agent environment variables listed

A new appendix, Appendix E, "Agent configuration and environment variables," on page 819, lists all the variables that define configuration parameters common to all IBM Tivoli Monitoring monitoring agents, including the new environment variables that enable and control agent autonomy. (For complete information about agent autonomy, see the*IBM Tivoli Monitoring: Administrator's Guide*.)

New Tivoli Enterprise Services User Interface Extensions

The tacmd CLI commands (the KUI component of IBM Tivoli Monitoring) are being deprecated in favor of the new Tivoli Enterprise Services User Interface Extensions (the new KUE component), as documented in the *IBM Tivoli Monitoring: Command Reference*.

These new user interface extensions are automatically installed when you install the Tivoli Enterprise Portal Desktop Client, the Tivoli Enterprise Portal Server, or the Tivoli Enterprise Monitoring Server after invoking the GUI installer. Additionally, they are an optional installable component of the Tivoli Enterprise Management Agent component.

Notes:

1. Though the tacmd commands have moved to a different component, KUE, and their packaging and the means for installing them have both changed, the tacmd commands themselves have not changed.
2. When installing an infrastructure agent using the silent-installation response file, you must explicitly install the optional KUE component if it is needed.

Improved user control of agent or server restart after reconfiguration

After reconfiguring a running monitoring agent or Tivoli Enterprise Monitoring Server, you are now asked if you want the agent or server restarted so your changes can take effect. Because it is not possible to reconfigure a Tivoli Enterprise Portal Server while running, before reconfiguring it you are asked if you want the portal server stopped; if you reply Yes, when the reconfiguration has completed, you are asked if you want the portal server restarted.

Note: These restart options do not apply to either multi-instance agents or silent agent/server reconfigurations.

A new column has been added to the Manage Tivoli Enterprise Monitoring Services main window: **Configuration**. With this column you can keep track of agents whose configuration has changed but has not yet been restarted. The **Configuration** column has the following possible values:

`up-to-date`
The agent is using the most recent configuration. In other words, either the agent was restarted after the last reconfiguration or it is not running.

`out-of-sync`
The agent is not using the most recent configuration. In other words, the agent was not restarted after the last reconfiguration.

`N/A`
Applies either to multi-instance agents (the restart options do not support multi-instance agents) or in cases where IBM Tivoli Monitoring cannot determine the configuration state.

Note: When you install a multi-instance Agent Builder agent, you don't need to run a command to configure it. You just need to provide the necessary .cfg file (if needed) and run the start command: `%CANDLE_HOME%\InstallITM\itmcmd.cmd agent start -o` *`instance product_code`*

Dynamic affinity affects agent coexistence with prior releases

The introduction of dynamic affinity for operating system agents affects your ability to use newly installed V6.2.2 agents with hub and remote Tivoli Enterprise Monitoring Servers and Tivoli Enterprise Portal Servers built for prior releases. Before your site implements a V6.2.2 monitoring agent, you need to ensure the monitoring server and the portal server (including the portal server's desktop clients) have been upgraded to the V6.2.2 level.

The supported configurations are:

- Tivoli Monitoring V6.2.2 agents require a V6.2.2 hub monitoring server and portal server. The V6.2.1 hub monitoring server and portal server do not support V6.2.2 agents.
- The V6.2.2 CLI can connect only to a V6.2.2 hub monitoring server.

Notes:

1. If your site does not update its desktop clients, they will exhibit problems when using the Query and Situation editors to modify the upgraded monitoring agents. In addition, the use of any V6.2.2 monitoring agents with a pre-V6.2.2 portal server means that workspaces will not be rendered correctly.
2. V6.2.1 agents work with both V6.2.1 and V6.2.2 monitoring servers and portal servers.

New installation parameters protect your customized configuration settings: When adding an agent's application support to the Tivoli Enterprise Monitoring Server, you are now asked to specify if you want to add the default managed system groups to its situations. The **All**, **New**, and **None** selections allow you to add the default managed system groups to all applicable situations, to only the ones being newly seeded, or to none of them.

Higher versions of the Firefox browser supported for Windows customers

If your site uses Mozilla's Firefox browser to run the Tivoli Enterprise Portal browser client, you can now use version 3.0.x as well as Firefox version 2.x. However, version 3.5.x is still not supported.

Simplified operating system selection for Linux and UNIX systems

Installers on Linux and UNIX systems are now presented the following selection screen when choosing the IBM Tivoli Monitoring operating system components to install:

```
Product packages are available for this operating system and component support categlories:
1) IBM Tivoli Monitoring components for this operating system
2) Tivoli Enterprise Portal Browser Client support
3) Tivoli Enterprise Portal Desktop Client support
4) Tivoli Enterprise Portal Server support
5) Tivoli Enterprise Monitoring Server support
6) Other operating systems
Type the number of type "q" to quit selection
(Number "1" or "IBM Tivoli Monitoring components for this operating system" is the default value.)
```

This new screen makes it easier and less error-prone to install the appropriate components for the operating system your current node is running.

To retrieve a list of other operating systems and releases, type **6**.

New installation option allows you to retain your customized seeding files

As part of the conversion of the operating system monitoring agents to dynamically assigned affinities, you now have the option, when adding their application support to the Tivoli Enterprise Monitoring Server, of either using the supplied files or retaining and using the ones you have modified.

Also, remote seeding is now supported.

Automatic installation of application support for Linux/UNIX monitoring servers

You now have the option to seed the monitoring server's application support during agent installation. In such cases, it is no longer necessary to manually seed the Tivoli Enterprise Monitoring Server after installation or upgrade.

New silent-response files simplify agent installations and updates

You now have the option of generating a silent response file when you successfully configure an agent. This response file can be used to install or deploy similar agents across the environment. This new option significantly speeds up the creation of silent response files and reduces the chances of including incorrect user input in a response file.

The response file can be generated from any successfully installed and configured agent.

Remote-deployment support extended to non-agent bundles

You can use new non-agent bundles to deploy components that need not connect to the Tivoli Enterprise Monitoring Server. Using V6.2.2, you can remotely deploy these new non-agent bundles, similar to the deployment of the agent bundles supported in prior releases.

Event integration of IBM Tivoli Monitoring with both IBM Tivoli Business Service Manager and Netcool/OMNIbus now supported

For users of both Tivoli Business Service Manager and Netcool/OMNIbus, integration of Tivoli Monitoring events is now supported for both products.

Upgrade procedure provided to Tivoli Event Synchronization V2.2.0.0: For users of Tivoli Event Synchronization version 2.0.0.0, an upgrade procedure has been added that converts your existing environment to V2.2.0.0.

OMEGAMON data warehouse migration tool no longer provided

OMEGAMON V350 and V360 customers who want to migrate their data warehouse to the new Tivoli Data Warehouse can use the migration tool provided with prior versions of IBM Tivoli Monitoring. Version 6.2.2 no longer includes that migration tool.

New in release 6.2.2 fix pack 1

Native 64-bit operating system agents available for 64-bit Windows environments: A new OS monitoring agent built as a native 64-bit binary is now available for sites that run 64-bit Windows Server 2003 or 2008 on x86-64 CPUs. If your site runs a 64-bit Windows operating system on such computers, you may want to install the new 64-bit agent from the Agents DVD using the standard agent installation process, via either the interactive installer or remote deployment. Their configuration is the same as the 32-bit Windows agent's.

To support mixed (that is, both 32-bit and 64-bit) agent environments on the same machine, you must install the agent compatibility (AC) component, which contains both the 32- and 64-bit agent frameworks and both the 32- and 64-bit GSKit libraries. The AC component is installed automatically when the installer detects it is running in an environment that contains both 32-bit and 64-bit IBM Tivoli Monitoring components. If the AC component is not installed, the installer will block the installation of either a 32-bit agent into a 64-bit Tivoli Monitoring environment or a 64-bit agent into a 32-bit Tivoli Monitoring environment.

Note: Installation of both the 32-bit and the 64-bit Windows agents on the same machine is not supported and will be blocked.

Event forwarding by autonomous agents: Since Tivoli Enterprise Monitoring Agents and Tivoli System Monitor Agents can now run completely independently of the IBM Tivoli Monitoring base servers (the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server), these agents have now been enhanced to enable them to directly forward events to either the Tivoli Enterprise Console or Netcool/OMNIbus. This ensures events trapped by these monitoring agents do not get lost when you elect to run them autonomously.

Support for DB2 Database for Linux, UNIX, and Windows version 9.7: Both the Tivoli Data Warehouse and the Tivoli Enterprise Portal Server now support V9.7 of DB2 for Linux, UNIX, and Windows.

Note: IBM Tivoli Monitoring V6.2.*x* still includes DB2 Workstation Server Edition V9.5 for use with the portal server and the data warehouse.

New in release 6.2.2 fix pack 2

64-bit System Monitor Agent now supported for Windows environments: IBM Tivoli Monitoring now provides native support for Windows-based (Windows 2003, Vista, 2008) System Monitor Agents running on x86_64 CPUs in 64-bit mode.

Note: If you take advantage of this new feature, note that any agents built with the Agent Builder that you install alongside the Windows System Monitor Agent must run in the same mode: a 32-bit Windows agent requires Agent Builder-built agents that run in 32-bit mode, and a 64-bit Windows agent requires Agent Builder-built agents that run in 64-bit mode.

Autonomous operation of the Warehouse Proxy Agent and the Summarization and Pruning Agent: To enable autonomous agents to write out their accumulated historical data to the Tivoli Data Warehouse without the intervention of the Tivoli Enterprise Monitoring Server and thereby prevent loss of such data, the warehouse agents have been enhanced to allow them to run autonomously as well.

32-bit DB2 for Linux, UNIX, and Windows no longer required for the Tivoli Enterprise Portal Server: The portal server's requirement for a 32-bit implementation of DB2 Database for Linux, UNIX, and Windows even when running on a 64-bit Windows computer has been removed. The portal server now can use either a 32-bit or a 64-bit implementation of DB2 for Linux, UNIX, and Windows.

Further enhancements to the autostart scripts: The behavior of the autostart scripts generated during installation and configuration on UNIX platforms has evolved; see "Changes in the behavior of the autostart scripts" on page 129.

Procedure for taking a snapshot of your Tivoli Enterprise Portal Server configuration settings now documented: If your site runs multiple hub Tivoli Enterprise Monitoring Servers and thus needs to switch the portal server from one hub to another, it is recommended you first take a snapshot of your site's current portal server customizations. The procedure for creating this snapshot and for restoring it later, "Connecting the Tivoli Enterprise Portal Server on Windows to a different monitoring server" on page 397, has been added to Chapter 15, "Additional Tivoli Enterprise Portal configurations."

Procedure for populating the data warehouse's ManagedSystem table now documented: If your site runs the Tivoli Data Warehouse, you must update its ManagedSystem table each time you add one or more monitoring agents to your site's Tivoli Monitoring environment; this is accomplished via the `populate_agents.sql` script. The procedures for invoking this script in a DB2 for Linux, UNIX, and Windows, SQL Server, or Oracle environment have been added to Chapter 9, "Installing IBM Tivoli Monitoring" in a new subsection, "Populating the data warehouse's ManagedSystem table" on page 262.

New in release 6.2.3

New 64-bit Warehouse Proxy Agent simplifies Tivoli Data Warehouse setup for Windows sites

To eliminate the need for a 32-bit ODBC driver, a 64-bit Warehouse Proxy Agent is now provided for users running 64-bit Windows environments. The 64-bit agent can run with the 64-bit ODBC driver.

Note that you must choose at installation time which Warehouse Proxy Agent to install on your Windows machines, as the 32-bit agent cannot coexist with the 64-bit agent. If you currently run the Tivoli Data Warehouse with a 32-bit Warehouse Proxy Agent on a 64-bit Windows system, and want to upgrade to the new 64-bit agent, you must first uninstall the 32-bit agent, and then install the 64-bit agent.

New prerequisite checking for IBM Tivoli Monitoring Agents

By using the new prerequisite-checking feature, you can perform prerequisite checking for agents before carrying out an installation. The two mechanisms available are a manually executed stand-alone prerequisite scanner, or a remote prerequisite scanner facility that extends the capabilities of IBM Tivoli Monitoring's remote deployment component.

The **tacmd checkprereq** command allows you to check the prerequisites required for deploying an agent to a managed system. The **tacmd checkprereq** command deploys a prerequisite checking tool to determine if the target system meets the requirements for the agent. A global transaction ID is immediately returned. You can then use the Deployment Status workspace in the Tivoli Enterprise Portal, or run the **tacmd getDeployStatus** command, to view the status of the queued operation. For more information, see "Prerequisite Checking for IBM Tivoli Monitoring agents" on page 59 and the *IBM Tivoli Monitoring: Command Reference*.

Self-describing monitoring agents

The self-describing agent feature makes it possible for new or updated IBM Tivoli Monitoring agents to become operational after installation, without having to perform additional product support installation steps. By default, the self-describing agent capability is disabled at the hub monitoring server. Self-describing agent environment variables are enabled by default for remote monitoring servers, portal servers, and self-describing agent enabled agents. However, these components disable the self-describing agent capability if connected to a hub monitoring server that has the self-describing agent capability disabled. For more information, see "Enabling self-describing agent capability at the hub monitoring server" on page 216.

Enabling the self-describing agent capability at the hub monitoring server controls the capability across all components. You can disable the capability individually at a remote monitoring server, portal server, or agent, but the best practice is to control the self-describing agent capability from the hub monitoring server.

If you prefer to manage application support in the same way as for earlier versions, you can disable the self-describing capability by editing the monitoring server and agent environment variables. For more information, see "Self-describing agent installation" on page 347.

New Tivoli Monitoring Startup Center

A new graphical user interface tool that uses topology diagrams to help you plan, configure, and deploy your IBM Tivoli Monitoring environment. The purpose of the Startup Center is to get an initial environment up and running for a new IBM Tivoli Monitoring installation. The Startup Center is not intended for upgrading existing IBM Tivoli Monitoring components. The Startup Center can run on both Windows and Linux Intel x86-32 systems. For more information, see Chapter 4, "Tivoli Monitoring Startup Center," on page 89.

Tivoli Performance Analyzer integrated as a base component of IBM Tivoli Monitoring

Tivoli Performance Analyzer is now a base component of IBM Tivoli Monitoring and part of the IBM Tivoli Monitoring installation media. Tivoli Performance Analyzer adds predictive capability to Tivoli Monitoring so you can monitor resource consumption trends, anticipate future performance issues, and avoid or resolve problems more quickly. For more information, see "Tivoli Performance Analyzer" on page 43.

DB2 agent and Oracle agent integrated as optional components of IBM Tivoli Monitoring

Two additional agents are now optional components of IBM Tivoli Monitoring and part of the IBM Tivoli Monitoring installation media. These agents might be used to monitor the database used for your Tivoli Data Warehouse, Tivoli Enterprise Portal Server, or both, without additional entitlement. These agents can also be used to monitor customer database management systems under the standard licensing agreements for IBM Tivoli Monitoring. The agents included are:

- IBM Tivoli Composite Application Manager Agent for DB2 provides you with the capability to monitor DB2 databases, and to perform basic actions with DB2 databases.
- IBM Tivoli Composite Application Manager Extended Agent for Oracle provides you with the capability to monitor Oracle databases, and to perform basic actions with Oracle databases.

For details on how to use these agents as well as prerequisite information, see the IBM Tivoli Composite Application Manager for Applications information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/index.jsp?topic=/com.ibm.itcama.doc_6.2.4/prerequisites.html.

Changes to event integration with Netcool/OMNIbus

See the following list of changes related to event integration with Netcool/OMNIbus:

- Support of EIF probe master rules file.
- Removal of the `tivoli_eif.rules` file. The `itm_event.rules` include file should be used instead with the EIF probe master rules file.
- Event integration requires Netcool/OMNIbus V7.2 or later and IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF version 10 or later.
- Chapter 26, "Setting up event forwarding to Netcool/OMNIbus," on page 675 will help you to configure the Netcool/OMNIbus EIF probe to optimize integration with IBM Tivoli Monitoring, Tivoli Business Service Manager, or both.

Dynamic affinity affects agent coexistence with earlier releases

The V6.2.3 CLI can connect only to a V6.2.3 hub monitoring server.

The IBM HTTP Server is the default web server used for communication between the portal client and server

The switch to the IBM HTTP Server occurs automatically with no action required on your part. The IBM HTTP Server provides increased scalability, performance, and reliability for portal clients. The IBM HTTP Server requires specified ports to be open in any firewall between the client and the server. For more information, or details on how to revert back from the IBM HTTP Server to an internal web server, see "Reverting from the IBM HTTP Server to the internal web server" on page 404.

This change can cause the Tivoli Business Service Manager IBM Tivoli Monitoring SSL data fetchers to stop working. This is because the SSL signer certificate will have changed as port 15201 is now the HTTP server. For more information on the changes and how to update Tivoli Business Service Manager with the new signer certificates, see "Tivoli Business Service Manager and Tivoli Enterprise Portal Server integration over SSL" on page 161.

The Portal client can communicate with the portal server by using only the HTTP or HTTPS protocol

HTTP communication can be used between the portal client and server without the need for the CORBA communication protocol. HTTP communication is required to integrate with security products such as Tivoli Access Manager. For more information, see "Configuring HTTP communication between the portal client and server" on page 405.

New performance-tuning information

Chapter 17, "Performance tuning," on page 421 has been enhanced with new information for optimizing the performance of several components within an IBM Tivoli Monitoring environment.

New in release 6.2.3 Fix Pack 1

See the following changes in this release:

Deploy Tivoli Netcool/OMNIbus in a multitiered configuration to increase performance and event handling capacity: You can use the Netcool/OMNIbus multitiered architecture if you need to add scalability to your Netcool/OMNIbus environment to increase performance and event handling. High availability is also supported by adding primary and backup servers to the tiers. For more information, see "Netcool/OMNIbus Multitiered and High-availability Architecture" on page 709.

New appendix for Agent configuration and Tivoli Monitoring environment variables: A new appendix called Appendix E, "Agent configuration and environment variables," on page 819 contains the following sections:

- “Configuration files preserved during an upgrade” on page 819
- “Product behavior with custom configuration settings” on page 823
- “Persistent configuration changes” on page 823

The appendix also contains a list of “Environment variables” on page 826 related to the components and monitoring agents of Tivoli Monitoring.

Self-describing agent application support installation is now supported in Hot-Standby (fault-tolerant operation, or FTO) configuration: For more information about enabling self-describing agent capability and requirements, see “Enabling self-describing agent capability at the hub monitoring server” on page 216.

New silent response file parameters to install products and support in one pass: You can install products and support in one pass by using the silent mode. For more information, see Table 163 on page 794.

Tivoli Performance Analyzer integrated with SPSS to support non-linear trending: You can now configure Tivoli Performance Analyzer to support non-linear trending and forecasting of capacity and performance metrics, using SPSS® Forecast Server expert modeler. The feature provides predictive analytics to forecast future performance based on past historical data. For information about configuring Performance Analyzer to use SPSS, see step 18 on page 196 of Installing Tivoli Monitoring.

New silent response file parameter for enabling protocols for the System Monitor Agent: You can specify protocols that are enabled and disabled for the System Monitor Agent. For example, you can enable the specific protocol that the System Monitor Agent uses to communicate with the Warehouse Proxy Agent. You can use the default protocol IP.PIPE, or switch to a more secure protocol such as IP.SPIPE. For more information, see *Contents of the silent response file* for both Windows and Linux/UNIX in Chapter 11, “Monitoring your operating system via a System Monitor Agent,” on page 349.

Sending private situation events to a Netcool/OMNIbus EIF receiver using SSL: You can now send your private situation events to a Netcool/OMNIbus EIF receiver probe using SSL communication. If you want to configure SSL, the destination Netcool/OMNIbus Probe for Tivoli EIF must be at version 12.0 or higher. For more information, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Java-based infrastructure components upgraded to the Java 6 runtime environment: All Java-based Tivoli Monitoring infrastructure components have been upgraded to the Java 6 runtime environment. Although this upgrade is transparent for most Tivoli Monitoring components, you might have to complete additional tasks to upgrade the Tivoli Enterprise Portal component depending on the deployment mode you use for the Tivoli Enterprise Portal. For more information, see “Installing and configuring IBM Java 6” on page 304.

Part 2. Planning your IBM Tivoli Monitoring deployment

This section contains information that assists you in assessing your environment and planning the deployment of product components. The successful use and availability of your IBM Tivoli Monitoring environment is the result of a well-planned and executed deployment. This section guides you through the deployment process for Tivoli Monitoring; it is divided into the following chapters:

- Chapter 2, “Pre-deployment phase,” on page 33 helps you plan for your installation. This chapter also provides you with an overview of the Tivoli Monitoring components, network considerations, sizing scenarios, platform support, as well as task and staffing estimates.
- Chapter 3, “Deployment phase,” on page 83 contains information, procedures, configuration, and reference documents to ensure a successful deployment of your Tivoli Monitoring environment.
- Chapter 5, “Post-deployment phase,” on page 117 provides you with the maintenance steps needed to ensure that your Tivoli Monitoring environment stays up and running. This chapter also provides you with daily, weekly, monthly, and quarterly health check procedures.

Tivoli Monitoring products use a set of service components (known collectively as Tivoli Management Services) that are shared by a number of other product suites, including IBM Tivoli OMEGAMON XE monitoring products, IBM Tivoli Composite Application Manager products, System Automation for z/OS, Web Access for Information Management, and others. Much of the information in this guide is also relevant to these products.

Chapter 2. Pre-deployment phase

Correctly planning your installation is probably the most important thing that you can do to achieve a smoothly running environment. By carefully planning the size of your servers, understanding your network topology, and understanding your key requirements, you can create a monitoring environment that meets your needs.

Careful planning is a team effort. Engage the networking team to ensure you understand all of the network firewall configurations and to open ports in the firewall. Understand the key requirements such as fault tolerance, executive dashboards, and reporting. The following sections outline the key planning items that you must complete to have a successful deployment.

Planning checklist

Use the Planning checklist in Table 2 to ensure that all important planning tasks are accomplished. Perform all of these tasks before starting your Tivoli Monitoring installation. The following sections of this guide provide the necessary information to complete this checklist.

Table 2. Planning checklist

Planning activity	Comments	Status
Attend IBM Tivoli Monitoring Administrator Training.	Sample: 2 people attended Administrator class	Sample: Complete
Determine the desired platforms for Tivoli Monitoring infrastructure components.	See “Hardware and software requirements” on page 138.	
Identify the number of Tivoli Monitoring agents (typically 3 monitoring agents per monitoring server).		
Determine high availability and disaster recovery requirements. Map those requirements to your Tivoli Monitoring deployment.		
Determine whether a firewall gateway is required and between which components.		
Determine the location of any remote monitoring servers.		
Determine that an adequate number of Warehouse Proxy Agents are planned.		
Provide Network administrators with a list of ports to open.		
Complete the Warehouse load projection spreadsheet. See “Locating and sizing the Warehouse Proxy Agent” on page 49.		
Determine the hardware required for your Tivoli Monitoring components.		
Download all the required software for your supported operating systems including all monitoring agents.		
Complete your deployment Verification Test Plan.		

Table 2. Planning checklist (continued)

Planning activity	Comments	Status
Review the *Readme and Documentation Addendum* and determine the installation steps based on your platform.		
Determine the method of user authentication. See paragraphs below.		

Tivoli Monitoring V6.2.3 supports integration with LDAP user registries for authentication. See "Security options" on page 136 for more information.

LDAP SSL requires some actions by an LDAP administrator that are not covered by the Tivoli Monitoring V6.2.3 documentation. Here are some LDAP SSL Web pages for working with LDAP servers:

- Configuring Microsoft Active Directory for SSL access
- Configuring Sun Java System Directory Server for SSL access
- Configuring the Tivoli Directory Server client for SSL access
- LDAP SSL will also require creating a GSKit keystore; see the *IBM Global Security Kit Secure Sockets Layer and iKeyman User's Guide*

Understanding Tivoli Monitoring and your network

For all successful deployments, you must understand the network infrastructure. When planning your Tivoli Monitoring deployment, take into account each of the following important factors, which are discussed in further detail in this guide:

- Locations of firewalls
- Whether you have NAT (network address translation)
- Network bandwidth between WAN (wide area network) links
- Number of agents that are located across WAN links

Tivoli Monitoring has several options for the communication protocols that are used by the product components. The IP PIPE and IP SPIPE are the protocols used when firewalls are used in the environment. Unlike the IP UDP (TCP) protocol, the communications for IP PIPE and IP SPIPE can be restricted to a single port.

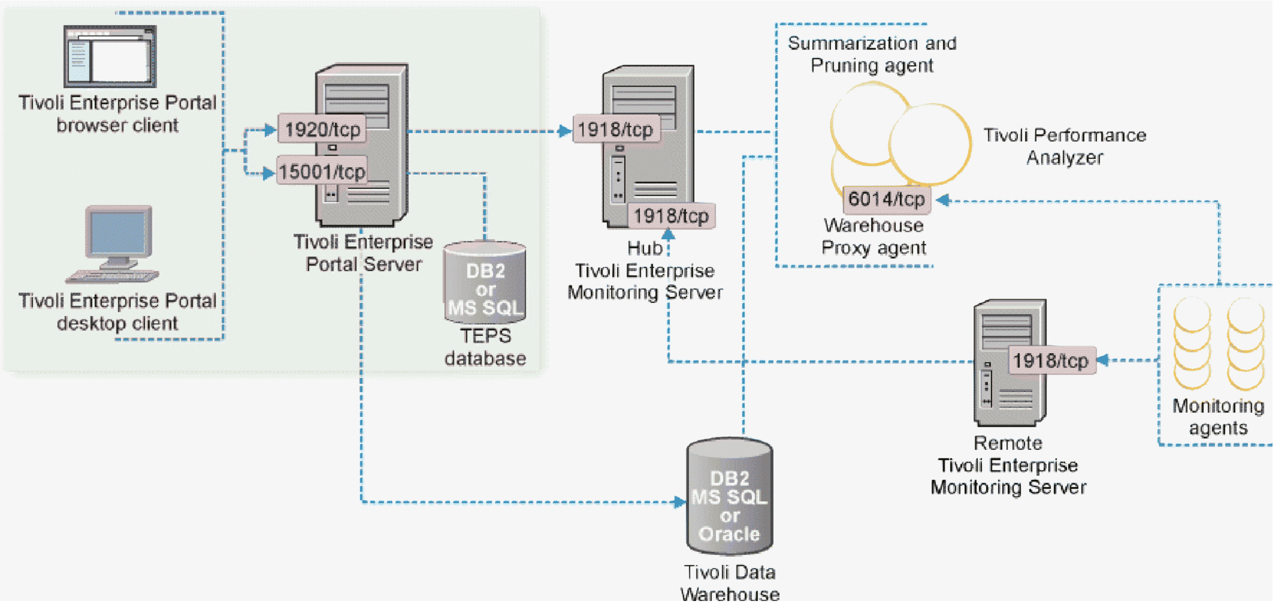

Tivoli Enterprise Portal browser client
Tivoli Enterprise Portal desktop client
1920/tcp
15001/tcp
Tivoli Enterprise Portal Server
DB2 or MS SQL
TEPS database
1918/tcp
1918/tcp
Hub Tivoli Enterprise Monitoring Server
Summarization and Pruning agent
Tivoli Performance Analyzer
6014/tcp
Warehouse Proxy agent
1918/tcp
Remote Tivoli Enterprise Monitoring Server
Monitoring agents
DB2 MS SQL or Oracle
Tivoli Data Warehouse

Figure 6. Tivoli Monitoring V6.2.3 communications model

The default port for SSL communication is 3660/TCP. (For non-SSL communication the default port is 1918/TCP.)

Determine if you require a firewall gateway

For most environments, using the firewall gateway is not required when deploying the Tivoli Monitoring software. However, in some cases, the firewall gateway is the only way to traverse the complex firewalls in a network. The following section describes the scenarios when the firewall gateway is required. In addition, the section outlines the optimal locations for the firewall gateway.

Use the firewall gateway for *any* of the following scenarios:

- A single TCP connection cannot be made to span between Tivoli Monitoring components. One example is when there are multiple firewalls between these components and a policy that does not allow a single connection to traverse multiple firewalls.
- Connection requirements do not allow the Tivoli Monitoring default pattern of connections to the hub Tivoli Enterprise Monitoring Server. One example is when agents that are located in a less-secure zone connect to the monitoring server located in a more-secure zone. Security policy allows a connection to be established only from a more-secure zone to a less-secure zone, but not the other way round.
- You must reduce open firewall ports to a single port or connection. For example, rather than opening the port for every system being monitored, consolidate the ports into a single *concentrator*.
- You must manage agent failover and Tivoli Enterprise Monitoring Server assignment symbolically at the hub monitoring server end of the gateway. Because gateway connections are made between matching service names, an administrator can change the failover and monitoring server assignment of downstream gateway agents by changing the client proxy bindings next to the hub monitoring server.

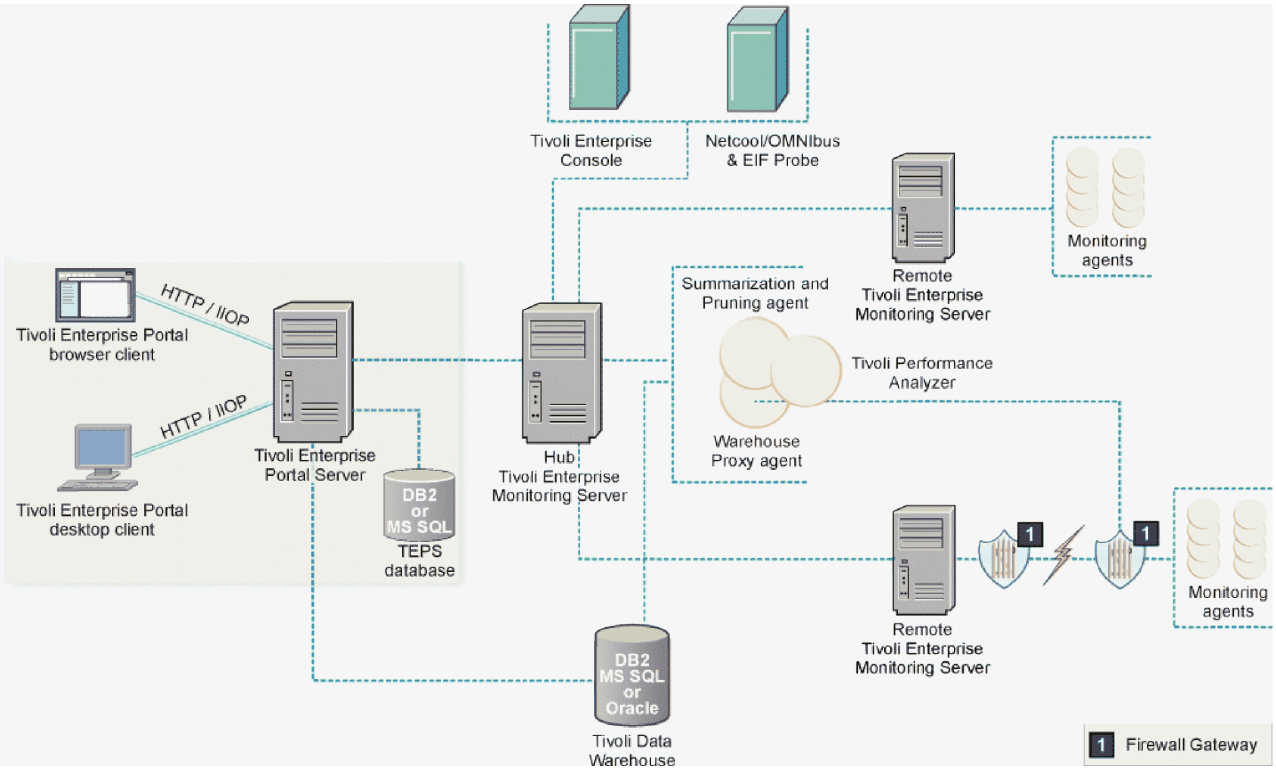
Network address translation (NAT) alone is not a reason to use the firewall gateway, which is content-neutral and can proxy any TCP connection. In most cases, NAT processing can be handled by the PIPE protocol (IP.PIPE or IP.SPIPE) without the firewall gateway.

For detailed information on installing and configuring the firewall gateway, see Appendix C, “Firewalls,” on page 799.

Determine where to place your Tivoli Monitoring components

There are a few factors to consider when determining the location of your Tivoli Monitoring components. The primary factors are firewalls and slow network connections. Before discussing the locations of these components in the data center, you must understand these components, the roles that they play, and what affects the load on these components.

The Tivoli Monitoring components and data paths to the Tivoli event management products are illustrated in Figure 7.



Tivoli Enterprise Console
Netcool/OMNIbus & EIF Probe
Monitoring agents
Remote Tivoli Enterprise Monitoring Server
Summarization and Pruning agent
Tivoli Performance Analyzer
Warehouse Proxy agent
HTTP / IIOP
Tivoli Enterprise Portal browser client
HTTP / IIOP
Tivoli Enterprise Portal desktop client
Tivoli Enterprise Portal Server
DB2 or MS SQL
TEPS database
Hub Tivoli Enterprise Monitoring Server
1
1
Monitoring agents
Remote Tivoli Enterprise Monitoring Server
DB2 MS SQL or Oracle
Tivoli Data Warehouse
1 Firewall Gateway

Figure 7. Tivoli Monitoring component architecture including firewall gateway

Tivoli Monitoring installation requires the following optional and mandatory components:

- "Tivoli Enterprise Monitoring Server" on page 37
- "Tivoli Enterprise Portal Server" on page 38
- "Tivoli Enterprise Portal client" on page 39
- "Warehouse Proxy Agent" on page 41
- "Warehouse Summarization and Pruning Agent" on page 41
- "Tivoli Data Warehouse" on page 42
- "Monitoring agent for IBM Tivoli Monitoring 5.x Endpoint" on page 42
- "Tivoli Enterprise Console integration" on page 42
- "Netcool/OMNIbus integration" on page 42
- "Firewall gateway" on page 43
- "IBM Tivoli Universal Agent" on page 43
- "Tivoli Performance Analyzer" on page 43

The specific hardware and software prerequisites for each of these components are listed in "Hardware and software requirements" on page 138.

Tivoli Enterprise Monitoring Server

The Tivoli Enterprise Monitoring Server, referred to as the monitoring server, is the first component installed to begin building the IBM Tivoli Monitoring Services foundation. The monitoring server is the key component on which all other architectural components directly depend. The monitoring server is the collection and control point for alerts received from agents. The monitoring server collects the performance and availability data of the agents.

The monitoring server is also responsible for processing *heartbeats* to track the online or offline status of monitoring agents. A monitoring agent sends a heartbeat every 10 minutes (this is configurable). If a heartbeat signal is not received from an agent within the heartbeat interval (plus a 3-minute grace period), the monitoring server considers the monitoring agent to be offline. The online or offline status of the monitoring agent is also called the *managed system status*.

In addition to the roles described in the previous paragraphs, the hub monitoring server also provides the following functions:

- Distributes situations down to the monitoring agents.
- Used for remote deployment and remote configuration and control of monitoring agents.
- Responsible for the processing of CLI and SOAP requests that can be used by users for automation of their Tivoli Monitoring environment.

There are two types of monitoring servers: the hub monitoring server and the remote monitoring server. The hub monitoring server is the focal point for the entire Tivoli Monitoring environment. The hub monitoring server is under a significant load. Work on the hub includes connections from the remote monitoring server, authentication, situations, policies, and workflows.

The hub monitoring server stores, initiates, and tracks all situations and policies and is the central repository for storing all active conditions and short-term data on every Tivoli Enterprise Monitoring Agent (monitoring agent). The hub monitoring server is also responsible for initiating and tracking all generated Take Action commands.

The monitoring server storage repository is a proprietary database format, referred to as the Enterprise Information Base (EIB), that is grouped as a collection of files located on the monitoring server. These files start with a file name prefix `qa1` and are located in the following directories:

- On **Windows**: *installation_dir*\cms
- On **UNIX and Linux**: *installation_dir*/tables/*tems_name*

where *installation_dir* specifies the Tivoli Monitoring installation directory and *tems_name* specifies the Tivoli Enrerprise Monitoring Server name.

Note: You can use the **CANDLEHOME** command on UNIX or Linux and **CANDLE_HOME** on Windows system to locate the home directory for Tivoli Monitoring.

Place the hub monitoring server inside the data center on a high-performance network (100 Megabits per second or higher). Connectivity between the Tivoli Enterprise Portal Server and hub monitoring server as well as between the hub monitoring server and most of the remote monitoring server must be fast and reliable. For large environments, use a multi-processor server. Some hardware configuration guidance is identified in “Sizing your Tivoli Monitoring hardware” on page 46.

The remote monitoring server is a collection point for the agents that are connected to that remote monitoring server. Certain types of situations run on the remote monitoring server. The load on the remote monitoring server is typically low. Load is driven higher if historical data is collected at the remote monitoring server instead of at the agents.

Placement of the remote monitoring server depends on a few factors. Plan your firewalls early to ensure that communications can be established between the Tivoli Monitoring components with only a modest number of holes in the firewall.

By locating the Warehouse Proxy Agent on the same computer as the remote monitoring server, you can negotiate NAT environments with their historical data collection. For remote locations connected over a slow network, place the remote monitoring server at the remote location if there are significant numbers of computers. For remote locations with just a few computers, it doesn't make sense to place a remote monitoring server at the remote location.

Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server, referred to as the portal server, is a repository for all graphical presentation of monitoring data. The portal server database consists of all user IDs and user access controls for the monitoring workspaces. The portal server provides the core presentation layer, which allows for retrieval, manipulation, analysis, and pre-formatting of data. The portal server manages data access through user workspace consoles.

The portal server keeps a persistent connection to the hub monitoring server, and can be considered a logical gateway between the hub monitoring server and the Tivoli Enterprise Portal client (portal client). Any disconnection between the two components immediately disables access to the monitoring data used by the portal client.

Locating the portal server in the same LAN segment as the hub monitoring server provides optimal performance when using the Tivoli Enterprise Portal graphical user interface (GUI). For users with multiple data centers, if the network connectivity is good, then one portal server is sufficient. If the network connection between data centers has significant latency, then additional portal servers can be placed at each data center.

Caution must be taken when using this approach because there can be only one read-write master portal server. Other portal servers must be used in a read-only manner. Read-only means that users must *not* create and edit objects like situations, workspaces, and policies. Customization must be replicated from the master portal server to the read-only portal server.

When you install the portal server, a proprietary integrated Web server is installed for use with the portal client in browser mode. Depending on the network topology and possible security implications, this can affect the construction of the solution. To avoid security issues, you may install an external Web server on the same computer where you installed the portal server. If there will be more than ten concurrent portal clients connected to the portal server, an external Web server can improve scalability of the portal server. For additional details, see “Configuring an external Web server to work with Tivoli Enterprise Portal” on page 400.

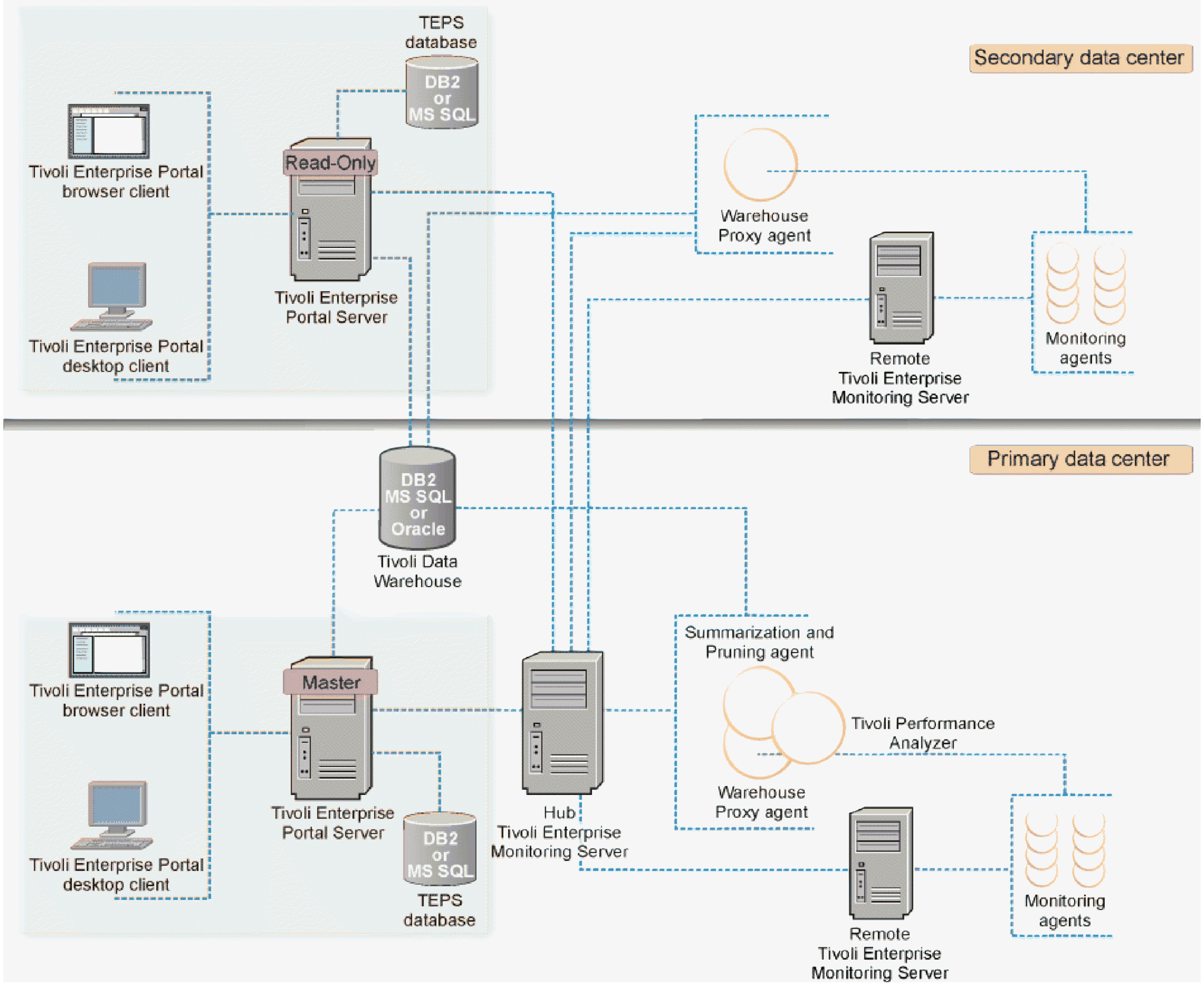

TEPS database
DB2 or MS SQL
Secondary data center
Tivoli Enterprise Portal browser client
Read-Only
Tivoli Enterprise Portal Server
Tivoli Enterprise Portal desktop client
Warehouse Proxy agent
Remote Tivoli Enterprise Monitoring Server
Monitoring agents
Primary data center
DB2 MS SQL or Oracle
Tivoli Data Warehouse
Summarization and Pruning agent
Tivoli Enterprise Portal browser client
Master
Tivoli Performance Analyzer
Tivoli Enterprise Portal Server
Hub Tivoli Enterprise Monitoring Server
Warehouse Proxy agent
Tivoli Enterprise Portal desktop client
DB2 or MS SQL
TEPS database
Monitoring agents
Remote Tivoli Enterprise Monitoring Server

Figure 8. Multiple data center environment

Tivoli Enterprise Portal client

The Tivoli Enterprise Portal client, also known as the portal client, is a Java-based user interface that connects to the portal server to view all monitoring data collections. The portal client is the user interaction component of the presentation layer. The portal client brings all of these views together in a single window so you can see when any component is not working as expected. The client offers three modes of operation, that all work with Java:

- Browser client
- Desktop client
- Java Web Start client

See Table 4 on page 51 for more information.

The browser client is automatically installed with the Tivoli Enterprise Portal Server (on the Web server that is integrated with the portal server). The desktop client is supported on Windows and Linux operating systems. You can install the desktop client from the IBM Tivoli Monitoring installation media or you can use IBM Web Start for Java to download the desktop client from the Tivoli Enterprise Portal Server. To find out what browsers are supported see the Software Product Compatibility Reports.

Using browser mode or Web Start clients allow you to perform maintenance updates in a single location. If the desktop client is installed from installation media, maintenance must be installed on each computer. If Web Start for Java is used to download and run the desktop client, you gain the performance advantage of the desktop client along with the convenience of centralized administration from the server. Additional performance gains can be made by modifying the Java heap settings. (For more details on the heap settings see “Locating and sizing the portal client” on page 51.) Unless you want a very secure environment where there are no downloads, use IBM Web Start for Java for obtaining the desktop client.

Note: To use Java Web Start to download the desktop client from the Tivoli Enterprise Portal Server, IBM Runtime Environment for Java, version 6.0 (also referred to as IBM JRE version 1.6) must be installed on the system to which you are downloading the client.

Many customers install the desktop client on Citrix for the following reasons. Citrix allows for better GUI performance for users at remote locations. In addition, some users do not allow the Tivoli Enterprise Portal client's prerequisite Java version to be installed on desktop computers. By using Citrix, users do not have to install Java on the user's desktop systems.

Tivoli Enterprise Monitoring Agents

Monitoring agents are installed on the system or subsystem that requires data collection and monitoring. Monitoring agents are responsible for gathering data on various properties, or *attributes*, of a monitored system, subsystem, application, or database, the *managed system*, and for sending that data to the monitoring servers. Monitoring agents test for specified conditions, or *situations*, by periodically comparing attribute values against specified thresholds. When the tested values match or exceed the thresholds, monitoring agents notify the monitoring server, and alert are displayed in the portal client.

The following scenarios prompt the monitoring server to gather data samples from the agents:

- Opening or refreshing a workspace that has data views (table or chart views):

  When a Tivoli Enterprise Portal workspace is opened or refreshed, the portal server sends a sampling request to the hub monitoring server. The request is passed directly to the monitoring agent, if there is a direct connection, or indirectly through the remote monitoring server to which the monitoring agent connects. The monitoring agent takes a data sampling and returns the results through the monitoring server and portal server to the portal workspace.
- The sampling interval for a situation (a test taken at your monitored systems):

  A situation can have a sampling interval as frequent as once every 30 seconds or as seldom as once every three months. When the interval expires, data samples are requested from the agent in which the returned values are compared with the condition described in the situation. If the values meet the condition, an event is generated and corresponding alerts or automation are initiated.
- Monitoring agents are configured to collect historical data:

  The collection of historical data can be configured to send the data to the remote monitoring server at regular intervals or, transfer data collections from the agent to the Warehouse Proxy Agent at hourly or daily intervals. If firewall restrictions are disabled or minimal, configure all of the agents to transfer directly to Warehouse Proxy Agent. If firewall security is an issue, you can either use the firewall gateway or place the Warehouse Proxy Agent on the remote monitoring server where a network connection is already established.

Typically, there are no unique considerations regarding the placement of monitoring agents. The main consideration is whether to store historical data on the agent or on the remote monitoring server. For environments with slow network connections between the agent and the remote monitoring server, storing the data on the monitoring server spreads out the network load of transmitting the historical data. Doing this places a higher demand on the remote monitoring server, and effectively lowers the number of agents that can be managed from the remote monitoring server.

For short-term historical requests of agent data less than 24 hours, the request does not have to cross the slow link, but this savings is offset by the remote monitoring server having to read a larger amount of data from disk (from all agents of that agent type connected to the remote monitoring server) to find the results for the desired agent. For environments with network events that cause agents to failover to a secondary monitoring server, this option may be a poor choice. When an agent fails over, it loses access to the short-term historical data because the data is located on the primary remote monitoring server and the agent is connected to the backup remote monitoring server.

By installing the operating system agent on the system first, the remaining agents can be deployed remotely using the **Add** agent capabilities.

Warehouse Proxy Agent

The Warehouse Proxy Agent is a unique agent that performs only one task: collecting and consolidating all historical data from the individual agents to store in the Tivoli Data Warehouse. If you are using the Tivoli Data Warehouse, at least one Warehouse Proxy Agent is required for each Tivoli Monitoring installation.

The Warehouse Proxy Agent uses ODBC (open database connectivity) on Windows and JDBC (Java Database Connectivity) on AIX and Linux to write the historical data to a supported relational database. The Warehouse Proxy Agent is typically placed on the same LAN segment as the warehouse database, which allows for the best throughput to the database. You can also place the Warehouse Proxy Agent on the same server as the warehouse database.

For larger environments or deployments with multiple data centers, use multiple Warehouse Proxy Agents. You can have as many as one Warehouse Proxy Agent per monitoring server in your monitoring environment. Placing the Warehouse Proxy Agent on the same server as the remote monitoring server reduces the number of servers running Tivoli Monitoring components. This placement also simplifies the configuration because the Warehouse Proxy Agent can be configured to service agents connected to the local monitoring server. If the server running the remote monitoring server goes down and the agent fails over to a secondary monitoring server, the agent should be able to upload historical data through the Warehouse Proxy Agent on the secondary monitoring server. This placement also has the advantage of limiting the number of agents that upload historical data through a single Warehouse Proxy Agent.

NAT environments require specific considerations. The agents receive the IP address of the Warehouse Proxy Agent from the monitoring server. In a NAT network, you must ensure that the agents receive the correct IP address of the Warehouse Proxy Agent. To ensure that the agents can communicate with the Warehouse Proxy Agent, place the Warehouse Proxy Agent on the same servers as the remote monitoring server. This placement is required only for the agents that are connected to a NAT network.

For information on configuring the historical collection, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.

Warehouse Summarization and Pruning Agent

The Summarization and Pruning Agent is a unique agent that performs the aggregation and pruning functions for the historical detailed data on the Tivoli Data Warehouse. The Summarization and Pruning Agent has advanced configuration options that enable customization of the historical data storage. One Summarization and Pruning Agent manages the historical data in the Tivoli Data Warehouse.

The Summarization and Pruning Agent can be placed on the Tivoli Data Warehouse database server to minimize network transmission delay during its processing. If the Summarization and Pruning Agent is placed on a separate server, ensure that it connects to the database server with a high-speed network connection of 100 Megabits per second or higher. For large environments, the database server must have at least four processors and a large number of disks, with maintenance by a skilled database administrator.

Performance enhancements were added into the Summarization and Pruning Agent for the IBM Tivoli Monitoring V6.2.1 release. Some of the enhancements require a new database schema. The warehouse will function without the schema changes, but will not take advantage of some of the performance improvements. For the enhancements that do not require a new schema, there are minor database changes that are documented in Chapter 19, “Schema Publication Tool,” on page 483. In a new Tivoli

Monitoring environment, the warehouse database schema will be created using the new schema. If the Tivoli Monitoring V6.2.3 environment was updated, the database will continue to use the old schema and will not benefit from the improvements. If you want to take advantage of the performance improvements in an upgraded environment, you will need to create a new warehouse database with a new schema. If desired, you can migrate the data from your old warehouse database into your new warehouse database.

Tivoli Data Warehouse

The Tivoli Data Warehouse is the storage database that contains all of the warehoused (long-term) historical data. A Warehouse Proxy Agent must be installed to leverage the Tivoli Data Warehouse function within the environment. In large-scale deployments, a Tivoli Data Warehouse can be shared among monitoring installations.

Note: If installing the warehouse database on a Microsoft SQL Server, you must also install the Tivoli Enterprise Portal Server on a Windows-based computer. This restriction applies even if the warehouse database and portal server are installed on separate computers. For example, a portal server on Linux does not support a warehouse database using Microsoft SQL Server.

Monitoring agent for IBM Tivoli Monitoring 5.x Endpoint

Also called IBM Tivoli Monitoring 5.x Endpoint Agent, this integration agent enables the collection and visualization of IBM Tivoli Monitoring 5.x resource models in the Tivoli Enterprise Portal. The visualization is the direct replacement for the Web Health Console.

Additionally, the agent provides the roll-up function of IBM Tivoli Monitoring 5.x metrics into the Tivoli Data Warehouse.

Tivoli Enterprise Console integration

Tivoli Enterprise Console events can be forwarded from Tivoli Monitoring V6.2.3 to Tivoli Enterprise Console Version 3.9. The events are forwarded from the hub monitoring server to the Tivoli Enterprise Console server or Tivoli Enterprise Console gateway. Make sure that firewall ports are opened between the hub monitoring server and Tivoli Enterprise Console servers. By default, the Tivoli Enterprise Console uses port 5529.

The Tivoli Enterprise Console event synchronization component sends updates to situation events back to the monitoring server and are then forwarded to the portal server. Actions performed at the Tivoli Enterprise Console for Tivoli Monitoring situations are reflected in the Tivoli Enterprise Portal Server, this is an optional component that must be installed on your Tivoli Enterprise Console server.

For information on forwarding events to Tivoli Enterprise Console and installing the event synchronization component, see Chapter 25, “Setting up event forwarding to Tivoli Enterprise Console,” on page 643.

Netcool/OMNIbus integration

If you are already using Netcool/OMNIbus to monitor events from other sources in your enterprise, you can also view and manage situation events from a Tivoli Enterprise Monitoring Server in the Netcool/OMNIbus console. Event integration requires Netcool/OMNIbus V7.2 or later and IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF version 10 or later.

Situation events are sent to Netcool/OMNIbus Probe for Tivoli EIF using the Tivoli Event Integration Facility (EIF) interface. The Netcool/OMNIbus EIF Probe receives events, maps them to the Netcool/OMNIbus events format, and then inserts into Netcool/OMNIbus ObjectServer. When an Netcool/OMNIbus user acknowledges, closes, or reopens a situation event, Netcool/OMNIbus sends those changes to the originating monitoring server through the event synchronization component.

For information on forwarding events to Netcool/OMNIbus and installing the event synchronization component, see Chapter 26, “Setting up event forwarding to Netcool/OMNIbus,” on page 675.

By default, the EIF Probe listens on the default port (9998).

Firewall gateway

Using the firewall gateway, you can traverse even the most complex firewalls. Using the IP PIPE or IP SPIPE protocols, the Tivoli Monitoring software can traverse most firewall configurations. In most cases, the firewall gateway is not necessary.

For detailed information on installing and configuring the firewall gateway, go to Appendix C, “Firewalls,” on page 799.

IBM Tivoli Universal Agent

The Tivoli Universal Agent is a customizable agent that allows you to monitor many different types of hardware, operating systems, and applications. The Tivoli Universal Agent has several different data providers that collect metric data. Some of the data providers gather data from a source running locally on the server (Script, File) while others (ODBC, SNMP, Socket, API, Post) collect data either locally or remotely.

From a networking and connection perspective, there are two aspects to the placement of a Tivoli Universal Agent. The Tivoli Universal Agent connects into a monitoring server just like any monitoring agent. You can use any of the supported protocols such as IP.UDP and IP.PIPE. In addition, for the data providers with remote monitoring capabilities, you must consider the network connection between the Tivoli Universal Agent and the monitored system. Each data provider uses a different access method, so you must consider different ports and protocols.

For optimal performance, place the Tivoli Universal Agent in close LAN proximity to the computer that is gathering the data. Some Universal Agents are very lightweight and can be configured to remotely monitor over slower WAN links. Other Universal Agents gather large volumes of data and can be run only on a LAN environment. As you create your custom monitoring solution, calculate the expected network traffic by examining the metrics being collected and the data collection interval.

For more information about the Tivoli Universal Agent, see the *IBM Tivoli Monitoring: Agent Builder User's Guide*.

Tivoli Performance Analyzer

Tivoli Performance Analyzer adds predictive capability to Tivoli Monitoring so you can monitor resource consumption trends, anticipate future performance issues, and avoid or resolve problems more quickly. For example, you can use Tivoli Performance Analyzer to predict application bottlenecks and create alerts for potential service threats.

Tivoli Performance Analyzer helps IT managers to answer the following questions so they can optimize IT capacity:

- When will my application fail to meet service levels?
- How will application performance change if I modify the infrastructure?
- What is the best hardware solution to meet my performance and cost goals?
- Where are my under-utilized servers and networks?
- Which servers and network components really need an upgrade?
- Which application will experience the next performance issue? When?

Providing accurate IT forecasts and appropriate IT capacity to meet business needs are two major goals in the capacity management area. You can use the following key performance indicators (KPIs) to measure the critical success factors:

- Total value of unplanned or unused capacity expenditures
- Percent of capacity forecasts that were accurate
- Number of inaccurate business forecast inputs provided
- Number of incidents related to capacity or performance issues

IBM Tivoli Agent Builder

The Agent Builder, introduced in IBM Tivoli Monitoring V6.2, is a wizard that makes it easier to build custom monitoring solutions. Using mostly point and click capabilities, the Agent Builder allows you to create a monitoring agent using multiple data providers such as WMI, Perfmon, Log scraping, scripts, and process monitoring. Over time additional data providers will be added to the Agent Builder.

Monitoring agents created using the Agent Builder have two advantages over agents based on the Tivoli Universal Agent. First, the Agent Builder monitoring agents tend to run with lower CPU utilization than an equivalent Tivoli Universal Agent. Second, you do not need to worry about Tivoli Universal Agent versioning.

For more information about Agent Builder, see the *IBM Tivoli Monitoring: Agent Builder User's Guide*.

Note to Windows users: Since the introduction of the Embedded Java Runtime and the Tivoli Enterprise Services User Interface Extensions (the KUE component) at IBM Tivoli Monitoring V6.2.2, the possibility has existed that, on nodes where only the Windows OS agent was installed, the Embedded Java Runtime and the User Interface Extensions were not also installed. If you later attempt to install an Agent Builder agent on such nodes, you may receive the error shown in Figure 63 on page 258. If this happens, complete the procedure outlined in “Installing the Embedded Java Runtime and the User Interface Extensions” on page 258, and retry the agent installation.

Additional ports used in the Tivoli Monitoring environment

If multiple components are installed on the same server, they may not share the same port number for IP PIPE or IP SPIPE communications. A scheme has been created to use additional ports for communications. The following section identifies the additional ports that are used.

In a typical environment, a user might have three monitoring agents that are located on a system. Each monitoring agent must use a unique port to ensure there are no communications conflicts. Tivoli Monitoring uses a methodology involving SKIP and COUNT to assign the ports. On a typical server with three monitoring agents, Tivoli Monitoring assigns ports 1918, 6014, 10110 to the three monitoring agents. The first monitoring agent to start uses port 1918, the second will use 6014, and the third will use 10110. If additional monitoring agents are installed and started, there will be additional ports assigned to those monitoring agents. A detailed explanation of the port assignments is described in the following paragraphs.

In environments behind firewalls it is critical that the Warehouse Proxy Agent listens on the same port every time the monitoring agent is started. This setup allows you to specify a port such as 6014 for communications between the monitoring agents and the Warehouse Proxy Agent. To guarantee that port 6014 is available for the Warehouse Proxy Agent, all monitoring agents running on the same server must use the Skip Count mechanism to specify a specific port. This ensures that port 6014 is available to the Warehouse Proxy Agent no matter what order the monitoring agents start.

Note: You are not required to use port 6014 for the Warehouse Proxy Agent, but this is the most commonly used port.

Understanding COUNT and SKIP options

COUNT:*N* is the mechanism for reserving IP.PIPE ports, where *N* is the number of IP.PIPE ports to reserve on a host, in addition to the monitoring server well-known port. COUNT:*N* accommodates multiple Monitored components on a single host in addition to the monitoring server. Monitoring servers with COUNT:*N* configured are assumed to require one of the *N* reserved ports. Monitoring components with SKIP:*N*+1 are assumed to not require one of the *N* reserved ports.

Note: When the IP.PIPE is referenced in this guide, the component uses TCP. If your site is using IP.PIPE, be aware of the following limitations:

- There can be at most 16 IP.PIPE processes per host.
- IP.PIPE uses one, and only one, physical port per process. Port numbers are allocated using a well-known port allocation algorithm. The first process for a host is assigned port 1918, which is the default.
- **KDC_PORTS** is not supported for IP.PIPE.

If **KDC_FAMILIES=IP.PIPE COUNT:1**, then you expect to allocate port (*the Well-known-port + (4096 * (1))*) or fail.

If **KDC_FAMILIES=IP.PIPE COUNT:2**, then you expect to allocate port (*the Well-known-port + (4096 * (1))*) or (*the Well-known-port + (4096 * (2))*) or fail.

If you need to reserve *N* ports (in addition to the well-known monitoring server port), then all servers on this host that are not permitted in the *reserved* range should specify SKIP:*N*+1 and all the servers on this host permitted in the *reserved* range should specify COUNT:*N*.

If you have **KDC_FAMILIES=IP.PIPE SKIP:1**, then the first port you want to use is (*Well-known-port + 4096 * (1)*) and you continue to try until you contact one or run out of ports to try.

If you have **KDC_FAMILIES=IP.PIPE SKIP:2**, then the first port you want to use is (*Well-known-port + 4096 * (2)*) and you continue to try until you contact one or run out of ports to try.

As an example, assume that you have the following scenario:

- A Windows system inside a firewall.
- You expect to run a monitoring server, monitoring agent, and a Warehouse Proxy Agent on this system.
- The monitoring server and the Warehouse Proxy Agent must be accessible from outside the firewall. Accessibility from outside the firewall means that you require IP.PIPE ports that must be permitted at the firewall, thus, these ports must be predictable.

Given this scenario, and to allow for the accessibility, one IP.PIPE port requires reservation and firewall permission in addition to the monitoring server. The monitoring server always receives the well-known port by default. The Warehouse Proxy Agent requires **KDC_FAMILIES=IP.PIPE COUNT:1** and the monitoring agent requires **KDC_FAMILIES=IP.PIPE SKIP:2** to make the necessary reservations for the Warehouse Proxy Agent. And if the monitoring server's well-known port is 1918, then the Warehouse Proxy Agent is assigned port (1918 + (4096 *(1)) 6014. The monitoring agent attempts to listen on port (1918 + (4096 *(2)) 10110.

The Warehouse Proxy Agent port is reserved only on 6014 if keyword **KDC_FAMILIES** is used in conjunction with the following COUNT option:

- COUNT specifies the number of offsets that the agent can use to retrieve a reserved port number.
- COUNT:*N* means that all offset ports calculated by the following formula starting from 1 up to *N* are allowed and the first free one is used:

  Well-known port 1918 + (*X* * 4096) for *X* ` {1,..,*N*}
- COUNT:1 means that only 1918+(1*4096)=6014 is used, thus the proxy port is fixed. For example:

  ```
  KDC_FAMILIES=IP.PIPE COUNT:1 PORT:1918 IP use:n SNA use:n IP.SPIPE use:n
  ```

If other Tivoli Monitoring components, such as the Tivoli Universal Agent, portal server, and other components, are running on the same system, the SKIP option must be used to instruct these components to skip some reserved ports:

- SKIP:*N*+1 means that only offset ports starting from multiplier *N*+1 are allowed:

  Well-known port 1918 + (*X* * 4096) for *X* ` {*N*+1,*N*+2,..,max_port_high}
- SKIP:2 allocates 1918+(2*4096)=10110 as the first port and there is no proxy port conflict. For example:

  ```
  KDC_FAMILIES=IP.PIPE SKIP:2 PORT:1918 IP use:n SNA use:n IP.SPIPE use:n
  ```

Configuring your firewalls

IBM Tivoli Monitoring has a firewall gateway component that allows you to traverse even the most complex firewalls. Figure 6 on page 35 outlines the ports and communications that must occur between the various Tivoli Monitoring components, and shows an example for a typical environment using IP.PIPE. If you are using IP.SPIPE, the diagram would look similar, but port 3660 would be used instead of 1918. The same skip counts apply as described in “Additional ports used in the Tivoli Monitoring environment” on page 44, but the initial port number is 3660.

Sizing your Tivoli Monitoring hardware

The following section outlines hardware scenarios for Tivoli Monitoring environments of various sizes. These are rough guidelines based on typical deployments. Because Tivoli Monitoring is highly customizable, this section also outlines some usage scenarios that drive additional load and might indicate different hardware requirements.

Part 3, “Installation and initial configuration of base components and agents,” on page 123 has a section describing the hardware requirements for the Tivoli Monitoring infrastructure components, including the processor, memory and disk requirements. The guidelines below are based on typical deployments, and supplement the hardware requirements described in “Hardware and software requirements” on page 138.

Locating and sizing the hub Tivoli Enterprise Monitoring Server

Always locate the hub monitoring server in a data center with good network reliability and throughput. Network connection speeds of 100 Megabits per second or higher are typically sufficient.

Except where noted, the server sizings assume that the hub monitoring server is installed on a separate server with no additional Tivoli components. Below are some basic guidelines. For detailed guidelines, see “Hardware and software requirements” on page 138.

Sizing your hub monitoring server hardware

Large environments (>2000 agents)

Run the hub monitoring server on its own fast dual-processor server. The server requires a minimum of 1 GB of total system memory. Intel processors should be 3 GHz or higher; RISC processors should be 1.5 GHz or higher.

Medium-sized environments (500-2000 agents)

Although the steady-state CPU utilization is usually low, the hub monitoring server uses a significant amount of system resources in processing transient workloads, such as startup, agent login, and situation distribution. A single processor system is adequate for the hub monitoring server, but a dual processor server can help during peak transient loads. It is reasonable to run other Tivoli Monitoring components on the same server such as the portal server, but allocate additional CPUs for the additional components. Assign a minimum of 1 GB of total system memory to the hub monitoring server.

Small environments (200-500 agents)

It is reasonable to combine components such as the monitoring server and portal server on a dual-processor server. With more than 200 agents, install one or more remote monitoring servers and offload some of the work of the hub monitoring server. When combining monitoring components on a single server, the different components' process memory requirements should be added together. See “Memory and disk requirements” on page 154 for more information.

Very small environments (<200 agents)

For environments with 200 or fewer agents, most of the components can be combined onto a single server. If you are going to use a small monitoring environment, choose a multiprocessor (2 or 4-way) for the monitoring server with at least 2 GB of total system memory. Configure the software and begin monitoring CPU and memory usage during periods of both normal and high volumes of situation events. If CPU or memory usage is constrained, consider deploying a separate server for the Tivoli Data Warehouse, the Warehouse Proxy Agent, and the Summarization and Pruning Agent. For more information see “Memory and disk requirements” on page 154.

When to use multiple hub monitoring servers

As environments grow in size, you must run multiple hub monitoring servers. The maximum limit for the number of agents for each hub monitoring server environment is 10000.

There are many factors that affect the scalability of a hub environment. The primary factors are network latency, number of situations running at the hub, the number of concurrent active users, and historical data collection. For environments approaching 10000 agents, review with the Application, Availability, and Business Service Management Solutions Enablement Best Practices team or the Tivoli Performance team by sending an e-mail to absmenbl@us.ibm.com.

Note: When multiple hub monitoring servers are used, correlation of events can be performed by using either Tivoli Enterprise Console or Netcool/OMNIbus.

Locating and sizing the remote Tivoli Enterprise Monitoring Server

When determining the location of your remote monitoring server, you must consider the number of connected agents, existence, number, and placement of firewalls, and network bandwidth. With the Tivoli Monitoring V6.2.3 release, a single remote monitoring server can support up to 1500 agents.

Although the steady state CPU utilization is usually low, a remote monitoring server uses a significant amount of system resources in processing transient workloads, such as startup, agent login, and situation distribution. A single processor system is adequate for a remote monitoring server, but a dual processor server can help during peak transient loads. To support 1500 agents, a server with 1.5 GB of memory is typically sufficient. If a Warehouse Proxy Agent will be running on the same server as the remote monitoring server, a dual-processor server with 2GB of memory should be used.

If the remote deployment capability is used, ensure that the remote monitoring server has sufficient network capabilities (100 Megabits per second or higher is typically sufficient). Because the remote deployment packages are large, in some cases hundreds of megabytes, you must ensure that you do not have a network bottleneck.

The network connection speed between the monitoring server and the monitoring agent will affect the elapsed time for remote deployment operations. One way to deal with slow network links is to deploy a remote monitoring server near the monitored servers. By co-locating the remote monitoring server with the agents, the Remote Deploy bundles need to be transferred across the slow link only once. Distribution to the agents can then be done on a high speed LAN segment.

If you use the Warehouse Proxy Agent with the remote monitoring server, ensure that you have sufficient network bandwidth, memory, and CPU capacity for both loads. The network load from the Warehouse Proxy Agent is described in “Locating and sizing the Warehouse Proxy Agent” on page 49.

Locating and sizing the remote deployment depot

Note: Remote deployment is possible only for monitoring agents that run on distributed platforms (Windows, UNIX, and Linux computers).

When you use the remote deployment for agents and patches, place the remote deployment depot on a separate directory from your Tivoli Monitoring installation directory. This allows for cleaner backup and restore operations. The remote deployment depot requires significant disk space, which varies depending on the number of agents and the number of platforms you use. Typically, allocate at least 2 GB of disk space for the depot.

Note: If you create a shared deployment repository named `depot` on the server hosting the deployment depot and you create this repository in a subdirectory of the `depot` directory, the monitoring server will not be able to find your deployment depot. Instead:

1. Create the repository at the `C:\IBM\ITM\CMS` level of the directory structure, *not* at the `C:\IBM\ITM\CMS\depot` level.
2. Then set the DEPOTHOME keyword to `DEPOTHOME=\\hubtems\centralrepository\depot` in the KBBENV file.

Otherwise you will get this error message:

```
KDY2077E: The specified agent bundle depot \\hubtems\depot is not a directory.
Either the agent bundle depot directory does not exist or it is not a directory.
The agent bundle depot directory does not exist because no bundles have been added.
```

One of the most important aspects of using remote deployment is ensuring that the files that are distributed are at the correct version. In large environments, there might be several remote monitoring servers that require maintenance to ensure that all of the depots are updated. To avoid the necessity of installing updates and maintenance on multiple computers, consider creating one depot and making it available using NFS or Windows shares.

If you decide to use a shared depot, make sure that the remote monitoring server is upgraded before you upgrade the agents that are connected to that remote monitoring server. When you deploy monitoring agents, make sure that you configure them to connect to the desired remote monitoring server.

Notes:

1. In addition to remotely deploying monitoring agents, as of V6.2.2, IBM Tivoli Monitoring lets you remotely deploy *non-agent bundles* (with which your site can employ Tivoli Monitoring components that need not connect to a Tivoli Enterprise Monitoring Server).
2. Remote deployment is not available for z/OS-based OMEGAMON agents, nor is it supported on nodes running the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or the Tivoli Enterprise Portal desktop or browser client.

Locating and sizing the Tivoli Enterprise Portal Server

Install the portal server on a very fast (at least 3 GHz for Intel, 1.5 GHz for RISC) server that has 2 to 4 GB of total system memory. For small environments with fewer than five concurrent users, a single-processor server is sufficient. With more than five users, use a dual-processor server. The Tivoli Monitoring V6.2.3 release has been tested with 50 concurrent users. For environments with more than ten concurrent portal clients, consider using an external Web server to increase the scalability of the portal server. See the “Tivoli Enterprise Portal Server” on page 426 for more details. For information on configuring the portal server to work with an external Web server, see “Configuring an external Web server to work with Tivoli Enterprise Portal” on page 400.

For optimal performance, place the portal server on the same LAN segment as the majority of your portal clients. Depending on firewall restrictions the location of the portal server has to be verified. If clients are not allowed to connect to a central segment they are not able to use the portal server.

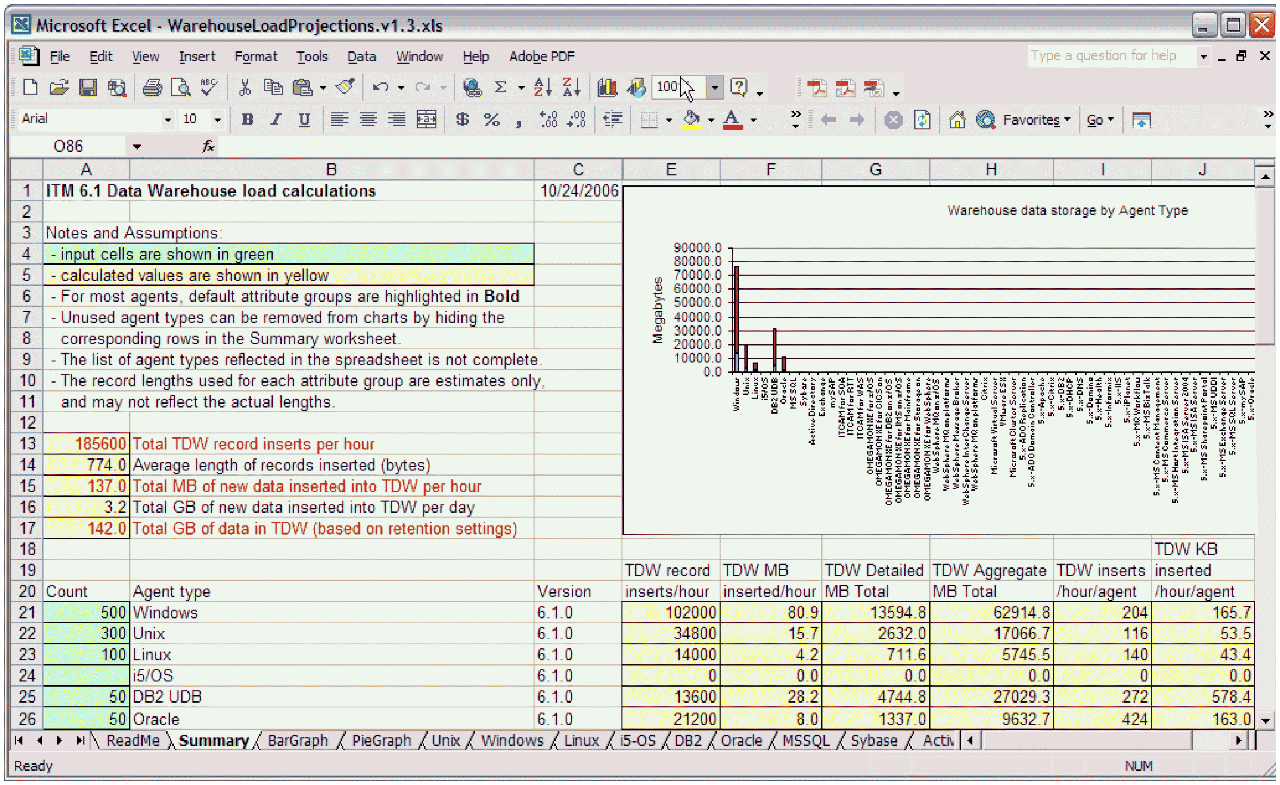
If you have multiple large data centers you might need to install a second read-only portal server. If you choose to use this option, be aware that only one portal server can be used to make customization changes such as editing workspaces and situations. See the section on high availability for details on how to configure multiple portal servers.

For more information see “Memory and disk requirements” on page 154.

Locating and sizing the Warehouse Proxy Agent

The Warehouse Proxy Agent is driven by the amount of historical data being collected. The Warehouse load projection spreadsheet can help you estimate both the number of database inserts and the volume of data being collected in the warehouse database.

You can find the Warehouse load projection spreadsheet in the Tivoli Integrated Service Management Library by searching for "warehouse load projections" or the navigation code "1TW10TM1Y" at http://www.ibm.com/software/tivoli/opal.



	A	B	C	E	F	G	H	I	J
1	**ITM 6.1 Data Warehouse load calculations**		10/24/2006						
2									
3	Notes and Assumptions:								
4	- input cells are shown in green								
5	- calculated values are shown in yellow								
6	- For most agents, default attribute groups are highlighted in **Bold**								
7	- Unused agent types can be removed from charts by hiding the								
8	corresponding rows in the Summary worksheet.								
9	- The list of agent types reflected in the spreadsheet is not complete.								
10	- The record lengths used for each attribute group are estimates only,								
11	and may not reflect the actual lengths.								
12									
13	185600	Total TDW record inserts per hour							
14	774.0	Average length of records inserted (bytes)							
15	137.0	Total MB of new data inserted into TDW per hour							
16	3.2	Total GB of new data inserted into TDW per day							
17	142.0	Total GB of data in TDW (based on retention settings)							
18									TDW KB
19				TDW record	TDW MB	TDW Detailed	TDW Aggregate	TDW inserts	inserted
20	Count	Agent type	Version	inserts/hour	inserted/hour	MB Total	MB Total	/hour/agent	/hour/agent
21	500	Windows	6.1.0	102000	80.9	13594.8	62914.8	204	165.7
22	300	Unix	6.1.0	34800	15.7	2632.0	17066.7	116	53.5
23	100	Linux	6.1.0	14000	4.2	711.6	5745.5	140	43.4
24		i5/OS	6.1.0	0	0.0	0.0	0.0	0	0.0
25	50	DB2 UDB	6.1.0	13600	28.2	4744.8	27029.3	272	578.4
26	50	Oracle	6.1.0	21200	8.0	1337.0	9632.7	424	163.0

Figure 9. Warehouse load projection spreadsheet

Figure 9 shows a sample spreadsheet summary created using the tool downloaded from the Tivoli Integrated Service Management Library site.

The amount of network traffic closely matches the amount of data collected in the Warehouse. The key planning numbers, highlighted in red, are based on sample data. Each Tivoli Monitoring environment is different, so fill out the spreadsheet based on your warehousing needs.

For some high-volume metrics, consider collecting only short-term historical data. For example, if you want to collect process data, there is one row of data per monitored process for every collection interval, which generates a significant amount of data. By retaining only 24 hours of short-term historical data, you do not overload your Warehouse server or network, but you can still perform trending analysis.

If historical data collection is started but a warehousing interval is not set, care must be taken to ensure that the local historical files do not grow indefinitely. This is only for distributed systems. For z/OS, Tivoli Management Services provides automatic maintenance for the data sets in the persistent data store (the dataset in which the short-term historical data is stored).

The key information in the spreadsheet includes the following data:

- Total Tivoli Data Warehouse Inserts per hour

  In this example, there are 185,600 inserts per hour. Most reasonable database servers can handle this rate without any problem.
- Total megabytes of new data inserted into Tivoli Data Warehouse per hour

  In the example, 137 MB of data per hour are inserted. This amount roughly translates to 137 MB of network traffic per hour coming into the Warehouse Proxy Agent from the connected agents, and roughly 137 MB per hour of network traffic from the Warehouse Proxy Agent to the warehouse database server.
- Total gigabytes of data in Tivoli Data Warehouse (based on retention settings)

  In this example, you expect the Tivoli Data Warehouse database to grow to 143 GB of data. Additional space must be allocated for the database logs. The database does not grow beyond 143 GB because you are summarizing the raw (detailed) metrics and pruning both the detailed and summarized data. The planning spreadsheet helps you determine your warehouse size based on your retention needs.

This warehouse database must be carefully planned and tuned. Where possible separate the largest tables into separate table spaces or data files. By filling in the Warehouse load projection spreadsheet, most users will find that 3 to 5 tables make up the majority of their warehouse data. These tables should be isolated from each other as much as possible so that they have optimal performance.

Multiple Warehouse Proxy Agents are required when the number of agents collecting historical data increases above approximately 1500 (to ensure that limits for the number of concurrent connections are not reached). If you use multiple Warehouse Proxy Agents, consider running them on the same servers running the remote monitoring servers, and configuring them to support agents connected to this monitoring server. This approach consolidates the Tivoli Monitoring infrastructure components, and limits the number of agents that connect to each Warehouse Proxy Agent.

For server sizing, see “Locating and sizing the Summarization and Pruning Agent.”

Locating and sizing the Summarization and Pruning Agent

The Summarization and Pruning Agent provides the ability to customize the length of time for which to save data (pruning) and how often to aggregate detailed data (summarization) in the Tivoli Data Warehouse database. Configuration of the Summarization and Pruning Agent can have a significant impact on the performance of your Warehouse. Install the Summarization and Pruning Agent on the warehouse database server. If that is not possible, ensure that low latency and high network bandwidth exists between your Summarization and Pruning Agent and the warehouse database.

Table 3 shows guidance for setting up the Summarization and Pruning Agent based on the amount of data inserted into the warehouse database per day. The server sizing is based on the warehouse database and Summarization and Pruning Agent running on the same server, and the Warehouse Proxy Agent running on a separate server. The assumption is that you have new and fast server-class hardware because the warehouse database server is a high-performing server.

For smaller environments, use a minimum of three dedicated disks because of the high disk I/O requirements of the warehouse database server.

Table 3. Warehouse database server considerations

Data inserted per day	Number of CPUs	Memory	Hard disk storage
0 to 100 K inserts per day	1 CPU	2 GB	3 or more dedicated SCSI drives
100 K to 500 K inserts per day	2 CPUs	2 GB	4 or more dedicated SCSI drives
0.5 to 2 Million inserts per day	2 CPUs	4 GB	RAID Array with 4 or more disks
2 to 10 Million inserts per day	4 CPUs	4 GB	Multiple RAID Arrays with 5 or more disks. Or, SAN storage

Table 3. Warehouse database server considerations (continued)

Data inserted per day	Number of CPUs	Memory	Hard disk storage
10 to 20 Million inserts per day	4 CPUs	8 GB	RAID Arrays with 15 to 20 dedicated disk drives or high-performance SAN storage
> 20 Million inserts per day	4 CPUs	8 GB	Multiple RAID Arrays with 20 to 25 dedicated disks or high-performance SAN storage

If your Summarization and Pruning Agent is installed on the warehouse database server, configure the agent so that it leaves some processing power for others to perform Warehouse queries and for the Warehouse Proxy Agent to insert data. Set the number of worker threads to 2 to 4 times the number of CPUs on the system where the Summarization and Pruning Agent is running. Start with 2 times the number of CPUs and then increase the number to see if it continues to improve your performance.

To set the number of worker threads, edit the Summarization and Pruning Agent configuration file (`KSYENV` on Windows, `sy.ini` on UNIX or Linux) and set the **KSY_MAX_WORKER_THREADS** to the number of desired threads. This parameter can also be set using the configuration dialog panels.

To assess the performance and throughput of the Summarization and Pruning Agent for your environment, you can use the following approach:

1. Start by enabling historical data collection for a small set of attribute groups which do not generate a large number of rows per data collection interval
2. Examine the Summarization and Pruning Agent Java log to see how many detailed records were read and pruned in a processing cycle, and the elapsed time for the processing cycle. Dividing the number of records read and pruned by the elapsed time will give you a rough measurement of the Summarization and Pruning Agent throughput (records read and pruned per second).
3. As long as the elapsed time for the Summarization and Pruning Agent processing cycle is acceptable, considering enabling historical data collection for additional attribute groups and repeat step 2.

The processing throughput as determined in step 2 is a rough and easily calculated measurement of the Summarization and Pruning Agent performance. Database tuning or additional worker threads can improve the throughput. See the “Tivoli Data Warehouse” on page 434 for more tuning information.

Locating and sizing the portal client

The portal client has three different deployment alternatives which are described in Table 4 below.

Table 4. Portal client deployment considerations

Portal client	Advantages	Disadvantages
Browser client	• Does not need to be installed on client machine. • No need to for maintenance to be applied for each client user. • Workspaces can be referenced using URLs.	• Slow initial download. • Requires tuning of Java settings.

Table 4. Portal client deployment considerations (continued)

Portal client	Advantages	Disadvantages
Desktop client	• Faster startup and performance over browser client. • Supported on Linux and Windows.	• Needs to be installed on each client machine. • Requires maintenance to be installed individually on each client machine. • Mismatch of versions between portal server and the client is not permitted.
Java Web Start client	• Similar performance to desktop client. • Faster download startup time than Browser client. • Supports multiple JREs. • Centralized administration for maintenance.	

- Try using the browser client first to see if it meets your response time needs. If you are using the browser client, it is ***imperative*** that you increase the Java heap size parameters for the Java Plug-in.
- Using the desktop client with Java Web Start may reduce response time for new workspace requests by about one second. If this additional response time reduction is important to you, then consider using Java Web Start with the desktop client. See “Using Web Start to download and run the desktop client” on page 321 for instructions on how to set this up.

The memory required by the portal client depends on the size of the monitoring environment and on the Java heap size parameters. The default maximum Java heap size is 256 MB for the desktop client. Using this default heap size and a medium to large monitoring environment, the portal client can be expected to use approximately 350 MB of memory.

For the browser client, it is imperative that the default Java plug-in parameters be increased. The preferred starting value for small to medium environments is **–Xms128m –Xmx256m** and **–Xms256m –Xmx512m** for larger environments. When modifying the maximum Java heap size, make sure there is adequate physical memory for the entire Java heap to be contained in memory. For more information see “Tuning the portal client JVM” on page 430.

You can also use the portal desktop client and leverage Java Web Start to minimize your maintenance costs. The portal desktop client requires less memory and offers better performance characteristics.

Locating and sizing the Tivoli Performance Analyzer

The Tivoli Performance Analyzer can be installed onto a stand-alone workstation, onto the same workstation as the Tivoli Monitoring environment if it is installed on a single machine, or onto the same workstation as any other Tivoli Monitoring element in a distributed environment. Before installing Analytic Agent make sure that your Tivoli Monitoring environment includes the listed products, that they are configured and that you have administrator access to the following components:

- Tivoli Data Warehouse
- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Summarization and Pruning Agent

Note: Tivoli Performance Analyzer analyzes summarized data. Therefore, the Summarization and Pruning Agent must be installed and active so that data is available for the agent to analyze.

- Warehouse Proxy Agent

Note: You must only install one instance of Tivoli Performance Analyzer for each Tivoli Monitoring environment.

The analytical calculations performed by Tivoli Performance Analyzer use the data in Tivoli Data Warehouse. Your Tivoli Monitoring installation must therefore be configured for History Collection and Warehouse Summarization and Pruning.

Elements to install

There are several distinct elements that must be installed to add Tivoli Performance Analyzer into your Tivoli Monitoring environment:

- Performance Analyzer Agent
- Tivoli Enterprise Monitoring Server Support
- Tivoli Enterprise Portal Server Support
- Tivoli Enterprise Portal Support

Note: If your Tivoli Enterprise Portal Support is at V6.2.3, then you must install V6.2.3 of Tivoli Performance Analyzer. If you attempt to configure an older version of Tivoli Performance Analyzer in the Tivoli Enterprise Portal, a database connectivity error message is displayed.

Support files must be installed on the same workstation as the components they are supporting. For example, Tivoli Enterprise Monitoring Server support should be installed on the same machine as Tivoli Enterprise Monitoring Server.

For information on installing the domain support to enable performance analytics for a broader set of systems, see "Installing domain definitions for Tivoli Performance Analyzer" on page 289.

Deployment scenario

Table 5 shows an example distributed installation of Tivoli Performance Analyzer. The agent is installed on the same workstation as Tivoli Data Warehouse.

Table 5. Example distributed installation

Machine 1	Machine 2	Machine 3
Tivoli Enterprise Monitoring Server	Tivoli Enterprise Portal Desktop Client	Analytic Agent
Tivoli Enterprise Portal Server		IBM Tivoli Data Warehouse

Table 6 shows the support files that must be installed on each workstation in the example distributed installation shown in Table 5.

Table 6. Support files to be installed

Machine 1	Machine 2	Machine 3
Tivoli Enterprise Monitoring Server Support	Tivoli Enterprise Portal Desktop Client Support	
Tivoli Enterprise Portal Server Support		
Tivoli Enterprise Portal Browser Client Support		

Software Product Compatibility Reports (SPCR)

When planning an installation, it is important to review the list of supported platforms to ensure that you are running on a supported operating system and database version. The Software Product Compatibility Reports that contain the current list of supported platforms can be found at http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html.

Configuring for high availability and disaster recovery

Among the most important considerations in setting up your Tivoli Monitoring environment is ensuring high availability of the product components and being able to recover quickly from failures. There are multiple components to consider when discussing high availability. In the following sections, each of the Tivoli Monitoring components is discussed, along with a strategy for achieving the desired level of high availability and disaster recovery. Ensuring high availability involves achieving redundancy for every Monitoring component. Disaster recovery means being able to recover from a major outage such as a data center going offline or losing its WAN link.

Configuring the hub monitoring server for high availability and disaster recovery

The hub monitoring server is a highly reliable component, and many users choose to run with a single hub monitoring server and use backup and restore operations to ensure that they have minimum downtime in case of a hardware failure. Other users require higher availability and less downtime and employ multiple hub monitoring servers to achieve either a high availability (HA) environment, disaster recovery (DR) environment, or a combination (high availability and disaster recovery) environment. The following section describes some of the strategies used to achieve the desired level of availability and downtime.

If you have a smaller environment and do not want to invest in additional hardware, you can set up a single hub monitoring server. Because the hub monitoring server is very reliable, there is no need to purchase any additional hardware. You can expect some downtime when patching the hub monitoring server. There are times when the monitoring server must be recycled. If you use one hub, it is important that you use a good backup and restore strategy. You can install the hub in a virtualized environment such as VMWare so that you can quickly bring up an identical virtual operating system to replace the original. In addition, there is a VMWare HA option with release 3.0.x that automates the start of a failing image on a different node.

If you want to achieve high availability, you have two options. The first option is to implement the Hot Standby feature that is built into the monitoring server. Extensive large scale testing has taken place to ensure that Hot Standby is a robust solution. The second option is to implement an operating system cluster. Extensive testing has been performed with some operating system clusters. The first two supported clustering options are Windows Cluster and High Availability Cluster Multi-Processing (HACMP™).

The main difference between the clustering options is the scripts to control the automated control of resources within the cluster. In this sense, Tivoli Monitoring is cluster-ready for other clustering solutions.

For detailed steps for configuring Clustering, including a detailed clustering scenario, see the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*. Also see this guide for instructions on configuring the Hot Standby option.

You can find more information about using a high availability z/OS hub monitoring server in the *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

Configuring for portal server high availability and disaster recovery

It is important to have multiple portal servers available in case of a hardware failure. While the hub monitoring server tracks the state of your Tivoli Monitoring environment it is not as critical to ensure that data is synchronized in real-time between multiple portal servers. The primary data that you want to protect is the customization that is stored in the portal server database, such as user-defined workspaces. Because this data does not change frequently, a good backup and restore strategy ensures a highly available environment.

Here are two different ways of achieving both high availability and disaster recovery with the portal server, depending on the amount of hardware you are willing to dedicate to your solution:

OS Cluster:

Many users set up an OS Cluster for their portal server. Depending on the clustering software used, the cluster can be set up across a WAN to achieve disaster recovery. For detailed information on setting up the monitoring server and portal server in an OS Cluster, see the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*.

Cold backup:

Some smaller users do not want to dedicate CPU cycles and memory to a live backup portal server. If that is the case in your environment, install a second portal server on another computer that serves as a production server. The backup portal server is typically shut down so that it does not use any CPU or memory. If the primary portal server goes down, the cold backup can be brought online. The key for a cold backup portal server is to periodically export the portal server database content and import it into the cold backup. In addition, ensure that the cold backup portal server is patched with the same software levels as the primary portal server.

Consider using a tool like the Tivoli System Automation for Multiplatforms to automate the process of backing up the resources.

As discussed previously, some users choose to implement a master read-write portal server and one or more read-only portal servers. When you implement multiple read-only portal servers, you can place a load balancer or edge server in front of the portal server and have users connect to the edge server. By doing this, you minimize the complexity and maximize the availability for the end user.

The strategy for backup and restore is to have one master Tivoli Enterprise Portal Server database where all customization is done. Then, periodically export the content from the “master” portal server and import the content into any other portal server. The import replaces the Tivoli Monitoring content in the portal server database, so be aware that any customization made in the secondary portal server environments will be overwritten during the import. The export and import of the portal server database can be done in two ways:

- Using RDBMS backup utilities such as DB2’s **db2 backup** and **db2 restore** commands
- Using the **migrate-export** and **migrate-import** command provided by the Tivoli Monitoring product

If the various portal server databases are not running on the same OS version, then the RDBMS backup and restore utilities will probably not work. In those cases, use the Tivoli Monitoring **migrate-export** and **migrate-import** commands as described in the product documentation.

Configuring for agent and remote monitoring server high availability and disaster recovery

All agents can be defined with a primary and secondary monitoring server, which allows the agent to connect to the secondary monitoring server if the primary is unavailable. Failover to the secondary monitoring server occurs automatically if the agent fails to communicate with the primary monitoring server.

If no other communication occurs between the agent and the monitoring server, the longest interval it should take for the failover to occur is the heartbeat interval, which defaults to 10 minutes.

The primary concern when building a high availability and disaster recovery configuration for the agents and remote monitoring servers is to determine how many agents to connect to each remote monitoring server. For Tivoli Monitoring V6.2.3, no more than 1500 monitoring agents should connect to each remote monitoring server.

The following information is important when planning your agents and remote monitoring servers:

- Ensure that failover does not result in many more than 1500 monitoring agents reporting to a single remote monitoring server. There are two strategies users typically take to avoid this situation.

  The *first* and preferred strategy involves having a spare remote monitoring server. By default, the spare remote monitoring server has no agents connected. When the monitoring agents that report to the primary monitoring server are configured, they are configured to use the spare remote monitoring server for their secondary monitoring server. Over time, network and server anomalies cause the agents to migrate.

  To manage this environment, write a situation to monitor how many agents are connect to the spare remote monitoring server. You can then use the situation to trigger a Take Action command that forces the agents back to their primary remote monitoring server by restarting them. Restarting the agents cause them to connect to their primary monitoring server. Ideally, migrate the agents back to their primary remote monitoring server when the number of agents connect to the spare monitoring server is greater than 20.

  The disadvantage to using a spare remote monitoring server is that you must dedicate a spare server to be the spare remote monitoring server. Some users choose to co-locate this server with the Warehouse Proxy Agent or run in a virtualized environment to minimize the extra hardware required.

  The *second* strategy is to evenly distribute the agents so that they failover to different remote monitoring servers to ensure that no remote monitoring server becomes overloaded. In the example below, there are four remote monitoring servers. In this example, configure one-third of the agents on each remote monitoring server to failover to a different remote monitoring server. Review the following scenario:

  RTEMS_1 has 1125 agents, RTEMS_2 has 1125 agents, RTEMS_3 and RTEMS_4 have 1125 agents.

  A third of RTEMS_1's agents failover to RTEMS_2, a third failover to RTEMS_3, and a third failover to RTEMS_4.

  This strategy ensures that none of the remote monitoring servers become overloaded. The problem with this strategy is that it requires a lot of planning and tracking to ensure that all of the remote monitoring servers are well-balanced.

- If you want your agent to failover to a remote monitoring server in another data center, ensure that you have good network throughput and low latency between the data centers.

Note: Connect a very small number of agents to the hub monitoring server. Typically, only the Warehouse Proxy Agent, Summarization and Pruning Agent, and any OS agents that are monitoring the monitoring server are connected to the hub monitoring server.

Use the Tivoli Monitoring V6.2.3 heartbeat capabilities to ensure that agents are running and accessible. The default heartbeat interval is 10 minutes. If an agent does not contact the monitoring server, a status of `MS_Offline` is seen at the monitoring server. An event can be generated when an agent goes offline. An administrator can evaluate whether the agent is having problems or whether there is another root cause. In addition, there is a solution posted on the Tivoli Integrated Service Management Library Web site that leverages the `MS_Offline` status and attempts to ping the server to determine if the server is down or whether the agent is offline. You can find more information by searching for "Perl Ping Monitoring Solution" or navigation code "1TW10TM0F" in the Tivoli Integrated Service Management Library.

Configuring for warehouse high availability and disaster recovery

When setting up the Warehouse for high availability and disaster recovery, the primary concern is backing up the data. The warehouse database can grow rapidly and has significant change, with many gigabytes of new data inserted per day plus summarization and pruning. Use the native database replication tools to achieve a high availability solution. All of the major database vendors provide data replication tools.

Configuring for Warehouse Proxy Agent high availability and disaster recovery

You need to achieve redundancy with the Warehouse Proxy Agent. Only one Warehouse Proxy Agent can be receiving historical data from a specific agent. You can encounter problems if two Warehouse Proxy Agents are configured to receive historical data from the same agent. To avoid problems, ensure that only one Warehouse Proxy Agent is responsible for collecting the historical data from a remote monitoring server.

To ensure that your Warehouse server performs optimally, ensure that the *WAREHOUSELOG* and *WAREHOUSEAGGREGLOG* tables are pruned on a regular basis.

Note: Beginning with Tivoli Monitoring V6.2.3 the tables *WAREHOUSELOG* and *WAREHOUSEAGGREGLOG* are disabled by default.

Pruning for these tables can be configured by specifying retention intervals in the configuration dialog for the Summarization and Pruning Agent or in the configuration file (`KSYENV` on Windows, `sy.ini` on UNIX or Linux). See "Historical data collection" on page 434 for more details.

Configuring for Summarization and Pruning Agent high availability and disaster recovery

Connect the Summarization and Pruning Agent to the hub monitoring servers. When the Hot Standby option is used, the Summarization and Pruning Agent must be configured with the standby hub as the secondary monitoring server. However, there are some additional considerations for achieving high availability with the Summarization and Pruning Agent.

Only one Summarization and Pruning Agent may be running against a warehouse database. Thus, it is important to ensure that there is data integrity within the database and that there is no database deadlock between two competing agents. So, by default, only one Summarization and Pruning Agent must be installed and running.

As in the Warehouse Proxy Agent set up, you want to install a second Summarization and Pruning Agent that serves as a cold backup to the primary Summarization and Pruning Agent. By default, the backup Summarization and Pruning Agent is disabled. Write a situation that detects when the primary Summarization and Pruning Agent is down and automatically starts up the backup Summarization and Pruning Agent through a Take Action command.

Care must be taken in writing the Take Action command to ensure that only one Summarization and Pruning Agent is running at any given time. To ensure the two Summarization and Pruning Agents are not running at the same time, perform the following steps:

1. Have the situation trigger only after the second or third missed heartbeat. Occasionally, there are temporary outages triggered by network problems or routine maintenance. You do not want the automated Take Action to occur during a temporary outage.
2. When starting up the backup Summarization and Pruning Agent using a Take Action command, the primary Summarization and Pruning Agent must be disabled so that it does not accidentally restart until an administrator manually corrects the problem.
3. Write up a documented procedure to ensure that only one of the Summarization and Pruning Agents is brought back online following a failover.

Configuring for Tivoli Performance Analyzer high availability and disaster recovery

The Tivoli Performance Analyzer must always be connected to the hub monitoring server. When you use the Hot Standby option, Tivoli Performance Analyzer Agent can be configured with the standby hub as the secondary monitoring server. Since there can only be one Tivoli Performance Analyzer Agent in your ITM environment (that is one Tivoli Performance Analyzer Agent per hub monitoring server) it is not possible to setup a secondary agent in Hot Standby mode. However, you can setup a second Tivoli Performance Analyzer Agent and keep it stopped, as long as the primary server is running. The secondary agent can only be started when the primary agent is stopped and disabled. The switch can be performed manually by the ITM administrator, or Take Action commands can be used. In both cases it is very important to ensure that only one agent is running at the same time. See "Configuring for Summarization and Pruning Agent high availability and disaster recovery" on page 57 for information on writing the Take Action command.

Agent deployments

When planning your installation, you need to determine how you want to deploy your agents. For very small environments, some users manually install agents on each server. For larger environments, automation tools must be used to deploy the agents and agent patches. A key decision point is to determine which deployment software to use. The Tivoli Monitoring V6.2.3 product has a remote deployment capability that allows you to initially deploy your operating system agents remotely, and then remotely add agents to your systems.

Each product and fix pack includes a remote deployment bundle that can be placed in the remote deployment depot for future agent distributions and patching. However, the remote deployment capability is not as efficient at distributing software as some purchasable distribution products. If you already have an enterprise-class software distribution product like Tivoli Configuration Manager or Tivoli Provisioning Manager, you might find it more efficient to distribute the agents and patches. Tivoli Monitoring V6.2.3 agents provide software package blocks that can be used by Tivoli Configuration Manager and Tivoli Provisioning Manager to distribute the agents.

The main advantage of using products such as Tivoli Configuration Manager and Tivoli Provisioning Manager are:

- Faster distribution times to speed large-scale deployments.
- Tivoli Configuration Manager and Tivoli Provisioning Manager can be tuned to utilize only a portion of the network bandwidth.
- Tivoli Configuration Manager and Tivoli Provisioning Manager can easily be configured for retries and tracking of success and failure.

Here is the location of the Software Packages for the IBM Tivoli Monitoring V6.1 monitoring agents: ftp://www.redbooks.ibm.com/redbooks/SG247143/.

The advantage of using remote deployment is no additional work is required to create and deploy agent patches.

If you have an enterprise-class software distribution product, use it for the initial software distribution and for the deployment of larger fix packs. For interim fixes and small fix packs, the remote deployment capability might require less time to configure and utilize.

One of the most important aspects of agent deployment is the agent prerequisite preparation. Ensure that the servers are prepared with the appropriate filesystems including adequate space. In addition to disk space, you must determine the user account to be used for the agent installation. By using an administrative account (administrator or root), you ease your agent deployment tasks.

If administrator or root are not allowed, then using **sudo** on a UNIX system is the next best choice. Without administrative authority, the installation becomes a multi-step process where the systems administrators need to be brought in to run commands such as **setperm** to setup the permissions. For planning purposes, plan for roughly 500 MB of disk space to allow space for the agent and historical logs.

Self-describing monitoring agents

The self-describing agent feature makes it possible for new or updated IBM Tivoli Monitoring agents to become operational after installation, without having to perform additional product support installation steps. Self-describing agents apply version updates to other components automatically without the need to recycle your hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or remote Tivoli Enterprise Monitoring Server. This new feature in V6.2.3 integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure.

For more information, see “Enabling self-describing agent capability at the hub monitoring server” on page 216 and the *IBM Tivoli Monitoring: Administrator's Guide*.

Prerequisite Checking for IBM Tivoli Monitoring agents

By using new tools in Tivoli Monitoring, you can perform prerequisite checking for agents before carrying out an installation. The two mechanisms available are a manually executed stand-alone prerequisite scanner and a remote prerequisite scanner facility that extends the capabilities of IBM Tivoli Monitoring's remote deployment component.

Note: Only the operating system (OS) agents are enabled with prerequisite checking. If you attempt to run remote prerequisite checking with an application agent, the operation fails because the application agent does not provide a prerequisite checker.

Stand-alone prerequisite checking

This section contains the following information:

- Locating the prerequisite checking tool
- Running the prerequisite checker by using the command-line interface
- A sample results file

Note: Prerequisite checker output for both stand-alone and remote prerequisite checking is available in English only. The prerequisite checker report (`results.txt` file), `precheck.log` file, `prereq_checker.bat` file, and `prereq_checker.sh` file are also available only in English.

Prerequisite Checker: The prerequisite checker is located on the IBM Tivoli Monitoring agent media. Platform-specific scripts are used by the prerequisite checker, so you must select the prerequisite checker that is appropriate for the platform where you are executing the prerequisite scan.

The prerequisite checker for Linux and UNIX platforms is located in the following directory:

```
<ITM_Agent_Media>/unix/prereqchecker
```

The prerequisite checker for the Windows platform is located in the following directory:

```
<ITM_Agent_Media>/WINDOWS/prereqchecker
```

Running the stand-alone Prerequisite Checker: Complete the following steps to run the stand-alone prerequisite checker:

1. Copy the prerequisite checker for the appropriate platform to the target computer.
2. Open a command-line interface:
   - On Windows systems, open a Command Prompt.
   - On Linux and UNIX systems, open a shell environment.

3. Change directory to the folder where you located the prerequisite checker and run the following command:
   - On Windows systems, run the `prereq_checker.bat` command.
   - On Linux and UNIX systems, run the `prereq_checker.sh` command.

   The following input parameters are displayed:

   ```
   prereq_checker.bat "<Product Code> [product version], <Product Code>
   [product version]..." [PATH=<Agent install path>] [detail]
   [-p <Product Code>.SECTION.NAME=VALUE pairs]
   ```

   Windows example: `prereq_checker.bat "KNT" detail PATH=d:\ibm\itm -p SERVER=IP.PIPE://mytems:1234`

   Linux and UNIX example: `./prereq_checker.sh "KLZ" detail PATH=/opt/IBM/ITM -p SERVER=mytems:1918`

The following paragraphs provide a detailed description of the input parameters:

- **Product code and version pairs**
  - At least one product code is required. For IBM Tivoli Monitoring, the code is the three-letter product code for the component or agent. Optionally, you can specify a version for the agent. If you do not specify a version, the prerequisite checker checks the latest version available.
  - You must enter the product codes in capital letters. For the Linux agent, for example, you must enter KLZ. If you enter lowercase klz, an error message is displayed.
  - You can specify multiple product codes separated by commas.
  - The product codes and versions must be enclosed in quotation marks, for example "KNT 06230000, KUD". In this example, the prerequisite check is made on the KNT agent version 06230000 and the latest version of the KUD agent.
  - Each component or agent has a `config` file with the naming convention `*.cfg`, for example, `KNT_0620000.cfg`. If you specify a product code that does not have a corresponding `*.cfg` file, that product code is ignored.
- **[PATH=<product install path>]**
  - The *PATH* parameter is optional. An example PATH is `PATH=D:\IBM\ITM`. If you do not specify a PATH parameter, the prerequisite checker checks the following default IBM Tivoli Monitoring installation paths:
    - On Windows systems, the default path is: `C:\IBM\ITM`.
    - On Linux and UNIX systems, the default path is: `/opt/IBM/ITM`.
- **[detail]**
  - The *detail* parameter is optional. This flag indicates that you want to see detailed results on the screen when you run the prerequisite checker.
  - Do not enclose the word *detail* in quotation marks.
  - If you do not specify *detail*, then only PASS or FAIL is displayed on the screen. Omitting the parameter is a simple way to see if your system meets the prerequisites.
  - See the following example output when the *detail* parameter is specified:

    ```
    Windows OS monitoring Agent [version 06210000] :
       Evaluation          PASS/FAIL       Expected Result           Result
       CPU Number          PASS            1                         1
       Available Memory    PASS            35MB                      1.02GB
       Available Disk      PASS            70MB                      1.09GB

       ALL COMPONENTS :
       Evaluation          PASS/FAIL       Expected Result           Result
       Available Memory    PASS            35MB                      1.02GB
       Available Disk      PASS            70MB                      1.09GB
    ```

- – The detailed output is written to a file called `precheck.log`, regardless of whether you specify *detail* or not. You can view this file to see the results of the prerequisite checker. In addition, you can view the `result.txt` file to see the detailed steps taken by the prerequisite checker.
- – You can specify optional parameters by using a `-p` flag. The parameters are formatted as follows:

  ```
  [-p <Product Code>.<instance>.<parameter>=<value>, <Product Code>.<instance>.<parameter>=<value>,]
  ```

  See the following example to check the ports:

  ```
  -p SERVER=IP.PIPE://mymachine:1918
  ```

Note: This script accepts the `-p` connection parameter in the same way as the **`tacmd createNode`** command, you can specify *SERVER, PROTOCOL, PORT, BACKUP, BSERVER,* and so on. All parameters are case sensitive, therefore you must use capital letters.

Sample prerequisite checker report:

```
IBM Tivoli Prerequisite Scanner
    Version  : 1.0.17
    Build    : 2010728
    OS Name  : Microsoft Windows Server 2003, Enterprise Edition
Service Pack 2
    User Name: Administrator
Machine Info
   Machine name : DEV-TAYLOR50
   Serial Number: KLXZL5G
   OS Serial    : 69713-640-1083907-45085

Windows OS monitoring Agent [version 06230000]:

Property                                Result  Found                                   Expected
========                                ======  =====                                   ========
# CPU                                   PASS    1                                       1
Memory                                  PASS    313MB                                   35MB
Disk                                    PASS    6.51GB                                  125MB
OS Version                              PASS

Microsoft Windows Server 2003, Enterprise Windows 2000 Server
Windows 2000 Advanced Server
Windows XP Professional Service Pack 1
Windows Server 2003 Datacenter Edition Service Pack 1
Windows Server 2003 Standard Edition Service Pack 1
Windows Server 2003 Enterprise Edition Service Pack 1
Windows Server 2003 R2 Enterprise x64 Edition
Windows Server 2003 R2 Standard x64 Edition
Windows Server 2003 R2 Data Center Edition x64 Edition
Windows Server 2003 Enterprise Itanium Edition
Windows Server 2008 Enterprise Edition
Windows Server 2008 Standard Edition
Windows Server 2008 Data Center Edition
Windows Server 2008 Enterprise x64 Edition
Windows Server 2008 Standard x64 Edition
Windows Server 2008 Data Center x64 Edition

ALL COMPONENTS :
Property            Result  Found         Expected
========            ======  =====         ========
Memory              PASS    313MB         35MB
c:\                 PASS    6667MB        125MB
```

Remote prerequisite checking

Remote prerequisite checking is a new feature in IBM Tivoli Monitoring V6.2.3. You must meet the following requirements in your IBM Tivoli Monitoring infrastructure to make use of this feature:

- The command-line component known as KUE (User Interface Extensions) must be at V6.2.3. Use this component to import the agent-specific prerequisite checkers. The KUE component also contains the new **tacmd checkprereq** command to run the remote prerequisite checking. You can use the **cinfo -i** command on Linux and UNIX systems or the **kincinfo -i** command on Windows systems to verify the installed level of KUE.

- The hub Tivoli Enterprise Monitoring Server must be at V6.2.3. By using V6.2.3 of the hub monitoring server, the new **tacmd checkprereq** command can be processed by the remote deployment component.
- For improved performance and parallelization, the tacmd checkprereq command uses the *SERVER* property, similar to tacmd createnode, to route the processing from the hub Tivoli Enterprise Monitoring Server to connect to the remote Tivoli Enterprise Monitoring Server. However, this routing occurs only if the supplied remote Tivoli Enterprise Monitoring Server is at V6.2.3. If the supplied remote Tivoli Enterprise Monitoring Server is at a version lower than V6.2.3, the remote prerequisite checking is performed by the hub Tivoli Enterprise Monitoring Server.

Importing the agent-specific prerequisite checker: As with IBM Tivoli Monitoring agents, an agent-specific prerequisite checker is delivered as a bundle in the IBM Tivoli Monitoring agent media. The IBM Tivoli Monitoring V6.2.3 tacmd addbundles command has been updated to implicitly import an agent's prerequisite checker bundle into the Tivoli Enterprise Monitoring Server depot if it exists on the agent media. No additional steps must be taken beyond adding the agent bundle to the Tivoli Enterprise Monitoring Server depot, as is done for remote deployment in previous releases. The IBM Tivoli Monitoring OS agents (KNT, KUX, KLZ) are the initial agents to use the new prerequisite checking feature.

Using remote prerequisite checking: New in IBM Tivoli Monitoring V6.2.3 is the ability to preemptively check for required prerequisites on a remote computer. This feature uses the same functionality as the tacmd createnode command to transfer and run on a target computer. By using the new **tacmd checkprereq** command, you can run a prerequisite scan for a specified agent on a specified target computer by including the -h | --host flag, or on a group of target computers by including the -g | --deploygroup flag.

You can also run the tacmd checkprereq command by using the -n | --node flag to perform remote prerequisite checking without endpoint credentials. Both the hub Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Monitoring Agent must be at V6.2.3 to use prerequisite checking without endpoint credentials. If a prerequisite check without endpoint credentials is run on a target that has a Tivoli Enterprise Monitoring Agent older than V6.2.3, the command fails. The -n | --node flag allows you to run a prerequisite check for an endpoint before you attempt a deployment, without specifying endpoint credentials. You can review the prerequisite check results through the tacmd getdeploystatus command or the Tivoli Enterprise Portal Deploy Status workspace. If a prerequisite check fails, you can review the results log on the Tivoli Enterprise Monitoring Server in the CANDLEHOME/logs/checkprereq_results directory.

The following example shows the execution of the checkprereq command to a single computer for the Windows OS Agent. After running the command, you receive a transaction ID, which is used to track the status of the prerequisite check by using the tacmd getdeploystatus command or the Tivoli Enterprise Portal Deployment Status Workspace.

Example:

```
tacmd checkprereq -h mysys1.mycompany.com -t NT -p SERVER=rtems1.mycompany.com
```

Note: You are prompted for the remote login credentials. Alternatively, you can use the optional flags to supply the credentials at the command-line.

The following example shows how you can use deployment groups to perform prerequisite checking for a group of target computers.

Example:

```
tacmd creategroup -g NewSystems -t DEPLOY

tacmd addgroupmember -g NewSystems -t DEPLOY -m
```

```
mysys1.mycompany.com -p SERVER=rtems1.mycompany.com
KDYRemote Execution and Access.Remote Execution and
AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution
and AccessPASSWORD=mypass

tacmd addgroupmember -g NewSystems -t DEPLOY -m
mysys2.mycompany.com -p SERVER=rtems2.mycompany.com
KDYRemote Execution and Access.Remote Execution and
AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution
and AccessPASSWORD=mypass

tacmd addgroupmember -g NewSystems -t DEPLOY -m
mysys3.mycompany.com -p KDYRemote Execution and
Access.Remote Execution and AccessUSERNAME=root
KDYRemote Execution and Access.Remote Execution and
AccessPASSWORD=mypassword
tacmd checkprereq -g NewSystems -t LZ
```

The steps in the previous section demonstrate the creation of a new deployment group called *NewSystems* and the addition of several deployment targets and the execution of the preemptive prerequisite checking. The first member is using the *SERVER* property to specify that the execution should be performed by the Tivoli Enterprise Monitoring Server running on `rtems1.mycompany.com`. If this specification is not a valid Tivoli Enterprise Monitoring Server, the prerequisite checking is performed by the hub Tivoli Enterprise Monitoring Server. The second member specifies a different executing Tivoli Enterprise Monitoring Server, and the third member does not supply the *SERVER* property, therefore the command runs on the hub Tivoli Enterprise Monitoring Server by default. When the preemptive prerequisite checking is run, a transaction ID is provided for deployment status tracking. After performing the `tacmd checkprereq` command, and resolving any prerequisite failures so that the `tacmd checkprereq` command results in a *SUCCESS* for all target machines, you can run the `tacmd createnode` command using the same deployment group, for example `tacmd createnode -g NewSystems`.

Note: For deployment groups with mixed platform types such as a combination of Linux and UNIX systems, and Windows systems, you can specify the product as *ITM*. In the case of the OS agents, the correct OS agent type is determined based on the platform type discovered when the connection is established. Even when the incorrect product code is supplied for the OS agents, the correct platform type is detected. This is a special case that applies to OS agents.

After running the preemptive prerequisite checking command, the example run can result as follows. In the following example, you can see that two of the endpoints successfully passed the prerequisite scan, however the `mysys2.mycompany.com` endpoint did not pass.

```
C:\IBM\ITM\bin>tacmd getdeploystatus -g 1282918238453000000000041

Transaction ID : 1282918238453000000000041
Command        : CHECKPREREQ
Status         : SUCCESS
Retries        : 0
TEMS Name      : RTEMS1
Target Hostname:  mysys1.mycompany.com
Platform       : li6263
Product        : LZ
Version        : 062300000
Error Message  : KDY4001I: The prerequisite checking operation was a success.

Transaction ID : 1282918238453000000000041
Command        : CHECKPREREQ
Status         : FAILED
Retries        : 3
TEMS Name      : RTEMS2
Target Hostname:  mysys2.mycompany.com
Platform       : li6263
Product        : LZ
Version        : 062300000
```

```
Error Message  : KDY4003E: A failure occurred while checking for
required prerequisites for bundle LZ on the
target host mysys2.mycompany.com.
Review the results from the prerequisite checking execution and
resolve the issues found on the endpoint.

Transaction ID : 128291823845300000000041
Command        : CHECKPREREQ
Status         : SUCCESS
Retries        : 0
TEMS Name      : HUB_TEMS
Target Hostname:  mysys3.mycompany.com
Platform       : li6263
Product        : LZ
Version        : 062300000
Error Message  : KDY4001I: The prerequisite checking operation was a success.
```

To correct the failures found on the mysys2.mycompany.com endpoint you can review the results log that was sent back to the executing Tivoli Enterprise Monitoring Server (RTEMS2 in this case). The results file is located in the CANDLEHOME/logs/checkprereq_results directory. The naming format of the results file is <SUCCESS|FAILED>_<host>_<transaction ID>.txt. In this instance, the name of the results file is FAILED_mysys2.mycompany.com_128291823845300000000041.txt. The preceding example shows the output of the prerequisite scan report. To review the results of both *SUCCESS* and *FAILED* prerequisite checks, you must specify the -c | -- collectall flag when running the tacmd checkprereq command. By default, all *FAILED* prerequisite check results are sent back to the executing Tivoli Enterprise Monitoring Server.

CLI and remote deployment prerequisite checking: You can use the remote deployment prerequisite checking feature as part of a tacmd addSystem, tacmd createnode, or tacmd updateAgent command execution when deploying an agent. Use the command in this way to run a prerequisite check in a single command-line execution. You have the option of allowing the deployment to continue only if the check is successful, or to ignore the results and attempt the deployment even if the prerequisite checker fails. The valid options are: COLLECTALL, EXECPREREQCHECK, and IGNOREPREREQCHECK. The values are to be specified in KEY=VALUE format.

Note: The prerequisite checking feature is available only on Tivoli Enterprise Monitoring Agents at Tivoli Monitoring V6.2.3 or later. If prerequisite checking is enabled during a tacmd addSystem or tacmd updateAgent command execution for a target that has a Tivoli Enterprise Monitoring Agent older than V6.2.3, the command fails with an error message.

An agent-specific prerequisite checker is delivered as a bundle in the IBM Tivoli Monitoring agent media. The IBM Tivoli Monitoring V6.2.3 tacmd addbundles command is used to import an agent's prerequisite checker bundle into the Tivoli Enterprise Monitoring Server depot if it exists on the agent media. No additional steps must be taken beyond adding the agent bundle to the Tivoli Enterprise Monitoring Server depot as was done for remote deployment in previous releases. Next, issue a tacmd addSystem, tacmd createnode, or tacmd updateAgent command with prerequisite checking enabled (EXECPREREQCHECK=Y). The following outcomes are possible:

- If the tacmd addSystem command runs successfully, a new application agent is installed on the endpoint and connected back. If the tacmd createnode command runs successfully, a new OS agent is installed on the endpoint and connected back. If the tacmd updateAgent command runs successfully, a new version of the agent is installed on the endpoint and connected back.
- If the prerequisite check fails, review the prerequisite check results log located on the deploying Tivoli Enterprise Monitoring Server in the CANDLEHOME/logs/checkprereq_results directory.
- The prerequisite check might fail but the installation still continues (due to the IGNOREPREREQCHECK option) and the installation fails. In this scenario, the data is received through the tacmd getdeploystatus command or the Tivoli Enterprise Portal Deploy Status workspace.

Background information about agent autonomy

By default, agents are configured for autonomous operation (in other words, the default value for configuration parameter IRA_AUTONOMOUS_MODE is **Y**). This means that when the connection to the Tivoli Enterprise Monitoring Server is lost or the monitoring server otherwise becomes unavailable, the agent continues to run all situations that can be evaluated exclusively at the agent. Situations that become true while disconnected from the monitoring server are stored persistently and are uploaded to the monitoring server when reconnected. If the agent was starting up when the communications break occurred, startup processing continues.

If you do not want autonomous agent operation, you can disable it by setting environment variable IRA_AUTONOMOUS_MODE to **N** in the agent configuration. In this case you must also ensure variable CT_CMSLIST is both specified and not blank; otherwise the agent will not start.

To configure an agent for fully autonomous operation (in other words, it runs without a connection to a monitoring server), the following agent configuration is required:

1. The CT_CMSLIST variable must not be set (that is, it should have no value).
2. At least one protocol must be defined in the KDC_FAMILIES variable.
3. The IRA_AUTONOMOUS_MODE variable does not need to be set, as its default value is **Y**. If IRA_AUTONOMOUS_MODE is currently set to **N**, it must be either changed to **Y** or removed completely.

Notes:

1. If you respond **NO** to question **Will this agent connect to a TEMS?** when running the UNIX installer, these parameters get set correctly for you.
2. In all cases, including fully autonomous mode, at least one active protocol *must* be defined by using the KDC_FAMILIES environment variable. If no protocols are defined, the agent will not start.

Event forwarding from autonomous agents

By incorporating the Event Integration Facility (EIF) into the autonomous agents, they can forward events directly to Netcool/OMNIbus via the Secure Sockets Layer (SSL) protocol. Available for the Windows, Linux, UNIX, and z/OS platforms, this integration provides another event emitter (similar to the SNMP trap emitter) that can emit EIF-format events directly from an autonomous agent to the event-management facility your site uses.

Currently, EIF event notification is supported only for locally defined private situations—those defined via a private configuration file local to the monitoring agent. (Events for enterprise situations must still be forwarded from the hub Tivoli Enterprise Monitoring Server.) These EIF events provide similar function and format to those provided by the event forwarder on the hub monitoring server, including the ability to send situation status. There is no support for dynamic refresh of changed EIF events (that is, event maps and event destinations); for such changes to be effective, you must recycle the agent.

You can customize agent-generated EIF events via a product-provided or user-defined event mapping file, which specifies these elements:

- The attributes included in the event.
- Custom message slot text with variable substitution.
- Mapping by situation name.
- Mapping by attribute group used in the situation.

Events generated will have a source slot value of ITM Agent:Private Situation to distinguish them from events originated at the monitoring server, which have a source slot value of ITM.

You can enable or disable event forwarding by setting IRA_EVENT_EXPORT_EIF to either Y or N in the event mapping file. Event forwarding is disabled automatically when no valid EIF event destination is defined. Note that a single event can be sent to up to five different destinations; these can be a mixture of both TEC and non-TEC event receivers.

A heartbeat function lets you use the event-forwarding feature to send events to either Tivoli Enterprise Console or Netcool/OMNIbus that notify you if the agent is online and running. The heartbeat interval determines how often the agent generates this event; it is configurable via the event mapping file. These EIF events, whose classname is `ITM_Heartbeat`, contain a slot called **interval** whose value is the heartbeat interval. You can customize the IBM-provided heartbeat rules or write your own to handle heartbeat events as your site's needs dictate.

For complete information about defining your own event mapping file, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Note: Existing TEC and OMNIbus event-synchronization rules function as usual, but bidirectional interactions with IBM Tivoli Monitoring are not possible, as the events do not originate with the Tivoli Enterprise Monitoring Server. However, the TEC and OMNIbus event receivers will still be able to update event status for events sent from autonomous agents.

Agentless monitoring versus monitoring agents

IBM Tivoli Monitoring provides operating system (OS) agents that monitor the availability and performance of the computers in your monitoring environment. An example of an OS agent is the monitoring agent for Windows, which can monitor Windows XP, Windows 2000, Windows 2003, and Windows 2008 operating systems. These full-function OS agents must reside on the same computers they are monitoring.

As of version 6.2.1, IBM Tivoli Monitoring also provides *agentless monitors*. An agentless monitor is a standard Tivoli Monitoring agent that can monitor the operating system running on multiple remote nodes that do not have the full-function OS agents running on them. An agentless monitor obtains data from nodes it is monitoring via a remote application programming interface, or API—in this case, SNMP, CIM, or WMI—running on the node being monitored. Since these interfaces provide information about either operating system functions or base application functions, no IBM Tivoli Monitoring component need be installed or deployed on the monitored node.

API **Function**

SNMP The Simple Network Management Protocol is a TCP/IP transport protocol for exchanging network management data and controlling the monitoring and operation of network nodes in a TCP/IP environment.

CIM The Common Information Model is an XML-based standard for defining device and application characteristics so system administrators and management programs can monitor and control them using the same set of tools, regardless of their differing architectures. CIM provides a more comprehensive toolkit for such management functions than the Simple Network Management Protocol.

WMI Microsoft's Windows Management Instrumentation API provides a toolkit for managing devices and applications in a network of Windows-based computers. WMI provides data about the status of local or remote computer systems as well as the tools for controlling them. WMI is included with the Windows XP and Windows Server 2003 and 2008 operating systems.

These APIs are supported by the Agent Builder, which enables you to build custom agentless monitoring solutions that are separate from the agentless monitors available on the Tivoli Monitoring installation media and that provide additional function.

Since an agentless monitor is a standard Tivoli Monitoring agent, it collects data and distributes it to a Tivoli Enterprise Monitoring Server and then on to a Tivoli Enterprise Portal Server. It also takes advantage of the various features of the IBM Tivoli Monitoring product, such as Tivoli Enterprise Portal workspace views, situations, remote deployment of the agentless monitors, policies, and so on. Detailed information can be found in the user's guide for each agentless monitor; see Table 8 on page 72.

Agentless monitoring does not provide the kind of deep-dive information your site may need for its core business servers; however, it does allow a small set of centralized servers to supervise the health of the operating nodes in your environment. There are five types of agentless monitors that cover the Windows, AIX, Linux, HP-UX, and Solaris environments.

The agentless monitors are multi-instance agents. After installing or deploying an agentless monitor on a machine, additional instances can be created via configuration. Each instance can communicate with up to 100 remote nodes.

Each type of agentless monitor can run on additional platforms beyond the type of platform it monitors. For example, the agentless monitor for Windows (which monitors only Windows operating systems) can run on any of the supported platforms: Windows, AIX, Solaris, HP-UX, Linux.

Specific operating system releases that a particular agentless monitor can monitor are detailed in Table 7 on page 69. Check the user's guide for each agentless monitor regarding platform-specific requirements for the operating systems that agentless monitors can run with.

A computer that has one or more agentless monitors running on it is referred to as an agentless monitoring server. Each server node can support up to 10 active agentless monitor instances, in any combination of agentless monitor types; for example, 2 AIX, 2 HP-UX, 2 Linux, 2 Solaris, 2 Windows; or 4 Windows, 3 AIX, 3 Linux; or 5 Windows, 5 Solaris; or 10 HP-UX. Each instance can communicate with up to 100 remote nodes, which means a single agentless monitoring server can support as many as 1000 monitored systems (10 instances * 100 remote nodes per instance). By adding more server nodes, the number of monitored nodes increases into the thousands.

Figure 10 illustrates the architecture of an IBM Tivoli Monitoring environment that employs agentless monitoring.

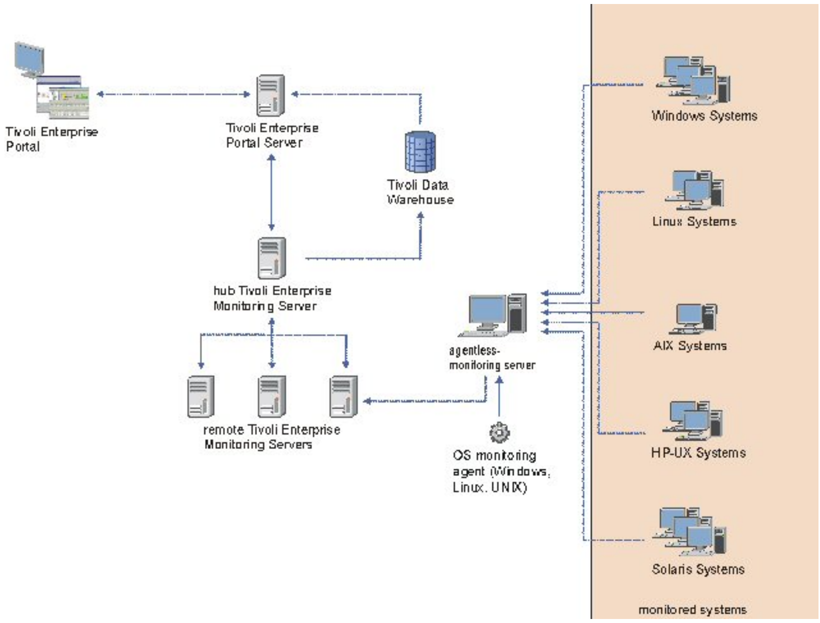

Tivoli Enterprise Portal
Tivoli Enterprise Portal Server
Tivoli Data Warehouse
hub Tivoli Enterprise Monitoring Server
remote Tivoli Enterprise Monitoring Servers
agentless-monitoring server
OS monitoring agent (Windows, Linux, UNIX)
Windows Systems
Linux Systems
AIX Systems
HP-UX Systems
Solaris Systems
monitored systems

Figure 10. Architecture of agentless monitoring

Agentless technology provides lightweight OS monitoring that targets key metrics along with basic situations meant to satisfy simple monitoring needs. Agentless monitoring provides speedy implementation and minimum agent deployment, including the deployment of updates; however, the need to poll the monitored node to retrieve its monitoring data increases network traffic, and real-time data availability is impacted both by the network delay and the reliance on polling. In addition, the implementation of Take Action commands for command and control is more powerful with the full-function agents than for agentless technology.

Key operating system metrics returned:
- Logical and physical disk utilization.
- Network utilization
- Virtual and physical memory
- System-level information
- Aggregate processor utilization
- Process availability

Default situations are provided for:
- Disk utilization
- Memory utilization
- CPU utilization
- Network utilization

You can use these situations as is or as models for custom situations that meet your site's specific needs.

The agentless monitors monitor the distributed operating systems listed in Table 7 on page 69. You can configure different data collectors for these environments, as shown.

Table 7. Data collectors usable with the various agentless monitors and releases supported

Agentless monitor	Product code	Data collectors supported	Operating system releases monitored
Agentless Monitoring for Windows OS	R2	• WMI[1] • Performance Monitor (PerfMon)[1] • Windows event log[1] • SNMP V1, V2c, V3	Monitors the following Windows releases: • Windows 2000 Professional Windows 2000 Server • Windows 2000 Advanced Server Windows XP Professional (32 bit) with SP1 or higher • Windows Server 2003 Standard Edition (32 bit) with SP1 or higher • Windows Server 2003 Standard Edition (32 bit) with R2 SP1 or higher • Windows Server 2003 Enterprise Edition (32 bit) with SP1 or higher • Windows Server 2003 Enterprise Edition (32 bit) with R2 SP1 or higher • Windows Server 2003 Datacenter Edition (32 bit) with SP1 or higher • Windows Server 2003 Datacenter Edition (32 bit) with R2 SP1 or higher • Windows 2003 Standard Edition (64 bit) with R2 SP2 or higher • Windows 2003 Enterprise Edition (64 bit) with R2 SP2 or higher • Windows Server 2003 Datacenter Edition (64 bit) with R2 SP2 or higher • Windows 2003 Server Enterprise Edition on Itanium2 (IA64) with R2 SP2 or higher • Windows Server 2008 Standard Edition (32 bit) • Windows Server 2008 Datacenter Edition (32 bit) • Windows Server 2008 Enterprise Edition (32 bit)Windows Server 2008 Standard Edition (64 bit) • Windows Server 2008 Enterprise Edition (64 bit) • Windows 2008 Enterprise Edition on Itanium2 (IA64) • Windows Vista Enterprise, Business, and Ultimate (32 bit) • Windows Vista Enterprise, Business, and Ultimate (64 bit) **Note:** IA64 machines running Windows are not supported.
Agentless Monitoring for AIX OS	R3	SNMP V1, V2c, V3	Monitors the following AIX releases: • AIX V5.2 (32 bit) with ML07 or later • AIX V5.2 (64 bit) with ML07 or later • AIX V5.3 (32 bit) with ML05 or later • AIX V5.3 (64 bit) with ML05 or later • AIX V6.x (64 bit)

Table 7. Data collectors usable with the various agentless monitors and releases supported (continued)

Agentless monitor	Product code	Data collectors supported	Operating system releases monitored
Agentless Monitoring for Linux OS	R4	SNMP V1, V2c, V3	Monitors the following Linux releases running on xSeries®, pSeries, and zSeries machines: • RedHat Enterprise Linux 4 Intel (32 bit) • RedHat Enterprise Linux 4 on x86-64 (64 bit) • RedHat Enterprise Linux 4 on Itanium (64 bit) • RedHat Enterprise Linux 4 on iSeries® and pSeries • RedHat Enterprise Linux 4 on zSeries (31 bit) • RedHat Enterprise Linux 4 on zSeries (64 bit) • RedHat Enterprise Linux 5 Intel (32 bit) • RedHat Enterprise Linux 5 on x86-64 • RedHat Enterprise Linux 5 on Itanium 64 bit • RedHat Enterprise Linux 5 on iSeries and pSeries • RedHat Enterprise Linux 5 on zSeries (31 bit) • RedHat Enterprise Linux 5 on zSeries (64 bit) • SuSE Linux Enterprise Server 9 Intel (32 bit) with SP3 or later • SuSE Linux Enterprise Server 9 on x86-64 (64 bit) with SP3 or later • SuSE Linux Enterprise Server 9 on Itanium (64 bit) with SP3 or later • SuSE Linux Enterprise Server 9 for iSeries and pSeries with SP3 or later • SuSE Linux Enterprise Server 9 for zSeries (31 bit) with SP3 or later • SuSE Linux Enterprise Server 9 for zSeries (64 bit) with SP3 or later • SuSE Linux Enterprise Server 10 Intel (32 bit) • SuSE Linux Enterprise Server 10 on x86-64 (64 bit) • SuSE Linux Enterprise Server 10 on Itanium (64 bit) • SuSE Linux Enterprise Server 10 for iSeries and pSeries (64 bit) • SuSE Linux Enterprise Server 10 for zSeries (64 bit)
Agentless Monitoring for HP-UX OS	R5	SNMP V1, V2c, V3	Monitors the following HP-UX releases: • HP-UX 11i v1 (B.11.11) (32/64) on PA-RISC • HP-UX 11i v2 (B.11.23) (64 bit) on PA-RISC • HP-UX 11i v3 (B.11.31) (64 bit) on PA-RISC • HP-UX 11i v2 (B.11.23) on Integrity (IA64) • HP-UX 11i v3 (B.11.31) on Integrity (IA64)
Agentless Monitoring for Solaris OS	R6	• CIM-XML • SNMP V1, V2c, V3	Monitors the following Solaris releases: • Solaris V8 (SPARC) (32/64bit) • Solaris V9 (SPARC) (32/64bit) • Solaris V10 (SPARC) (32/64 bit) • Solaris V10 (x86-64) (64 bit) • Solaris V10 (Opteron) (64 bit)

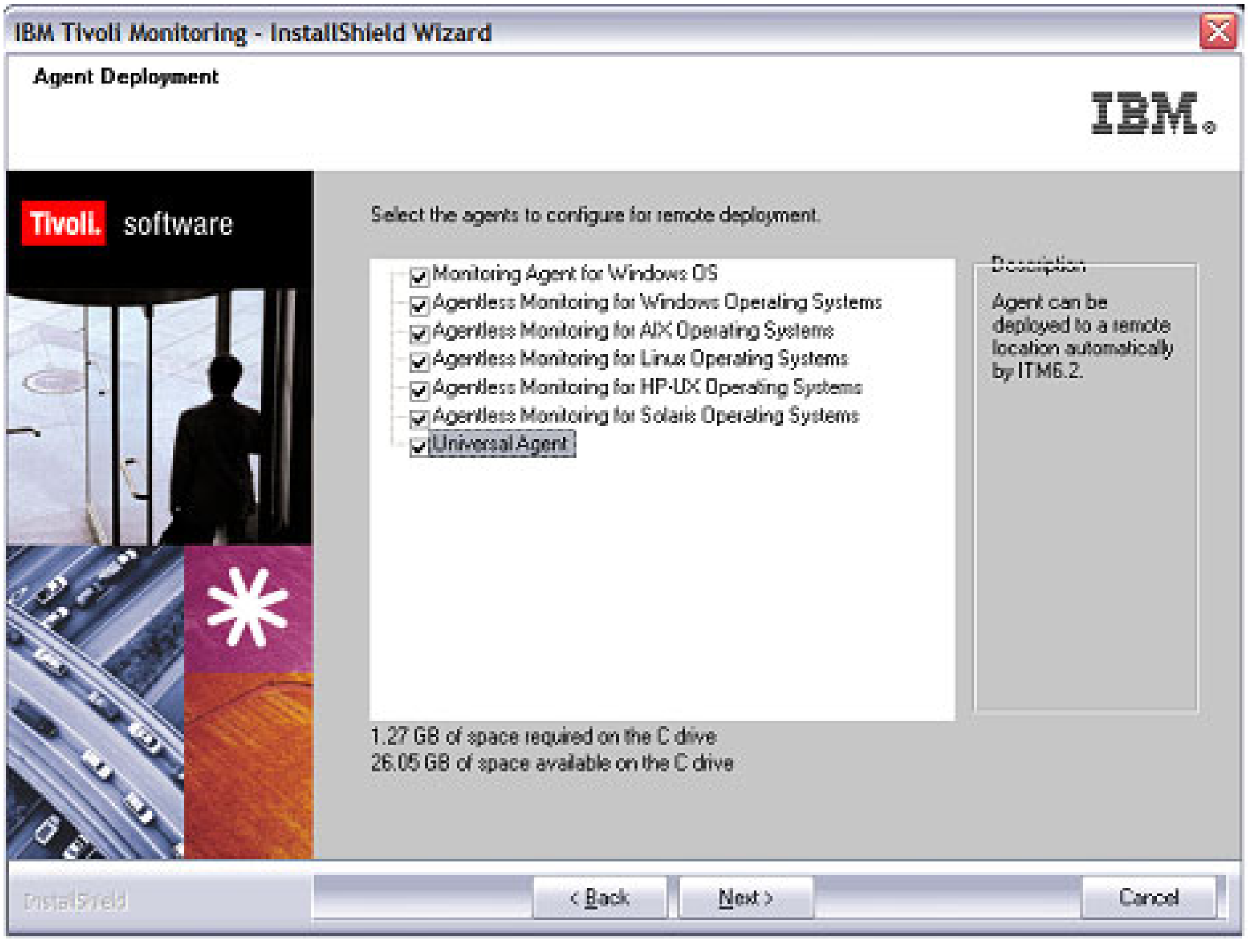
Notes:

1. To use one of the native Windows data collectors (WMI, PerfMon, the event log), the agentless monitoring server must run under Windows.

IBM recommends that you deploy a full-feature operating system agent to each agentless monitoring server to watch the CPU, memory, and network consumption of the agentless monitors themselves.

Deployment options for agentless monitors

As with other IBM Tivoli Monitoring agents, you can either install agentless monitors or deploy them from your site's deployment depot. When installing Tivoli Monitoring, you can add the agentless monitors to your site's deployment depot, as shown in .Figure 11



Select the agents to configure for remote deployment.

- ☑ Monitoring Agent for Windows OS
- ☑ Agentless Monitoring for Windows Operating Systems
- ☑ Agentless Monitoring for AIX Operating Systems
- ☑ Agentless Monitoring for Linux Operating Systems
- ☑ Agentless Monitoring for HP-UX Operating Systems
- ☑ Agentless Monitoring for Solaris Operating Systems
- ☑ Universal Agent

Description

Agent can be deployed to a remote location automatically by ITM6.2.

1.27 GB of space required on the C drive
26.05 GB of space available on the C drive

Figure 11. Adding agentless monitors to the deployment depot

You also have the full range of remote-deployment options, as explained in Chapter 3, “Deployment phase,” on page 83, at your disposal when planning how best to deploy agentless monitors across your environment. These include:

- The Tivoli Enterprise Portal's deployment features.
- The tacmd CLI commands.

As required for the deployment of any IBM Tivoli Monitoring agent, remote deployment of an agentless monitor to an agentless monitoring server requires that an OS agent be running on that machine. For example, if the agentless monitor runs on an AIX operating system, the IBM Tivoli Monitoring AIX agent must first be running on it to remotely deploy that agentless monitor. In addition, the OS agent is required to configure a server's agentless monitors via the Tivoli Enterprise Portal.

The agentless monitors are included on the same Agents DVD as the traditional OS agents and the Universal Agent.

Documentation resources for agentless monitoring

Table 8 lists the IBM Tivoli Monitoring manuals that detail the configuration and usage of the agentless monitors.

Table 8. User's guides for the agentless monitors

Title	Document number
IBM Tivoli Monitoring: Agentless Monitoring for Windows Operating Systems User's Guide	SC23-9765
IBM Tivoli Monitoring: Agentless Monitoring for AIX Operating Systems User's Guide	SC23-9761
IBM Tivoli Monitoring: Agentless Monitoring for Linux Operating Systems User's Guide	SC23-9762
IBM Tivoli Monitoring: Agentless Monitoring for HP-UX Operating Systems User's Guide	SC23-9763
IBM Tivoli Monitoring: Agentless Monitoring for Solaris Operating Systems User's Guide	SC23-9764

Problem-diagnosis tools available for agentless monitoring

Log files are available on both the remote system being monitored and the agentless monitoring server that is doing the polling. The following agent log files are available in these locations on the monitoring server:

- Windows: `C:\IBM\ITM\TMAITM6\logs`
- Linux and UNIX: `/opt/IBM/ITM/logs`

On the remote system, the following log resources are at your disposal, depending on the type of system being monitored and the monitoring API used:

- Windows: event logs
- Linux and UNIX: SNMPD or CIM log

Tivoli Universal Agent deployments

There are some special considerations for deploying the Tivoli Universal Agent. This section describes some strategies and techniques for ensuring a successful deployment of the Tivoli Universal Agent.

This section does not attempt to address the creation of Tivoli Universal Agents. Detailed information is available in the *IBM Tivoli Universal Agent User's Guide* and the *IBM Tivoli Universal Agent API and Command Programming Reference Guide*.

Tivoli Universal Agent versioning considerations

One of the unique challenges of deploying Tivoli Universal Agent solutions is the Tivoli Universal Agent versioning. As schema changes are made to the Tivoli Universal Agent, the Tivoli Universal Agent version is incremented. Not all changes cause the version to be incremented: for example, adding one ore more new attributes to the end of an existing attribute group does not trigger a version change. But when you do things like rename, resize, or delete an attribute, the version is updated.

A typical Tivoli Universal Agent looks like `AGENT00` in the portal client. If the Tivoli Universal Agent is updated, the Tivoli Universal Agent name is changed to `AGENT01` and then if updated again, to `AGENT02`. As the agent version is updated, the situations and workspaces must be modified to work with the new agent name. A custom workspace written for AGENT00 does not work with AGENT01. Therefore, use the following strategy:

Create and thoroughly test your Tivoli Universal Agent solutions in your test environment. After your Tivoli Universal Agent solution has been completed, reviewed, and signed off by the stakeholders, you can begin the process of moving it into production. If the Tivoli Universal Agent is finalized, you can install a duplicate Tivoli Universal Agent into production. If you want to have identical Tivoli Universal Agent versions in test and production, you can reset your test version to `00` using the steps documented in the *IBM Tivoli Universal Agent User's Guide*.

A solution used by many users to handle the complexities of versioning is to develop your agent using a different agent name during your development phase. Develop the agent using a different name and after the agent is working and optimized, rename the agent in your test environment before promoting it to production.

To avoid issues with Tivoli Universal Agent versioning entirely, consider developing your custom monitoring solution with the Agent Builder.

Tivoli Universal Agent firewall considerations

The Tivoli Universal Agent is similar to any other agent. The Tivoli Universal Agent connects to the monitoring server using your selected protocol (IP.PIPE, IP.UDP, or IP.SPIPE). However, in addition to communication between the Tivoli Universal Agent and the monitoring server, there can also be communications from the Tivoli Universal Agent to the monitored system.

There are three data providers that frequently have remote connections (ODBC, SNMP, and Socket). You can have remote connections with other data providers like the API data provider as well. When you are in an environment with firewalls, you must ensure that you can establish communications between the monitored system and the Tivoli Universal Agent.

- For SNMP, the default port is 161. The Tivoli Universal Agent contacts the managed system on this port.
- For SNMP traps, the Tivoli Universal Agent listens on port 162.
- For ODBC, the default port varies depending on the database server. For example, the port for DB2 is 50000.
- For the Socket Data Provider, the data collection program writes to a socket on the Tivoli Universal Agent using default port of 7500.

Large-scale deployment strategies

When deploying large numbers of similar Universal Agents you need to consider strategies so that you do not have to manually install a Tivoli Universal Agent on dozens or hundreds of servers. There are a couple of solutions to large scale deployments.

One strategy is to use Tivoli Monitoring remote deployment to distribute Universal Agents to multiple servers. When distributing the Tivoli Universal Agent, remote deployment distributes the Tivoli Universal Agent `MDL` file and associated scripts and files.

There are challenges to `MDL` file updates and control of the file rights. Additionally deploying version `05` to a new server creates version `00` on the new server. This challenge can be controlled by using your own packages which have a target version and verifies the agent is at the correct version level. Additionally, the default permissions on the scripts associated with the Tivoli Universal Agent are 777. Perform testing in your environment and change the permissions to reduce the filesystem permissions.

Another strategy to use in large scale environments is to create one or more metafile servers, using the KUMP_META_SERVER parameter. By specifying **KUMP_META_SERVER**, you tell the Tivoli Universal Agent which server is the Metafile Server. Then, when updates are made to the `MDL` file, only the Metafile Server needs to be updated. When using Metafile Servers, the Tivoli Universal Agent opens a socket to the Metafile Server. So, when using the Metafile Server in environments with firewalls, it is sometimes necessary to create multiple Metafile Servers (typically one for each network zone). Metafile Servers are well documented in the *IBM Tivoli Universal Agent User's Guide*.

Using Universal Agents with remote monitoring servers

There is a known limitation when using the Tivoli Universal Agent with both hub and remote monitoring servers. If you connect your Tivoli Universal Agent to remote monitoring servers, certain required files (kum.cat and kum.atr) are not automatically propagated to the hub monitoring servers. There are two options to address this limitation.

- The preferred approach is to connect your Tivoli Universal Agent to the hub prior to performing the **kumpcon import** and **um_console import** command. This approach causes the required files to be created on the hub monitoring servers. For monitoring to work as desired, move the Tivoli Universal Agent to the remote monitoring server.
- You can also leave the Universal Agent connected to a remote monitoring server and copy the required files from the monitoring server to the hub monitoring server. Because this approach requires a recycle of the hub monitoring server, it is the less desirable approach.

Mainframe users

Mainframe environments have some unique considerations. There are features that are available only when running a z/OS hub monitoring server and features that are available only when running a distributed hub monitoring server. This section outlines those considerations so that you can make the best choice for your environment.

Unique z/OS hub monitoring server features

The z/OS hub monitoring server allows you to take advantage of RACF® authentication. However, the OMEGAMON for MQ Configuration product has some specific integration with RACF that requires a z/OS hub in order to take advantage of RACF authentication within the OMEGAMON for MQ Configuration product.

The z/OS hub does not provide the Hot Standby feature. High availability is achieved using a movable hub solution as described in the *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

Note: The z/OS environment does not support the Tivoli Universal Agent.

Unique distributed hub monitoring server features

Two features that are provided on a distributed hub monitoring server environment are not available on z/OS hub monitoring server environments.

- Remote deployment
- Hot Standby

  The distributed hub monitoring server has a feature called Hot Standby to assist with high availability and disaster recovery scenarios. Many users choose not to use the Hot Standby feature and instead deploy OS Clusters for high availability and disaster recovery.

Linux on z/VM systems

Many mainframe users run Linux on their z/VM systems. Many different Tivoli Monitoring components can be installed in Linux on z/VM environments, including the monitoring server, portal server, monitoring agent and warehouse-related components. Each Linux environment can be configured with monitoring server or portal server software, or both.

Multi-hub environments

Large users who go beyond the limits of a single hub monitoring server environment must consider additional factors. Tivoli Monitoring V6.2.3 has been tested with environments with as many as 10000 agents. Some users need multiple hub monitoring servers to handle the tens of thousands of agents in their environment. Following are some considerations to take into account when deploying multiple hub monitoring servers.

Sharing a warehouse database:

You can share a single warehouse database with multiple hub monitoring servers, but there are additional considerations when choosing this deployment option. First, you must take into account scalability of the Warehouse Proxy and Summarization and Pruning Agents. Use the Warehouse load projection spreadsheet, which can be found in the Tivoli Integrated Service Management Library by searching for "warehouse load projections" or the navigation code "1TW10TM1Y."

With multiple hubs and more than 10000 agents, you increase the likelihood of exceeding the capacity of the Warehouse. Be aware of how much data you are collecting to ensure that you do not exceed the capacity of the warehouse database. To get an idea of the capacity of the Summarization Pruning agent with your warehouse database, consider using the measurement approach discussed in “Locating and sizing the Summarization and Pruning Agent” on page 50.

In addition to scalability, there are specific deployment requirements when a warehouse database is shared between hub monitoring servers. First, you can run only one Summarization and Pruning Agent in only one of the two monitoring server environments. The single Summarization and Pruning Agent is responsible for summarizing and pruning the data for all of the data in the Warehouse. The summarization and pruning configuration settings are maintained by the portal server that is specified in the Summarization and Pruning Agent configuration dialog.

Due to the complexity and potential scalability issues of sharing a warehouse database across multiple hub monitoring servers, you might want to maintain multiple warehouse databases. To build reports across the databases, use Federation capabilities or create a data mart that merges that content from multiple warehouse databases.

You cannot set up different summarization and pruning schedules for each of the hub monitoring server environments. In addition, you must also ensure that the hub with the Summarization and Pruning Agent is patched and maintained so that it is a superset of the two monitoring servers. If you install the database agents in one hub, then you must install the application support for the database agents on the hub monitoring server and portal server in the hub environment with the Summarization and Pruning Agent. If you install a fix pack on one hub, then you must ensure that it is also installed on the hub with the Summarization and Pruning Agent, which ensures that the Summarization and Pruning Agent is aware of all attribute groups and attributes that can be collected.

Sharing customization:

When using multiple hubs, most customization can be shared between the two hub environments. Customization includes situations, policies, workspaces, managed systems lists, and Tivoli Universal Agent solutions. In the Tivoli Monitoring V6.2.3 release a number of CLIs were added to the product to do bulk imports and exports of situations, policies, and workspaces. For details on the new CLIs, see the *IBM Tivoli Monitoring: Command Reference*. Most of the customization can be cleanly exported from one monitoring server environment to another monitoring server environment using tools that are identified in “Maintaining an efficient monitoring environment” on page 119.

Accelerating your custom monitoring

There are ways of accelerating the creation and deployment of custom monitoring solutions. First, you can access many solutions already available in the Tivoli Integrated Service Management Library. Check the Tivoli Integrated Service Management Library site before creating any custom solutions. Even if the Tivoli Integrated Service Management Library solution does not meet all of your needs, it is probably easier to extend the capabilities of an existing solution rather than create a new one. If you need to create a custom monitoring solution, the following search criteria for the Tivoli Integrated Service Management Library might provide solutions that can help you:

- "SNMP MIB to Universal Agent utility" or navigation code "1TW10TM3P"
- "Uses of the Universal Agent" or navigation code "1TW10TM25"

Planning and project management

Before you begin the implementation phase of your deployment, you must develop a detailed project plan. Chapter 3, “Deployment phase,” on page 83 helps you understand the detailed steps required to successfully install and configure your Tivoli Monitoring V6.2.3 environment. This section describes some additional tasks that must take place before the initial implementation takes place.

The Project Management and Planning phase must be led by a certified project manager working with a deployment architect. The project manager ensures that all required tasks are identified and documented. Thus, providing a framework for tracking the progress of the deployment.

The most important task to complete before proceeding to the implementation phase is to identify and document the key requirements. These requirements are typically business objectives that have been identified by senior management. After the business objectives have been identified, the Tivoli Monitoring requirements and implementation details can be created by architect. These details include the following:

- High-level design for the Tivoli Monitoring components.
- Naming conventions for situation, workspaces, queries, managed systems lists, and Take Actions commands.
- Any changes to existing production environments such as firewall changes, and server prerequisites.
- Monitoring strategy

  The monitoring strategy should specify exactly what will be monitored, and how; what situations and automation policies will be created; what events will be forwarded to other event management systems; and similar details. Working out the specifics of the monitoring strategy is perhaps the most challenging aspect of planning.

The details provided in this guide are intended to help the architect design the right solutions to meet the needs of all Tivoli Monitoring stake holders, given the constraints of the environment and the business objectives.

Estimating deployment tasks

The following estimating approach provides details of each logical task so that the estimates can be applied to any piece of work. No distinction is made between the tasks carried out in test and the tasks carried out in production in terms of how long the tasks take to complete. The only difference is that in production there are more barriers to starting the work because of the need for access to user IDs, firewalls, and change control.

In each section a description of the tasks necessary to deploy those components is provided.

- “Install server components on Windows and UNIX” on page 77
- “Install server components on z/OS” on page 77
- “Install data warehousing components” on page 77
- “Install and configure event integration components” on page 77
- “Install and configure monitoring agents” on page 78
  - “Self-describing monitoring agents” on page 79
- “Setting up situation-based monitoring” on page 79
- “Creating policies and workflows” on page 79
- “Creating workspaces” on page 79
- “Creating and deploying Tivoli Universal Agent applications” on page 80
- “Transferring skills” on page 80
- “Scheduling the initial deployment” on page 80
- “Scheduling for fix packs” on page 80

Install server components on Windows and UNIX

The first step is to install the Tivoli Monitoring server components. In general, for planning purposes, allow one day for this task. In some cases, installing the server components takes longer than one day due to change control processes or lack of access to key personnel such as database administrators and network administrators. By arranging for key personnel to be available before beginning the installation, you ensure a timely deployment.

Tivoli Monitoring includes the following server components:

- DB2
- Monitoring server
- Portal server
- Local operating system agent

Note: Although IBM Java is included with the Tivoli Monitoring V6.2.3 distribution, special considerations might need to be made if there are existing versions of Java required on the server where Tivoli Monitoring V6.2.3 is to be installed. Thus, Java might be a specific component that also needs to be addressed as part of your planning process.

Install server components on z/OS

Most users with a z/OS monitoring server deploy both z/OS agents and a monitoring server. The estimates in the following paragraph include the time requirements for installing both the agents and the monitoring server components.

In general, you need to allow a day for loading the software off tape and performing the receive, apply, and accept processing. The elapsed time depends on the number of products and the amount of processing power allocated to the installation LPAR, typically a test environment. If you have more than two products, then allow more time. As a rule of thumb allow half a day per product with a minimum of one day.

For more information search for "SMP/E installation on z/OS" or navigation code "1TW10TM3M" at the Tivoli Integrated Service Management Library.

Install data warehousing components

In general, allow a day to install and configure the Warehouse Proxy Agent and the Summarization and Pruning Agent. The installation is straightforward but then you need to verify that data is being collected and summarized successfully, which has inherent delays because the minimum warehouse interval is one hour and there is no way to force the warehousing of data.

Install and configure event integration components

Allow one day to install the event synchronization component and configure event forwarding to either Tivoli Enterprise Console or Netcool/OMNIbus. This estimate includes time to confirm that events are being forwarded to Tivoli Enterprise Console or Netcool/OMNIbus and synchronized back to Tivoli Monitoring.

Because this step requires the installation of new baroc files for the Tivoli Monitoring V6.2.3 events (Table 9 on page 78 lists the update history for these files), you must plan for at least one recycle of your Tivoli Enterprise Console server. Since the installation of new baroc files for Tivoli Monitoring V6.2.3 requires that the Tivoli Enterprise Console Server be recycled, users must plan for an outage in their Tivoli Enterprise Console Server. This means you must use your normal change control processes to schedule a change window. The estimated one-day duration to implement the Tivoli Enterprise Console changes does not include creating any custom rules or baroc files. The one day includes implementing the default baroc files included with Tivoli Monitoring V6.2.3.

When using the Tivoli Enterprise Console, you need a baroc file for each type of agent. For the packaged agents, a baroc file is automatically installed on your monitoring server. The baroc files are placed in the *CANDLE_HOME*\cms\TECLIB directory. For Tivoli Universal Agent solutions, it is necessary to create a baroc file for each Tivoli Universal Agent solution. A tool available on the Tivoli Integrated Service Management Library generates the baroc file for the Tivoli Universal Agent solutions. To find this tool search for "BAROC file generator" or navigation code "1TW10TM43" at the Tivoli Integrated Service Management Library.

Table 9. Update history for the baroc files for IBM Tivoli Monitoring agents and components

IBM Tivoli Monitoring agent/component	Last updated for which release?
Tivoli Enterprise Monitoring Server (kib.baroc)	V6.2.3 fix pack 1
Remote deployment (kdy.baroc)	v6.2.1
Warehouse Proxy Agent (khd.baroc)	V6.2.3
Summarization and Pruning Agent (ksy.baroc)	V6.2.3
Windows agent (knt.baroc)	V6.2.3
Linux agent (klz.baroc)	V6.2.3
UNIX agent (kux.baroc)	V6.2.3 fix pack 1
Unix Log Alert agent (kul.baroc)	V6.2.1
i5/OS agent (ka4.baroc)	V6.2.3 fix pack 1
Tivoli Performance Analyzer (kpa.baroc)	V6.2.3 fix pack 1
Tivoli Performance Analyzer OS agent domain (kp3.baroc)	V6.2.3 fix pack 1

Note: File omegamon.baroc contains the base event class definitions for all Tivoli Monitoring events; it is automatically installed on Tivoli Enterprise Console when event synchronization is installed.

Install and configure monitoring agents

Windows and UNIX systems:

The time required to install and configure an agent differs depending on the agent. Using operating system agents as a baseline, you should be able to install a monitoring agent in 10 minutes. Taking operating system agents as a baseline, then in theory for UNIX and Windows you can install them in 10 minutes. However, when planning, allow time to debug some failures. Ideally, in an 8-hour day you should be able to install 48 error-free agents. If scripts are allowed to run overnight, nearly 150 agents can be installed per day. Even so this is probably not realistic for a couple of reasons.

First, you must spend time debugging problems and due to issues with access to user IDs that have sufficient authority to perform the installation and access to the installation image. In most environments a single person can reasonably aim for 50 agents per day if user access is provided in a timely manner. Even if there are multiple agents to be installed per computer, then this is still a reasonable estimate for installing an operating system agent, a Tivoli Universal Agent, and a database agent.

Note: If software distribution is being used, then after a package has been built, the number of agents that can be deployed in a given time frame increase.

z/OS systems:

If you take CICS, DB2, and z/OS agents as an example, in general allow one day to configure each agent, assuming one CICS region and one DB2 subsystem. For additional CICS regions, this means more CICS table changes. Additional DB2 subsystems are defined in ICAT and you can estimate one hour per subsystem.

Upgrading your environment to allow for additional LPARs is typically quicker because you can use batch mode replication. Allow one day per LPAR.

For more information search for "SMP/E installation on z/OS" or navigation code "1TW10TM3M" at the Tivoli Integrated Service Management Library.

Self-describing monitoring agents

The self-describing agent feature in Tivoli Monitoring V6.2.3 makes it possible for new or updated IBM Tivoli Monitoring agents to become operational after installation, without having to perform additional product support installation steps. Self-describing agents apply version updates to other components automatically without the need to recycle your hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or remote Tivoli Enterprise Monitoring Server. This feature integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure.

For more information, see “Self-describing agent installation” on page 347.

Setting up situation-based monitoring

It is difficult to estimate the time required to set up situation-based monitoring. It is easy to turn on various predefined situations if they are not already running, but not always practical. Before starting or turning off predefined situations, you need to review them to determine which ones are appropriate for your environment. Moreover, you need to review the default thresholds for the situations that you decide to use and adjust them to suit your site policies and requirements.

Note: Any changes made to predefined situations are overwritten when application support is updated. If you modify predefined situations, save them with a new name, and use them instead of the original predefined situation for your monitoring.

For creating new situations, allow for 20 situations per day, including testing.

You may also want to create managed system groups, which are lists of managed systems to which you want to distribute the same situations.

When creating situations and managed systems lists, it is extremely important to choose a naming convention that is meaningful. A similar naming convention must be used for situations, managed systems lists, workspaces, and queries. “Customizing your environment” on page 85 describes some options when considering the situation and managed system group naming conventions.

Creating policies and workflows

Policies can be used to perform complex monitoring and automation. Time must be allocated to determine when situations are inadequate to provide complex thresholding and Take Actions. When you have determined which complex monitoring and automation cannot be accomplished, put proper plans in place to ensure effective policies are created. When planning policies, choose a naming convention that is similar to your convention for naming situations. Plan enough time to adequately develop and test your policies.

Creating workspaces

Again this is difficult to estimate. Typically, you not know exactly what you want to monitor. Be careful not to create something that will be hard to maintain in the future. Workspaces vary in complexity a great deal, so a graphic view with icons does not take long, but a workspace with links using variables takes quite a while to set up and test. On balance, aim for ten workspaces per day along with associated custom navigator items.

Creating and deploying Tivoli Universal Agent applications

Installing Tivoli Universal Agent applications is almost impossible to estimate because of the varied nature of what the Tivoli Universal Agent can monitor. If you are using user-provided scripts, for instance, then you can create an application in half an hour. If you have to create your own scripts then that can take days depending on your skill level with scripting and the complexity of the script. Likewise, log file monitoring can be straightforward or complex depending on the format of the log file.

Adjust the time estimates on a case-by-case basis. Be very careful in moving metafiles from development and test to production because of the versioning nature of the Tivoli Universal Agent. Depending on how they are implemented, changes to the metrics being collected can cause the version of Tivoli Universal Agent to increment, resulting in loss of all customization done to the Tivoli Universal Agent.

Make sure that all the desired metrics are added to metafile and maybe two or three placeholder generic metrics can be added to accommodate future modifications without needing a version change. Another option is to use two very generic metrics such as MetricName and MetricValue and write situations such as `scan for string with in a string` to catch the data. When the metafiles are imported, make sure the Tivoli Universal Agent is connected to the hub monitoring server first and then later it can be moved to any desire remote monitoring server.

Transferring skills

Much of the skills transfer can occur during the installation and configuration process if you are working closely with key staff. You can still factor in an additional three days to cover the same subjects as in the IBM Tivoli Monitoring Administration course. Also add one to two days for the Tivoli Universal Agent depending on the complexity of the monitoring requirements. Other skills transfer can be estimated at one day per agent type. For some of the z/OS-based agents this can be a bit longer because of the two different 3270 mainframe and the portal client interfaces, so allow two days for CICS and DB2 agents.

These time estimates correspond roughly to the number of days it takes to deliver formal training.

Scheduling the initial deployment

The initial deployment of the Tivoli Monitoring environment is the most time-consuming. The majority of the customization is performed during or immediately after the initial deployment. Customization includes modifying workspaces, modifying situation thresholds, building managed systems lists, and defining Take Action commands.

Scheduling for fix packs

In a typical user environment, allocate one day to upgrade the Tivoli Monitoring infrastructure components (hub monitoring server, portal server, Warehouse Proxy Agent, Summarization and Pruning Agent). In a multi-hub environment, allow one day for each hub. Then, time must be allocated to upgrade the agents. Typically, the schedule is driven largely by Change Control windows and not by the time it takes to deploy the fix pack to the agents. However, for planning purposes use the following time estimates.

These times assume that there is adequate network bandwidth between the remote monitoring server deployment depot and the agents. In environments with slow network links between the remote monitoring server and the agents, evaluate the size of the fix pack files to calculate the rate of transfer. Because each fix pack is a different size and each network has unique characteristics, calculate this for your environment before starting the upgrade.

Performing agent upgrades in parallel can be done with the itmpatchagents tool. For environments with adequate network bandwidth and two people performing upgrades in parallel, plan to upgrade approximately 500 agents per day.

Finally, following the installation of a fix pack, plan time to test your environment. For Tivoli Monitoring infrastructure fix packs, spend two person days thoroughly testing your environment. For application agents, there is little risk that the Tivoli Monitoring infrastructure components were affected by the fix pack. Therefore, no more than one person day need be allocated to test the environment following the installation of an application fix pack.

Staffing

The table below lists the staffing and time estimates for various Tivoli Monitoring tasks.

Table 10. Staffing estimates

Tivoli Monitoring tasks	**Hours required**	**Number of people required**	**Skills required**
Discussing the project plan including implementation and support in place by IBM	16	2	Medium
Downloading the correct media for installation	8 hours	1	Medium
Completing the hardware check	4 hours	1	Medium
Creating the warehouse database with sufficient space, user IDs and tuning parameters	4	1	High (DBA)
Configuring firewall for ports	8	1	High (Network Team)
Installing core components (portal server, monitoring server, Warehouse Proxy Agent, and Summarization and Pruning Agent) 1 on page 82	20	2	High
Disabling the default situations	2	1	Medium
Creating the managed systems list	8	1	Medium
Creating the custom situations	32	2	Medium
Deploying first 50 mix of Windows OS, UNIX and Linux OS and different application agents	40	1	High
Deploying situations to appropriate managed systems lists	8	1	Medium
Verifying all agents and core components are active and working correctly	40	2	High
Verifying events flowing to Tivoli Enterprise Console server	4	1	Medium
Backing up the environment (core components)	4	1	High
Deploying agents in chunks of 100 agents, depending upon the requirement. Sizing includes adding computers to the managed systems lists.	24	2	Medium
Enabling only required attribute groups for warehousing as mentioned in the project plan	8	1	Medium
Verifying warehouse data is flowing from agents to warehouse database	8	1	High
Performing frequent health checks of the environment, more frequent in first few months and then later less frequent checks	4	1	High
Creating and deploying custom monitoring solutions with the Universal Agents - *simple agents*	8	1	Medium
Creating and deploying custom monitoring solutions with the Universal Agents - *complex agents*	40	1	High
Configuring Performance Analyzer Agent	.5	1	Medium
Adding application support to Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server and Tivoli Enterprise Portal Client	.5	1	Medium

Table 10. Staffing estimates (continued)

Tivoli Monitoring tasks	**Hours required**	**Number of people required**	**Skills required**
Reconfiguring Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal Client	.5	1	Medium
Configuring historical data collection	2	1	Medium
Moving the Tivoli Performance Analyzer configuration database 2	4	1	Medium
Installing Tivoli Performance Analyzer domain support in console mode or in GUI mode	2	1	Medium
Installing reports	1	1	Medium
Inspection, Validation, and Enabling or Disabling of the default tasks to suit the environments and requirements	2	1	Medium
Customizing the default tasks to suit the environment and requirements	2	1	Medium
Creating custom tasks	2	1	Medium

Notes:

1. Installing core components depends on the size of the environment and the complexity regarding multiple Warehouse Proxy Agents.
2. In some situations you might want to move the Tivoli Data Warehouse database to a different computer or Relational Data Base Management System. The migration procedure described in Tivoli Data Warehouse manuals does not cover Performance Analyzer configuration data. For more information, see *Moving the Tivoli Performance Analyzer configuration database* in the IBM Tivoli Monitoring information center.

Use the following Project Plan as a template to ensure that all tasks are planned during your installation. This Project Plan is located on the SAPM Technical Exchange wiki at http://www.ibm.com/developerworks/wikis/pages/viewpageattachments.action?pageId=9595.

Chapter 3. Deployment phase

Proper planning ensures that the deployment phase goes smoothly. The following section documents the steps necessary to deploy and configure your Tivoli Monitoring V6.2.3 environment. During the initial installation, use a phased approach. Perform these steps in development and test environments before applying them in the production environment.

Pre-installation checklist

Use the following checklist for your pre-installation:

- Verify that your hardware meets the requirements for Tivoli Monitoring V6.2.3 core components.
- Verify that the correct media is downloaded (not the upgrade install media).
- Verify that you have the correct media for your hardware architecture. Some operating systems support both 32-bit and 64-bit kernels. To check which kernel version your system is running, use the following commands:

Table 11. Commands for determining your system's kernel version

System	Command
AIX	**bootinfo -K get 64** or **bootinfo -K get 32**
HP	**getconf KERNEL_BITS**
Linux	**rpm -q libstdc++ l grep libstdc++-2.9**
Solaris	**isainfo -b**

- Verify prerequisites are met for the databases.
- Confirm you have the appropriate authority on the system to be installed.
- Confirm the network team is contacted for required ports to be opened between the portal server, monitoring server, monitoring agents, and Warehouse Proxy Agents.
- Ensure the team responsible to deploy and implement Tivoli Monitoring solution is equipped with sufficient skills to complete the tasks successfully.

Give special consideration to Windows installations where you are using terminal services or some other *remote control* software to access hosts. In a perfect world, install this software while you are physically sitting at and logged in to the system console. As this might be impractical, you need to have LOCAL ADMINISTRATOR authority (domain administrator authority itself does not suffice).

If using something like terminal services, you need to be a member of the local user security policy called *Allow Logon Locally* as well as *Allow Logon through Terminal Services*. In some cases, you might need to work with the Windows engineers to help you maneuver through local and AD security issues. Security problems can initially manifest themselves as subtle inconsistencies but ultimately can inhibit a successful installation.

Installing the infrastructure components

Before installing any Tivoli Monitoring agents, you must first install the Tivoli Monitoring infrastructure components, which include the hub monitoring server, portal server, remote monitoring server, Warehouse Proxy Agents, and Summarization and Pruning Agent.

For the first phase of your deployment, deploy a relatively small environment, which provides you with the opportunity to test your production environment and ensure that it is running smoothly. Deploy the hub monitoring server and only two remote monitoring servers initially. The remote monitoring servers can successfully monitor up to 1500 agents, so using two remote monitoring servers during your initial deployment is not a problem. Firewall considerations might necessitate additional remote monitoring servers.

If you plan to use clustering to achieve a high availability configuration, configure the cluster before connecting any agents. Otherwise, it is necessary to reconfigure the agents to connect to the cluster rather than connecting to one of the nodes in the cluster. For information on setting up the hub monitoring server and portal server in an OS, see the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*.

Because the installation of application agent support on the monitoring server and portal server requires them to be recycled, install the application support on the monitoring server, portal server, and portal client for all agents that you expect to deploy in the next six to nine months. This likely includes the IBM Tivoli Monitoring for Databases, IBM Tivoli Monitoring for Messaging and Collaboration, ITCAM for SOA and any other products you plan to implement.

With the self-describing agent automated application support feature in Version 6.2.3, a new type of automated installation process is available which is internal to the IBM Tivoli Monitoring agent, Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal client. The self-describing agent capability must be correctly configured for a product before deployment. For more information, see "Enabling self-describing agent capability at the hub monitoring server" on page 216 and the *IBM Tivoli Monitoring: Administrator's Guide*.

To take advantage of the self-described agent capability, you must have the Tivoli Management Services at Version 6.2.3 (or higher).

During the initial installation phase, install the Warehouse Proxy Agent and Summarization and Pruning Agent, but do not begin using them until after you have successfully deployed your first 50 agents. Installing in this way gives you an opportunity to assess the health of your environment before adding to the complexity of your environment.

Configuration checklist

Use the following checklist for your configuration:

- Install all Tivoli Monitoring V6.2.3 core components (portal server, monitoring server, Warehouse Proxy Agent, and Summarization and Pruning Agent) so there can be a section for each component.
- Verify the correct protocols are selected. If SPIPE is chosen, make sure the encryption key string used is the same across the Tivoli Monitoring V6.2.3 enterprise environment.
- Verify the correct configurations are performed regarding data warehousing.

After installing the hub and remote monitoring server, ensure that you do not attempt to start a second `kdsmain` instance, which can corrupt your environment. Modify the monitoring server startup *CandleServer* script so that it looks as follows:

```
#
#Local change to check for another running kdsmain
#
if [ "$action" = "start" ]
then
  if ps -ef | grep -v grep | grep kdsmain
    then
      echo "There is a KDSMAIN running already"
      exit
  fi
fi
```

Some users run multi-hub monitoring servers on a single server for the development and test environments using different ports. If that is the case, then the previous script cannot work because `kdsmain` is running for the other hub. If that is the case, use extreme care to ensure that you do not accidentally start a second `kdsmain` for a given hub.

Do *not* enable warehousing at this time. Wait until all the agents for phase one have been installed and started, and the situations have been distributed.

Disable all the default situations by unassigning the managed system group and any agents present in the **Assigned** check box.

Create the managed system groups before creating the situations.

Distribute all newly created situations with naming conventions to the customized managed system group and not to `*NT_SYSTEM`, `*ALL_UNIX`. You must customize your situation thresholds before forwarding your events to Tivoli Enterprise Console or OMNIbus, which ensures that you do not cause any event storms.

You can enable Tivoli Enterprise Console forwarding at this time. Install your first 50 agents.

Using the installation method of your choice, install several OS monitoring agents. These can be installed locally on the server or through the remote deployment mechanism. Detailed steps on remote deployment are included in the following sections.

Note: Because the remote deployment of application agents depends on having an OS agent running on the server, always deploy the OS agents first.

Customizing your environment

Before progressing any further, you must perform customization to your environment. The following sections describe this customization.

Customizing situations, workspaces, and queries:

Perform some customizations of your situations to ensure that this first set of agents do not generate event storms. Many users choose to disable the default situations and then create their own situations. Careful thought must be put into the naming conventions of your situations. Many users include the following elements in their situation names.

- OS type
- Agent name or type, or both
- Business unit
- Physical location
- Severity

When choosing a name, keep in mind that the situations are sorted alphabetically in the Situation Editor. A typical situation might look like:

- `East_UNIX_High_CPU_Crit`

For more information on disabling the default situations and performing bulk work on situations see the *IBM Tivoli Monitoring: Command Reference*.

Choose similar naming conventions for any custom queries, managed system groups, and workspaces. Choose the same criteria such as physical location, business unit, agent type, and so on that you used for situations.

Another important consideration when creating situations is the **Display Item**, which by default, is not enabled. If you want to generate a unique Tivoli Enterprise Console Event for each item that triggered a situation or want to run a Take Action command against each item that triggered the situation, then you want to select the **Display Item** and choose the appropriate attribute.

Changing the default monitoring server configuration settings

There are a few monitoring server configuration settings that you might want to change from the default values. Consider changing the following settings to ensure a smooth running environment even as your environment scales.

- **KDCFC_RXLIMIT**
  - This is the buffer used for return queries, specified in KB. The default value is 2048 KB (equivalent to 2 MB). A value of 8192 KB (equivalent to 8 MB) seems to work well for most users.
  - Recommendation: 8192 KB
- **DEPOTHOME**
  - The location of the depot. The default location is

    Windows: `%CANDLE_HOME%\CMS\Depot`

    Linux and UNIX: `$CANDLEHOME/tables/`*`hub_tems_name`*`/depot`
  - Relocating the depot directory enables you to backup your Tivoli Monitoring environment without having to backup the very large depot directory. In addition, relocating the depot directory ensures that the depot will not fill up the filesystem where Tivoli Monitoring is running.

    The target directories listed below are examples:

    If `CANDLE_HOME` on Windows is located at `C:\IBM\ITM`, then relocate the depot to `D:\ITM\depot`

    If `CANDLE_HOME` on Linux and UNIX is located at `/opt/IBM/ITM`, then relocate the depot to `/data/ITM/depot`
- Bind a specific IP address

  KDEB_INTERFACELIST=192.100.100.100

  Note: Use this option only if the monitoring server and portal server are on separate servers. If they are on the same computer, there are going to be problems due to the multiplexed port 1920: The tacmd command will not be able to find the monitoring server, and portal server clients will not be able to find the portal server.
- Bind a specific host name

  KDEB_INTERFACELIST=caps001
- Bind the first IPV4 address associated with the current host name to be the default interface.

  KDEB_INTERFACELIST=!*

There are many other optional parameters documented in the *IBM Tivoli Monitoring: Administrator's Guide*. Review those parameters and determine whether any are required for your environment. See “Tivoli Enterprise Monitoring Server” on page 421 for information on monitoring server parameters for performance tuning.

Enabling historical collection of CandleNet Command Center logs

Some users choose to enable the historical data collection of the CandleNet Command Center® logs for the Tivoli Data Warehouse so that you can collect historical information about your situations, which can be very useful when debugging your environment. The CandleNet Command Center logs track status of internal Tivoli Monitoring components such as situations. Keep in mind that this requires warehouse space and adds load to your Tivoli Monitoring environment.

Installing your first 50 agents

Using the installation method of your choice, install several OS monitoring agents. These agents can be installed locally on the server or through the remote deployment mechanism. Because the remote deployment of application agents depends on having an OS agent running on the server, always deploy the OS agents first.

Remote deployment of OS agents

> Tivoli Monitoring V6.2.3 provides default **CreateNode** and **AddSystem** flags using the **tacmd** command to deploy OS agents as well as application agents remotely from a central location. It is very important that the depots be created on each monitoring server (hub and remote) and that you make sure the level of code, bundles, and packages in those installation depots is consistent. You can also use the shared depot. If a shared directory is accessible to all monitoring servers, it can be mounted across the monitoring server environment. This reduces the maintenance workload, since you have only one directory to maintain in a shared location, rather than maintaining depot directories on each monitoring server.

Postinstallation checklist

Use this postinstallation checklist to ensure the following items have been completed:

- Monitoring server (situations created)
- Portal server (check all aspects of portal server functionality such as workspaces)
- Perform a complete backup of all Tivoli Monitoring components
- If the self-describing agent feature is enabled, use the command `tacmd listappinstallrecs -t <pc>` to verify that self-describing agent support installation was successful after the agent connected to the infrastructure.

Configuring your warehouse agents

Now that you have your first 50 agents up and running smoothly, it is time to configure your Warehouse Proxy Agent and Summarization and Pruning Agent. There are the two major components to configuring your warehousing.

- First, configure the two agents and specify all of the steps necessary for the agents to communicate with the database server. These steps are done through the configuration panels that appear when you reconfigure the agents. Follow the steps in the install guide when performing this configuration. Choose the summarization and shift options that you chose during the planning phase.
- The second aspect to warehousing is to decide which attribute groups you are going to collect and at what intervals. Always use the Warehouse load projection spreadsheet before enabling additional historical collection. Thus, ensuring that you do not overload your warehouse environment. Start slowly and incrementally add attribute groups to your historical collection. This allows you to confirm that everything is working properly before you add more data to your warehouse.

  Enable one attribute group at a time and verify that the data is being collected, summarized and pruned properly. At this point, you can also see how many rows are getting written per data collection interval for the attribute group, by examining the entries in the WAREHOUSELOG table. The number rows written per interval is an important input parameter for the Warehouse load projection spreadsheet. If the WAREHOUSELOG is disabled, you can look at the self monitoring Warehouse Proxy Agent workspace. The workspace displays the top ten nodes with the greatest number of exports since the Warehouse Proxy Agent started, and the ten most recent errors in the last 24 hours.

At this point, install the warehouse monitoring solution that is available at the Tivoli Integrated Service Management Library by searching for "Data Warehouse DB activity" or navigation code "1TW10TM1X."

In addition, you must create a critical situation that monitors both your Warehouse Proxy Agent and Summarization and Pruning Agent to ensure that they are running. If they are not running, the situation must run a Take Action to automatically restart those agents.

When configuring and starting your Summarization and Pruning Agent, do not forget to set the **KSY_MAX_WORKER_THREADS** to the appropriate number for your environment. See “Locating and sizing the Summarization and Pruning Agent” on page 50 for recommendations on the number of worker threads.

If you have configured some or all of your monitoring agents to run autonomously, you might want to configure the Warehouse Proxy and Summarization and Pruning agents to run autonomously as well. See “Running the warehouse agents autonomously” on page 609 for more information.

Installing additional agents

At this point in your deployment, keep things simple by deploying a single hub monitoring server, a portal server, the warehouse components, and a couple of remote monitoring servers. If warehousing is critical, it can be enabled at this point. The environment must be evaluated thoroughly before moving beyond the initial 50 agents.

After the first 50 agents are deployed, deploy additional agents in groups of approximately 200, which allows you to verify all of the agents before progressing to the next step. As you scale up your environment and become comfortable with the deployment and validation of agents, you can increase the number of agents being deployed to 400 or 500 at a time.

Chapter 4. Tivoli Monitoring Startup Center

The IBM Tivoli Monitoring Startup Center is a graphical user interface tool that guides you through the setup of a new IBM Tivoli Monitoring environment. The Startup Center is not intended for upgrading existing IBM Tivoli Monitoring components. You use topology diagrams to configure and deploy an initial base IBM Tivoli Monitoring environment, which can be expanded later. The Startup Center reduces complexity, increases transparency, and simplifies your IBM Tivoli Monitoring deployment.

The Startup Center can run on both Windows and Linux Intel x86-32 systems. On Linux systems you must have installed the *GTK* support packages to run the tool. The IBM Tivoli Monitoring product image location that is used in the deployment must be accessible.

In the event of a failed installation, the Startup Center log files can help to diagnose the problem. The log files can be found in the following locations:

- On Windows systems: `%USERPROFILE%\.STARTUP\logs\`
- On Linux systems: `${HOME}/.STARTUP/logs`

Startup Center platform support

Table 12 and Table 13 include a list of platform support for launching the Startup Center or installing it locally on Windows and Linux Intel x86-32 operating systems. For remote installation on distributed platforms and the supported monitoring components, see “Supported operating systems” on page 139.

Notes:

1. On Windows systems, the Startup Center installs only 32-bit components to 64-bit systems.
2. The Startup Center is not supported on z/OS installations.

Table 12. Startup Center: Supported Windows operating systems

Operating system	Launch Startup Center or Install locally
Windows XP on x86-32	X
Windows Vista on x86-32	X
Windows 7 on x86-32	X
Windows 7 on x86-64	X

Supported Linux operating systems

Table 13. Startup Center: Supported Linux operating systems

Operating system	Launch Startup Center or Install locally
RedHat Enterprise Linux 5.0 Intel x86-32 with gtk support	X
SuSE Linux Enterprise Server 11 Intel x86-32 with gtk support	X

Startup Center prerequisite requirements

For a distributed installation, you must have DB2 already installed on the target machine where you want to install the Tivoli Enterprise Portal Server and the Warehouse Proxy Agent.

Remote Execution and Access (RXA) is an IBM developer toolkit that provides classes and methods to create an application that can establish a connection with a remote computer. The Startup Center relies on RXA to establish a connection with a remote computer. Therefore you must enable the target machines for RXA.

Enabling the RXA toolkit on UNIX and Linux systems

- You must ensure that the SSH protocol is installed and enabled on any target that you want to access by using SSH protocol. OpenSSH 3.71 (or higher) contains security enhancements that are not available in earlier releases.
- RXA cannot establish connections with any UNIX target that has all remote access protocols (rsh, rexec, or ssh) disabled.
- For RXA to communicate with Linux and other SSH targets by using password authentication, you must edit the `/etc/ssh/sshd_config` file on target computers and set the parameter `PasswordAuthentication` to `yes`. The default setting is `no`. After changing this setting, stop and restart the SSH daemon by using the following commands:

  ```
  /etc/init.d/sshd stop
  /etc/init.d/sshd start
  ```

Enabling RXA on Windows systems:

- Interprocess communications share (IPC$) is required to enable RXA on Windows systems. You can use the `net share` command to check the status of IPC$:

  ```
  C:\Documents and Settings\Administrator>net share
  ```

  If IPC$ is not set correctly, you can use the command `net share C$=C:` to set the IPC$.
- On Windows Server 2008 and Windows Vista, you might need to disable User Account Control if your account is not a domain user account.
- The account used to install IBM Tivoli Monitoring components on target machines must be in the Administrator group.

For more information, see "Remote Execution and Access" on page 112.

Deployment procedure

1. First select the appropriate Tivoli Monitoring Startup Center CD for your platform:
   - CD for Windows systems: `IBM Tivoli Monitoring V6.2.3 Startup Center for Windows 32-bit, Multilingual`.
   - CD For Linux systems: `IBM Tivoli Monitoring V6.2.3 Startup Center for Linux 32-bit, Multilingual`.
2. If you are downloading the Startup Center from the Passport Advantage® Web site, you must first unpack the Startup Center packages. To launch the Startup Center:
   - For Windows systems, use: `launchStartup.bat`.
   - For Linux systems, use: `launchStartup.sh`.

Note: All screens illustrated in this chapter are for Windows systems.
You have two types of installation for you to choose from: **local** or **distributed**.

Local installation

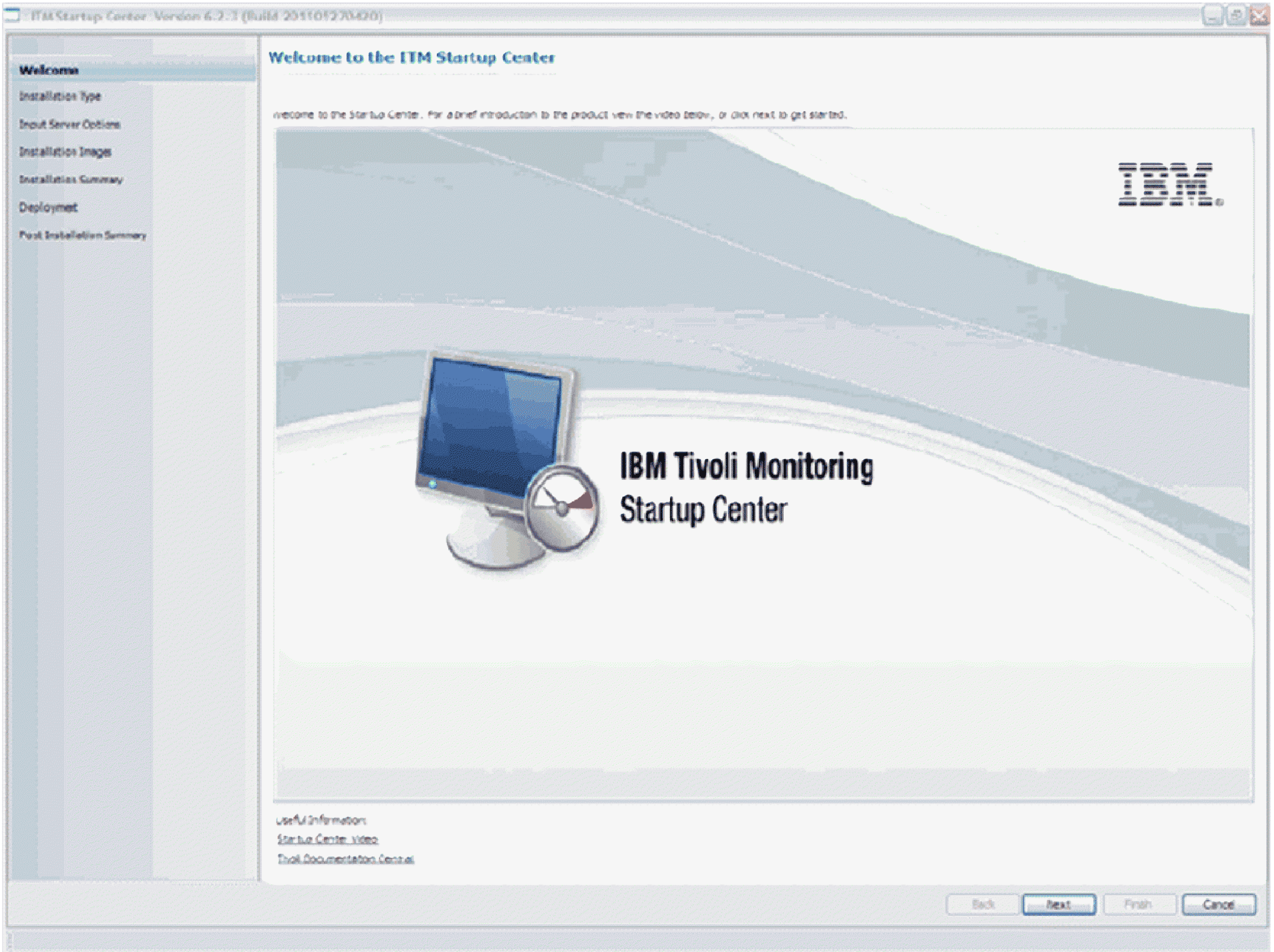
A predefined set of components are installed on the local machine. A local installation is the simplest form of an IBM Tivoli Monitoring installation and is best suited to small environments and evaluations. You can install the following components on single systems:

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server

- Tivoli Enterprise Portal desktop client
- Tivoli Enterprise Portal browser client
- Operating system agent

Complete the following steps to perform a local installation using the Startup Center:

1. Launch the Startup Center.
2. Click **Next** on the Welcome page.

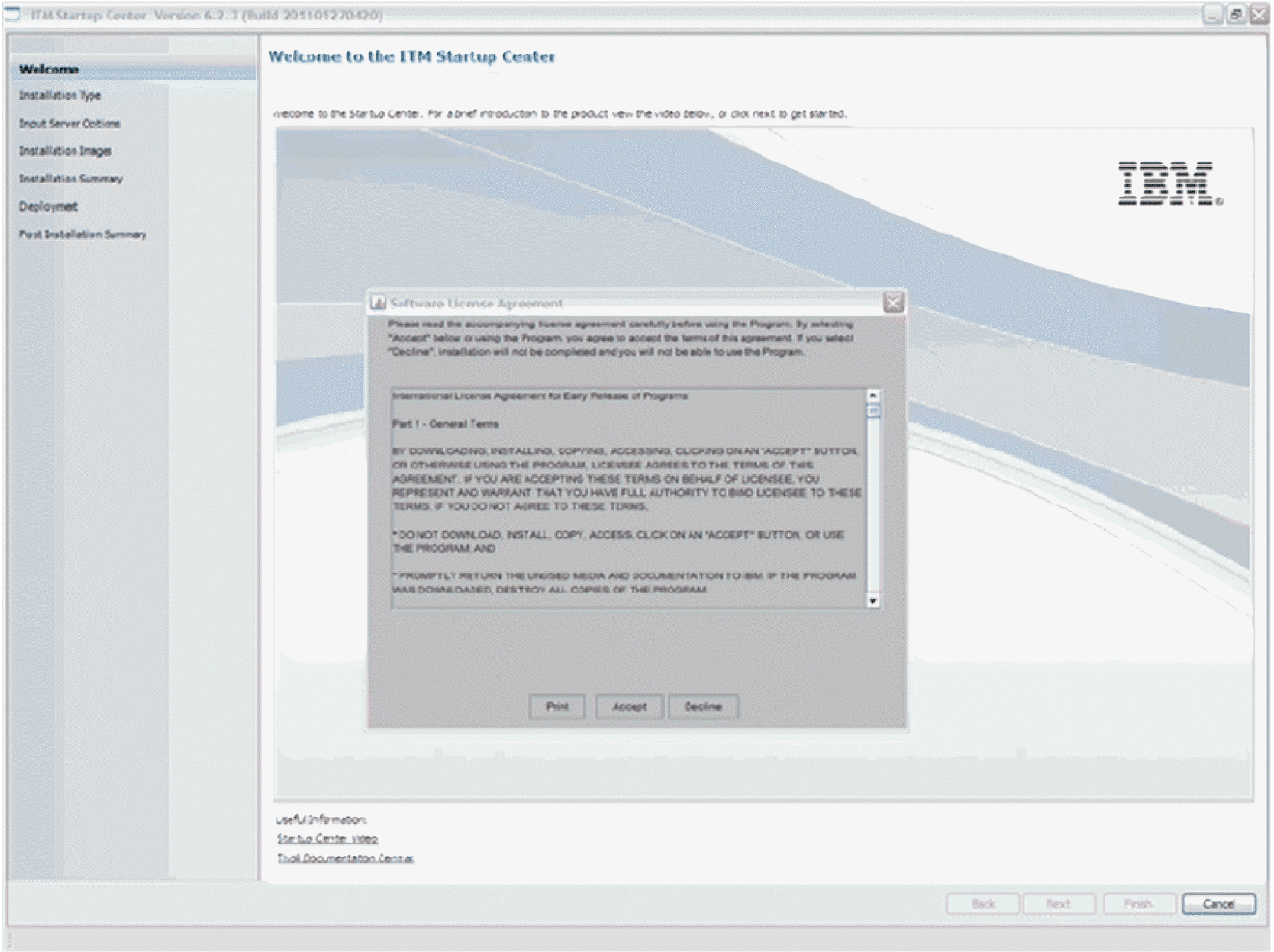


Welcome to the ITM Startup Center

Welcome to the Startup Center. For a brief introduction to the product view the video below, or click next to get started.

Figure 12. Tivoli Monitoring Startup Center Welcome page

3. Click **Accept** to accept the license agreement.



Welcome to the Startup Center. For a brief introduction to the product view the video below, or click next to get started.

Please read the accompanying license agreement carefully before using the Program. By selecting "Accept" below or using the Program, you agree to accept the terms of this agreement. If you select "Decline", installation will not be completed and you will not be able to use the Program.

International License Agreement for Early Release of Programs

Part 1 - General Terms

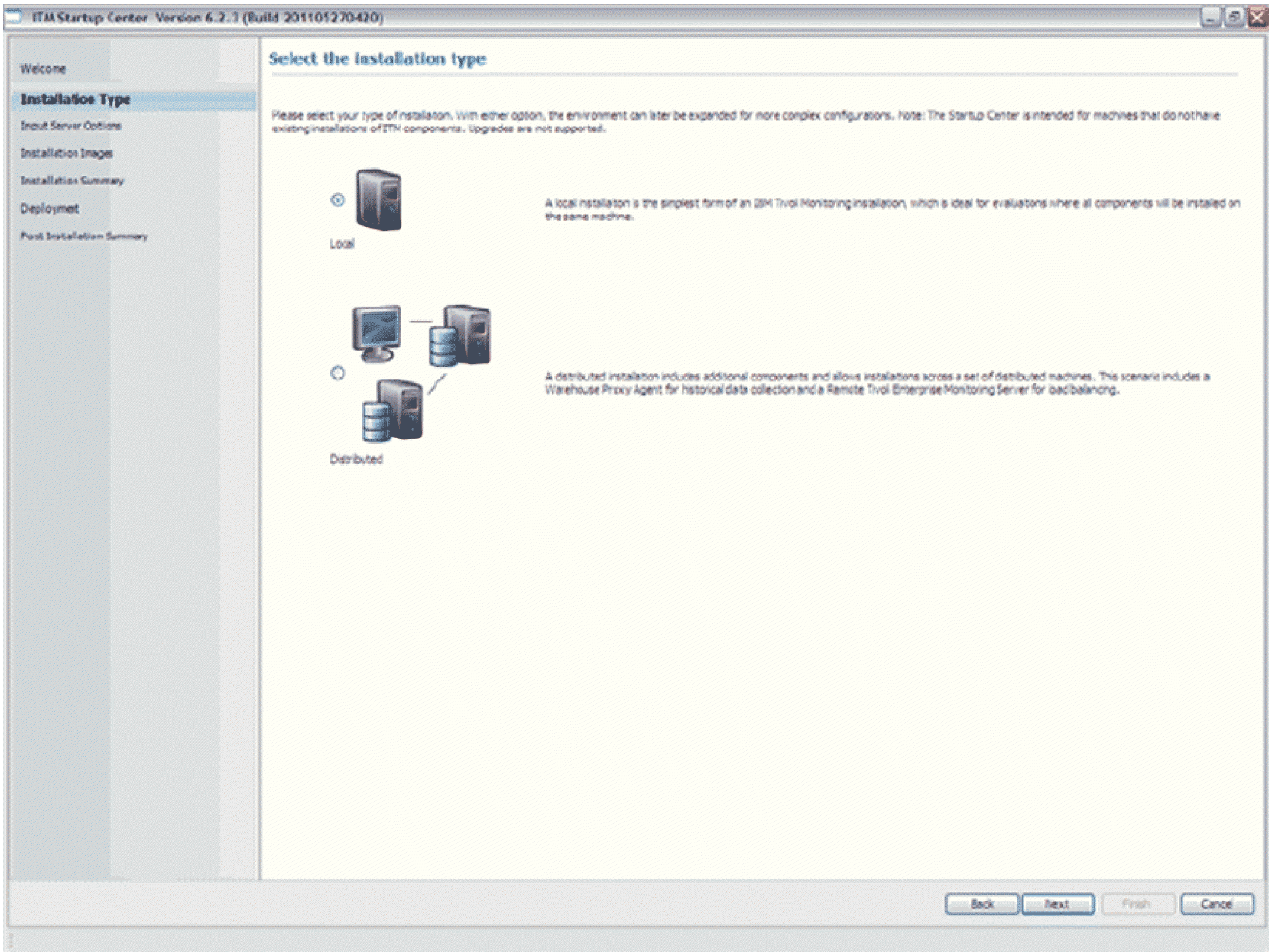
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON, OR USE THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA AND DOCUMENTATION TO IBM. IF THE PROGRAM WAS DOWNLOADED, DESTROY ALL COPIES OF THE PROGRAM.

Figure 13. Tivoli Monitoring Startup Center Software License Agreement window

4. Select the **Local** radio button and click **Next**.



Please select your type of installation. With either option, the environment can later be expanded for more complex configurations. Note: The Startup Center is intended for machines that do not have existing installations of ITM components. Upgrades are not supported.

A local installation is the simplest form of an IBM Tivoli Monitoring installation, which is ideal for evaluations where all components will be installed on the same machine.

A distributed installation includes additional components and allows installations across a set of distributed machines. This scenario includes a Warehouse Proxy Agent for historical data collection and a Remote Tivoli Enterprise Monitoring Server for load balancing.

Figure 14. Tivoli Monitoring Startup Center Installation Type window: Select the installation type

5. Enter the installation options for the components in the topology diagram and click **Next**.

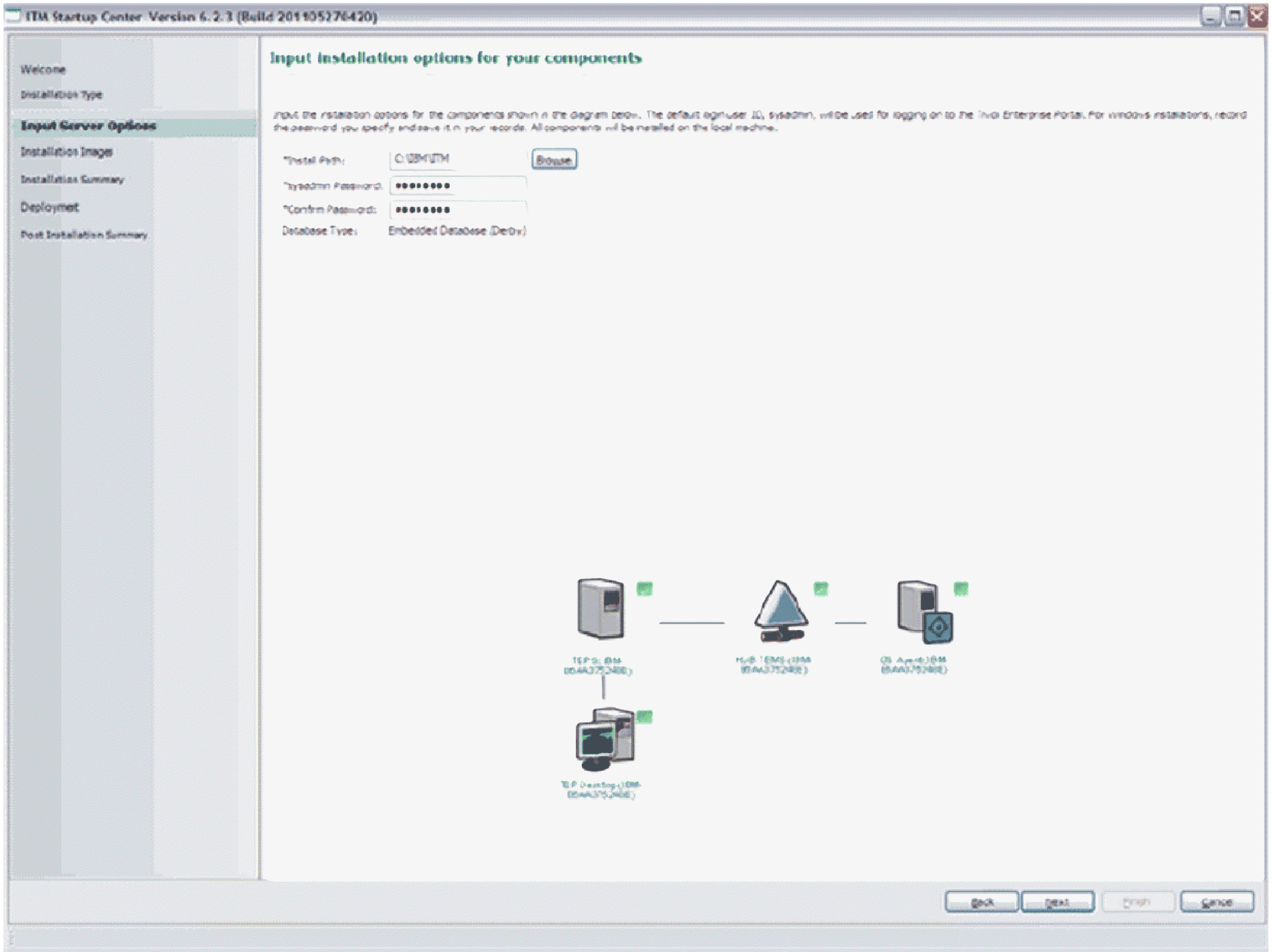
   Notes:

   a. Running multiple installations on a Windows machine specifying different directories is not supported.

   b. On Linux systems, you are not required to enter a password for the embedded Derby database.

   In the event of a failed installation, the Startup Center log files can help to diagnose the problem. The log files can be found in the following locations:

   - On Windows systems: `%USERPROFILE%\.STARTUP\logs\`
   - On Linux systems: `${HOME}/.STARTUP/logs`



ITM Startup Center Version 6.2.3 (Build 201105276420)

Input installation options for your components

Input the installation options for the components shown in the diagram below. The default login user ID, sysadmin, will be used for logging on to the Tivoli Enterprise Portal. For Windows installations, record the password you specify and save it in your records. All components will be installed on the local machine.

*Install Path: C:\IBM\ITM

*sysadmin Password: •••••••••

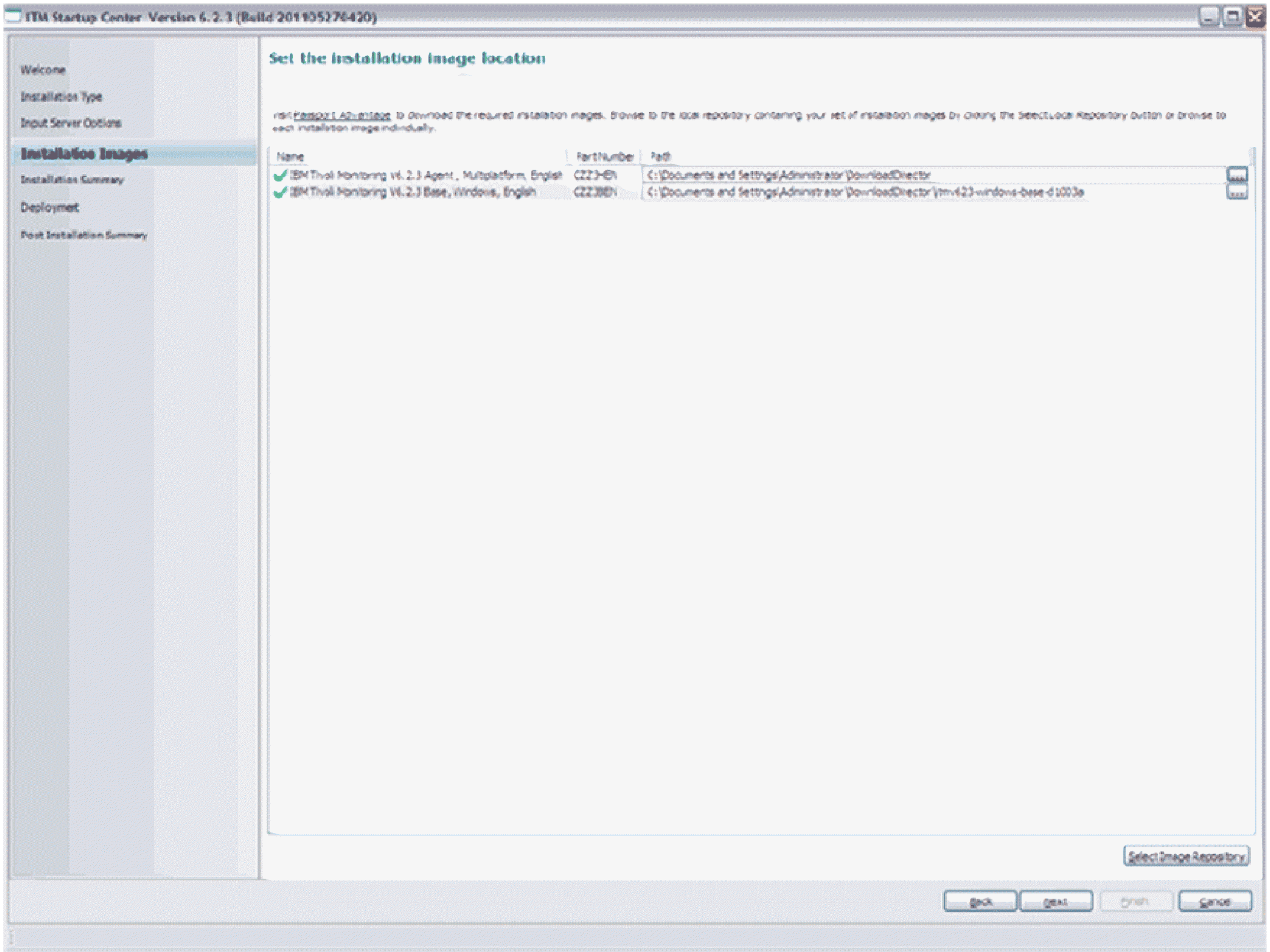
*Confirm Password: •••••••••

Database Type: Embedded Database (Derby)

Figure 15. Tivoli Monitoring Startup Center Input Installation Options window: Input installation options for your components

6. You can place all of your installation images on one directory and click **Select Image Repository** to browse to the directory. You can also click the ellipsis button at the end of the Path field to browse to each installation image individually. The directory path containing installation images cannot contain commas or spaces. Click **Next** to continue.

   Note: The installation images must be extracted before you set the image location.



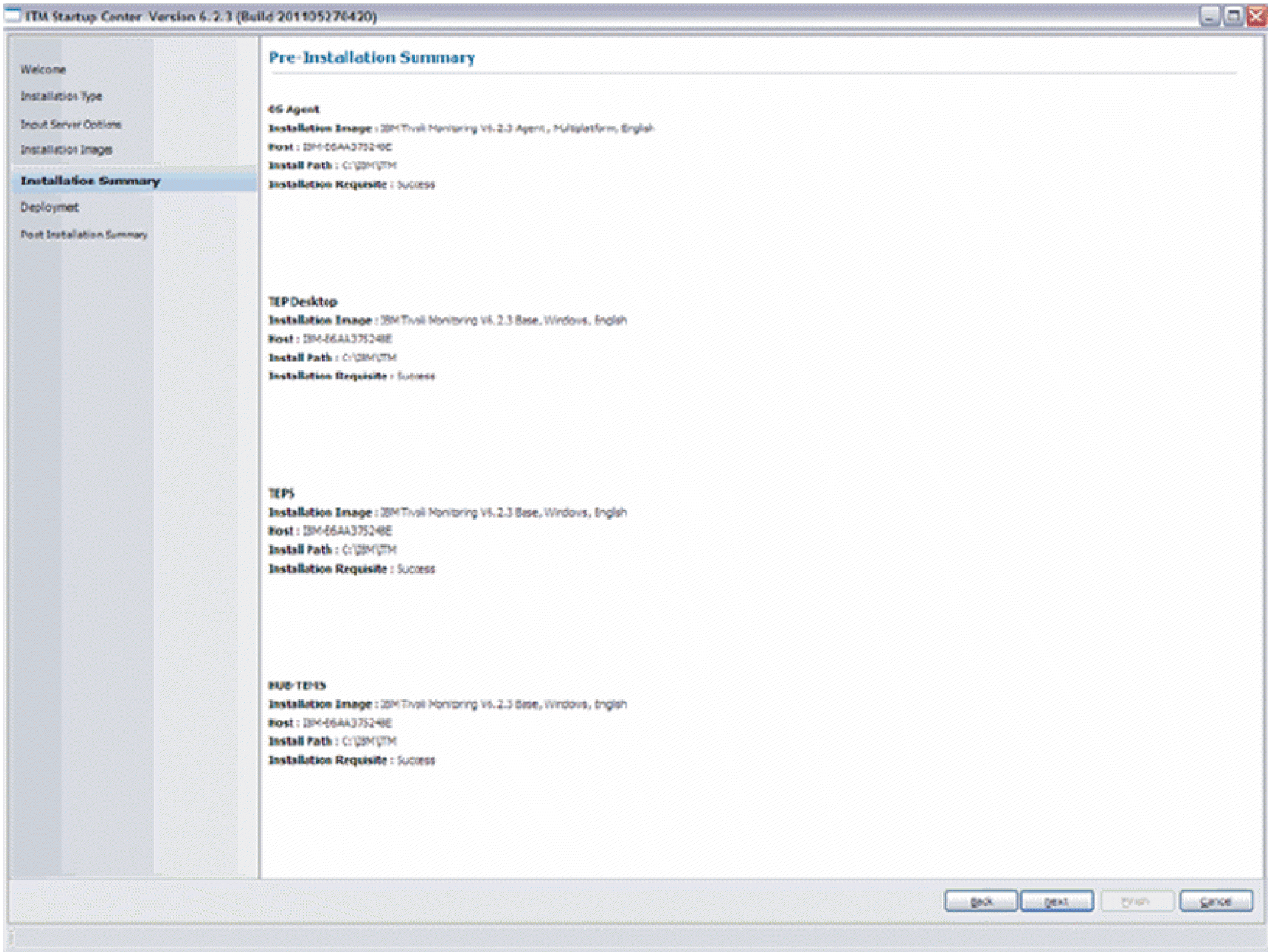
Set the installation image location

Visit Passport Advantage to download the required installation images. Browse to the local repository containing your set of installation images by clicking the Select Local Repository button or browse to each installation image individually.

Name	Part Number	Path
✓ IBM Tivoli Monitoring V6.2.3 Agent, Multiplatform, English	CZ23HEN	C:\Documents and Settings\Administrator\DownloadDirector
✓ IBM Tivoli Monitoring V6.2.3 Base, Windows, English	CZ23BEN	C:\Documents and Settings\Administrator\DownloadDirector\itm623-windows-base-d1003a

Figure 16. Tivoli Monitoring Startup Center Installation Image window: Set the installation image location

7. The Pre-Installation Summary screen is displayed. Any installation errors are displayed here. Review the summary information in this screen window and click **Back** to take corrective action, or click **Next** to continue.



Pre-Installation Summary

OS Agent
Installation Image : IBM Tivoli Monitoring V6.2.3 Agent, Multiplatform, English
Host : IBM-66AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

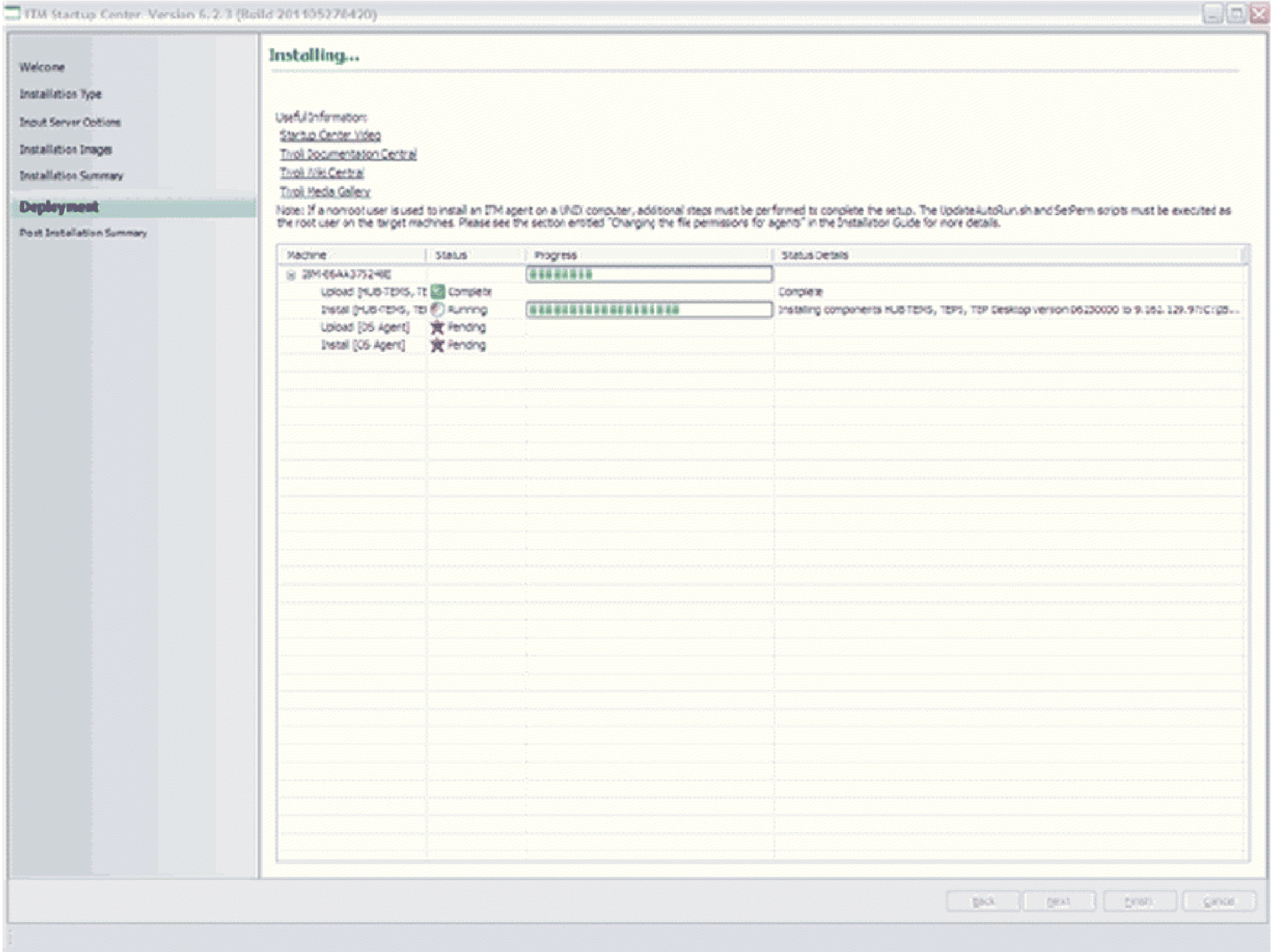
TEP Desktop
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-66AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

TEPS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-66AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

HUB-TEMS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-66AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

Figure 17. Tivoli Monitoring Startup Center Installation Summary window

8. The Deployment window shows the deployment status of each component and a progress indicator.



ITM Startup Center Version 6.2.3 (Build 201105276420)

Installing...

Useful Information:

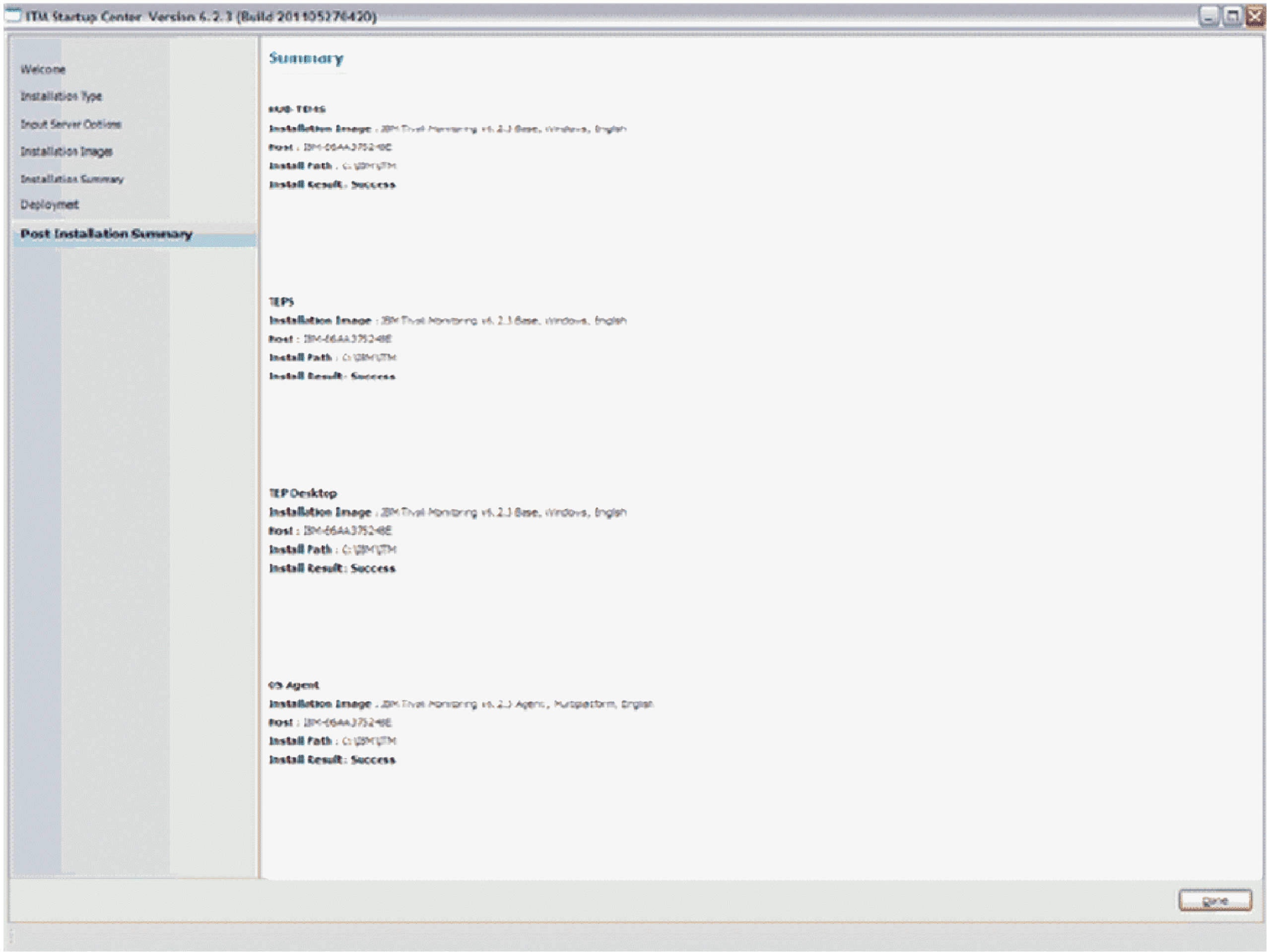
Startup Center Video
Tivoli Documentation Central
Tivoli Wiki Central
Tivoli Media Gallery

Note: If a nonroot user is used to install an ITM agent on a UNIX computer, additional steps must be performed to complete the setup. The UpdateAutoRun.sh and SetPerm scripts must be executed as the root user on the target machines. Please see the section entitled "Changing the file permissions for agents" in the Installation Guide for more details.

Machine	Status	Progress	Status Details
⊟ IBM-E6AA375248E			
Upload [HUB-TEMS, TE	✓ Complete		Complete
Install [HUB-TEMS, TEI	Running		Installing components HUB TEMS, TEPS, TEP Desktop version 06230000 to 9.162.129.97:C:\IB...
Upload [OS Agent]	Pending		
Install [OS Agent]	Pending		

Figure 18. Tivoli Monitoring Startup Center Deployment window

Any installation errors are displayed in the Postinstallation Summary window. Click **Next** when the deployment of each component has completed.



Summary

HUB-TEMS
Installation Image : IBM Tivoli Monitoring v6.2.3 Base, Windows, English
Host : IBM-66AA3752 48E
Install Path : C:\IBM\ITM
Install Result: Success

TEPS
Installation Image : IBM Tivoli Monitoring v6.2.3 Base, Windows, English
Host : IBM-66AA3752 48E
Install Path : C:\IBM\ITM
Install Result: Success

TEP Desktop
Installation Image : IBM Tivoli Monitoring v6.2.3 Base, Windows, English
Host : IBM-66AA3752 48E
Install Path : C:\IBM\ITM
Install Result: Success

OS Agent
Installation Image : IBM Tivoli Monitoring v6.2.3 Agent , Multiplatform, English
Host : IBM-66AA3752 48E
Install Path : C:\IBM\ITM
Install Result: Success

Figure 19. Tivoli Monitoring Startup Center Postinstallation Summary window

Distributed installation

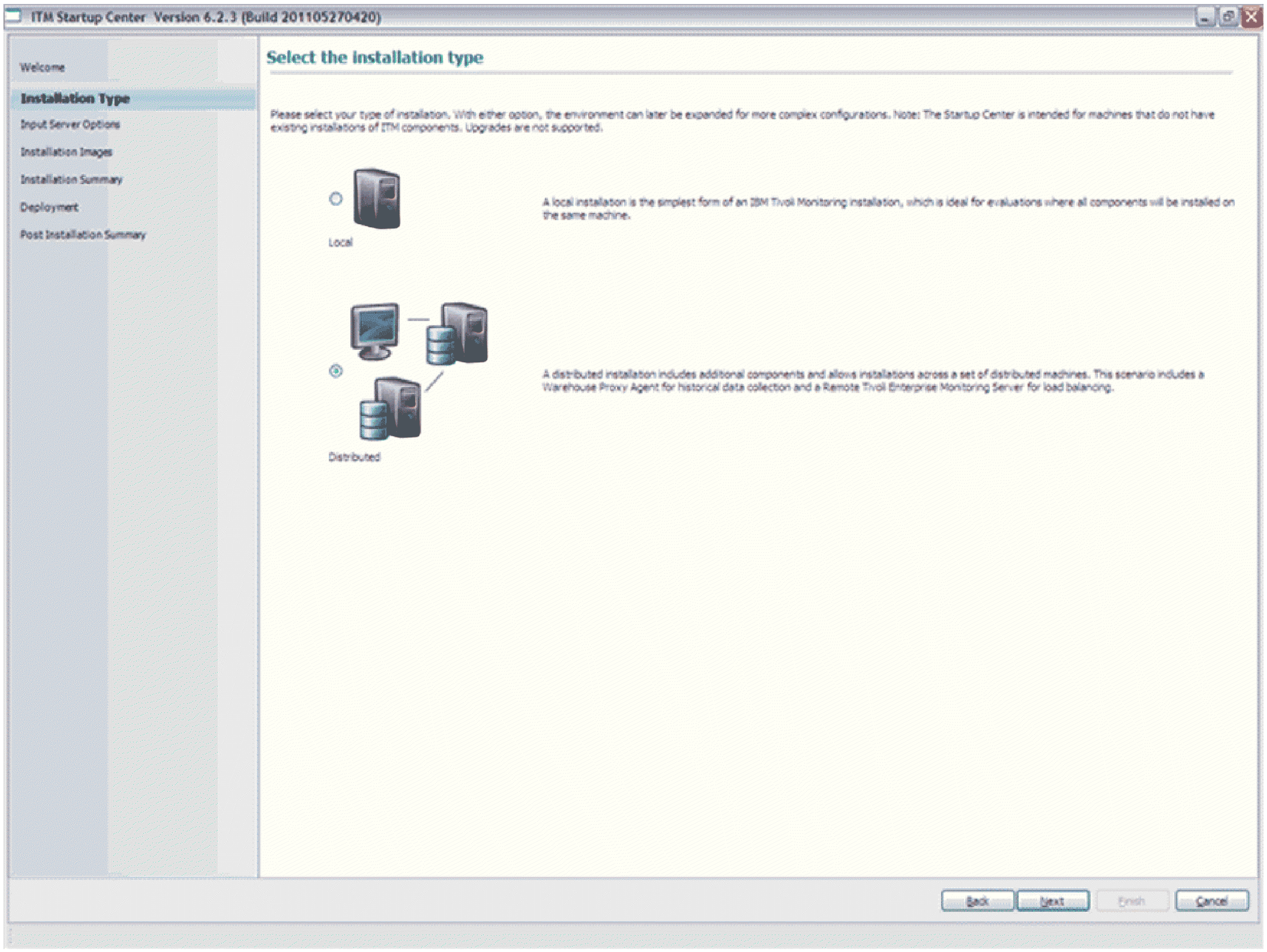
A predefined set of components are installed on a set of distributed machines. A distributed installation is recommended for medium and large environments that are monitoring more than 500 systems. A distributed installation offers you the most expansion possibilities as your environment grows. The distributed installation is the ideal configuration if you are looking to set up a robust foundation of IBM Tivoli Monitoring. You can install the following components with a distributed installation:

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal browser client
- Operating system agent
- Warehouse Proxy Agent
- Remote Tivoli Enterprise Monitoring Server (optional component)
- Tivoli Enterprise Portal desktop client (optional component)

Note: The TEPS and WAREHOUS databases are created automatically by the Startup Center tool. The `itmuser` account should have a profile that sets up the DB2 environment correctly before running the Startup Center tool.

Complete the following steps to perform a distributed installation by using the Tivoli Monitoring Startup Center:

1. Launch the Startup Center.
2. Click **Next** on the Welcome screen.
3. Click **Accept** to accept the license agreement.
4. Select the **Distributed** radio button and click **Next**.



ITM Startup Center Version 6.2.3 (Build 201105270420)

Select the installation type

Please select your type of installation. With either option, the environment can later be expanded for more complex configurations. Note: The Startup Center is intended for machines that do not have existing installations of ITM components. Upgrades are not supported.

○ Local

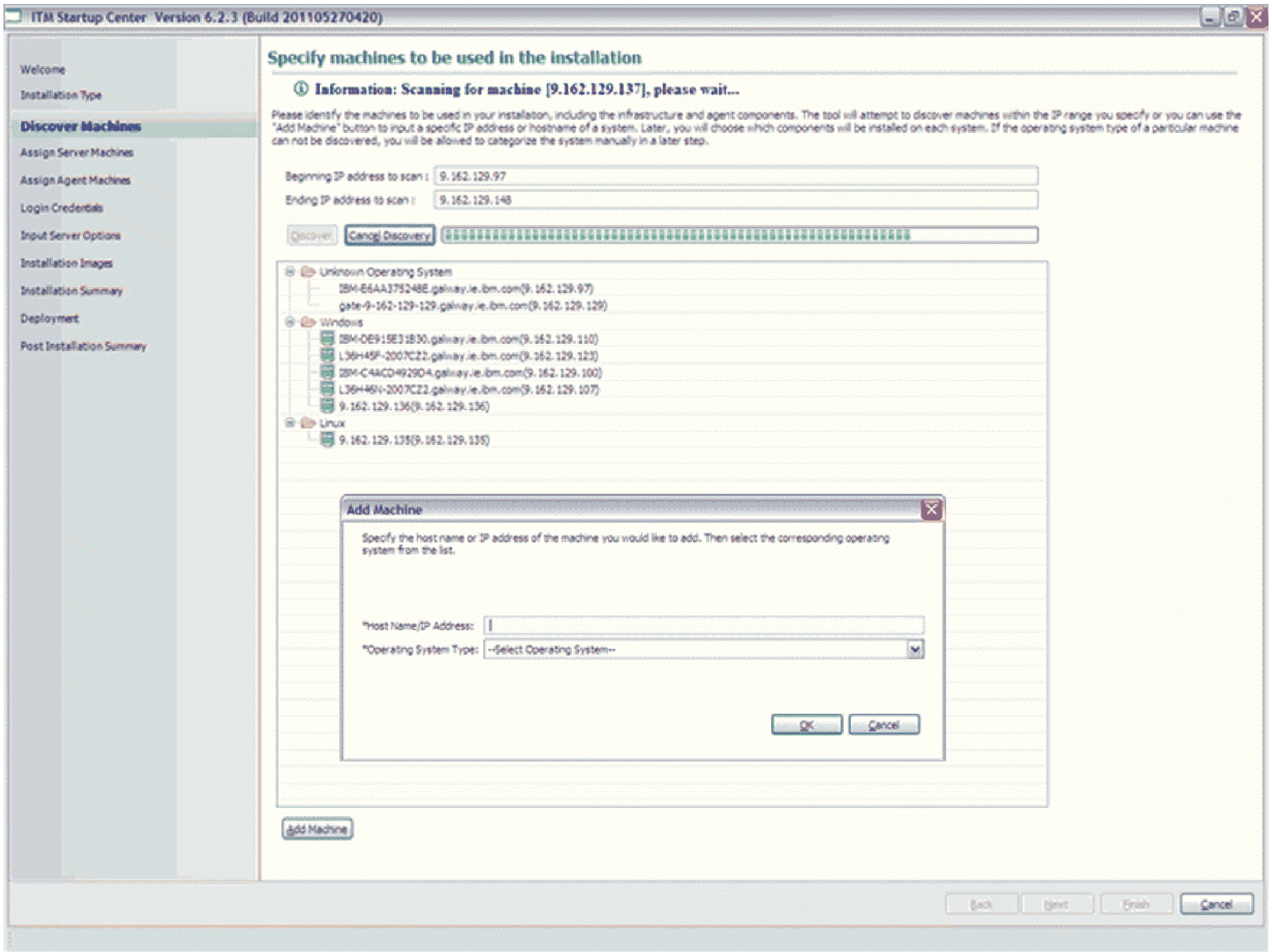
A local installation is the simplest form of an IBM Tivoli Monitoring installation, which is ideal for evaluations where all components will be installed on the same machine.

● Distributed

A distributed installation includes additional components and allows installations across a set of distributed machines. This scenario includes a Warehouse Proxy Agent for historical data collection and a Remote Tivoli Enterprise Monitoring Server for load balancing.

Figure 20. Tivoli Monitoring Startup Center Installation Type window

5. Your environment can now be scanned to discover machines within the IP range that you specify. You can also click **Add Machine** to specify a system host name or IP address. If you specify a machine by using the **Add Machine** option you must ensure that the machine is online. Click **Next** to continue.



Specify machines to be used in the installation

Information: Scanning for machine [9.162.129.137], please wait...

Please identify the machines to be used in your installation, including the infrastructure and agent components. The tool will attempt to discover machines within the IP range you specify or you can use the "Add Machine" button to input a specific IP address or hostname of a system. Later, you will choose which components will be installed on each system. If the operating system type of a particular machine can not be discovered, you will be allowed to categorize the system manually in a later step.

Beginning IP address to scan : 9.162.129.97

Ending IP address to scan : 9.162.129.148

- Unknown Operating System
  - IBM-E6AA375248E.galway.ie.ibm.com(9.162.129.97)
  - gate-9-162-129-129.galway.ie.ibm.com(9.162.129.129)
- Windows
  - IBM-DE915E31B30.galway.ie.ibm.com(9.162.129.110)
  - L36H45F-2007CZ2.galway.ie.ibm.com(9.162.129.123)
  - IBM-C4ACD4929D4.galway.ie.ibm.com(9.162.129.100)
  - L36H46N-2007CZ2.galway.ie.ibm.com(9.162.129.107)
  - 9.162.129.136(9.162.129.136)
- Linux
  - 9.162.129.135(9.162.129.135)

Add Machine

Specify the host name or IP address of the machine you would like to add. Then select the corresponding operating system from the list.

*Host Name/IP Address:

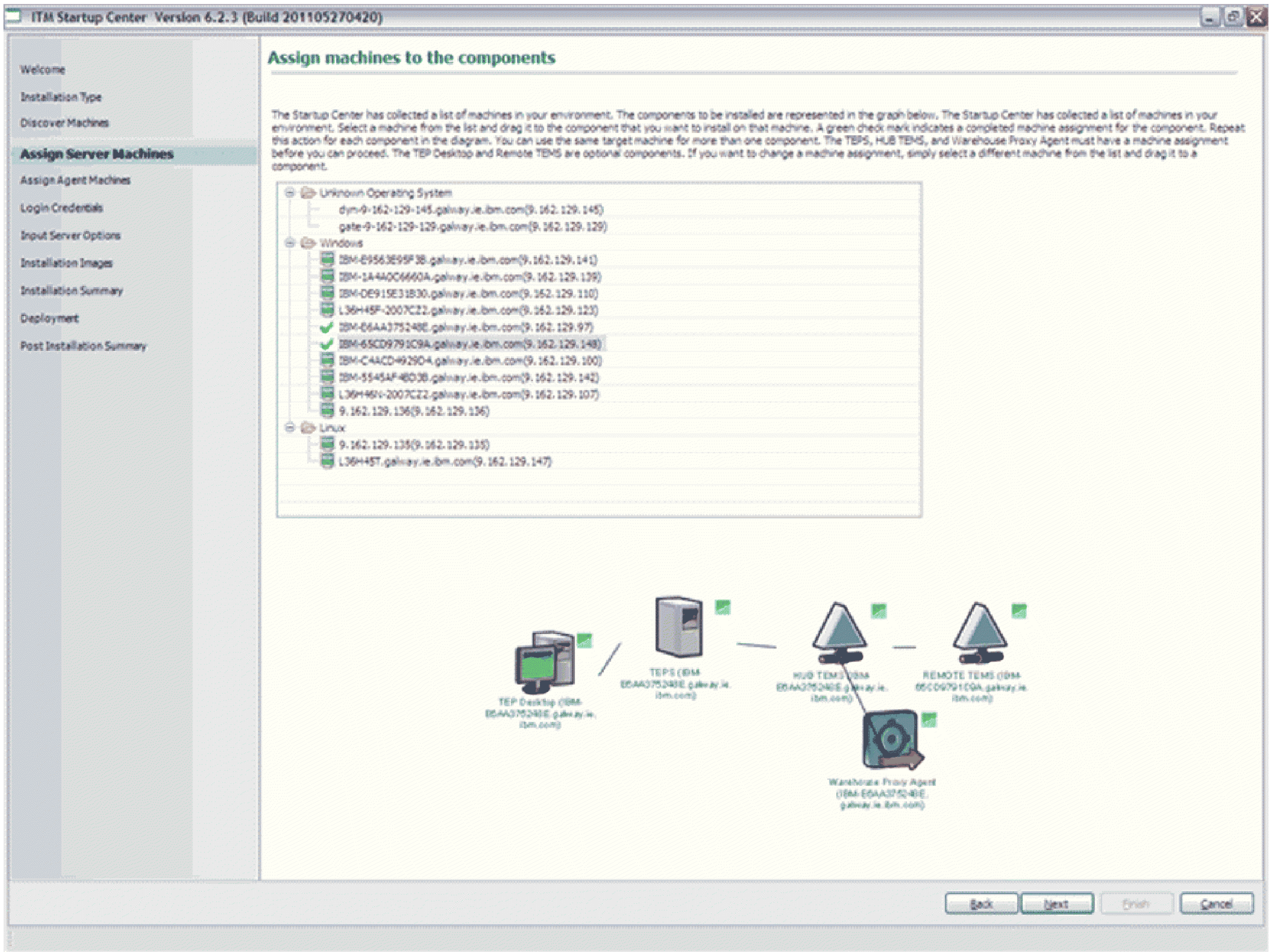
*Operating System Type: --Select Operating System--

Figure 21. Tivoli Monitoring Startup Center Discover Machines window

Note: For some systems, the Startup Center might not identify the type of operating system. These systems are listed under the **Unknown Operating System** category. For more information, see the *IBM Tivoli Monitoring: Troubleshooting Guide*.

6. The Startup Center has discovered the machines in your environment. The components to be installed are represented in a graph. Now you must specify the components to install on a target machine. Select a machine from the list and drag it to the component that you want to install on that machine. You can also press the **Enter** key and select the component from the pop-up dialog box. The Tivoli Enterprise Portal Server and the Warehouse Proxy Agent are assigned to the same machine. If you want to change a machine assignment, select a different machine from the list and drag it to a component.

   A green check mark indicates a completed machine assignment for the component. Repeat this action for each component in the diagram. You can use the same target machine for more than one component. The Tivoli Enterprise Portal Server, hub Tivoli Enterprise Monitoring Server, and Warehouse Proxy Agent must have a machine assignment before you can proceed. The Tivoli Enterprise Portal Desktop and remote Tivoli Enterprise Monitoring Server are optional components. Click **Next** to continue.


ITM Startup Center Version 6.2.3 (Build 201105270420)
Welcome
Installation Type
Discover Machines
Assign Server Machines
Assign Agent Machines
Login Credentials
Input Server Options
Installation Images
Installation Summary
Deployment
Post Installation Summary
TEP Desktop (IBM-E6AA375248E.galway.ie.ibm.com)
TEPS (IBM-E6AA375248E.galway.ie.ibm.com)
HUB TEMS (IBM-E6AA375248E.galway.ie.ibm.com)
REMOTE TEMS (IBM-65CD9791C9A.galway.ie.ibm.com)
Warehouse Proxy Agent (IBM-E6AA375248E.galway.ie.ibm.com)
Back
Next
Finish
Cancel

Assign machines to the components

The Startup Center has collected a list of machines in your environment. The components to be installed are represented in the graph below. The Startup Center has collected a list of machines in your environment. Select a machine from the list and drag it to the component that you want to install on that machine. A green check mark indicates a completed machine assignment for the component. Repeat this action for each component in the diagram. You can use the same target machine for more than one component. The TEPS, HUB TEMS, and Warehouse Proxy Agent must have a machine assignment before you can proceed. The TEP Desktop and Remote TEMS are optional components. If you want to change a machine assignment, simply select a different machine from the list and drag it to a component.

- Unknown Operating System
  - dyn-9-162-129-145.galway.ie.ibm.com(9.162.129.145)
  - gate-9-162-129-129.galway.ie.ibm.com(9.162.129.129)
- Windows
  - IBM-E9563E95F3B.galway.ie.ibm.com(9.162.129.141)
  - IBM-1A4A0C6660A.galway.ie.ibm.com(9.162.129.139)
  - IBM-DE915E31B30.galway.ie.ibm.com(9.162.129.110)
  - L36H45F-2007CZ2.galway.ie.ibm.com(9.162.129.123)
  - ✓ IBM-E6AA375248E.galway.ie.ibm.com(9.162.129.97)
  - ✓ IBM-65CD9791C9A.galway.ie.ibm.com(9.162.129.148)
  - IBM-C4ACD4929D4.galway.ie.ibm.com(9.162.129.100)
  - IBM-5545AF4BD3B.galway.ie.ibm.com(9.162.129.142)
  - L36H46N-2007CZ2.galway.ie.ibm.com(9.162.129.107)
  - 9.162.129.136(9.162.129.136)
- Linux
  - 9.162.129.135(9.162.129.135)
  - L36H45T.galway.ie.ibm.com(9.162.129.147)

Figure 22. Tivoli Monitoring Startup Center Assign Server Machines window

Notes:

a. If you assign a component to an unknown operating system, you are prompted to specify your operating system.

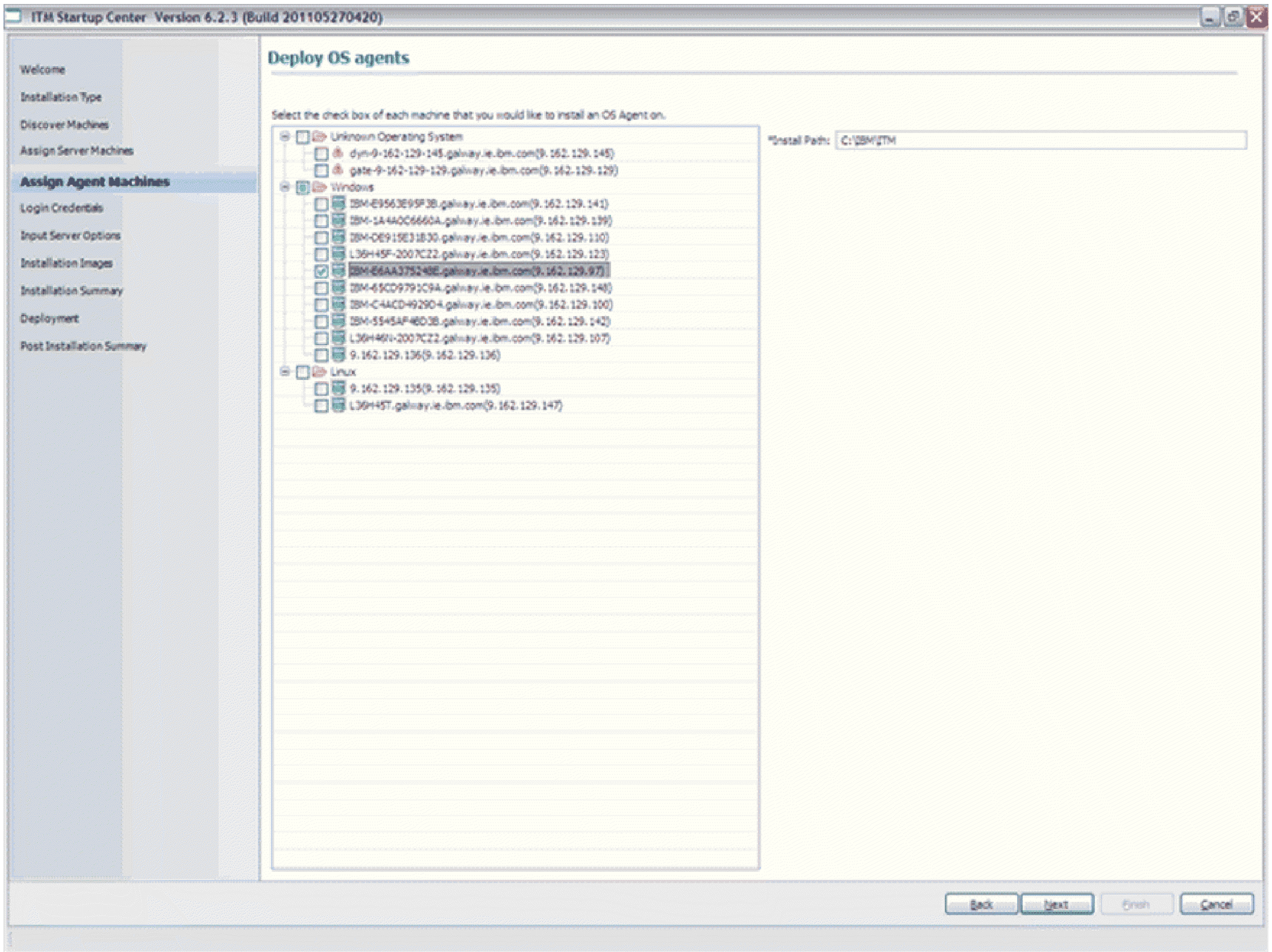
b. By default, when you assign a machine to the Tivoli Enterprise Portal Server, the Startup Center assigns the Warehouse Proxy Agent to the same machine. Likewise, when you assign a machine to the Warehouse Proxy Agent, the Startup Center also assigns the Tivoli Enterprise Portal Server.

c. You cannot assign a hub monitoring server and remote monitoring server on the same system.

d. You can change a machine assignment but you cannot remove an assignment. If you want to change any of the machine assignments, select another machine from the list and drag it to the component.

7. The Deploy OS Agents screen is displayed. Select the check box of each machine that you want to install an OS Agent on. Enter a path for the target installation directory or accept the default. You must select at least one machine before continuing.

   Note: The installation of OS Agents is not supported on Windows Itanium systems.
   Click **Next** to continue.



ITM Startup Center Version 6.2.3 (Build 201105270420)

Deploy OS agents

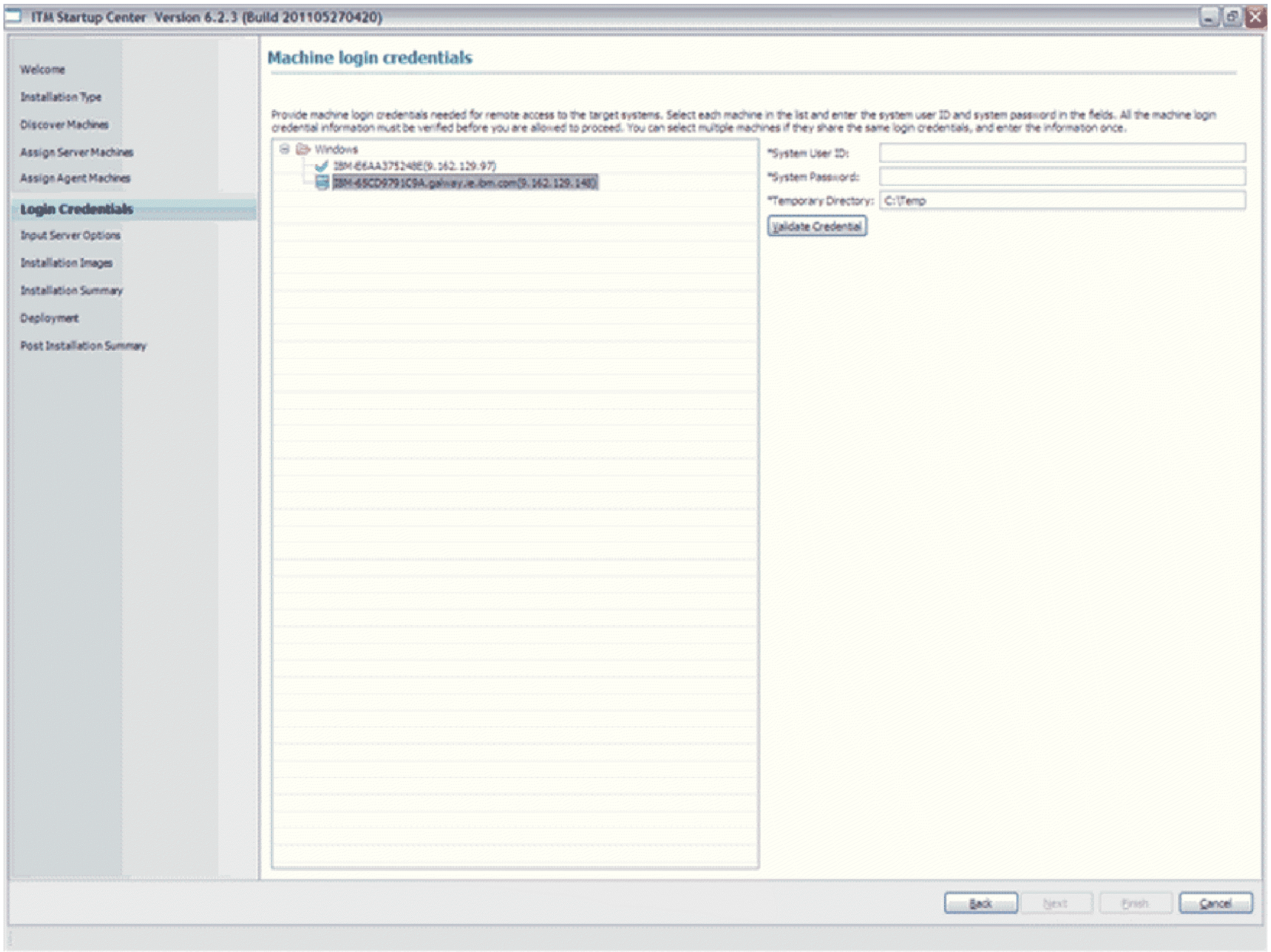
Select the check box of each machine that you would like to install an OS Agent on.

- ☐ Unknown Operating System
  - ☐ dyn-9-162-129-145.galway.ie.ibm.com(9.162.129.145)
  - ☐ gate-9-162-129-129.galway.ie.ibm.com(9.162.129.129)
- ☐ Windows
  - ☐ IBM-E9563E95F3B.galway.ie.ibm.com(9.162.129.141)
  - ☐ IBM-1A4A0C6660A.galway.ie.ibm.com(9.162.129.139)
  - ☐ IBM-OE915E31B30.galway.ie.ibm.com(9.162.129.110)
  - ☐ L36H45F-2007CZ2.galway.ie.ibm.com(9.162.129.123)
  - ☑ IBM-E6AA375248E.galway.ie.ibm.com(9.162.129.97)
  - ☐ IBM-65CD9791C9A.galway.ie.ibm.com(9.162.129.148)
  - ☐ IBM-C4ACD49290 4.galway.ie.ibm.com(9.162.129.100)
  - ☐ IBM-5545AF4BD3B.galway.ie.ibm.com(9.162.129.142)
  - ☐ L36H46N-2007CZ2.galway.ie.ibm.com(9.162.129.107)
  - ☐ 9.162.129.136(9.162.129.136)
- ☐ Linux
  - ☐ 9.162.129.135(9.162.129.135)
  - ☐ L36H45T.galway.ie.ibm.com(9.162.129.147)

*Install Path: C:\IBM\ITM

Figure 23. Tivoli Monitoring Startup Center Deploy OS Agents window

8. Now you must provide the machine login credentials that are required to remotely access target systems. Select each machine in the list and enter the System user ID and System password in the fields provided. For information on non-root user credentials, see “Non-root user support” on page 110. You can select multiple machines if they share the same login credentials and enter the information once. All credentials must be validated before you can click **Next**.



Machine login credentials

Provide machine login credentials needed for remote access to the target systems. Select each machine in the list and enter the system user ID and system password in the fields. All the machine login credential information must be verified before you are allowed to proceed. You can select multiple machines if they share the same login credentials, and enter the information once.

- Windows
  - ✓ IBM-E6AA375248E(9.162.129.97)
  - IBM-65CD9791C9A.galway.ie.ibm.com(9.162.129.148)

*System User ID:

*System Password:

*Temporary Directory: C:\Temp

Validate Credential

Figure 24. Tivoli Monitoring Startup Center Login Credentials window

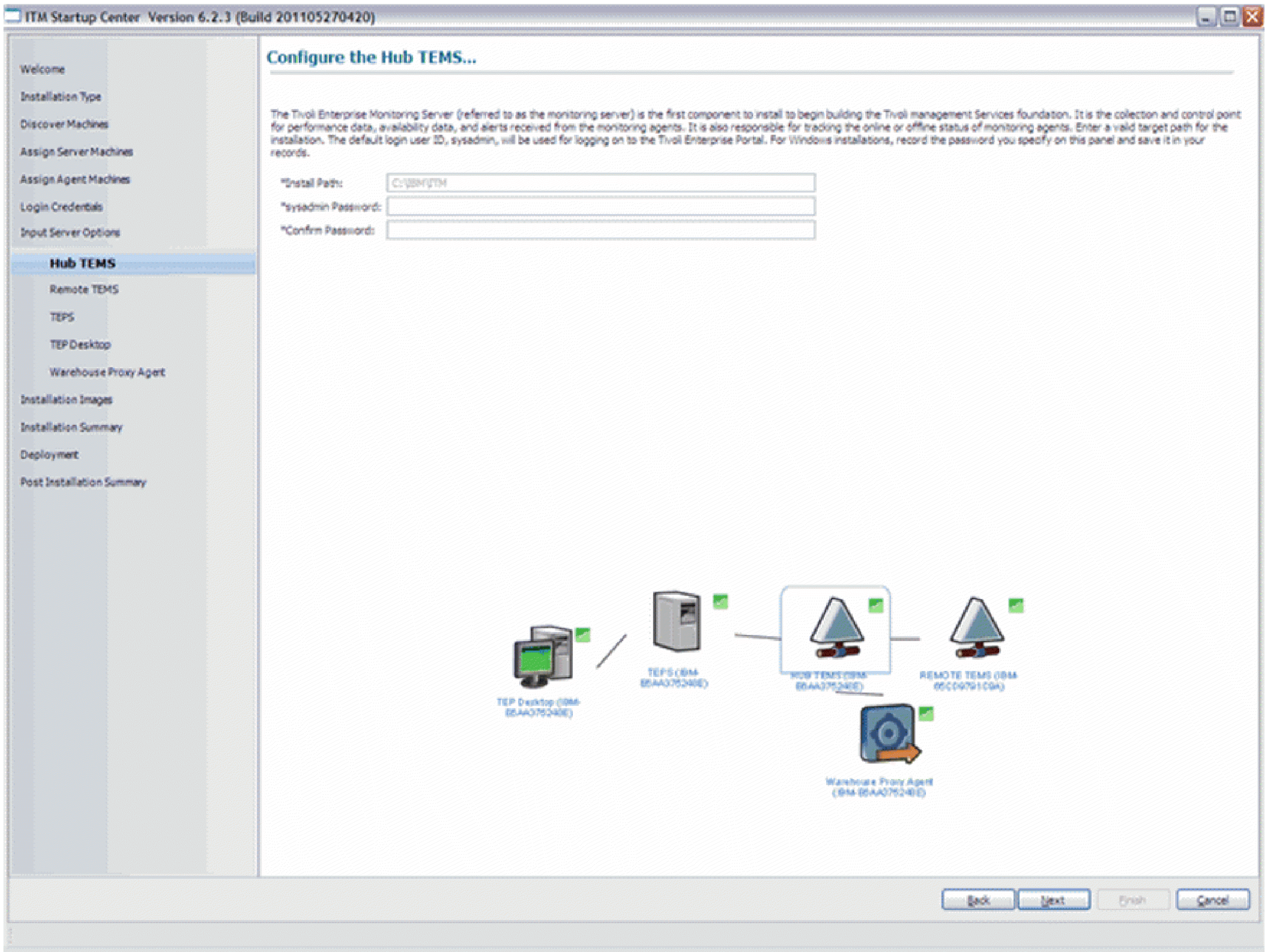
Notes:

a. If you encounter a problem at this step, it is probably RXA related. For information about RXA, see “Remote Execution and Access” on page 112.

b. The **Temporary Directory** must have sufficient space to accommodate the transferred installation image.

c. In the Turkish locale, a user ID containing a lowercase "i" causes machine login credentials to fail. You can change to a locale other than Turkish, or else use only capital 'I' letters in user IDs, for example *AdmInIstrator* instead of *Administrator*.

9. The Input Server Options screen is displayed. The following components must be configured:
   - Tivoli Enterprise Monitoring Server: Enter the installation path for the monitoring server and click **Next**.



Configure the Hub TEMS...

The Tivoli Enterprise Monitoring Server (referred to as the monitoring server) is the first component to install to begin building the Tivoli management Services foundation. It is the collection and control point for performance data, availability data, and alerts received from the monitoring agents. It is also responsible for tracking the online or offline status of monitoring agents. Enter a valid target path for the installation. The default login user ID, sysadmin, will be used for logging on to the Tivoli Enterprise Portal. For Windows installations, record the password you specify on this panel and save it in your records.

*Install Path: C:\IBM\ITM

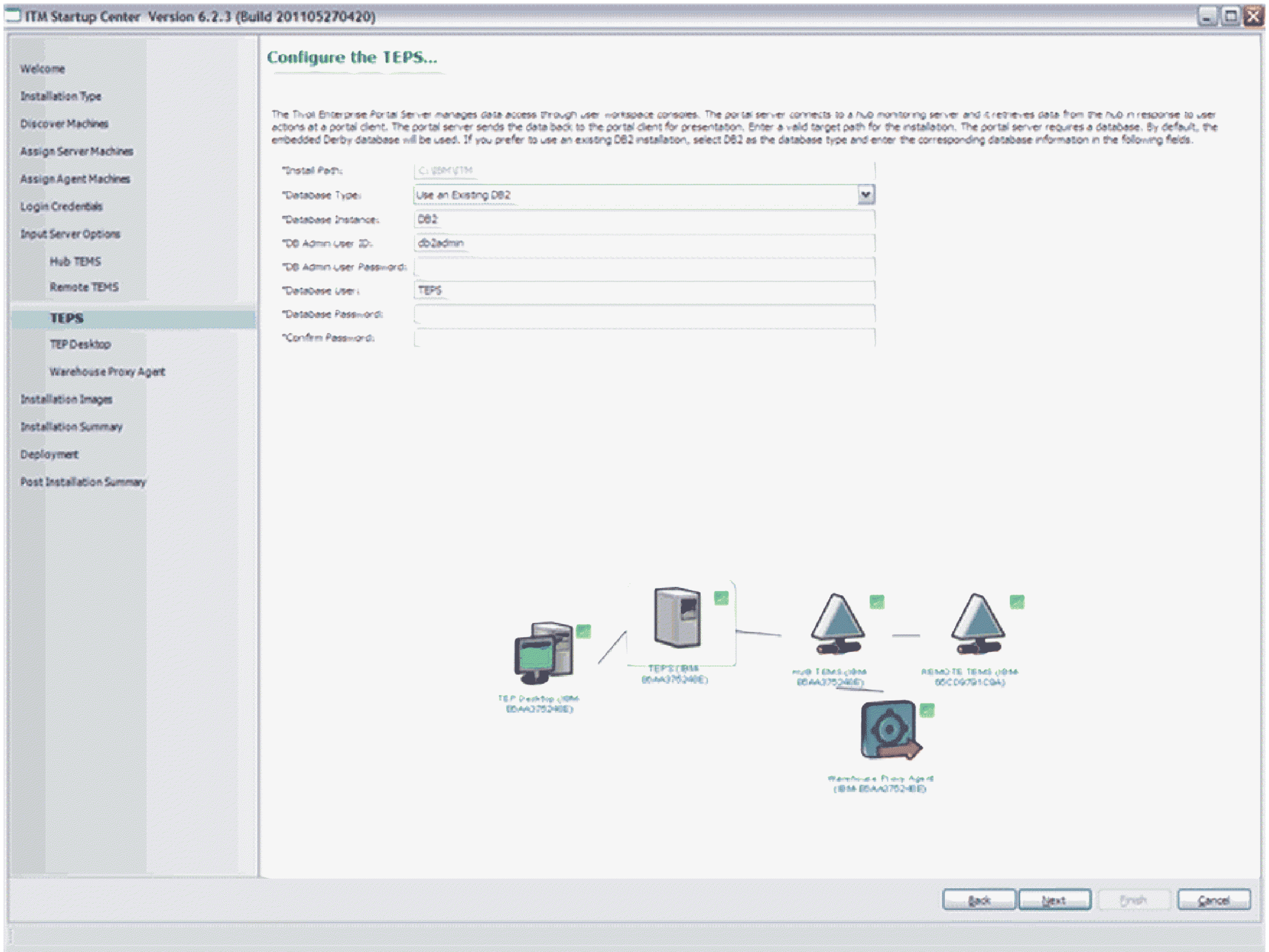
*sysadmin Password:

*Confirm Password:

Figure 25. Tivoli Monitoring Startup Center Configure the hub Tivoli Enterprise Monitoring Server window

   - Remote Tivoli Enterprise Monitoring Server: Enter the installation path for the remote monitoring server and click **Next**.

- Tivoli Enterprise Portal Server:
  a. Enter the installation path, password, and database type for the Tivoli Enterprise Portal Server. The database types that are currently supported are embedded Derby and DB2. If you select DB2 as your database type, you must enter the corresponding database information in the fields provided.
  b. Click **Next**. The database connection is automatically validated.



Configure the TEPS...

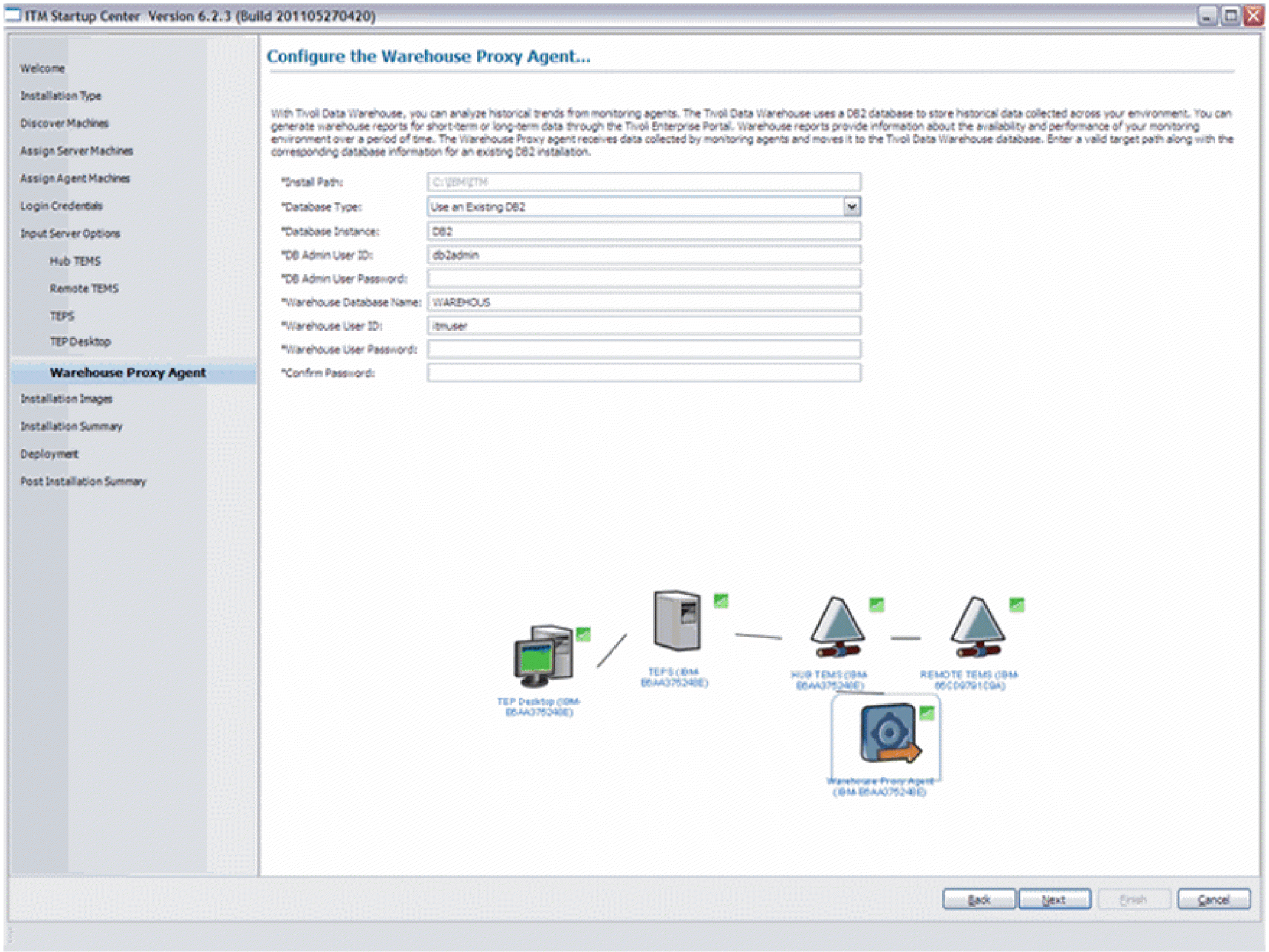
The Tivoli Enterprise Portal Server manages data access through user workspace consoles. The portal server connects to a hub monitoring server and it retrieves data from the hub in response to user actions at a portal client. The portal server sends the data back to the portal client for presentation. Enter a valid target path for the installation. The portal server requires a database. By default, the embedded Derby database will be used. If you prefer to use an existing DB2 installation, select DB2 as the database type and enter the corresponding database information in the following fields.

*Install Path: C:\IBM\ITM
*Database Type: Use an Existing DB2
*Database Instance: DB2
*DB Admin User ID: db2admin
*DB Admin User Password:
*Database User: TEPS
*Database Password:
*Confirm Password:

Figure 26. Tivoli Monitoring Startup Center Configure the Tivoli Enterprise Portal Server window

- Tivoli Enterprise Portal Server Desktop: Enter the installation path for the Tivoli Enterprise Portal Server Desktop and click **Next**.

- Warehouse Proxy Agent: Enter a valid installation path along with the corresponding database information and click **Next**.



Configure the Warehouse Proxy Agent...

With Tivoli Data Warehouse, you can analyze historical trends from monitoring agents. The Tivoli Data Warehouse uses a DB2 database to store historical data collected across your environment. You can generate warehouse reports for short-term or long-term data through the Tivoli Enterprise Portal. Warehouse reports provide information about the availability and performance of your monitoring environment over a period of time. The Warehouse Proxy agent receives data collected by monitoring agents and moves it to the Tivoli Data Warehouse database. Enter a valid target path along with the corresponding database information for an existing DB2 installation.

*Install Path: C:\IBM\ITM
*Database Type: Use an Existing DB2
*Database Instance: DB2
*DB Admin User ID: db2admin
*DB Admin User Password:
*Warehouse Database Name: WAREHOUS
*Warehouse User ID: itmuser
*Warehouse User Password:
*Confirm Password:

Figure 27. Tivoli Monitoring Startup Center Configure the Warehouse Proxy Agent

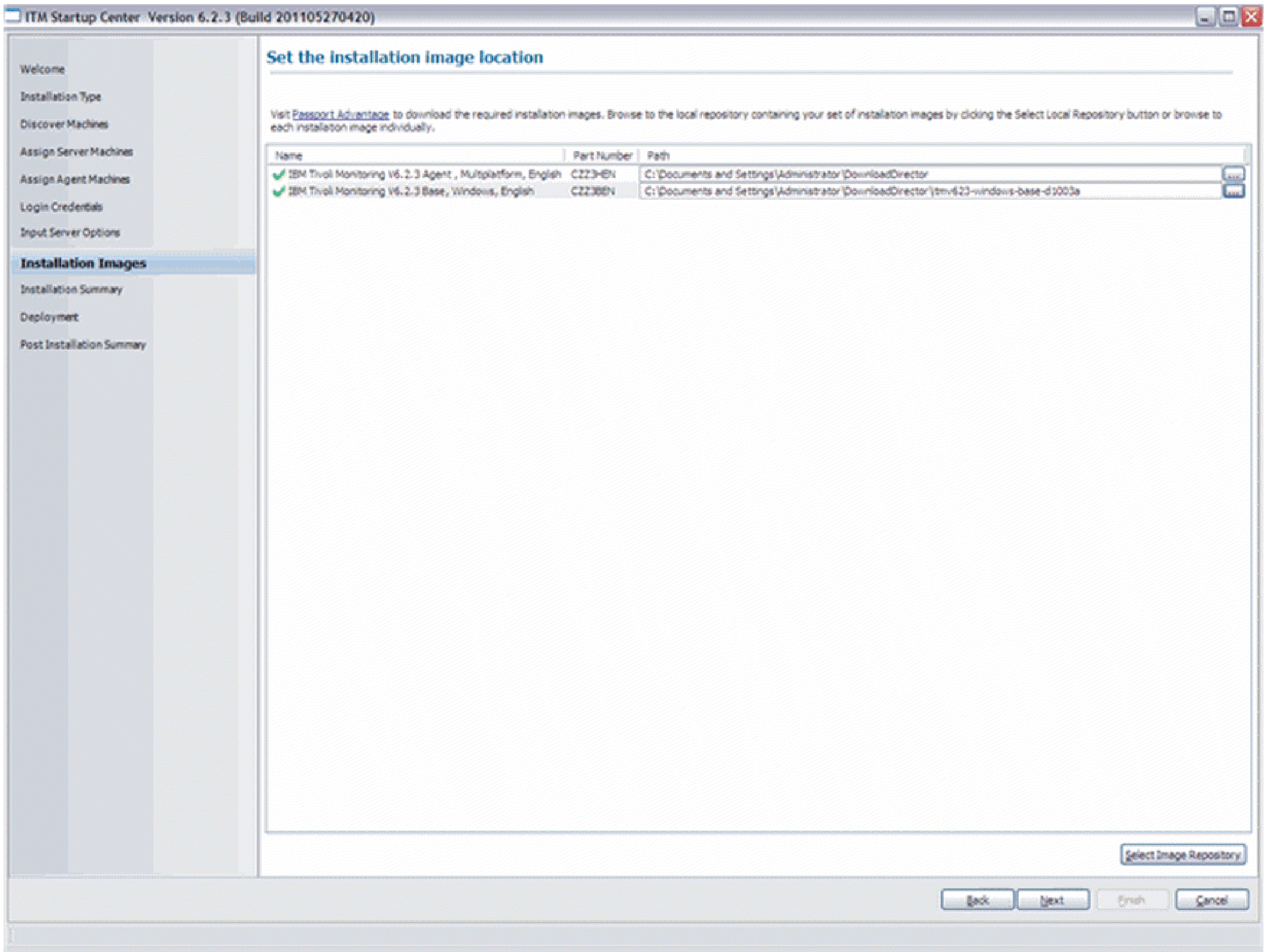
Note:

At this point, an error might be displayed regarding your connection settings. The RXA tool is used to establish a connection with a remote computer. You must ensure that your target system meets the requirements to establish remote execution and access. For more information see “Remote Execution and Access” on page 112.

If the Startup Center fails to create a warehouse database or database user, see the *IBM Tivoli Monitoring: Troubleshooting Guide* for more information.

10. You can place all of your installation images on one directory and click **Select Image Repository** to browse to the directory. You can also click the ellipsis button at the end of the Path field to browse to each installation image individually. The directory path containing installation images cannot contain commas or spaces. Click **Next** to continue.

    Note: The installation images must be extracted before you set the image location.



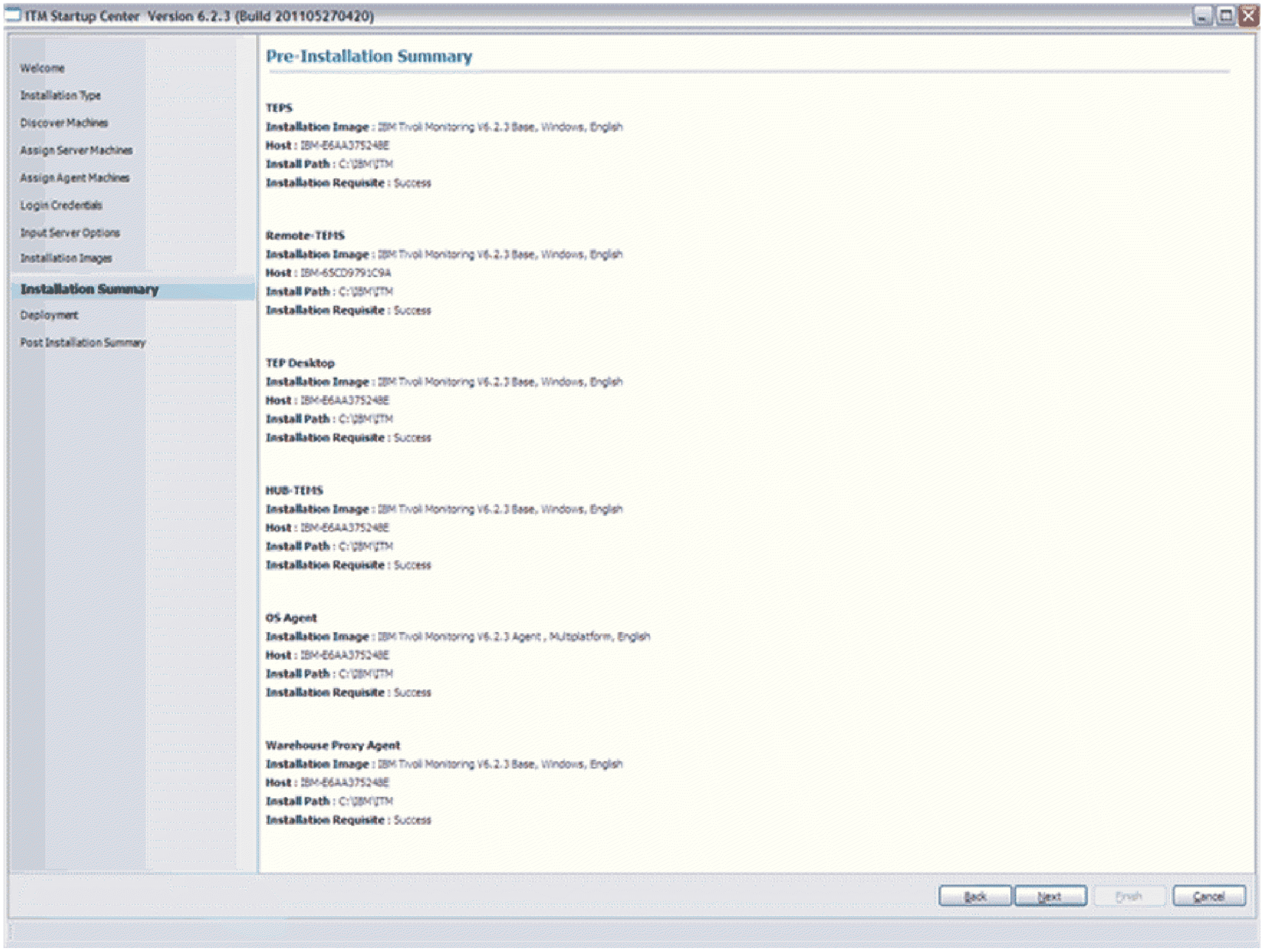
Set the installation image location

Visit Passport Advantage to download the required installation images. Browse to the local repository containing your set of installation images by clicking the Select Local Repository button or browse to each installation image individually.

Name	Part Number	Path
✓ IBM Tivoli Monitoring V6.2.3 Agent , Multiplatform, English	CZZ3HEN	C:\Documents and Settings\Administrator\DownloadDirector
✓ IBM Tivoli Monitoring V6.2.3 Base, Windows, English	CZZ3BEN	C:\Documents and Settings\Administrator\DownloadDirector\itmv623-windows-base-d1003a

Figure 28. Tivoli Monitoring Startup Center Installation Images window

11. The Pre-Installation Summary screen is displayed. Any installation errors are displayed here. Review the summary information on this screen and click **Back** to take corrective action, or click **Next** to continue.



Pre-Installation Summary

TEPS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

Remote-TEMS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-65CD9791C9A
Install Path : C:\IBM\ITM
Installation Requisite : Success

TEP Desktop
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

HUB-TEMS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

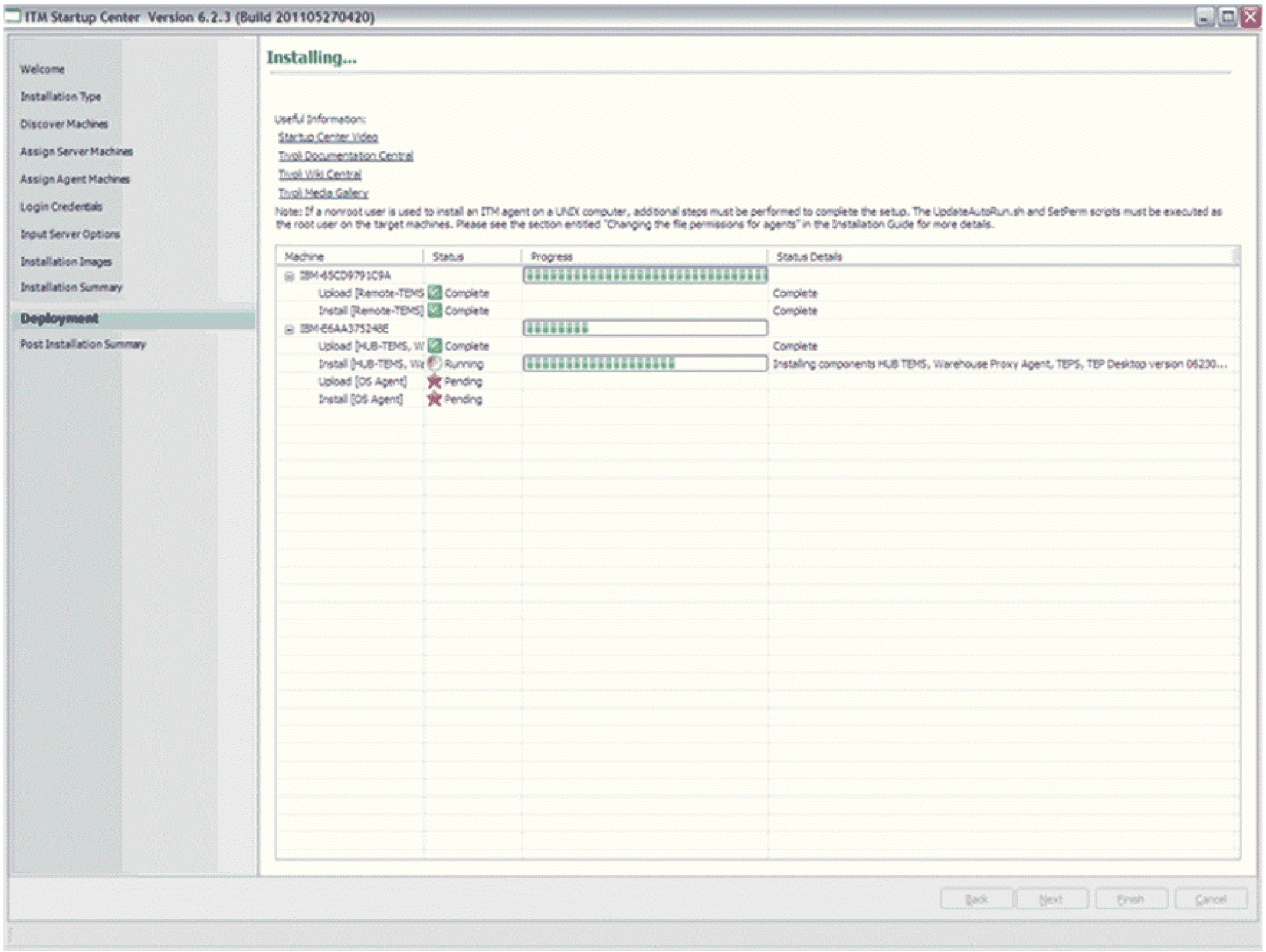
OS Agent
Installation Image : IBM Tivoli Monitoring V6.2.3 Agent , Multiplatform, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

Warehouse Proxy Agent
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Installation Requisite : Success

Figure 29. Tivoli Monitoring Startup Center Installation Summary window

Note: An error message is displayed on the Pre-Installation Summary screen if there is not enough space in the installation directory or the temp directory. Make sure that the directories you specified have enough free space.

12. The Deployment screen displays the deployment status of each component and a progress indicator.



Installing...

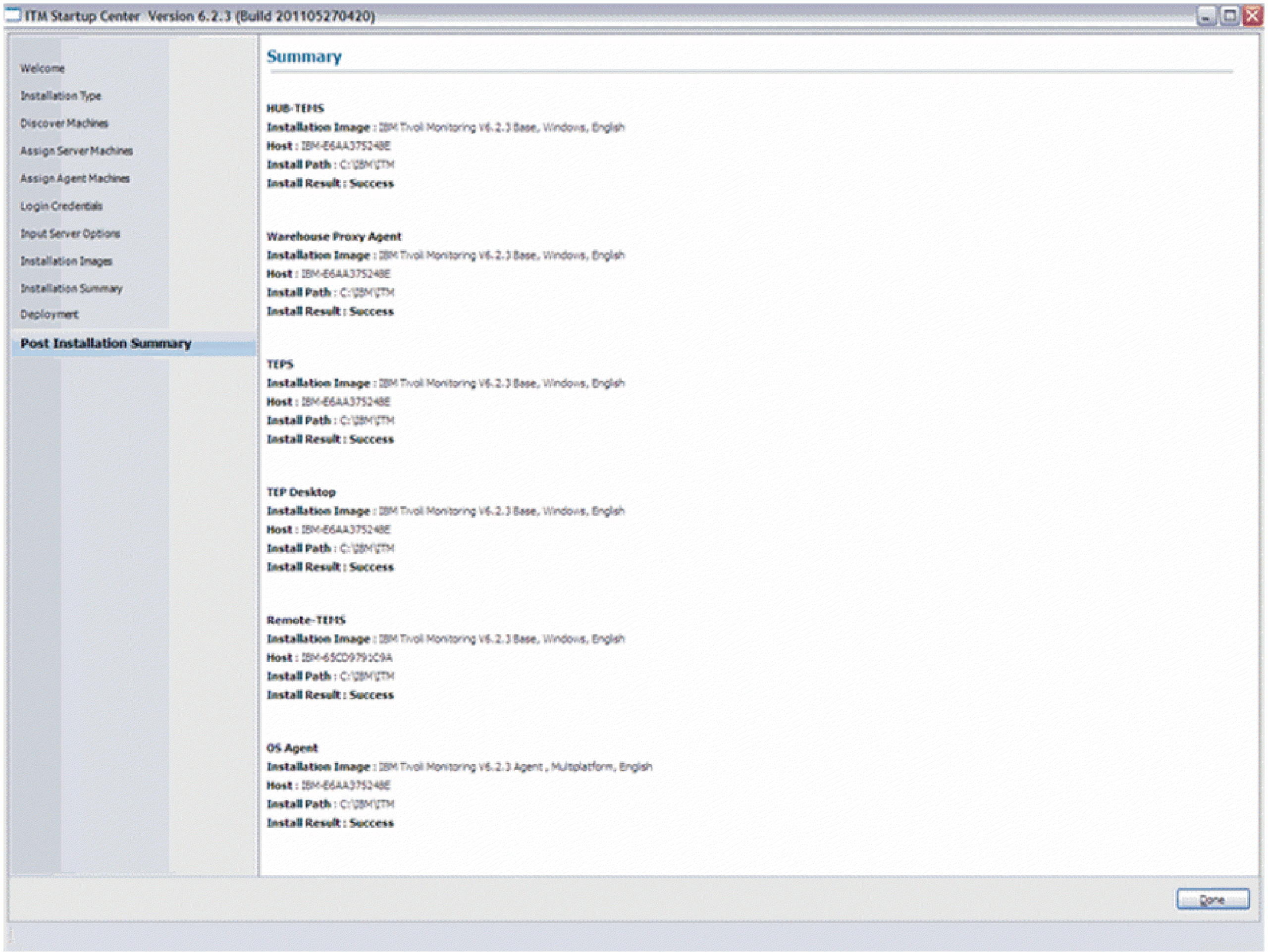
Useful Information:
Startup Center Video
Tivoli Documentation Central
Tivoli Wiki Central
Tivoli Media Gallery

Note: If a nonroot user is used to install an ITM agent on a UNIX computer, additional steps must be performed to complete the setup. The UpdateAutoRun.sh and SetPerm scripts must be executed as the root user on the target machines. Please see the section entitled "Changing the file permissions for agents" in the Installation Guide for more details.

Machine	Status	Progress	Status Details
⊟ IBM-65CD9791C9A			
Upload [Remote-TEMS	✓ Complete		Complete
Install [Remote-TEMS]	✓ Complete		Complete
⊟ IBM-E6AA375248E			
Upload [HUB-TEMS, W	✓ Complete		Complete
Install [HUB-TEMS, Wa	Running		Installing components HUB TEMS, Warehouse Proxy Agent, TEPS, TEP Desktop version 06230...
Upload [OS Agent]	★ Pending		
Install [OS Agent]	★ Pending		

Figure 30. Tivoli Monitoring Startup Center Deployment window

When the deployment of each component has completed, click **Next** to display the Post Installation Summary page.



Summary

HUB-TEMS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Install Result : Success

Warehouse Proxy Agent
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Install Result : Success

TEPS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Install Result : Success

TEP Desktop
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Install Result : Success

Remote-TEMS
Installation Image : IBM Tivoli Monitoring V6.2.3 Base, Windows, English
Host : IBM-65CD9791C9A
Install Path : C:\IBM\ITM
Install Result : Success

OS Agent
Installation Image : IBM Tivoli Monitoring V6.2.3 Agent , Multiplatform, English
Host : IBM-E6AA375248E
Install Path : C:\IBM\ITM
Install Result : Success

Figure 31. Tivoli Monitoring Startup Center Post Installation Summary window

Non-root user support

Be aware of the following requirements when using a non-root user ID for Startup Center configuration:

- If you use a non-root user to log on to a target machine and get a connection failed error, it might be because the password provided has expired. Ensure that passwords have not expired before you attempt to install components by using the Startup Center.
- If you install a monitoring agent on a UNIX computer as a non-root user, the file permissions are initially set to a low level. Additional steps must be performed to complete the setup. The `UpdateAutoRun.sh` and `SetPerm` scripts must be executed as the root user on the target machines. For more information, see the procedure in “Postinstallation steps for nonroot installations” on page 261.
- For Windows machines, any non-root user must be a member of the administrator group.
- If you select DB2 as your Tivoli Enterprise Portal Server database, you must reconfigure the Tivoli Enterprise Portal Server independently by using root authority. Complete the following steps to reconfigure the Tivoli Enterprise Portal Server:
  1. Log in as a root user and check that your installation path has the correct permissions. If the installation path belongs to a non-root user, change the access permission level to 755. For example, if the candle home directory is `/home/tester/ITM`, the existing `home` and `tester` folders must have permission 755. Use the following command to change the permission level:

     ```
     chmod 755/home/tester
     ```

where:

755
Is the new permission `rwxr-xr-x`.

/home/tester
Is the folder where IBM Tivoli Monitoring is installed.

2. Use the CLI tool or the Manage Tivoli Enterprise Monitoring Services window to reconfigure the Tivoli Enterprise Portal Server:
   - In the command-line interface, run the following command:

     ```
     <Candle_Home>/bin/itmcmd config -A cq
     ```
   - In the Manage Tivoli Enterprise Monitoring Services, right-click **Tivoli Enterprise Portal Server** and click **Configure**.
3. The required DB2 user and database parameters have already been set up in the Startup Center. Complete the configuration process by using the default values provided.
4. Use the following commands to add a new group, such as **itmgroup**, and add both the non-root user and DB2 administrator user to the group:

   ```
   groupadd itmgroup
   usermod -a -G itmgroup tester
   usermod -a -G itmgroup db2inst1
   ```

   where:

   tester
   Is the non-root user.

   db2inst1
   Is the DB2 administrator user.
5. After you have made your changes, use the following command to run the secureMain utility before you start the Tivoli Enterprise Portal Server:

   ```
   ./secureMain -g itmgroup lock
   ```

   Note: Whenever you reconfigure the Tivoli Enterprise Portal Server, you should run the secureMain utility before you restart the Tivoli Enterprise Portal Server.

   For more information on the secureMain utility, see Appendix G, "Securing your IBM Tivoli Monitoring installation on Linux or UNIX," on page 851.
6. Restart the Tivoli Enterprise Portal Server as a non-root user.

- For Tivoli Data Warehouse: Because the non-root user cannot create a DB2 user or reset a DB2 user's password, you must perform these tasks manually as a root user. You then add the user to the DB2 administrator group and re-configure the Tivoli Data Warehouse.

Default values taken by Startup Center components

By default, the Tivoli Monitoring Startup Center names the hub Tivoli Enterprise Monitoring Server as `machinename_TEMS` and the remote Tivoli Enterprise Monitoring Server as `machinename_RTEMS`. The following default values are used for all components:

INSTALL_ENCRYPTION_KEY=IBMTivoliMonitoringEncryptionKey
Encryption key.

SEED_TEMS_SUPPORTS=true
Seed this support on the monitoring server.

DEFAULT_DISTRIBUTION_LIST=NEW
Default seeding option for the *situation distribution definition* is NEW.

FIREWALL=NO
The Startup Center component does not connect to the Tivoli Enterprise Monitoring Server through a firewall.

PRIMARYIP=none
The primary IP address is not provided.

FTO=NO
Does not configure a connection for a secondary Tivoli Enterprise Monitoring Server.

NETWORKPROTOCOL=ip.pipe
The connection mode is IP_PIPE.

IPPIPEPORTNUMBER=1918
The default IP_PIPE port number is 1918.

The following default values are used for the Warehouse Proxy Agent:

- KHD_DB2_JDBCDRIVER=com.ibm.db2.jcc.DB2Driver
- KHD_BATCH_USE=true
- KHD_DB_COMPRESSION=false
- KHD_WAREHOUSE_TEMS_LIST=
- KHD_SERVER_Z_COMPRESSION_ENABLE=false
- KHD_SERVER_DIST_COMPRESSION_ENABLE=true

The following default value is used for the Tivoli Enterprise Portal Server:

- DB2ATTR=CONNECTION_LIMIT32

Remote Execution and Access

The Startup Center uses Remote Execution and Access (RXA) to establish a connection with a remote computer. RXA is an IBM developer toolkit that provides classes and methods to create an application that can establish a connection with a remote computer, log on to a remote computer, run commands and scripts on a remote computer (including installation and uninstallation), and manipulate the remote computer's files and directories. RXA does not require the remote computer to have a software agent, such as the Tivoli management agent, installed on the remote machine. Instead of relying on a remote agent for communications with a remote computer, RXA provides access to the remote machine by using rsh, rexec, SSH, Windows (Server Message Block/Common Internet File System), and AS/400® host server protocols.

The default connection timeout setting for RXA in the Startup Center is 180000 milliseconds. You can alter this setting by changing the CONNECTION_TIME_OUT setting located here:

- On Windows systems: `%USERPROFILE%\.STARTUP\workspace\userdata\custom.properties`
- On Linux systems: `${HOME}/.STARTUP/workspace/userdata/custom.properties`

You must restart the Startup Center after you change this setting.

Windows Targets

Some RXA operations rely on VBScript and Windows Management Instrumentation (WMI) calls to execute scripts on Windows targets. If the Windows Scripting Host (WSH) or the WMI service is disabled on the target, or if VBScript is otherwise disabled, some WindowsProtocol methods will not work.

If you intend to access Windows targets by using the SMB protocol over NetBIOS, which is determined by setSMBTransportType(), then port 139 or the port specified by setNetBIOSPort(), must not be blocked by firewalls or IP security policies. The Enable NetBIOS over TCP/IP must also be selected in the Control

Panel settings for the machine's network connections properties (**Control Panel** → **Network and Dial-Up Connections** → **<some connection>** → **Properties** → **Internet Protocol (TCP/IP)** → **Advanced** → **WINS** → **Enable NetBIOS over TCP/IP**).

Consult the documentation for your firewall to determine that these ports are not blocked for inbound requests.

To determine if security policies are blocking these ports, click **Start** → **Settings** → **Control Panel** → **Administrative Tools**. Depending on whether your policies are stored locally or in Active Directory, the next steps are as follows:

- Locally stored policies: **Administrative Tools** → **Local Security Policy** → **IP Security Policies on Local Computer**.
- Policies stored in Active Directory: **Administrative Tools** → **Default Domain Security Settings** → **IP Security Policies on Active Directory**.

Examine the IP security policies and edit or remove filters that block the ports listed above. Table 14 lists the ports reserved for NetBIOS. Ensure that all ports currently used by RXA are not blocked.

Table 14. NetBIOS Reserved Ports

Port number	Use
135	NetBIOS Remote procedure call. At this time, RXA does not use this port.
137	NetBIOS name service.
138	NetBIOS datagram. At this time, RXA does not use this port.
139	NetBIOS session (file/print sharing).
445	CIFS (On XP and Win2K).

A utility program (testconn) is provided in the RXA .zip and JAR files. It can be used to determine whether a remote Windows target is configured to run Server Message Block protocol on top of NetBIOS (NetBIOS over TCP/IP) by using port 139, or whether the target is configured to run SMB on top of TCP/IP (without the NetBIOS layer) by using port 445. The target machine must have the Remote Registry service started (which is the default configuration) in order for RXA to connect to the target machine. A utility program (getregkey) is provided in the RXA zip and JAR files. It can be used to determine whether a Windows target (local or remote) has Remote Registry management enabled.

RXA requires access to the hidden remote administrative disk share for access to the system %TEMP% and other directories. Access to the Interprocess Communications share (IPC$) is also required for RXA to access remote registries. Before you access the Interprocess Communications share (IPC$), make sure the Server service is started (**Control Panel** → **Administrative Tools** → **Services** → **Server**). You can use the `testconn.exe` utility (in the `\diagtools` directory) to verify that the administrative share is accessible. RXA requires Simple File Sharing to be disabled. The next section details information specific to operating systems.

Disabling User Account Control to facilitate RXA

You might have to disable the User Account Control to enable RXA to connect to your Windows operating system.

Windows XP: Windows XP systems must have Simple File Sharing disabled for RXA to work. Simple Networking forces all logins to authenticate as "guest". A guest login does not have the authorizations required for RXA to function.

To disable Simple File Sharing, start Windows Explorer and click **Tools → Folder Options**. Select the **View** tab and scroll through the list of settings until you find **Use Simple File Sharing**. Remove the check mark next to **Use Simple File Sharing**, and then click **Apply** and **OK**.

Windows XP includes a built-in firewall called the Internet Connection Firewall (ICF). By default, ICF is disabled on Windows XP systems. Windows XP Service Pack 2 includes the Windows ICF set to ON by default. If either firewall is enabled on a Windows XP or Vista target, the firewall blocks access by RXA. On XP Service Pack 2, you can select the **File and Printer Sharing** box in the **Exceptions** tab of the Windows Firewall configuration to allow access.

Windows 2003: Windows 2003 systems must have Simple File Sharing disabled for RXA to work. Check that the firewall settings are the same as outlined for Windows XP above.

Windows Server 2008: On Windows Server 2008 you might need to disable User Account Control if your account is not a domain user account. See the section on Windows Vista to learn how to disable User Account Control.

Windows Vista: The new User Account Control feature in Windows Vista requires users to perform several steps before RXA applications can communicate with Vista targets. If you have a domain user account, ensure that the local and the target machine are both members of a Windows domain.

If you are a member of a local administrators group and you use a local user account, complete the three steps below to be able to perform administrative tasks on the target machine:

1. Enable the built-in Administrator account and use it to connect to the target system. To enable the built-in Administrator account, open the Windows Control Panel and click **Administrative Tools → Local Security Policy → Security Settings → Local Policies → Security Options**. Then double-click **Accounts: Administrator account status** and select **enable**.
2. Disable User Account Control if a different Administrator user account is to be used to connect to the Vista target. To disable User Account Control, open the Windows Control Panel and click **Administrative Tools → Local Security Policy → Security Settings → Local Policies → Security Options**. Then double-click **User Account Control: Run all administrators in Admin Approval Mode** and select **disable**. Changing this setting requires a system restart.
3. Disable User Account Control when you administer a workstation with a local user account (Security Account Manager user account). Otherwise, you cannot connect as a full administrator and cannot complete administrative tasks. To disable User Account Control, complete the following steps:
   a. Click **Start**, click **Run**, type `regedit`, and then press **ENTER**.
   b. Locate and then click the following registry subkey:
   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
   CurrentVersion\Policies\System
   ```
   c. If the LocalAccountTokenFilterPolicy registry entry does not exist, follow these steps:
      1) On the Edit menu, point to New, and then click **DWORD** Value.
      2) Type `LocalAccountTokenFilterPolicy`, and then press **ENTER**.
      3) Right-click **LocalAccountTokenFilterPolicy**, and then click **Modify**.
      4) In the Value data box, type `1`, and then click **OK**.
      5) Restart your computer.

   Alternatively, you can modify the registry entry manually by typing the following command at a command prompt:
   ```
   cmd /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\
   system /v LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
   ```

Windows 7: On Windows 7, the default startup type for the Remote Registry service is manual. The Remote Registry service must be running to enable RXA.

To check whether the Remote Registry service is enabled and started:

1. Go to **Start**.
2. In the Start Search box, type `services.msc`. Press **ENTER**.
3. When Microsoft Management Console starts, in the console pane, ensure that the service status is: started. If not, right-click **Remote Registry**, and click **Start**.

To avoid problems with the manual startup, it is advisable to set the Remote Registry service startup type to automatic. To automatically start the service after the server starts:

1. Right-click **Remote Registry** and select **Properties**.
2. In the Startup type option, choose **Automatic**.
3. Click **Apply** and **OK**.

When the system starts up, Remote Registry automatically starts.

Windows Vista FDCC (Federal Desktop Core Configuration): With Windows Vista FDCC custom security settings, it is not possible to connect to this operating system by using RXA.

Complete the following steps on Windows Vista FDCC to enable RXA to connect to the operating system:

1. Allow File and Printer Sharing with the Firewall by enabling the inbound file and printer exception by using the Local Group Policy Editor:
   a. Go to **Start**.
   b. In the **Start Search** box, type `gpedit.msc`. Press ENTER.
   c. Go to: **Local Computer Policy** → **Computer Configuration** → **Administrative Templates** → **Network** → **Network Connections** → **Windows Firewall** → **Standard Profile and enable Windows Firewall: Allow inbound file and printer sharing exception**
2. Turn off the User Account Control.
3. Start the Remote Registry service.

Chapter 5. Post-deployment phase

Now that you have your Tivoli Monitoring V6.2.3 environment in production, there are some important steps that need to be taken to ensure your environment stays up and running and healthy. This includes applying routine maintenance and performing periodic health checks. The post-deployment phase is divided into two sections: "Applying maintenance" and "Maintaining an efficient monitoring environment" on page 119.

Applying maintenance

This section outlines the planning and implementation steps necessary to install maintenance in your Tivoli Monitoring V6.2.3 environment. Routine maintenance is outlined in the following sections:

- "Planning an upgrade"
- "Upgrade steps"
- "Post-upgrade health check" on page 118

Planning an upgrade

Use the following checklist to plan your upgrade:

- Check that the plan is in place for upgrading the environment (upgrade the environment incrementally). Follow a formal change management plan for Tivoli Monitoring upgrades and include, at minimum, both a deployment and tested backout plan.
- Download the correct upgrade media, *not* the fresh installation media.
- Back up all Tivoli Monitoring core components such as monitoring server and portal server.
- Carefully review the appropriate Fix Pack Readme and Documentation Addendum for any prerequisites.

Attention: Before upgrading your infrastructure components and beginning the upgrade process, perform a cold backup of your hub monitoring server, portal server, portal client, Warehouse Proxy Agents, Summarization and Pruning Agents, and remote monitoring server. Back up the following key components:

- Portal server database
- Warehouse database
- Full system backups and file system backups for installed Tivoli Monitoring components

Upgrade steps

When performing upgrades read this *Installation and Setup Guide* or the supplied fix pack readme carefully. Perform your install in the following order:

Note: This order might vary depending on the content of the release and the fix pack.

1. Event Synchronization
2. Warehouse including Warehouse Proxy Agent and Summarization and Pruning Agent
3. Hub Tivoli Enterprise Monitoring Server
4. Remote Tivoli Enterprise Monitoring Server
5. Tivoli Enterprise Portal Server
6. Run any scripts necessary to update the Warehouse schema
7. Tivoli Enterprise Portal desktop client
8. Update monitoring agents

   In some cases, when you upgrade the infrastructure components you will also be upgrading the monitoring agents on those servers. It is okay to upgrade those monitoring agents at that time.

Self-describing monitoring agents is a new feature in V6.2.3 that integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure. For more information, see "Self-describing agent installation" on page 347.

For more information about installing fix packs, see "Installing product maintenance" on page 324.

Post-upgrade health check

Use the following checklist for your post-upgrade health check:

- Check if the predefined and user-created workspaces that were present prior to the upgrade are still in place.

  Log in to the portal server by using the portal browser or desktop client and browse through the different workspaces for each product type.
- If you enabled the self-describing agent capability for your upgrade, run the `tacmd listappinstallrecs` to ensure that all products upgraded by the self-describing agent feature completed the installation without errors. Check the Audit Log messages for the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server to ensure there are no self-describing agent errors related to the latest product upgrades.
- Check if the predefined and user-created situations that were present prior to the upgrade are still in place.

  Log in to the portal server by using the portal browser or desktop client and browse through different situations in the Situation Editor or run the **tacmd listsit** command.
- Check if all the catalogs are the same on each monitoring server (hub and remote). Try one of the following two approaches:
  - Run grep on each monitoring server (hub and remote). For example:

    ```
    grep @ * | awk '{print $2, $3, $4, $5}' | sort
    ```
  - Run the following SQL against each monitoring server in a portal server view:

    ```
    "SELECT APPL_NAME, TIMESTAMP FROM SYSTEM.SYSAPPLS AT ('REMOTE_TEMS') ORDER BY APPL_NAME"
    ```
- Check if the depots populated on each monitoring server (hub and remote) are the same.

  Run these commands from the hub monitoring server.
  - **tacmd viewdepot**
  - **tacmd viewdepot -j** *remote_tems*
- Check if the warehouse data is visible through the workspace views, meaning the portal server still has the correct connection to the warehouse database.

  Select the attribute group for which history collection is enabled by checking that view and making sure the data can be pulled for more than 24 hours.
- Check if the agents are online and connected to the expected remote monitoring server.

  Run the **tacmd listsystems** command.
- Check if the situations are firing and events are being forwarded to Tivoli Enterprise Console or OMNIbus.

  Run the command on the Tivoli Enterprise Console server using **wtdumprl** or drag the Tivoli Enterprise Console icon to any view to view the events.
- Check if the historical configuration is active.

  Log in to the portal server by using the portal browser or desktop client and click **History Configuration**. Browse through the desired attribute groups to see if they are still active.

  Or you can run this query: `"SELECT NODEL, OBJNAME, LSTDATE FROM O4SRV.TOBJACCL WHERE OBJNAME LIKE 'UADVISOR*'"`
- Check if the Warehouse Proxy Agent and Summarization Pruning agents correctly started, meaning the agents made successful connections to warehouse database.

You can examine the WAREHOUSELOG table to see the last updates by each attribute group. See sample below:

```
SELECT ORIGINNODE AS "Agent Hostname", OBJECT AS "Attribute Group",
EXPORTTIME AS "Export Time", ROWSRECEIVED AS "Received Rows",
ROWSINSERTED AS "Inserted Rows", ROWSSKIPPED AS "Skipped Rows",
ERRORMSG AS "Error Message" FROM WAREHOUSELOG
```

Note to Windows users: If you attempt to run a tacmd CLI command and either the Embedded Java Runtime or the User Interface Extensions are not available on the node where you invoke the command, you will receive the error shown in Figure 63 on page 258. If this happen, complete the procedure outlined in “Installing the Embedded Java Runtime and the User Interface Extensions” on page 258, and retry the tacmd command.

Maintaining an efficient monitoring environment

This section covers the daily, weekly, monthly, and quarterly routine health checks on the Tivoli Monitoring V6.2.3 enterprise environment.

- “Daily health checks” on page 120
- “Weekly health checks” on page 120
- “Monthly health checks” on page 121
- “Quarterly health checks” on page 121

By performing these routine procedures in addition to your daily, weekly, monthly and quarterly health checks, you ensure that your Tivoli Monitoring environment continues to run smoothly.

- Run the **taudit.js** tool which can be found in the Tivoli Integrated Service Management Library by searching for "Web SOAP scheduled reporting tools" or navigation code "1TW10TM0U." This tool provides an overall status of the environment. Run this tool every day.
- Take a monitoring server backup every 24 hours in early stages and then move to weekly backups. If you have snapshot software, you can take backups with the monitoring server or portal server, or both online. Otherwise, shutdown the monitoring server and portal server before taking a backup. Test these backups after you first develop the process and at least twice a year thereafter by restoring to a monitoring server in your test environment to ensure the backups are successfully backing up the data you need to restore your production monitoring server in the event of an outage or need for rolling back to a previous state.
- Make sure the portal server database backup is in the plan and is being made daily as the environment is being rolled out and then weekly as the environment matures and less frequent changes are made to the environment. Test these backups after you first develop the process and at least twice a year thereafter by restoring to a portal server in your test environment to ensure the backups are successfully backing up the data you need to restore your production portal server in the event of an outage or need to rollback to previous state.
- Make sure the DB2 warehouse backup is in the plan and is being made weekly. The reason you need to do this weekly is because of huge database size.
- Check daily that the warehouse agent is performing by looking at the warehouse logs (`hostname_hd_timestamp-nn.log`).
- Check daily that the Summarization and Pruning Agent is performing by looking at the (`hostname_sy_timestamp-nn.log`) logs.
- Check the monitoring server (hostname_ms_timestamp-nn.log) and portal server logs (`hostname_cq_timestamp-nn.log`) for any obvious errors and exceptions.
- Check that there are no monitoring servers overloaded with agents. One way to do this is by checking the "Self-Monitoring Topology" workspace, which has a "Managed Systems per TEMS" view showing the number of agents reporting to each monitoring server.
- For DB2, run the **REORGCHK** and **RUNSTATS** on the warehouse database daily.

- For DB2, run the **REORGCHK** and **RUNSTATS** on the portal server database weekly.
- Check that events are reaching the Tivoli Enterprise Console server or OMNIbus. It is also important to check that events from custom Universal Agents reach the event server.
- Check that all the fired situations are answered with a response and are not in open state for a long period of time.
- Check that the Tivoli Enterprise Monitoring Server can communicate to each of the monitoring agents. The easiest way to test this is to run the taudit.js tool (mentioned above).
- Check the core components process memory and CPU usage and that you have situations created to monitor them.

Daily health checks

Use the following list to perform daily health checks.

Tasks

- Make sure all of the systems from the day before are still online. A quick look at the Managed System Status workspace shows the status of each managed system. If you encounter managed systems that are offline investigate them individually.

  There are several reasons why a managed system can go offline. The agent might have gone offline for some reason, there may be communication problems between the agent and the monitoring server that it is connected to or the agent was decommissioned. In any case the cause of the problem must be found and addressed. Run a script every morning that provides a report on ONLINE and OFFLINE systems, **Taudit.js** can be used for this purpose.
- You might find situations that are in open status that have not been addressed (acknowledged). Determine if the problem reported by the situation is valid. Determine if there is really a problem or is it a case where the situation does not have the correct thresholds and is producing a false positive that is being ignored by the monitoring staff. Make sure your situations are reflecting real events, which helps train the monitoring staff to react to each event that goes true in the Tivoli Monitoring environment.
- If you have decided to collect historical data and are using the Tivoli Data Warehouse, make sure the Warehouse Proxy Agent and Summarization and Pruning Agents are up and running. Check the logs for both to make sure the agents are collecting and summarizing data on the intervals you have set. To find a solution that allows you to monitor the warehouse activity to ensure that it is functioning properly, search for "Data Warehouse DB activity" or navigation code "1TW10TM1X" in the Tivoli Integrated Service Management Library.
- Spot-check the workspaces for several different monitoring agent types to make sure report data is being returned.

Weekly health checks

Use the following list to perform weekly health checks.

Tasks

- Include all of the items in the daily health check.
- Check system backups. Have a mechanism in place for backing up your core Tivoli Monitoring components in the event of recovery. The portal server, monitoring server, and warehouse must be backed up on a regular interval. That interval must be decided by you, but make it no less than a weekly backup. Daily backups of the portal server and monitoring server databases are highly recommended because of their constant change.
- Check warehouse data. Make sure you are collecting the last week's worth of data and it is correct. You can accomplish this by using the portal client to run the weekly summarization reports. The summarized data returned must accurately reflect the past week's worth of summarized data.

Monthly health checks

Use the following list to perform monthly health checks.

Tasks

- Include all of the checks from the daily and weekly item checklist.
- If you are collecting historical data and storing it in the warehouse make sure your monthly summarization is correct. Validate this by running reports from the portal client to ensure you have the correct monthly summarized data.
- Check the list of managed systems deployed in your Tivoli Monitoring environment. Take note of their maintenance levels. Check with IBM Software Support or your IBM account representative to see if new fix packs and interim fixes are available. If so, determine what has been fixed so you can decide if you want to deploy the patches to your environment or just wait until the next major fix pack.
- Once again check your situation thresholds to make sure you don't have false positive events. In large user environments there are many factors that can have an affect on how a system performs. The change in performance in any system can change the way Tivoli Monitoring V6.2.3 reports status for any given system. Make sure the events active in Tivoli Monitoring are real.
- Take inventory of the systems being managed by Tivoli Monitoring. There might be a need to deploy additional agents on new systems or systems where new applications have been added.
- Assess the capacity of the infrastructure systems for resource CPU, memory and disk utilization to continually plan for overall workload balancing. As new versions of applications are introduced into the environment, their affect on resources typically change. This ongoing effort helps ensure the correct hardware is in place. Confirm the number of agents connected to each remote monitoring server to ensure that you have not exceeded the recommended limit of 1500 agents.

Quarterly health checks

Use the following list to perform quarterly health checks.

Tasks

- Include all of the checks from the daily, weekly, and monthly checklist.
- Discuss the usage of the historical data with the end users who rely on the data for various reasons. You might find they are not looking at some of the data being collected and no longer need it. In this case, turn off the collection so you are not using unnecessary resources. The same holds true for the reverse. There might be data missing that is needed and historical collection can then be activated for the necessary information.
- Check with IBM Software Support or your IBM account representative for fix packs on all IT components. Regular maintenance fix packs for each component are typically delivered on a quarterly basis. Look through the Readme files and decide if you feel it is necessary to install the fix pack. Install the latest fixes for any of the components.

Part 3. Installation and initial configuration of base components and agents

The chapters in this section provide instructions for installing and configuring base components and agents. If you are installing Tivoli Monitoring for the first time, review the chapters in the preceding section before you use the chapters in this section.

Chapter 6, “Preparing for installation,” on page 125, helps you collect the information you will need during installation and configuration and details the hardware and software requirements for various components and configuration of supported databases.

Chapter 7, “Upgrading from a previous installation,” on page 159, provides instructions for upgrading an existing installation of OMEGAMON Platform 350/360 or IBM Tivoli Monitoring version 6.1 to version 6.2.

Chapter 8, “Installing IBM Tivoli Monitoring on one computer,” on page 187, contains instructions for installing the base components on a single Windows computer. This scenario is useful for creating a test or teaching environment or for monitoring a small environment.

Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207, contains instructions for installing each of the base components on Windows, UNIX and Linux (command and GUI installations) computers and for completing the initial configuration of those components. It also contain procedures for installing and configuring the base agents and for installing support for those agents on the Tivoli Enterprise Portal and the Tivoli Enterprise Monitoring Server.

Chapter 10, “Deploying monitoring agents across your environment,” on page 325, provides instructions for deploying distributed monitoring agents from a monitoring server.

Chapter 6. Preparing for installation

The following sections provide information to help you prepare to install your IBM Tivoli Monitoring environment.

Overview of the installation process

The following table provides an overview of the steps required to fully install and deploy an IBM Tivoli Monitoring environment.

Table 15. Installation and configuration steps

Step	Where to find detailed information
Assess your monitoring needs to determine the best deployment of IBM Tivoli Monitoring components.	Chapter 2, “Pre-deployment phase,” on page 33
Ensure you have the required hardware and software.	“Hardware and software requirements” on page 138
Gather any information required for successful installation (such as DB2 user information and security specifications).	“Specific information to have ready” Appendix A, “Installation worksheets,” on page 777
Install the Tivoli Enterprise Monitoring Server.	“Installing and configuring the hub Tivoli Enterprise Monitoring Server” on page 208
Install the Tivoli Enterprise Portal Server.	“Installing the Tivoli Enterprise Portal Server” on page 228
Install the management agent software.	“Installing monitoring agents” on page 253 “Self-describing agent installation” on page 347
Install support for IBM Tivoli Enterprise Console.	Chapter 25, “Setting up event forwarding to Tivoli Enterprise Console,” on page 643
Install the portal desktop client on any system where you want to use it.	“Installing the Tivoli Enterprise Portal desktop client” on page 263
Start the portal client to verify that you can view the monitoring data.	“Starting the Tivoli Enterprise Portal client” on page 320

If you are upgrading from IBM Tivoli Monitoring V6.1 or OMEGAMON Platform 350 or 360 and CandleNet Portal 195, see Chapter 7, “Upgrading from a previous installation,” on page 159 before installing any IBM Tivoli Monitoring components.

If you are upgrading from Tivoli Distributed Monitoring to IBM Tivoli Monitoring, see the *IBM Tivoli Monitoring: Upgrading from Tivoli Distributed Monitoring* guide.

If you are upgrading from IBM Tivoli Monitoring V5.x to V6.2, see *IBM Tivoli Monitoring: Upgrading from V5.1.2*.

If you plan to use firewalls in your environment, see Appendix C, “Firewalls,” on page 799 for an overview of the IBM Tivoli Monitoring implementation of firewalls.

Specific information to have ready

During installation, you must supply the following information:

- Name of the monitoring server you are installing or to which the agent will connect
- Host name for the computer where you are installing the product (a monitoring server or one instance of an agent)

- Whether the monitoring server being installed or being connected to is configured as a hub or remote monitoring server
- Hub monitoring server host name
- Port number

Use the worksheets in Appendix A, “Installation worksheets,” on page 777 to collect this information for each component that you want to install.

Information to gather for event forwarding

You need the following additional information to successfully install and configure event forwarding and synchronization between a hub Tivoli Enterprise Monitoring Server and either IBM Tivoli Enterprise Console or Netcool/OMNIbus:

- Host names, user IDs, and passwords for the monitoring servers that you want to receive events from.
- The amount of free space in your temporary directory. The installation requires 200 MB of temporary space.
- Simple Object Access Protocol (SOAP or Web Services) information to send events to a monitoring server (the URL, the rate to send requests to the server).

  By default, all monitoring servers are configured as SOAP servers. If you did not change this configuration to make it unique for your environment, you can accept the default values during the installation.

  If you did change this configuration, use the SOAP information unique to your configuration.
- For TEC, the host of the event server or servers to which events are being forwarded and the port on which it is listening. For Netcool/OMNIbus, the host of the EIF probe and the port on which it is listening.
- For TEC, event rule base information (either the name of a new rule base to create or the name of an existing rule base to use)

  Notes:

  1. For a Windows event server, any existing rule base that you use must indicate a relative drive letter (such as C:\) as part of its associated path. To verify that your existing rule base contains a relative drive letter, run the following command from a bash environment on your event server:

     ```
     wrb -lsrb -path
     ```

     If the returned path includes something like *hostname*:*Rulebase_directory*, with no drive letter (such as C:\), copy the `ESync2300Win32.exe` file from the \TEC subdirectory of the IBM Tivoli Monitoring installation image to the drive where the rule base exists and run the installation from that file.
  2. If you are using a Windows event server, if you have any rule base with an associated path that does not contain a relative drive letter and that has the Sentry2_0_Base class imported, copy the `ESync2300Win32.exe` file from the \TEC subdirectory of the IBM Tivoli Monitoring installation image to the drive where the rule base exists and run the installation from that file.

     To verify if you have any rule bases that have an associated path containing no relative drive letter, run the **wrb -lsrb -path** command as described in the previous note.

     To determine if your rule bases have the Sentry2_0_Base class imported, run the following command against all of your rule bases:

     ```
     wrb -lsrbclass rule_base
     ```

     where *rule_base* is the name of the rule base.

Naming your monitoring server

You must decide how the monitoring servers are to be named. In general, create names that are short but meaningful within your environment. Use the following guidelines:

- Each name must be unique. One name cannot match another monitoring server name for its entire length. (For example, "ibm" and "ibmremote" are unique and permitted.)
- Each name must begin with an alpha character. No blanks or special characters ("$#@") can be used.
- Each name must be between 2 and 32 characters in length.
- Monitoring server naming is case-sensitive on all platforms.

Choose between IPv6 and IPv4

You can now enable IP version 6 (IPv6) communication between any two IBM Tivoli Monitoring components; possible configurations include communications between the portal server and the hub monitoring server, a remote monitoring server and the hub, or an agent and a hub. Both components need to be at version 6.2, with the exception of agents, which can be at version 6.1. IPv6 is not supported for sending EIF events from the hub monitoring server or from monitoring agents.

To use this capability, your IBM Tivoli Monitoring environment must be configured and enabled for IPv6 communication. IPv6 communication over IPv4-only networks is not supported.

Before components can be enabled for IPv6 communication, the following requirements must be met:

1. The host on which the components are located must be enabled for IPv6.
2. If the component needs to communicate using IPv6 with some components and IPv4 with others, the host must be enabled for dual-stack operation.

   Note: A dual-stack host provides two discrete network layers. The term *stack* here refers to the protocol stack, a suite of protocols used in computer networking software.
3. The host must have DNS or hosts file entries for both IPv4 and IPv6 addresses. The host must also be configured to resolve host names and IP addresses from the DNS or from the hosts file.
4. The network path between the two components must be enabled for IPv6 traffic. Tunneling of IPv6 traffic over IPv4 networks is not supported.

Table 16 shows the supported combinations of IPv6 with IPv4 across the various IBM Tivoli Monitoring components.

Table 16. Supported IBM Tivoli Monitoring configurations using the IPv6 communications protocol

Valid configurations	Portal client	Portal server	Hub monitoring server	Agents	Remote monitoring server	Agents (connected to the remote monitoring server)
IPv6 only	IPv6	IPv6	IPv6	IPv6	IPv6	IPv6
IPv6 with IPv4	IPv4	IPv4 or IPv6	IPv4 or IPv6	IPv4[1]	IPv4 or IPv6	IPv4[1]

Notes:

1. All agents running on a computer must be configured to use the same protocol, either IPv4 or IPv6.
2. In scenarios where some agents are on IPv4 only computers or the network between the agents and the monitoring servers they report to is IPv4 only, these agents need to communicate with the monitoring servers over IPv4. The monitoring servers therefore may communicate with some agents over IPv4 and with others over IPv6.
3. The portal server does not support IPv6 on the Windows platform. If the portal server is on Windows, the browser and desktop clients need to communicate with it using IPv4.
4. Components do not operate in dual-stack mode on the Solaris platform. Components can be configured to communicate using either IPv4 or IPv6. Thus, if a hub server on a Solaris host is configured to use IPv6, the portal server, all remote servers, and all agents connecting to the hub must be configured to use IPv6 for communicating with the hub.

5. On HP-UX, patch PHNE_29445 is required for IPv6 support.
6. Components do not operate in dual-stack mode on the HP-UX HP9000 platform. Dual-stack mode is supported on the HP-UX Integrity platform.
7. On Linux computers, a minimum kernel level of 2.6 is required for IPv6 support.

Monitoring components, when installed and configured using the appropriate platform-specific configuration tools, are initially configured only for IPv4 communication on all platforms except z/OS (where your ICAT settings govern the protocol used). On all other platforms, you must perform supplemental configuration steps to reconfigure Tivoli Monitoring components to communicate using IPv6.

For more information, see Chapter 14, “Configuring IBM Tivoli Monitoring components for IPv6 communication,” on page 387.

Required order of installation or upgrade of IBM Tivoli Monitoring component products

If any of the following products will be installed on the same computer as monitoring agents, they must be installed before the agent is installed:

- Hub Tivoli Enterprise Monitoring Server
- Remote monitoring server (if necessary)
- Tivoli Enterprise Management Agent Framework
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal desktop client

In addition, these products must be installed on at least one computer before the agent can be properly configured. The appropriate Tivoli Enterprise Management Agent Framework is installed when an agent is installed.

Windows installation considerations

The following sections provide information about issues unique to Windows installations.

User authority

To install IBM Tivoli Monitoring on a Windows computer, you must have Administrator privileges on that computer. You must also run the IBM Tivoli Monitoring components as a user with Administrator privileges.

32 bit versus 64 bit

If your site runs either Windows 2003 or 2008 on 64-bit x86-64 CPUs, you must decide whether to install the 32-bit operating system agent or the 64-bit agent. The new 64-bit agent supports 64-bit operations and can coexist with other 32-bit agents in your Tivoli Monitoring environment.

Notes:

1. Direct upgrade of the 32-bit Windows agent to the 64-bit agent is not supported. When upgrading a 32-bit agent from a prior release to the current release, you can upgrade only to the current 32-bit agent.
2. This new support does not extend to native 64-bit applications (including the operating system) running on the Itanium IA64 architecture.
3. This 64-bit support has not been extended to the IBM Tivoli Monitoring servers (the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server).
4. Support was added for a 64-bit Java Runtime Environment (JRE) to the V6.2.3 release of IBM Tivoli Monitoring. Prior releases supported 32-bit JRE only. The 64-bit JRE can only work on 64-bit machines. The ITM Installer does not allow a 64-bit JRE to be installed on a 32-bit machine. The installation summary dialog informs you of the JRE type being installed.

Installation using a Citrix client

If you are using a Citrix client to access the IBM Tivoli Monitoring installation program for Windows through Microsoft Windows Terminal Services, you must manually change Terminal Services to install mode before running the installation. To change Terminal Services to install mode, run the **change user /install** command before starting the installation. After installation, run the **change user /execute** command to return Terminal Services to normal mode.

Linux or UNIX installation considerations

The following sections provide information about issues unique to Linux and UNIX installations:

- "Changes in the behavior of the autostart scripts"
- "Create an IBM Tivoli account for installing and maintaining the installation directory" on page 133
- "Host name for TCP/IP network services" on page 133
- "Use of fully qualified path names" on page 133
- "Multiple network interface cards" on page 133
- "Installing into an NFS environment" on page 134
- "Installing into Solaris zones" on page 134
- "File descriptor (maxfiles) limit on UNIX and Linux systems" on page 135

Changes in the behavior of the autostart scripts

The behavior of the autostart scripts generated during installation and configuration on UNIX platforms has evolved.

- In V6.1 fix pack 3, the installation process produced an autostart script with only one entry using a generic `CandleAgent start all` command. Users modified this file as needed.
- In V6.1 fix pack 4, the installation process generated individual entries for each application in a particular installation, but the values captured in the file could not be overridden.
- In V6.1 fix pack 5, the multiple entries remain, and an override capability has been added.
- In V6.2.2 fix pack 2, the multiple entries remain, and the override capability has been significantly enhanced.

The autostart script generated by an installation, upgrade, or configuration and named ITMAgents*N* or rc.itm*N* (depending on the UNIX platform) contains an entry for each application in a particular installation. The entries look similar to this:

```
su - USER -c "ITM_Install_Home/bin/itmcmd agent start product_code"
```

Or:

```
su - USER -c "ITM_Install_Home/bin/itmcmd agent –o Instance start product_code"
```

Where:

USER
Is the ID that the application will be started as. By default, *USER* is the owner of the bin directory for the application. For the UNIX Log Alert agent, *USER* is the owner of the *ITM_Install_Home*/*PLAT*/ul/bin directory.

N Is an integer specific to each installation on a system.

ITM_Install_Home
Is the full path to the IBM Tivoli Monitoring version 6.*x* installation directory.

product_code
Is the two-character code for this application. See Appendix D, "IBM Tivoli product, platform, and component codes," on page 815 for a list of the codes for the common components and the base agents. See the product documentation for other product codes.

instance
Is the instance name required to start this application.

PLAT
Is the platform directory where the application is installed.

Components are started in the order listed in the autostart script. This order is based on the dependencies between components, rather than any logical sequence.

The kcirunas.cfg file was added to allow overrides to the default processing. The kcirunas.cfg file is delivered in the root directory of the installation media, in the same location as install.sh. During installation, this file is copied to the *ITM_Install_Home*/config directory (but is not overwritten if this file already exists). This file is provided as a sample file with each section commented out. You do not have to modify this file if you want the autostart script to be generated with the default processing.

For local installation usage, you may modify the kcirunas.cfg file in the root directory of the media if you want to use the same set of values for multiple installations on similar systems from this image. You may also modify the kcirunas.cfg file in the *ITM_Install_Home*/config directory if you want to use a specific set of values for each individual installation from this image.

For remote deployment usage, you can modify the kcirunas.cfg file in the root directory of the media. You can also modify the kcirunas.cfg file in the Tivoli Enterprise Monitoring Server depot after populating the depot from this image. If you have set the **DEPOTHOME** variable in the tables/TEMS_NAME/KBBENV file, you must use that value as the base when searching for the depot location. To determine if you have set **DEPOTHOME**, run the following commands:

```
cd ITM_Install_Home
DEPOTHOME=`find tables -name KBBENV -exec grep DEPOTHOME {} \; 2> /dev/null | cut -d= -f2`
echo $DEPOTHOME
```

If DEPOTHOME is not empty, run the following commands to locate kcirunas.cfg in the monitoring server depot:

```
cd ITM_Install_Home
DEPOTHOME=`find tables -name KBBENV -exec grep DEPOTHOME {} \; 2> /dev/null | cut -d= -f2`
find $DEPOTHOME -name kcirunas.cfg -print
```

If DEPOTHOME is empty, run the following commands instead:

```
cd ITM_Install_Home
find tables -name kcirunas.cfg -print
```

The file kcirunas.cfg contains a superset of the XML syntax and structure in the *ITM_Install_Home*/config/*HOST*_kdyrunas.cfg file (where *HOST* is the short hostname for this system) produced by remote configurations, such as remote deployment or Tivoli Enterprise Portal-based agent configuration.

The entries in kcirunas.cfg do not affect the actions performed for remote deployment, remote configuration, remote starting or stopping, or any Tivoli Enterprise Portal-initiated agent action. The entries in *HOST*_kdyrunas.cfg affect the generation of the reboot script. The entries in kcirunas.cfg also affect the generation of the reboot script, and they override any entries for the same component in *HOST*_kdyrunas.cfg.

The following is the default kcirunas.cfg file with all **<productCode>** entries commented:

```
<agent>

  <!productCode>ux</productCode>
  <instance>
   <user>itmuser</user>
  </instance>

  <!productCode>ul</productCode>
  <instance>
   <user>root</user>
  </instance>

  <!productCode>lz</productCode>
  <instance>
   <user>itmuser</user>
  </instance>

  <!productCode>ud</productCode>
  <instance>
   <name>db2inst1</name>
   <user>db2inst1</user>
  </instance>
  <instance>
   <name>db2inst2</name>
   <user>root</user>
  </instance>

  <!productCode>ms</productCode>
  <instance>
   <name>HUB17</name>
   <user>itmuser</user>
  </instance>

  <!productCode>cq</productCode>
  <instance>
   <user>itmuser</user>
  </instance>

  <!productCode>cj</productCode>
  <instance>
   <user>itmuser</user>
  </instance>

 </agent>
```

By default, each **<productCode>** section in the `kcirunas.cfg` file is disabled by making the product code a comment, such as **<!productCode>**. To activate a section, do the following:

1. Remove the comment indicator (the exclamation point, !) so that the **<!productCode>** item looks like **<productCode>**.
2. Copy a **<productCode>** section.
3. Rather than creating new sections from scratch, customize each **<productCode>** section, and activate it.

Commented, or *deactivated*, sections are ignored. Uncommented, or *activated*, sections for applications that are not installed are ignored. For agents that do not require an instance value, specify only:

```
<productCode>product_code</productCode>
  <instance>
   <user>USER</user>
  </instance>
```

For agents that do require an instance value, like the DB2 monitoring agent (product code ud), specify the *product_code*, *instance*, *user*, and *name*:

```
  <productCode>ud</productCode>
  <instance>
   <name>db2inst1</name>
   <user>db2inst1</user>
  </instance>
  <instance>
   <name>db2inst2</name>
   <user>root</user>
  </instance>
```

Two items that are supported in the `kcirunas.cfg` file that are not supported in the *HOST*`_kdyrunas.cfg` file are the **<default>** section and the **<autoStart>** flag. The **<autoStart>** flag can be used in the **<default>** section and in the **<instance>** section. The **<default>** section is specified as follows:

```
<productCode>product_code</productCode>
<default>
 <user>db2inst1</user>
</default>

<productCode>product_code</productCode>
<default>
 <autoStart>no</autoStart>
</default>

<productCode>product_code</productCode>
<default>
 <user>db2inst1</user>
 <autoStart>no</autoStart>
</default>
```

The **<autoStart>** flag is specified as follows:

```
<productCode>product_code</productCode>
<default>
 <autoStart>no</autoStart>
</default>

<productCode>product_code</productCode>
<instance>
 <autoStart>no</autoStart>
</instance>
```

A section similar to the following can be used to not automatically start the default MQ Monitoring instance and to automatically start all other instances as the mqm user:

```
<productCode>mq</productCode>
<default>
 <user>mqm</user>
</default>
<instance>
 <name>None</name>
 <autoStart>no</autoStart>
</instance>
```

A set of sections similar to the following can be used to avoid automatically starting the set of installed agents and servers. You need one section for each agent or server component installed:

```
<productCode>product_code</productCode>
<default>
 <autoStart>no</autoStart>
</default>
```

Where *product_code* is the two-character product code for the individual agent or server component (See Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815).

Notes:

1. Any changes made directly to the autostart script (`ITMAgentsN` or `rc.itmN`, depending on the platform) will not be preserved and will be overwritten the next time you install, configure, or upgrade an application.
2. Any changes made to the `AutoRun.sh` script will not be preserved and will be overwritten the next time you apply higher maintenance.

Create an IBM Tivoli account for installing and maintaining the installation directory

Create an IBM Tivoli account for installing and maintaining the installation directory. For best performance, follow these guidelines:

Note: This option does not apply to configuring the portal server on Linux systems where you may use a non-root IBM Tivoli Monitoring user ID to install the portal server. If you do, you must then use the root user ID to configure the portal server because the DB2 installation or administrator ID may lack the necessary privileges.

If, however, you use either the root user ID, the DB2 installation ID, or the DB2 administrator ID to install the portal server, you must use that same user ID to configure it. You can then use your non-root Tivoli Monitoring user ID to run the portal server.

- You can use any valid name.

  You can install the IBM Tivoli Monitoring software as the root user, but you do not have to. If you do not install as a root user, you must follow the steps outlined in "Postinstallation steps for nonroot installations" on page 261 after you install any monitoring agents.
- Use the same user to install all components.
- If you are using NFS or a local file system, establish your installation directory according to the guidelines used in your environment.
- Only the Korn shell is supported for the execution of the installation and runtime scripts. Consider using the Korn shell as the default environment for your IBM Tivoli login account.

Host name for TCP/IP network services

TCP/IP network services such as NIS, DNS, and the /etc/hosts file must be configured to return the fully qualified host name (for example: HostName.ibm.com). Define the fully qualified host name after the dotted decimal host address value and before the short host name in the /etc/hosts.

On a machine that will be running the Tivoli Enterprise Portal Server, you should verify that the /etc/hosts file has only one line containing the short name of the host machine. Having multiple entries (that is different network cards) resolving to the same short name of the machine can result in the Tivoli Enterprise Portal Server generating an error when either starting, reconfiguring, or running the buildPresentation, InstallPresentation.sh, or migrate-export scripts.

Use of fully qualified path names

Because of the wide variety of UNIX operating systems and possible user environments, use fully qualified path names when entering a directory during the installation process (do not use pattern-matching characters). IBM scripts use the Korn shell. When a new process or shell is invoked, use of symbolic links, environmental variables, or aliases can potentially cause unexpected results.

Multiple network interface cards

When more than one network interface card (NIC) exists in the computer on which the monitoring server is installed, you need to identify which NIC to use when specifying the monitoring server name and host name. Additionally, the host name of the system might not match the interface name, even when only one NIC exists. In either of these cases, to establish connectivity between the monitoring server and agents, you must specify an additional variable when configuring the monitoring server or agents. This variable is listed under the **Optional Primary Network Name** option in the configuration windows or during the installation.

If the host of the Tivoli Enterprise Portal Server has more than one NIC, you need to configure an additional interface for each one.

Installing into an NFS environment

IBM supports installing IBM Tivoli Monitoring in NFS environments. Using NFS, you can concentrate your software and data in a specific location, minimizing maintenance, administrative overhead, and disk space.

Although using NFS to support multiple hosts simplifies the maintenance of installed IBM Tivoli products, its use can impact performance. If you are installing into an NFS environment, consider the administrative savings to the possible impact on the performance of your network.

Consider the number of hosts that share a single installation directory, as well as the effects of network congestion and file system performance on the overall response time of your IBM Tivoli products.

NFS also has some trade-offs in how you manage your environment. While you can have your entire IBM Tivoli Monitoring in one place, there might be additional configuration required to define the use of specific products or processes in your installation directory. Will every product on every host system execute using the same configuration; or will you tailor the configuration to the particular environment?

Note: If installing from images on an NFS mount, the NFS mount needs **world** execute permissions to be accessible by the installation processes.

Installing into Solaris zones

There are limitations that must be taken into consideration when installing into zones that share resources.

Big local zones share no resources with other zones, so there are no limitations on installing into a Big Local zone. For local zones that share resources, there are the following limitations:

- Anything installed outside of $CandleHome must:
  - Be installed in the global zone, OR
  - Grant local zone write access to the global zone directory tree
- Any devices used by an agent to obtain statistics must be linked to the Local zone.
- During installation of the Tivoli Enterprise Monitoring Server, the GSKit library can be installed from an alternate directory within the local zone. To accomplish this, edit the ms.ini file for the monitoring server, and add the extra library path to the LD_LIBRARY_PATH concatenation.

  GSKit, the Global Security Toolkit, provides SSL (Secure Sockets Layer) processing within protocols such as SPIPE and HTTPS.

The following sections describe the types of zones and the installation possibilities for each.

Global Zone:

The main administration zone. Both local and remote installation is possible. GSKit installs into the standard location for Solaris, and the links are created in the standard location for Solaris.

Big Local Zone:

A local zone with no shared file system resources. Local and remote installation is possible. GSKit installs into the standard location for Solaris, and the links are created in the standard location for Solaris. These locations are local to this zone and not shared with any other zone.

Small Local Zone:

A local zone with some shared file system resources. Local and remote installation is possible if /opt is not a shared resource. GSKit installs into the standard location for Solaris, but the links are created in the new default location of $CANDLEHOME/gsklinks. These locations are local to this zone and not shared with any other zone.

Default Small Local Zone:

A local zone with the default set of shared file system resources. This is a Small Local Zone with /sbin /lib /usr and /export directories shared with global zone. These directories are read-only file systems in local zone. Local and remote installation is possible. GSKit installs into the standard location for Solaris, but the links are created in the new default location of $CANDLEHOME/gsklinks. These locations are local to this zone and not shared with any other zone.

Small Local Zone with /opt directory shared:

Local and remote installation is not possible. Tivoli Monitoring installation always requires read-write access to the /opt directory. This is not only a GSKit issue. Even if CANDLEHOME is specified as the nondefault directory, read-write access to /opt/IBM/ITM/tmaitm6/links is still needed.

Note: In all supported Small Local Zone configurations, the Tivoli monitoring interactive command-line installation prompts you for the parent directory where the links will be created. For example, if you enter /tmp for the directory, the links will be created in the /tmp/usr/lib and /tmp/usr/bin directories. The default directory for this prompt is $CANDLEHOME/gsklinks. During remote installation, the default directory is always used.

It is very difficult to predict all the possible shared-resource policies for small local zones and the possible side effects. It is the responsibility of the system administrator to create these policies without causing unintentional side effects between different zones and software installed.

Architecture and product codes

On UNIX and Linux operating systems, some IBM Tivoli Monitoring files are placed in directories whose paths include a code for the platform architecture. You can find the codes for the supported architectures in the registry directory. **archdsc.tbl** contains the architecture codes and descriptions. You can find the product codes and descriptions for supported components and products in the same directory, in **proddsc.tbl**.

File descriptor (maxfiles) limit on UNIX and Linux systems

The monitoring server and Warehouse Proxy Agent can use a large number of file descriptors, especially in a large environment. On UNIX and Linux systems, the maximum number of file descriptors available to a process is controlled by user limit parameters. To display the user limits, run the following command:

```
ulimit -a
```

The "nofiles" parameter is the number of file descriptors available to a process. For the monitoring server process (kdsmain), the "nofiles" parameter should be set larger than the maximum number of agents that will be connecting to the monitoring server. If the monitoring server is unable to get file descriptors when needed, unexpected behavior can occur, including program failures. Consider increasing the value to 8000 file descriptors or more.

There are other user limit parameters that control how much data, stack and memory are available to a process. For large environments, consider increasing these memory-related user limit parameters for the monitoring server (kdsmain) process. Configuring the user limit parameters usually requires root access, and involves changing system startup files which are operating system specific. Consult the operating system manuals for information on how to configure the user limit parameters.

Installing into an existing installation

Installing components or agents in an existing CANDLEHOME or installation directory is supported as long as the user ID used to run the installation is always the same. Installing components or agents in an existing CANDLEHOME or installation directory using different user IDs is not supported.

Security options

User IDs and passwords sent between Tivoli Management Services components are encrypted by default. Other communication between components can be secured by configuring the components to use secure protocols. See “Communication between components.”

Access to the Tivoli Enterprise Portal (*authorization*) is controlled by user accounts (IDs) defined to the Tivoli Enterprise Portal Server. The hub Tivoli Enterprise Monitoring Server can be configured to validate, or *authenticate*, user IDs through either the local system registry or an external LDAP-enabled registry. Alternatively, authentication by an external LDAP registry can be configured through the Tivoli Enterprise Portal Server. If authentication is not configured through either the monitoring server or the portal server, no password is required to log on to the Tivoli Enterprise Portal. See “Authorization and authentication” on page 137.

User IDs that require access to the SOAP Server, including user IDs that issue commands that invoke SOAP methods, must be authenticated through the hub monitoring server. If user authentication is *not* enabled on the hub monitoring server, anyone can make requests to the SOAP Server. If user authentication *is* enabled on the hub, the SOAP Server honors only requests from user IDs and passwords authenticated by the local or external registry. If type of access is specified for specific users, only requests from those users for which access is specified are honored. See “SOAP server security” on page 138.

User IDs that require the ability to share credentials with other Web-enabled Tivoli applications must be authenticated through the Tivoli Enterprise Portal Server, which must be configured for single sign-on. See “Single sign-on capability” on page 137. If you have previously enabled authentication through the hub monitoring server and want to change to the portal server, see the *IBM Tivoli Monitoring: Administrator's Guide*.

A binary '0' is automatically added as the final character in the 16 byte user password field. There is a limitation of 15 characters or fewer on the remainder of the password.

Notes:

1. The Tivoli Directory Server (TDS) LDAP client used by the Tivoli Enterprise Monitoring Server does not support LDAP referrals, such as those supported by Microsoft Active Directory.
2. The IBM Tivoli Monitoring Service Console enables you to read logs and turn on traces for remote product diagnostics and configuration. The Service Console performs user authentication using the native operating system security facility. This means that if you use the Service Console on z/OS, your user ID and password are checked by the z/OS security facility (such as RACF/SAF). If you use the Service Console on Windows, you must pass the Windows workstation user ID and password prompt. A password is always required to access the Service Console. Even if a user ID is allowed to log into the operating system without a password, access to the Service Console will be denied. If necessary, you must create a password for the user ID that is being used to log in to the Service Console. For more information about the Service Console, see the *IBM Tivoli Monitoring: Troubleshooting Guide*.

Communication between components

To secure communication between agents, monitoring servers, and the Tivoli Enterprise Portal, use SPIPE as the protocol when you configure communications between the Tivoli Enterprise Portal Server and the hub Tivoli Enterprise Monitoring Server, between hub and remote monitoring servers, and between agents and monitoring servers.

Two additional protocols are used to secure communication between clients and the portal server:

- Secure Hypertext Transport Protocol (HTTPS) to retrieve files and Interoperable Object Reference (IOR). The integrated browser in the client provides HTTPS support on the client side; for the server, consider using a third party Web server that supports HTTPS, such as the IBM HTTP Server. See "Configuring an external Web server to work with Tivoli Enterprise Portal" on page 400 for more information.
- Internet Inter-ORB Protocol (IIOP) to secure the communications between the portal server and client. This uses Secure Sockets Layer (SSL) provided by VisiBroker. This secure communication uses public key cryptography. See "Using SSL between the portal server and the client" on page 398 for more information.

Authorization and authentication

Access to the Tivoli Enterprise Portal is controlled by user accounts defined to the portal server. In addition to defining the user IDs that are authorized to log on to the Tivoli Enterprise Portal, these accounts define the permissions that determine the Tivoli Enterprise Portal features a user is authorized to see and use, the monitored applications the user is authorized to see, and the Navigator views (and the highest level within a view) the user can access. An initial **sysadmin** user ID with full administrator authority is provided during installation so you can log in to the Tivoli Enterprise Portal and add more user accounts. (For information on creating user accounts and setting user permissions, see the *IBM Tivoli Monitoring: Administrator's Guide*.) No password is required to log on to the Tivoli Enterprise Portal, unless user authentication is enabled.

User authentication may be enabled through either the hub Tivoli Enterprise Monitoring Server, or the Tivoli Enterprise Portal Server.

If authentication is enabled through the hub monitoring server, user IDs can be authenticated either by the local system registry or by an external LDAP-enabled central registry. User IDs that need to make SOAP Server requests (including user IDs that issue CLI commands that invoke SOAP server methods) can be authenticated only through the hub monitoring server.

If authentication is enabled through the Tivoli Enterprise Portal, user IDs are authenticated against an external LDAP-enabled registry. User IDs that require single sign-on (SSO) capability must be authenticated through the portal server and mapped to unique user identifiers in an LDAP registry shared by all SSO-eligible Tivoli applications.

User authentication should not be enabled until at least a basic installation of Tivoli Management Services components and IBM Tivoli Monitoring base agents has been completed and tested. For instructions on enabling authentication, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Single sign-on capability

The Tivoli Enterprise Portal single sign-on (SSO) feature provides users the ability to launch out of the portal to other Tivoli Web-based or Web-enabled applications, or to launch into the portal from those applications, without having to re-enter their user IDs and passwords. For SSO to be enabled, authentication must be configured through the Tivoli Enterprise Portal Server and the LDAP registry defined to the portal server must be a central registry shared by all participating Tivoli applications. All the participating applications must be configured for SSO and must belong to the same security domain and realm.

For instructions on using SSO, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Note: A binary '0' is automatically added as the final character in the 16 byte user password field. There is a limitation of 15 characters or fewer on the remainder of the password.

SOAP server security

You can control access to the SOAP server in two ways:

- You can control *who* is permitted to make requests by enabling user authentication on the hub monitoring server.

  If the **Security: Validate User** option is *not* enabled, the SOAP server honors all requests regardless of the sender. If the **Security: Validate User** option on the hub monitoring server is enabled, the SOAP server honors requests only from users defined to the operating system or security authorization facility of the host of the monitoring server.
- You can control *what type of requests* users are permitted to make by configuring the SOAP server.

  If you specify a specific type of access for any users, the SOAP server honors requests only from those users, regardless of whether Security: Validate User is enabled.

For information on configuring the security on the SOAP server, see Chapter 16, "Configuring IBM Tivoli Monitoring Web Services (the SOAP Server)," on page 415.

Global Security Toolkit

IBM Tivoli Monitoring includes the Global Security Toolkit (GSKit) for SSL processing as used in SPIPE and HTTPS. GSKit is installed by default on all distributed components, and its utilities are used to create and manage the encryption of data between components through the use of digital certificates.

Note: Do not uninstall or manipulate the GSKit during installation. This may cause functional regression in other products or make them inoperable. The GSKit will automatically install the most recent build if another version GSKit already exists.

On z/OS, GSKit is known as the Integrated Cryptographic Service Facility, or ICSF. If ICSF is not installed on the z/OS system, the monitoring server uses an alternative, less secure encryption scheme. Since both components must be using the same scheme, if the hub system does not use ICSF, you must configure the Tivoli Enterprise Portal to use the less secure scheme (EGG1) as well. For more information, see *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

A default certificate and key are provided with GSKit at installation. A stash file provides the database password for unattended operation. You can also use the key management facilities in GSKit to generate your own certificates. For more information regarding GSKit and iKeyMan, including information about creating and using security certificates, see the GSKit product documentation located at http://www-128.ibm.com/developerworks/java/jdk/security/50/.

Notes:

1. The IBM Tivoli Monitoring installer no longer modifies the system GSKit. If necessary, it installs a local copy of GSKit that is private to Tivoli Monitoring.
2. In 64-bit environments, both the 64-bit and the 32-bit GSKit are installed, to support both 64-bit and 32-bit Tivoli Monitoring components.

Hardware and software requirements

The following sections provide specific information about the memory, software, and hardware requirements for installing IBM Tivoli Monitoring.

Note: This section does not show agent-specific requirements (such as supported application levels or any hardware requirements unique to a certain agent). For this information, see the user's guide (for agents on distributed system) or configuration guide (for agents on mainframe systems) for the agent that you are installing.

Supported operating systems

The following tables show which operating systems are supported for the different IBM Tivoli Monitoring components: monitoring server, portal server, portal client, monitoring agent, Warehouse Proxy, Warehouse Proxy Summarization and Pruning Agent, and Tivoli Performance Analyzer.

For the latest information about the supported operating systems, see http://www-306.ibm.com/software/sysmgmt/products/support/Tivoli_Supported_Platforms.html.

Note: In the following tables, an X indicates a supported component on the platform. If the platform being discussed supports 64-bit applications but only a 32-bit version of the Tivoli Monitoring component is supported on that platform, the X is replaced by "32 bit."

Table 17 shows the support for monitoring components on Windows computers.

Table 17. Supported Windows operating systems

Operating system	Monitoring server	Portal server [1]	Portal client[2]	OS monitoring agent[3]	Warehouse Proxy Agent	Summarization and Pruning Agent	Tivoli Performance Analyzer
Windows XP Professional on Intel x86-32 (32 bit)			X	X	X	X	X
Windows XP Professional with FDCC on Intel x86-32 (32 bit)			X	X	X	X	X
Windows Server 2003 Standard Edition R2 on Intel x86-32 (32 bit)	X	X	X	X	X	X	X
Windows Server 2003 Standard Edition R2 on x86-64 (64 bit)	32 bit	32 bit	32 bit	X	X	32 bit	32 bit
Windows Server 2003 Standard Edition R2 on Itanium IA64 (64 bit)				32 bit			
Windows Server 2003 Enterprise Edition R2 on Intel x86-32 (32 bit)	X	X	X	X	X	X	X
Windows Server 2003 Enterprise Edition R2 on x86-64 (64 bit)	32 bit	32 bit	32 bit	X	X	32 bit	32 bit
Windows Server 2003 Enterprise Edition R2 on Itanium IA64 (64 bit)				32 bit			
Windows Server 2003 Datacenter Edition R2 on Intel x86-32 (32 bit)				X			
Windows Server 2003 Datacenter Edition R2 on Intel x86-64 (64 bit)				X			
Windows Server 2003 Datacenter Edition R2 on Itanium IA64 (64 bit)				32 bit			

Table 17. Supported Windows operating systems (continued)

Operating system	Monitoring server	Portal server [1]	Portal client[2]	OS monitoring agent[3]	Warehouse Proxy Agent	Summarization and Pruning Agent	Tivoli Performance Analyzer
Windows Vista Enterprise Edition on Intel x86-32 (32 bit)			X	X			
Windows Vista Enterprise Edition on Intel x86-64 (64 bit)			32 bit	X			
Windows Server 2008 Standard Edition on Intel x86-32 (32 bit)[2,4,]	X	X	X	X	X	X	X
Windows Server 2008 Standard Edition on Intel x86-64 (64 bit)[2]	32 bit	32 bit	32 bit	X	X	32 bit	32 bit
Windows Server 2008 Standard Edition on Itanium IA64 (64 bit)[2,]				32 bit			
Windows Server 2008 Enterprise Edition on Intel x86-32 (32 bit)[2,4,]	X	X	X	X	X	X	X
Windows Server 2008 Enterprise Edition on Intel x86-64 (64 bit)[2,]	32 bit	32 bit	32 bit	X	X	32 bit	32 bit
Windows Server 2008 Enterprise Edition on Itanium2 (IA64)				32 bit			
Windows Server 2008 Datacenter Edition on Intel x86-32 (32 bit)[2]				X			
Windows Server 2008 Datacenter Edition on Intel x86-64 (64 bit)[2]				X			
Windows Server 2008 Datacenter Edition on Itanium IA64 (64 bit)[2]				32 bit			
Windows 7 Enterprise Edition on Intel x86-32 (32 bit)			X	X			
Windows 7 Enterprise Edition on Intel x86-64 (64 bit)			32 bit	X			
Windows 7 Professional Edition on Intel x86-32 (32 bit)			X	X			
Windows 7 Professional Edition on Intel x86-64 (64 bit)			32 bit	X			
Windows 7 Ultimate Edition on Intel x86-32 (32 bit)			X	X			

Table 17. Supported Windows operating systems (continued)

Operating system	Monitoring server	Portal server[1]	Portal client[2]	OS monitoring agent[3]	Warehouse Proxy Agent	Summarization and Pruning Agent	Tivoli Performance Analyzer
Windows 7 Ultimate Edition on Intel x86-64 (64 bit)			32 bit	X			
Windows Server 2008 R2 Standard Edition on Intel x86-64 (64 bit)[2,]	32 bit	32 bit	32 bit	X	X	32 bit	32 bit
Windows Server 2008 R2 Standard Edition on Itanium IA64 (64 bit)[2,]				32 bit			
Windows Server 2008 R2 Enterprise Edition on Intel x86-64 (64 bit)[2,]	32 bit	32 bit	32 bit	X	X	32 bit	32 bit
Windows Server 2008 R2 Enterprise Edition on Itanium2 (IA64)				32 bit			
Windows Server 2008 R2 Datacenter Edition on Intel x86-64 (64 bit)[2]				X			
Windows Server 2008 R2 Datacenter Edition on Itanium IA64 (64 bit)[2]				32 bit			
Windows Server 2008 R2 Server Core on Intel x86-64 (64 bit)				X			

Notes:

1. The Tivoli Enterprise Portal server runs an instance of embedded WebSphere® Application Server, so the system must also meet the minimum requirements for WebSphere. For information on the minimum service pack level required to install and run the embedded WAS in Tivoli Enterprise Portal Server, see http://www.ibm.com/support/docview.wss?rs=180&uid=swg27012421.
2. The Tivoli Enterprise Portal desktop client is supported on marked platforms. However, the browser client can be accessed only from Windows, Linux, and AIX computers running the browser versions documented in Table 25 on page 156.
3. The **OS monitoring agent** column indicates the platforms on which an operating system monitoring agent is supported. This column does not indicate that any agent runs on any operating system. For example, to monitor a Linux computer, you must use a Linux monitoring agent, not a Windows monitoring agent.

   For information about the operating systems supported for non-OS agents, see the documentation for the specific agents you are using in your environment.
4. Windows Server 2008 (32-bit) is supported for these Tivoli Management Services components:
   - The monitoring server
   - The portal server
   - The agent infrastructure and the OS agent
   - The portal desktop client and the browser client under Internet Explorer. (Mozilla Firefox is not supported.)
   - The Warehouse Proxy Agent
   - The Tivoli Performance Analyzer

Table 18 shows the support for monitoring components on UNIX (non-Linux), i5/OS, and z/OS computers.

Table 18. Supported UNIX, i5/OS, and z/OS operating systems

Operating system	Monitoring server	Portal server	Portal client	OS monitoring agent[1,2]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer
AIX V5.3 (32 bit)[4]	X	X		X	X	X	X
AIX V5.3 (64 bit)[4]	X	X		X	X	X	32 bit
AIX V6.1 (64 bit)[5]	X	X		X	X	X	32 bit
AIX V7.1 (64 bit)[5]	X	X		X	X	X	32 bit
Solaris V8 (SPARC) (32/64bit)[6]				X			
Solaris V9 (SPARC) (32/64bit)[7]	32 bit			X	32 bit	32 bit	32 bit
Solaris V10 (SPARC) (32/64 bit)	32 bit			X	32 bit	32 bit	32 bit
Solaris V10 (Intel x86-64) (64 bit)	32 bit			X			
Solaris Zones (SPARC) (32/64 bit)[8]	32 bit			X	32 bit	32 bit	32 bit
Solaris Zones (Intel x86-64) (64 bit)[8]	32 bit			X			
HP-UX 11i v1 (32/64) on PA-RISC[13,14]				X			
HP-UX 11i v2 (64 bit) on PA-RISC[14]				X			
HP-UX 11i v3 (64 bit) on PA-RISC[14]				X			
HP-UX 11i v2 on Integrity (IA64)[12,14]	32 bit			X	32 bit	32 bit	
HP-UX 11i v3 on Integrity (IA64)[12,14]	32 bit			X	32 bit	32 bit	
i5/OS 5.4 (64 bit)[9]				X			
IBM i 6.1 (64 bit)[9]				X			
IBM i 7.1 (64 bit)[9]				X			
z/OS 1.10 (31/64 bit)[10,11]	31 bit			X			
z/OS 1.11 (31/64 bit)[10,11]	31 bit			X			
z/OS 1.12 (31/64 bit)[10,11]	31 bit			X			

Table 18. Supported UNIX, i5/OS, and z/OS operating systems (continued)

Operating system	Monitoring server	Portal server	Portal client	OS monitoring agent[1,2]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer

Note:

1. The **OS monitoring agent** column indicates the platforms on which an operating system monitoring agent is supported. This column does not indicate that any agent runs on any operating system. For example, to monitor a Linux computer, you must use a Linux monitoring agent, not a Windows monitoring agent.

   For information about the operating systems supported for non-OS agents, see the documentation for the specific agents you are using in your environment.
2. If you are installing the OMEGAMON XE for Messaging agent on a 64-bit operating system, you must install the 32-bit version of the agent framework.
3. Configuration of the Warehouse Proxy Agent requires an X Window System (also known as the X11 GUI) on the computer where you are configuring it. Alternatively, you can run the following command to use an X terminal emulation program (such as Cygwin) that is running on another computer:

   ```
   export DISPLAY=my_windows_pc_IP_addr:0.0
   ```

   where *my_windows_pc_IP_addr* is the IP address of a computer that is running an X terminal emulation program.
4. Supported AIX systems must be at the required maintenance level for IBM Java 1.5. See the following Web site for the Java 5 AIX maintenance level matrix: http://www-128.ibm.com/developerworks/java/jdk/aix/service.html

   Component `xlC.aix50.rte` must be at level 8.0.0.4 (or higher). See the following Web site for installation instructions: http://www-1.ibm.com/support/docview.wss?uid=swg1IY84212

   The Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server require AIX 5.3 TL5 SP3 or newer. The other components need AIX 5.3 TL3 or newer, but if they are at AIX 5.3 TL5, they too require SP3.

   Version 8 of the AIX XL C/C++ runtime must be installed. To determine the current level, run the following AIX command:

   ```
   lslpp -l | grep -i xlc
   ```

Table 18. Supported UNIX, i5/OS, and z/OS operating systems (continued)

Operating system	Monitoring server	Portal server	Portal client	OS monitoring agent[1,2]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer

5. Component `xlC.aix61.rte` must be at level 10.1.0.2 or higher. See the following Web site for installation instructions: http://www-1.ibm.com/support/docview.wss?uid=swg1IY84212

   Version 10 of the AIX XL C/C++ runtime must be installed. Additionally on AIX 6.1, the xlC supplemental runtime for aix50 (`xlC.sup.aix50.rte` at level 9.0.0.1 or higher) must be installed. To determine the current level, run the following AIX command:

   ```
   lslpp -l | grep -i xlc
   ```

   The output should be something similar to the following:

   ```
   xlC.aix61.rte    10.1.0.2
   xlC.cpp       9.0.0.0
   xlC.msg.en_US.cpp   9.0.0.0
   xlC.msg.en_US.rte   10.1.0.2
   xlC.rte       10.1.0.2
   xlC.sup.aix50.rte   9.0.0.1
   ```
6. Solaris V8 32 bit requires patches 108434-17 and 109147-07. Solaris V8 64 bit requires 108435-17 and 108434-17. Both 32-bit and 64-bit versions require 111721-04.
7. Solaris V9 32 bit requires patch 111711-11. Solaris V9 64 bit requires 111712-11 and 111711-11. Both 32-bit and 64-bit versions require 111722-04.
8. There are some limitations on installing into Solaris 10 when zones are configured. See “Installing into Solaris zones” on page 134.
9. SNMP version 3 is not supported on i5/OS.
10. For information about installing the monitoring server on z/OS, see the program directory that comes with that product.
11. The OS monitoring agent for z/OS computers is part of the IBM Tivoli OMEGAMON for z/OS product.
12. You cannot upgrade either the OS or Log Alert agents that you currently have running on an HP-UX 11i v2 (B.11.23) on an Integrity (IA64) computer in PA-RISC mode prior to v6.1 fix pack 4. Fix packs prior to fix pack 4 did not run in native 64-bit mode by default. You must first uninstall the agent if the version is previous to the fix pack 4 version.
13. The 32-bit kernel still requires a 64-bit processor. Ensure that any HP-UX managed system is based on PA-RISC2 architecture. From the native kernel mode (for example, 64 bit if the system is 64-bit based), run the following command:

    ```
    file /stand/vmunix
    ```

    This returns the native architecture type. For example:

    ```
    /stand/vmunix:  PA-RISC1.1 executable -not stripped
    ```

    Verify the architecture is at least PA-RISC2.
14. Prerequisite patches for HP-UX R11: PHSS_26946, PHSS_33033, and PHCO_34275 or later.
15. Desktop client deployed using Java Web Start 32-bit mode only.
16. The Tivoli Enterprise Portal server runs an instance of embedded WebSphere Application Server, so users of AIX 5.3 (32 bit or 64 bit) or AIX 6.1 must also meet the minimum requirements for WebSphere. For information on the minimum service pack level required to install and run the embedded WAS in Tivoli Enterprise Portal Server, see http://www.ibm.com/support/docview.wss?rs=180&uid=swg27012389.

Table 19 shows the monitoring components supported on Linux operating systems.

Table 19. Supported Linux operating systems

Operating system	Monitoring server	Portal server	Portal client[1]	OS monitoring agent[2,5]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer
Asianux 2.0 for Intel x86-32 (32 bit) [9]	X	X	X	X	X	X	
Asianux 2.0 on Intel x86-64 (64 bit) [10]	32 bit	32 bit	browser client only	X	32 bit	32 bit	
Asianux 2.0 on Itanium IA64 (64 bit)			browser client only	X			
Asianux 3.0 for Intel x86-32 (32 bit) [9]	X	X	X	X	X	X	
Asianux 3.0 on Intel x86-64 (64 bit) [10]	32 bit	32 bit	browser client only	X	32 bit	32 bit	
Asianux 3.0 on Itanium IA64 (64 bit)			browser client only	X			
Red Flag 4.1 for Intel x86-32 (32 bit) [9]	X	X	X	X	X	X	
Red Flag 5.0 for Intel x86-32 (32 bit) [9]	X	X	X	X	X	X	
RedHat Enterprise Linux 4 Intel x86-32 (32 bit) [9]	X	X	X	X	X	X	X
RedHat Enterprise Linux 4 on Intel x86-64 (64 bit) [10]	32 bit	32 bit [12]	browser client only	X	32 bit	32 bit	32 bit
RedHat Enterprise Linux 4 on Itanium IA64 (64 bit)[6]			browser client only	X			
RedHat Enterprise Linux 4 on iSeries and pSeries (64 bit)				X			
RedHat Enterprise Linux 4 on zSeries (31 bit) [11]	X	X	browser client only	X	X	X	
RedHat Enterprise Linux 4 on zSeries (64 bit) [11]	X	X[8]	browser client only	X	X	X	
RedHat Enterprise Linux 5 Intel x86-32 (32 bit)[7]	X	X	X	X	X	X	X
RedHat Enterprise Linux 5 on Intel x86-64 (64 bit)[7] [10]	32 bit	32 bit [12]	browser client only	X	32 bit	32 bit	32 bit

Table 19. Supported Linux operating systems (continued)

Operating system	Monitoring server	Portal server	Portal client[1]	OS monitoring agent[2,5]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer
RedHat Enterprise Linux 5 on Itanium IA64 (64 bit)[6]			browser client only	X			
RedHat Enterprise Linux 5 on iSeries and pSeries (64 bit)[7]				X			
RedHat Enterprise Linux 5 on zSeries (31 bit)[7] [11]	X	X	browser client only	X	X	X	
RedHat Enterprise Linux 5 on zSeries (64 bit)[7] [11]	X	X[8]	browser client only	X	X	X	
RedHat Enterprise Linux 6 Intel x86-32 (32 bit)[7] [9]	X	X	X	X	X	X	X
RedHat Enterprise Linux 6 on Intel x86-64 (64 bit)[7] [10]	32 bit	32 bit [12]	browser client only	X	32 bit	32 bit	32 bit
RedHat Enterprise Linux 6 on iSeries and pSeries (64 bit)[7]				X			
RedHat Enterprise Linux 6 on zSeries (31 bit)[7] [11]	X	X	browser client only	X	X	X	
RedHat Enterprise Linux 6 on zSeries (64 bit)[7] [11]	X	X[8]	browser client only	X	X	X	
SuSE Linux Enterprise Server 9 Intel x86-32 (32 bit)[4] [9]	X	X	X	X	X	X	
SuSE Linux Enterprise Server 9 on Intel x86-64 (64 bit)[4] [10]	32 bit	32 bit [12]	browser client only	X	32 bit	32 bit	
SuSE Linux Enterprise Server 9 on Itanium IA64 (64 bit)[4,6]			browser client only	X			
SuSE Linux Enterprise Server 9 for iSeries and pSeries (64 bit)[4]				X			
SuSE Linux Enterprise Server 9 for zSeries (31 bit)[4] [11]	X	X	browser client only	X	X	X	
SuSE Linux Enterprise Server 9 for zSeries (64 bit)[4] [11]	X	X[8]	browser client only	X	X	X	

Table 19. Supported Linux operating systems (continued)

Operating system	Monitoring server	Portal server	Portal client[1]	OS monitoring agent[2,5]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer
SuSE Linux Enterprise Server 10 Intel x86-32 (32 bit)[4] [9]	X	X	X	X	X	X	
SuSE Linux Enterprise Server 10 on Intel x86-64 (64 bit) [4] [10]	32 bit	32 bit [12]	browser client only	X	32 bit	32 bit	
SuSE Linux Enterprise Server 10 on Itanium IA64 (64 bit)[4,6]			browser client only	X			
SuSE Linux Enterprise Server 10 for iSeries and pSeries (64 bit)[4]				X			
SuSE Linux Enterprise Server 10 for zSeries (64 bit)[4] [11]	X	X[8]	browser client only	X	X	X	
SuSE Linux Enterprise Server 11 on Intel x86-32 (32 bit)[4] [9]	X	X	X	X	X	X	
SuSE Linux Enterprise Server 11 on Intel x86-64 (64 bit) [46] [10]	32 bit	32 bit [12]	browser client only	X	32 bit	32 bit	
SuSE Linux Enterprise Server 11 on Itanium IA64 (64 bit)[4,6]			browser client only	X			
SuSE Linux Enterprise Server 11 for iSeries and pSeries (64 bit)[4]				X			
SuSE Linux Enterprise Server 11 for zSeries (64 bit)[4] [11]	X	X[8]	browser client only	X	X	X	
VMWare ESX Server 3.0.1 on Intel x86-32 (32 bit)				native Linux OS agent			
VMWare ESX Server 3.0.1 on Intel x86-64 (64 bit)				native Linux OS agent			
VMWare ESX Server 3.5 on Intel x86-32 (32 bit)				native Linux OS agent			

Table 19. Supported Linux operating systems (continued)

Operating system	Monitoring server	Portal server	Portal client[1]	OS monitoring agent[2,5]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer
VMWare ESX Server 3.5 on Intel x86-64 (64 bit)				native Linux OS agent			
VMWare ESX Server 4.0 on Intel x86-32 (32 bit)				native Linux OS agent			
VMWare ESX Server 4.0 on Intel x86-64 (64 bit)				native Linux OS agent			

Notes®:

1. Both the Tivoli Enterprise Portal desktop and browser clients are supported on the marked platforms. The browser client may work with Firefox on other Linux platforms that have not yet been certified by IBM. See Table 25 on page 156 for the list of supported Firefox versions.
2. The **OS monitoring agent** column indicates the platforms on which an agent is supported. This column does not indicate that any agent runs on any operating system. For example, to monitor a Linux computer, you must use a Linux monitoring agent, not a Windows monitoring agent. An "X" symbol in the column indicates that an operating system agent is available for the specific operating system that is named in the row where the "X" is located.
3. Configuration of the Warehouse Proxy Agent requires an X Window System (also known as the X11 GUI) on the computer where you are configuring it. Alternatively, you can run the following command to utilize an X terminal emulation program (such as Cygwin) that is running on another computer: `export DISPLAY=`*`my_windows_pc_IP_addr`*`:0.0` where *my_windows_pc_IP_addr* is the IP address of a computer that is running an X terminal emulation program.
4. SuSE Linux Enterprise Server 9 must be at SP3 or higher. SuSE 10 must be at pdksh-5.2.14 or higher.
5. The Linux OS Monitoring Agent requires the installation of the latest versions of the following libraries:

   ```
   libstdc++
   libgcc
   compat-libstdc++
   libXp
   ```

   These libraries are available on the Linux operating system installation media and Service Packs. Each library can have multiple packages, and each should be installed.

Table 19. Supported Linux operating systems (continued)

Operating system	Monitoring server	Portal server	Portal client[1]	OS monitoring agent[2,5]	Warehouse Proxy Agent[3]	Summarization and Pruning Agent	Tivoli Performance Analyzer

6. The following rpm files are prerequisites when running IBM Tivoli Monitoring V6.2.3 under Linux on Itanium:
   ia32el-1.1-20.ia64.rpm
   glibc-2.3.4-2.36.i686.rpm
   glibc-common-2.3.4-2.36.i386.rpm
   libgcc-3.4.6-8.i386.rpm
7. RedHat Enterprise Linux V5 now enables SELinux (security-enhanced Linux) by default, which interferes with the installation, configuration, and operation of IBM Tivoli Monitoring. To ensure the proper operation of the V6.1.*x* and V6.2.*x* releases, the SELinux setting must be either disabled, or changed from `enforcing` mode to `permissive` mode when enabled. When the `permissive` mode is chosen, the system log will contain entries regarding which Tivoli Monitoring binaries have triggered the SELinux security condition. However, under the `permissive` mode, these entries are for auditing purpose only, and will operate normally.
8. On zSeries systems running a 32-bit DB2 V9 instance under Linux, you must install a 32-bit Tivoli Enterprise Portal Server. On zSeries systems running a 64-bit DB2 V9 instance under Linux, you can install *either*:
   - A 64-bit Tivoli Enterprise Portal Server.
   - A 32-bit Tivoli Enterprise Portal Server if the 32-bit DB2 V9 client libraries are also installed.
9. The Tivoli Enterprise Portal server runs an instance of embedded WebSphere Application Server, so the system must also meet the minimum requirements for WebSphere. For information on the minimum service pack level required to install and run the embedded WAS in Tivoli Enterprise Portal Server, see http://www.ibm.com/support/docview.wss?rs=180&uid=swg27012414.
10. The Tivoli Enterprise Portal server runs an instance of embedded WebSphere Application Server, so the system must also meet the minimum requirements for WebSphere. For information on the minimum service pack level required to install and run the embedded WAS in Tivoli Enterprise Portal Server, see http://www.ibm.com/support/docview.wss?rs=180&uid=swg27012415.
11. The Tivoli Enterprise Portal server runs an instance of embedded WebSphere Application Server, so the system must also meet the minimum requirements for WebSphere. For information on the minimum service pack level required to install and run the embedded WAS in Tivoli Enterprise Portal Server, see http://www.ibm.com/support/docview.wss?rs=180&uid=swg27012416.
12. When installing as a nonroot user, 32 bit libpam is required for the Tivoli Enterprise Portal Server on a 64 bit Intel Linux installation.

The following table lists the operating system patches required for the IBM Global Security Toolkit (GSKit), which is used to provide security between monitoring components. GSKit is installed automatically when you install Tivoli Management Services components.

Table 20. Operating system requirements for IBM GSKit

Operating system	Patch required
Solaris V8	108434-14, 111327-05, 108991, 108993-31, 108528-29, 113648-03, 116602-01, 111317-05, 111023-03, 115827-01
Solaris V9	111711-08
Solaris V10	none
HP-UX V11i	PHSS_26946, PHSS_33033, PHCO_34275
AIX V5.x	xlC.aix50.rte.6.0.0.3 or later
AIX V6.x	xlC.aix61.rte 10.1.0.2 or later
Windows Server 2003	none
Red Hat Enterprise Linux 4 Intel	compat-gcc-32-c++-3.2.3-46.1.i386.rpm compat-gcc-32-3.2.3-46.1.i386.rpm compat-libstdc++-33-3.2.3-46.1.i386.rpm compat-libstdc++-296.2.96.i386 compat-libgcc-296.2.96.i386 compat-libstdc++-33.3.2.3.x86_64 compat-libstdc++-33.3.2.3.i386 libXpm-3.5.5-3
Red Hat Enterprise Linux 6 x86 (32 bit)	ksh-20100621-2.el6.i686.rpm compat-libstdc++-33-3.2.3-69.el6.i686.rpm
Red Hat Enterprise Linux 6 x86-64 (64 bit)	ksh-20091224-1.el6.x86_64.rpm compat-libstdc++-33-3.2.3-69.el6.i686.rpm compat-libstdc++-33-3.2.3-69.el6.x86_64.rpm glibc-2.12-1.7.el6.i686.rpm libgcc-4.4.4-13.el6.i686.rpm nss-softokn-freebl-3.12.7-1.1.el6.i686.rpm
SuSE Linux Enterprise Server 9 Intel	none

Supported databases for Tivoli Enterprise Portal Server and Tivoli Data Warehouse

The following tables show the supported databases for the portal server and the Tivoli Data Warehouse.

Table 21 shows the supported databases for the portal server. Note that the database and the portal server must be installed on the same computer.

Note: IBM Tivoli Monitoring V6.2.3 includes DB2 Workstation Server Edition V9.7 fix pack 4 for use with the portal server and the Tivoli Data Warehouse. IBM Tivoli Monitoring V6.2.1 and V6.2.2 includes DB2 Workstation Server Edition V9.5 for use with the portal server and the Tivoli Data Warehouse. (Version 6.2 and its fix packs provided a restricted-use version of DB2 Database for Linux, UNIX, and Windows V9.1.)

Table 21. Supported databases for the portal server

Portal server operating system	**Portal server database ("TEPS")[1],[2]**	
	IBM DB2 for Linux, UNIX, and Windows[3]	**MS SQL Server**
AIX	• V9.1 with fix pack 4 or higher • V9.5 and fix packs • V9.7 and fix packs • V9.8 and fix packs	
Linux[4]	• V9.1 and fix packs[note 8 for Table 19] • V9.5 and fix packs[note 8 for Table 19] • V9.7 and fix packs[note 8 for Table 19] • V9.8 and fix packs	
Windows	• V9.1 and fix packs • V9.5 and fix packs • V9.7 and fix packs • V9.8 and fix packs	• MS SQL Server 2000 SP3[5] • MS SQL Server 2005[6] • MS SQL Server 2008 • MS SQL Server 2008 R2

Notes:

1. "TEPS" is the default database name for the database used by the portal server.
2. Your portal server database must be located on the computer where the portal server is installed.
3. If, in your environment, you are using products whose licenses require you to collect software use information and report it to IBM using IBM Tivoli License Manager, you must ensure that use of this instance of IBM DB2 for Linux, UNIX, and Windows is not included in the report. To do this, create a Tivoli License Manager license, select a license type that does not involve reporting to IBM, and associate this instance of the product with it.
4. On Linux, the portal server database must be installed with the operating system language set to UTF-8.
5. IBM Tivoli Monitoring supports Microsoft SQL Server 2000 only if the data is limited to codepoints inside the Basic Multilingual Plane (range U+0000 to U+FFFF). This restriction does not apply to IBM DB2 for Linux, UNIX, and Windows.
6. This assumes the portal server runs on Windows. See Technote 1240452 for further information on support for this application.
7. The base installation CD includes, among its Tivoli Enterprise Portal Server files, a version of the embedded Derby database that you can use instead of either DB2 for Linux, UNIX, and Windows or SQL Server (but note the limitations listed in “Derby now supported as a portal server database” on page 22). This version of Derby is the one supported by and included with the version of eWAS required for the portal server.
8. If you transition from one supported database system to another for the Tivoli Enterprise Portal Server, your existing portal server data is not copied from the first system to the new one.
9. The Tivoli Enterprise Portal Server automatically prunes closed events out of the database after 24 hours, to prevent long-term database growth.

Table 22 shows the supported databases for the Tivoli Data Warehouse. Note that if you run the database for the Tivoli Enterprise Portal Server and the database for the warehouse in the same instance of IBM DB2 for Linux, UNIX, and Windows, you must follow the support requirements in Table 21 on page 151.

Table 22. Supported databases for the Tivoli Data Warehouse

Tivoli Data Warehouse database ("WAREHOUS")[1]			
IBM DB2 for Linux, UNIX, and Windows	**IBM DB2 on z/OS[6]**	**MS SQL Server**	**Oracle**
Supported versions: • V9.1 and fix packs • V9.5 and fix packs[5] • V9.7 and fix packs	Supported versions: version 9.1 or subsequent. Support applies to any Windows, Linux, or UNIX platform that can run the Warehouse Proxy Agent. DB2 Connect™ Server Edition is also required on the workstation.	Supported versions: • MS SQL Server 2000 Enterprise Edition[4] • MS SQL Server 2005 Enterprise Edition • MS SQL Server 2008 Enterprise Edition • MS SQL Server 2008 R2 Enterprise Edition	Supported versions: • 10g Release 1 • 10g Release 2 • 11g Release 1 • 11g Release 2

Notes:

1. "WAREHOUS" is the default database name for the database used by Tivoli Data Warehouse. Support is for 32-bit or 64-bit databases. Your Tivoli Data Warehouse database can be located on the same computer as your monitoring server or on a remote computer.
2. See the Oracle company support Web site (www.oracle.com) for information about installing and configuring Oracle on Solaris V10.
3. Do not use DB2 for Linux, UNIX, and Windows V9.1 fix pack 2 for the Tivoli Data Warehouse. Use of DB2 for Linux, UNIX, and Windows V9.1 FP2 can cause the Warehouse Proxy Agent and the Summarization and Pruning Agent not to function properly. Use an earlier version, such as DB2 for Linux, UNIX, and Windows V9.1 fix pack 1, or upgrade to a level that contains the fix for APAR JR26744, such as DB2 for Linux, UNIX, and Windows V9.1 fix pack 3.
4. IBM Tivoli Monitoring supports Microsoft SQL Server 2000 only if the data is limited to code points inside the Basic Multilingual Plane (range U+0000 to U+FFFF).
5. Users of DB2 Database for Linux, UNIX, and Windows V9.5 fix pack 1 must update their JDBC driver to version 3.57.82. You can download this updated driver here: http://www-01.ibm.com/support/docview.wss?rs=4020&uid=swg21385217
6. DB2 on z/OS is not supported for the Tivoli Enterprise Portal Server.

Table 23 shows the supported databases for the Tivoli Data Warehouse database compression feature.

Table 23. Supported databases for Tivoli Data Warehouse database compression

Tivoli Data Warehouse database compression		
IBM DB2	MS SQL Server	**Oracle**
V9.1 or higher, with the IBM DB2 Storage Optimization feature.	MS SQL Server 2008 Enterprise Edition or higher.	11g Release 1 or higher, with the Oracle Advanced Compression feature.

Note: DB2 on z/OS is not supported for this feature.

Required hardware for distributed systems

The following sections describe the processor, disk, memory, and other hardware requirements for the IBM Tivoli Monitoring infrastructure components on distributed systems. A *distributed system* is defined here as any hardware that is not zSeries.

Following is a list of the IBM Tivoli Monitoring components covered in this section:
- Hub monitoring server
- Remote monitoring server
- Portal server
- Portal client
- Tivoli Data Warehouse
- Warehouse Proxy Agent
- Summarization and Pruning Agent

Processor requirements

For best performance, processor speeds should be at least 1.5 GHz for RISC architectures and 3 GHz for Intel architectures. Choosing faster processors should result in improved response time, greater throughput, and lower CPU utilization.

Except for the Tivoli Data Warehouse, single-processor systems are suitable when an IBM Tivoli Monitoring infrastructure component is installed on a separate computer from the other components. The infrastructure components (monitoring server, portal server, portal client) run as multithreaded processes and are able to run threads concurrently across multiple processors if they are available. CPU utilization for most components is bursty, and steady-state CPU utilization is expected to be low in most cases. For components supporting large environments, using multiprocessor systems can improve throughput.

You should also consider using multiprocessor systems in the following scenarios:
- You want to run the Tivoli Enterprise Portal client on a computer that is also running one of the server components.
- You have a monitoring environment of 1000 or more monitored agents, and you want to install multiple server components on the same computer. For example:
  - Portal server and hub monitoring server
  - Monitoring server (hub or remote) and Warehouse Proxy Agent
  - Warehouse Proxy and Summarization and Pruning Agents
- You have a small environment and you want to include all of the server components (monitoring server, portal server, Warehouse Proxy Agent, and Summarization and Pruning Agent) on a single computer.
- Except in very small environments, use a multiprocessor system for the Tivoli Data Warehouse database server. You can run the Warehouse Proxy Agent and the Summarization and Pruning Agent on the Warehouse database server to eliminate network transfer from the database processing path:
  - If you install the Warehouse Proxy Agent on the Warehouse database server, consider using a two-way or four-way processor.
  - If you install the Summarization and Pruning Agent on the Warehouse database server (with or without the Warehouse Proxy Agent), consider using a four-way processor. For large environments where more CPU resources might be needed, you can run the Summarization and Pruning Agent on a computer separate from the Warehouse database server. In this case, ensure that a high-speed network connection exists (100 Mbps or faster) between the Summarization and Pruning Agent and the database server.

Memory and disk requirements

The following table shows estimated memory and disk storage for IBM Tivoli Monitoring components on distributed systems.

Table 24. Estimated memory and disk storage for IBM Tivoli Monitoring components on distributed systems

Component	Process memory requirements[1]		Disk storage requirements[4]
	Small environment[2]	Large environment[3]	
Hub monitoring server	70 MB	400 MB	Windows: 1.1 GB Linux and UNIX: 1.3 GB plus 300 MB free in the /tmp directory[5 7]
Remote monitoring server	100 MB	400 MB	900 MB[5 7]
Portal server	2 GB without the embedded Derby database[6]	2 GB without the embedded Derby database[6]	1.2 GB plus an additional 1.2 GB in your computer's temporary directory to install the eWAS server and the Eclipse Help Server [7]
Portal client (browser or desktop)	150 MB	300 MB	150 MB
Tivoli Data Warehouse	2 - 4 GB depending on database configuration parameters	4 - 8 GB depending on database configuration parameters	See “Estimating the required size of your database” on page 469.
Warehouse Proxy Agent	100 MB	200 MB	150 MB
Summarization and Pruning Agent	100 MB	200 MB	150 MB
Tivoli Performance Analyzer	100 MB	2 GB	Depends on Tivoli Data Warehouse configuration for Tivoli Performance Analyzer attribute groups

Table 24. Estimated memory and disk storage for IBM Tivoli Monitoring components on distributed systems (continued)

Component	Process memory requirements[1]		Disk storage requirements[4]
	Small environment[2]	Large environment[3]	

Notes:

1. The memory and disk sizings shown in this table are the amounts required for the individual component beyond the needs of the operating system and any concurrently running applications. For the total system memory required by small, medium-size, and large environments, see "Sizing your Tivoli Monitoring hardware" on page 46.
2. A small environment is considered to be a monitoring environment with 500 to 1000 agents, with 100 to 200 monitored agents per remote monitoring server and 20 clients or fewer per portal server.
3. A large environment is considered to be a monitoring environment with 10,000 agents or more monitored agents, with 500 to 1500 monitored agents per remote monitoring server, and with 50 clients or more per portal server.
4. The disk storage estimates apply to any size monitoring environment and are considered high estimates. The size of log files affect the amount of storage required.
5. The storage requirements for the hub and remote monitoring servers do not include storage for the agent depot, which can require an additional 1 GB or more.
6. The memory requirement for the portal server does not include database processes for the portal server database, which require up to 400 MB of additional memory, depending on configuration settings.
7. Tivoli Monitoring components require additional disk storage when the self-describing agent feature is enabled. The self-describing agent disk storage usage sizes should be multiplied by the number of products in your environment that are expected to perform self-describing agent installations. This estimate is only applicable when the self-describing agent feature is enabled. The disk storage is used by the directory indicated in the Tivoli Enterprise Monitoring Server variable `TEMS_MANIFEST_PATH`, or the Tivoli Enterprise Portal Server variable `TEPS_MANIFEST_PATH`.
   - The hub monitoring server requires an estimated 3 to 5 MB of additional disk storage for each product that is performing self-describing agent installations. WebSphere and OMEGAMON agents require more storage, so use the 5 MB per product estimate if you are installing those agents. Otherwise, you can use the 3 MB average size per product.
   - The remote monitoring server requires an estimated 2 MB of additional disk storage for each product that is performing self-describing agent installations.
   - The portal server requires an estimated 3 to 6 MB of additional disk storage for each product that is performing self-describing agent installations since it maintains the new and previous version of an agent product's application support. WebSphere and OMEGAMON agents require more storage, so use the 6 MB per product estimate if you are installing those agents. Otherwise, you can use the 3 MB average size per product.

Add the sizings for individual components to calculate a total for more than one component installed on the same computer.

For example, if the hub monitoring server and portal server are installed on the same computer in a small monitoring environment, the initial requirement is 170 MB of memory and 900 MB of disk space beyond the needs of the operating system and other applications. If you add 400 MB of memory for the portal server database and 1 GB of storage for the agent depot, the total requirement for IBM Tivoli Monitoring components comes to 570 MB of memory and 1.9 GB of storage.

Additional requirements

- The best network connection possible is needed between the hub monitoring server and portal server and also between the Tivoli Data Warehouse, Warehouse Proxy Agent, and Summarization and Pruning Agent.
- A video card supporting 64,000 colors and 1024 x 768 resolution is required for the portal client.

Required hardware for System z

The Tivoli Enterprise Monitoring Server can be installed on either a z/OS or Linux operating system running on System z® hardware. The Tivoli Enterprise Portal Server is supported on Linux for zSeries, but not z/OS. The supported product versions for z/OS and Linux for zSeries are listed in Table 18 on page 142.

The following paragraphs summarize the hardware requirements for IBM Tivoli Monitoring components that run on zSeries:

- Tivoli Enterprise Monitoring Server on z/OS

  The Tivoli Enterprise Monitoring Server can run natively on any zSeries hardware under z/OS V1.10 or later.

  If you enable the self-describing agent feature in your environment, your monitoring servers must have access to a HFS or zFS file system through UNIX System Services (USS). A recommendation is to allocate 25 to 50 MB for this file system. The amount of storage required for the packages of a self-describing agent depends on the monitoring server type:

  – Hub monitoring server = 3.25 MB

  – Remote monitoring server = 2 MB

  This average number takes into account the following factors:

  – The average amount of storage needed by the monitoring server self-describing agent process to backup the existing monitoring server product files in the `$TEMS_MANIFEST_PATH/SDMBACKUP` directory.

  – The hub monitoring server self-describing agent process stores all existing self-describing agent package files including a self-describing agent's TEP(S) JAR files, which are the largest JAR files, often in the 1 to 2 MB range.

  – The remote monitoring server only has to store a self-describing agent's monitoring server JAR files, which are typically around 50K.

  The amount of USS disk space required is also influenced by the following additional factors:

  – The amount of space fluctuates depending on how many self-describing agents register with the z/OS monitoring server, and whether multiple self-describing agent installations are in progress.

  – The JAR files are extracted into individual files in subdirectories under the monitoring server USS home directory, and those individual files are copied to `&rhilev.&rte.RKANDATV` and then automatically deleted. You must allocate enough space for both the JAR files and the extracted contents of those JAR files. The recommended USS disk space must be sufficient to handle these spikes in disk usage.

- Tivoli Enterprise Monitoring Server or Tivoli Enterprise Portal Server on Linux for zSeries

  Any zSeries hardware provides adequate processing capability for a monitoring server or portal server running under Linux for zSeries. The memory and disk requirements for the Tivoli Monitoring components on Linux for zSeries are similar to those for Linux running on distributed systems; these are documented in Table 24 on page 154.

Required software

The following table lists the software required for IBM Tivoli Monitoring.

Table 25. Required software for IBM Tivoli Monitoring

Product	Supported version	Component where the software is required				
		Monitoring server	Portal server	Portal desktop client	Portal browser client	Monitoring agent
IBM JRE 1.5				X	X[18]	

Table 25. Required software for IBM Tivoli Monitoring (continued)

Product	Supported version	Component where the software is required				
		Monitoring server	Portal server	Portal desktop client	Portal browser client	Monitoring agent
Distributed systems: IBM Runtime Environment for Java	JRE V1.6	[3]	[3]	[3]	X[17]	
Sun Java SE Runtime Environment	JRE V1.6.xx			X	X[7]	
z/OS systems: IBM 31-bit or 64-bit SDK for z/OS, Java Technology Edition	IBM Java SDK for z/OS at V6 or later	X[4]				
z/OS systems: z/OS UNIX System Services		X[4 6]				
For Linux computers: a Korn shell interpreter	ksh package provided by the Linux distribution or pdksh-5.2.14	X	X	X		X[2]
For RedHat Enterprise Linux computers: libXp.so.6						X
AIX only: xlC Runtime Environment	Component xlC.aix50.rte must be at 8.0.0.4 or higher fix pack	X	X			X
Microsoft Internet Explorer	V6.0 through V9.0, with all critical Microsoft updates applied				X	
Mozilla Firefox	V3.0.x,V3.5.x				X[5]	
Database[1]	A supported RDBMS is required for the Tivoli Enterprise Portal Server and the Tivoli Data Warehouse. Supported database platforms for the portal server are listed in Table 21 on page 151. Supported database platforms for the Tivoli Data Warehouse are listed in Table 22 on page 152. Each database requires a driver. For detailed information, see Chapter 18, “Tivoli Data Warehouse solutions,” on page 465 and subsequent chapters about the Tivoli Data Warehouse.					

Table 25. Required software for IBM Tivoli Monitoring (continued)

Product	Supported version	Component where the software is required				
		Monitoring server	Portal server	Portal desktop client	Portal browser client	Monitoring agent

Notes:

1. If the JRE is not installed on the computer on which the browser is launched, you are prompted to download and install it from the portal server. The Windows user account must have local administrator authority to download and install the JRE.
2. The Korn shell (any version) is also required when installing the monitoring agent on AIX, HP-UX, or Solaris systems.
3. IBM JRE 1.6 is installed with the monitoring server, portal server, and portal desktop client components on distributed systems.
4. If the self-describing agent function is enabled, you must install this software on the z/OS systems where your monitoring servers are installed.
5. The portal browser client does not support the Mozilla Firefox browser on Windows Server 2008. JRE 1.5 must be used with the Mozilla Firefox browser.
6. A file system (either HFS or zFS) and user security must be configured for UNIX System Services on the z/OS systems where monitoring servers are installed in order to use the self-describing agent function.
7. IBM and SUN JRE 1.6 cannot be used with the Mozilla Firefox browser.
8. JRE 1.5 must be used when the TEP Browser Client is used with the Mozilla Firefox browser.

Required software for event integration with Netcool/OMNIbus

The following products must be installed and configured before you install event synchronization and configure event forwarding for Netcool/OMNIbus:

- IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF version 10 or later and the non-native probe version 12 or later
- Netcool/OMNIbus 7.2.0 FP10 or later fix pack
- Netcool/OMNIbus 7.2.1 FP10 or later fix pack
- Netcool/OMNIbus 7.3.0 FP5 or later fix pack
- Netcool/OMNIbus 7.3.1 or later fix pack

Notes:

1. If you are installing IBM Tivoli Monitoring event synchronization on a Linux operating system, you must ensure that the libXp shared library is installed on the computer system before installing event synchronization.
2. If you want to configure SSL between the probe and monitoring agents, version 12.0 or later of the Netcool/OMNIbus Probe for Tivoli EIF is required.
3. If you are using the virtualization rules and triggers and predictive rules and triggers provided with Netcool/OMNIbus, you must use Netcool/OMNIbus Version 7.3.0 FP 6 or later, or Netcool/OMNIbus Version 7.3.1 FP2 or later, as those fix packs have the required updates to the rules files and triggers for integration with IBM Tivoli Monitoring.

Software and memory requirements for non-linear trending in Tivoli Performance Analyzer

The following are prerequisite criteria for using the non-linear trending feature in Performance Analyzer:

- You must install SPSS Forecast Server V20.
- A minimum of 100 MB of space is required in the ITPA agent home directory for the output data files generated by SPSS.

Chapter 7. Upgrading from a previous installation

This chapter provides information to help you upgrade from a previous installation.

- "Upgrade scenarios" helps you identify the scenario to use as a guide when you upgrade and where to find information about it.
- "Planning your upgrade" on page 160 identifies the platforms no longer supported, describes two methods for components need to be upgraded, provides instructions for backing up your existing environment, and specifies the required order for upgrading components.
- "Upgrading from IBM Tivoli Monitoring V6.1 or V6.2" on page 173 provides an overview of the process of upgrading from V6.1 to V6.2 of IBM Tivoli Monitoring.
- "Upgrading from OMEGAMON Platform V350 and V360" on page 177 provides an overview of the process of upgrading from OMEGAMON Platform 350 or 360 to IBM Tivoli Monitoring V6.2. For more information, see "Installing into an existing installation" on page 136.

Upgrade scenarios

Use one of the following scenarios as a guide when you upgrade from a previous installation:

Upgrading from Tivoli Distributed Monitoring: The new IBM Tivoli Monitoring is not dependent on the Tivoli Management Framework. An upgrade toolkit is provided to facilitate your move from a Tivoli Distributed Monitoring environment to the new IBM Tivoli Monitoring. For information, see *IBM Tivoli Monitoring: Upgrading from Tivoli Distributed Monitoring*, document number GC32-9462.

Upgrading from IBM Tivoli Monitoring V5.1.2: An upgrade toolkit is provided to facilitate your move from IBM Tivoli Monitoring V5.1.2 to IBM Tivoli Monitoring V6.2.

The IBM Tivoli Monitoring version 5 to version 6 Migration Toolkit provides a starting point for migrating your IBM Tivoli Monitoring version 5 environment to IBM Tivoli Monitoring version 6. This tool assesses a Tivoli Monitoring V5 environment and produces an inventory of your site's current monitoring architecture and its monitors. In addition, a draft Tivoli Monitoring V6 configuration is produced for your review and modification. While the tool migrates what it can from IBM Tivoli Monitoring V5 to IBM Tivoli Monitoring V6, optimizations and improvements might be available that would greatly improve your new Tivoli Monitoring V6 performance and scale, but you must perform these manually.

For information about using the toolkit and some best-practices guidelines, see *IBM Tivoli Monitoring: Upgrading from V5.1.2*.

IBM Tivoli Monitoring V5.x interoperability: Using the IBM Tivoli Monitoring 5.x Endpoint agent, you can view data from IBM Tivoli Monitoring 5.x resource models in the Tivoli Enterprise Portal and warehouse granular data in the Tivoli Data Warehouse. You can use this visualization to replace the Web Health Console used in IBM Tivoli Monitoring V5.1. For information about using this visualization, see the *Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint User's Guide* (document number SC32-9490).

Upgrading from IBM Tivoli Monitoring V6.1: To upgrade from a previous installation of IBM Tivoli Monitoring V6.1, use the planning, upgrading, and postinstallation configuration instructions provided in this chapter (see "Planning your upgrade" on page 160 and "Upgrading from IBM Tivoli Monitoring V6.1 or V6.2" on page 173).

Upgrading from OMEGAMON Platform V350 and V360: Migration tools are provided to facilitate the upgrade process for your site's custom situations, policies, and queries to the formats used by IBM Tivoli Monitoring V6.1 and V6.2.

Many of the existing OMEGAMON Platform V350 and V360 agents have equivalent IBM Tivoli Monitoring agents. For any that do not yet have an IBM Tivoli Monitoring counterpart, you can continue to monitor those agents in your new IBM Tivoli Monitoring environment. Use the planning and upgrade instructions in this chapter (see “Planning your upgrade” and “Upgrading from OMEGAMON Platform V350 and V360” on page 177).

OMEGAMON V350/V360 users upgrading to IBM Tivoli Monitoring V6.2.3

There is no direct upgrade path from the OMEGAMON V350 or V360 platform to Tivoli Monitoring V6.2.3. You must first upgrade to either Tivoli Monitoring V6.1 (with any fix pack), V6.2, or V6.2 with fix pack 1, then upgrade to V6.2.3. The recommended upgrade path is OMEGAMON V350/V360 → Tivoli Monitoring V6.1 FP5 → Tivoli Monitoring 6.2.3. Although the IBM Tivoli Monitoring V6.2 images are not included with the V6.2.3 media, you can install the V6.1 fix pack 5 distribution provided with previous releases of the OMEGAMON, NetView®, and ITCAM products. If during installation you find you need OMEGAMON V350/V360, Tivoli Monitoring, or DB2 Database for Linux, UNIX, and Windows images, go to this IBM Web Membership (IWM) site: https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-migrate-itm.

This restriction applies only to these distributed OMEGAMON V350/V360 components:

- the CandleNet Portal desktop and browser clients
- the CandleNet Portal Server
- the Candle Management Server running on a distributed platform
- the OMEGAMON Monitoring Agents
- the Candle Data Warehouse proxy agent

The z/OS Installation and Configuration Tool does migrate z/OS-based Tivoli Management Services processes from OMEGAMON V350/V360 to IBM Tivoli Monitoring V6.2.x; thus this two-step upgrade process does not apply to a z/OS-based Candle Management Server nor to any z/OS monitoring agent.

Planning your upgrade

The following sections provide planning information for upgrading to IBM Tivoli Monitoring V6.2/V6.2.3:

- “Prerequisites for IBM Tivoli Monitoring V6.2.3”
- “IBM Tivoli Monitoring V6.2.*x* coexistence and interoperability” on page 170
- “Components to upgrade” on page 162
- “Required order of upgrade” on page 163
- “Migrated information when upgrading from a previous version” on page 165
- “Backing up IBM Tivoli Monitoring” on page 165

Note: In previous versions there was no validation on hub and remote monitoring server protocols and stand-by configuration, so it is possible that your configuration has incorrect entries. Consider reconfiguring all monitoring servers, both hub and remote. During this reconfiguration you will be warned about incorrect entries. Reconfiguration may be particularly useful if you are experiencing long startup and stopping times for monitoring servers. You can reconfigure a hub monitoring server as a remote, but reconfiguring a remote monitoring server from a remote to a hub is currently not supported. In this case you must reinstall the monitoring server.

Prerequisites for IBM Tivoli Monitoring V6.2.3

Some of the prerequisites have changed for the current version of IBM Tivoli Monitoring. Review the “Hardware and software requirements” on page 138 for each component and platform and apply all necessary upgrades and fix packs before you begin your upgrade.

Upgrading and Migrating DB2 Database for Linux, UNIX, and Windows

IBM Tivoli Monitoring V6.2.3 includes a limited-use version of IBM DB2 Workgroup Server Edition V9.7 fix pack 4 for use with the Tivoli Enterprise Portal Server and the Tivoli Data Warehouse. IBM Tivoli

Monitoring V6.2.*x* includes a limited-use version of IBM DB2 Workgroup Server Edition V9.5 for use with the Tivoli Enterprise Portal Server and the Tivoli Data Warehouse. IBM Tivoli Monitoring V6.2 and its fix packs included a limited-use version of IBM DB2 Enterprise Server Edition V9.1.

As long as you're using a supported version of DB2 Database for Linux, UNIX, and Windows, upgrading and migrating to IBM DB2 Workstation Server Edition V9.5 or V9.7 is not required. See "Supported databases for Tivoli Enterprise Portal Server and Tivoli Data Warehouse" on page 151 for supported database versions.

If you elect to upgrade and migrate from an earlier version of DB2 for Linux, UNIX, and Windows, you may receive a warning during instance migration that the new edition of DB2 is different from the edition prior to upgrade (Workgroup Edition versus Enterprise Server Edition). As a result, certain DB2 GUI utilities such as **db2cc** may fail to start after migration, possibly due to the DB2 **jdk_path** parameter being restored to its default value during the migration. To resolve this, **jdk_path** may need to be reset: as the db2inst1 user, run the following DB2 command:

```
db2 update dbm cfg using jdk_path /home/db2inst1/sqllib/java/jdk32
```

See the *DB2 Version 9.5 for Linux, UNIX, and Windows Migration Guide* for more information on the **jdk_path** parameter. A complete suite of DB2 product documentation is provided in the IBM DB2 Database information center at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp.

Tivoli Business Service Manager and Tivoli Enterprise Portal Server integration over SSL

In previous releases, HTTP was only used to transfer static content such as help files from the embedded HTTP server in IBM Tivoli Monitoring, or for single sign-on using the embedded HTTP server in the WAS component. As of version 6.2.3, the Tivoli Enterprise Portal Server installation contains a required IBM HTTP Server front end for all HTTP and HTTPS traffic. The IBM HTTP Server provides increased scalability, performance, and unifies HTTP(S) access to the Tivoli Enterprise Portal Server to a single pair of ports.

By default, the IBM HTTP Server is assigned ownership of the HTTP and HTTPS ports 15200 and 15201. These ports are currently used by the embedded WAS HTTP server in the Tivoli Enterprise Portal Server. This ensures that existing Tivoli Enterprise Portal Server eWAS clients, configured for single sign-on, migrate to the new architecture. However, with the use of HTTPS you must now consider the SSL/TLS certificate of the server.

A trusted signing authority must sign the IBM Tivoli Monitoring Certificate. You must import the certificate of that signing authority into any web applications that are accessing IBM Tivoli Monitoring data from the Tivoli Enterprise Portal Server using HTTPS. If this certificate is missing, and your Tivoli Enterprise Portal Server clients attempt to connect to port 15201 using SSL/TLS, the error message *no trusted certificate found* is displayed.

The following applications are affected:

- The Tivoli Business Service Manager Charts services for IBM Tivoli Monitoring, that is provided with Tivoli Business Service Manager 4.2.2.
- IBM Tivoli Monitoring policy-based data fetcher, that is provided with Tivoli Business Service Manager 4.2 and 4.2.1.
- Other Tivoli Integrated Portal clients using the Tivoli Integrated Portal WebService.

Ensuring web applications connecting to the embedded WAS HTTP server in the portal server using SSL continue to work after upgrading to V6.2.3: Before you upgrade to V6.2.3, you must generate a new pair of keys and sign the IBM Tivoli Monitoring Certificate with the key of your signer. You then import the public key certificate of your signer into the truststores of your client applications, before the upgrade to V6.2.3.

Follow these steps to apply this method and minimize the potential for outages:

1. For instructions on how to create a new public/private key pair, requesting a certificate signing, and saving the signed certificate, see the *IBM Tivoli Monitoring: Administrator's Guide*.
2. You must update the truststore of the client with the signer's certificate, if this has not already been done. The instructions in this section are specific to the Tivoli Business Service Manager Charts services for IBM Tivoli Monitoring that is provided with Tivoli Business Service Manager 4.2, 4.2.1 Fix Pack 2.

For the Tivoli Business Service Manager data server:

1. Log in to the server where the Tivoli Business Service Manager data server is installed.
2. Change directory to `TIP_HOME/bin`.
3. Issue the following command:
   ```
   ./wsadmin.sh ?profileName TBSMProfile ?user tipadmin
   ```
4. When prompted, enter the password for the `tipadmin` user.
5. From the wsadmin prompt, issue the following command:
   ```
   $AdminTask retrieveSignerFromPort { -host tepshostname -port 15201 -keyStoreName
   NodeDefaultTrustStore -certificateAlias alias-name }
   ```

   Where:

   `tepshostname`
   Is the hostname of the Tivoli Integrated Portal Server.

   `alias-name`
   Is an alias for the certificate in the NodeDefaultTrustStore, for example *TEP-IHS*.
6. Save the configuration:
   ```
   wsadmin>$AdminConfig save
   ```
7. Exit the wsadmin prompt:
   ```
   wsadmin>quit
   ```
8. Stop and restart the Tivoli Business Service Manager data server for the changes to take effect.

For the Tivoli Business Service Manager Dashboard server:

1. Log in to the Tivoli Business Service Manager or Tivoli Integrated Portal Dashboard server.
2. Expand **Security** and click **SSL certificate and key management**.
3. Click **Keystores and certificates**.
4. Click **NodeDefaultTrustStore**.
5. Click **Signer certificates**.
6. Click **Retrieve from port**.
7. Enter the host name of the Tivoli Integrated Portal Server.
8. Enter port **`15201`**.
9. Enter an alias name, for example *TEP-IHS*.
10. Click **Retrieve signer information**.
11. Click **OK**.
12. Stop and restart the Tivoli Business Service Manager dashboard server for the changes to take effect.

Components to upgrade

You can determine what components to upgrade by using either of two methods. Use the Enterprise level of the Managed Systems Status list to display the version of any connected agent, as well as its status (online/offline). Identifying an agent's status is especially important as you plan remote deployments. An alternative method, available at the Enterprise level of the Navigator, is to use the Tivoli Management Services Infrastructure view. This topology view visually expresses the relationships and linking of monitoring agents and other components to the hub monitoring server. Use hover-help (help flyovers) in the Tivoli Enterprise Portal topology view to determine the current version of your installed monitoring agents and other components.

1. Access the Tivoli Enterprise Portal through the desktop client or the browser client:
   - **Desktop client:**
     a. In the Windows **Start** menu, select **Programs** → **IBM Tivoli Monitoring** → **Tivoli Enterprise Portal**. The login message is displayed.
     b. Type the name (**sysadmin**) and password of the account that you created during installation.
     c. Click **OK**.
   - **Browser client:**
     a. Start the browser.
     b. Type the following URL for the Tivoli Enterprise Portal into the Address field of the browser:

        ```
        http://systemname:1920///cnp/client
        ```

        where the *systemname* is the host name of the computer where the Tivoli Enterprise Portal Server is installed, and 1920 is the port number for the browser client. 1920 is the default port number for the browser client. Your portal server might have a different port number assigned.
     c. Click **Yes** in response to any security prompts.
     d. In the login prompt, type the user name and password of the **sysadmin** account that you created during installation.
     e. Click **OK**.
     f. Click **Yes** to accept Java Security certificates for this browser session.
2. Click the Navigator, the view that is located by default in the upper-left corner of the portal. The first display that you see in the Navigator view is Enterprise Status workspace.
3. Select the **Enterprise** node at the top of the navigation tree, if it is not already selected.
4. Right-click the node and select **Workspace** → **Self-Monitoring Topology** to display the default topology view.
5. Click on the **TMS Infrastructure - Base view**. This view contains the topology display.
6. Click on the Maximize icon in the upper-right corner of the portal. The topology view occupies the entire display area of the workspace.

Required order of upgrade

Upgrade the products that comprise your configuration in the order described below. If multiple components are being installed within a single installation directory, upgrade all components at once rather than running the upgrade multiple times.

1. Event synchronization component

   If you use event synchronization, upgrade the event synchronization component first. If you are using Netcool/OMNIbus, refer to “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746. If you are using Tivoli Enterprise Console, refer to “Upgrading to Tivoli Event Synchronization version 2.3.0.0” on page 669.
2. Tivoli Data Warehouse
   a. Stop all instances of the Warehouse Proxy Agent and the Summarization and Pruning Agent.
   b. If you are upgrading from V6.1.*x* and are collecting warehouse data from any of the agents listed in Table 26 on page 164, you must migrate your warehouse database before you upgrade other components. See “Upgrading the warehouse” on page 169 and the user's guide for each agent for details.

Table 26. Agents requiring warehouse database migration

- Windows OS agent
- Unix OS agent
- Monitoring for Databases: DB2 Database for Linux, UNIX, and Windows agent
- Monitoring for Databases: Sybase agent
- Monitoring for Databases: mySAP agent
- Monitoring for Databases: Oracle agent

Install the Tivoli Enterprise Portal Server (step 5) before you upgrade the warehouse components to ensure the SQL scripts needed for the warehouse upgrade are available. Ensure you upgrade *all* your site's Warehouse Proxy Agents and Summarization and Pruning Agents before performing step 8.

Note: Running the upgrade scripts a second time may cause loss of warehouse data.

If you are upgrading from V6.2.*x*, you do not need to perform these migration steps.

3. Hub Tivoli Enterprise Monitoring Server

   Upgrading the hub monitoring server is not required, but if the hub is not upgraded, new features of upgraded agents will not be available. If the hub is upgraded, it must be upgraded first. If the Hot Standby feature is used, the primary and secondary hub monitoring servers must be upgraded in that order before the remote monitoring servers are upgraded.

   Note: Installing a back-level OS agent into an existing V6.2.3 environment is not supported. If this is the desired configuration, install the back-level OS agent first.
4. Remote monitoring servers (if necessary)
5. Tivoli Enterprise Portal Server
6. Tivoli Enterprise Portal desktop client

   The client and the portal server must be at the same level for the client to connect successfully.
7. Tivoli Performance Analyzer:
   - If you are upgrading from V6.2.2 of Tivoli Performance Analyzer, you must first upgrade to V6.2.2 fix pack 2 or V6.2.3 before installing V6.2.3 fix pack 1.
   - For configuration information, see step 15 on page 192 in Chapter 8.
   - For information about historical data collection settings when upgrading from V6.2.2 to V6.2.3, see note 4 on page 290 in “Installing domain definitions for Tivoli Performance Analyzer” on page 289.
8. Monitoring agents

   Note: The appropriate level of the Tivoli Enterprise Management Agent Framework is installed when an agent is installed.

   Self-describing monitoring agents is a new feature in V6.2.3 that integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure. For more information, see “Self-describing agent installation” on page 347.

Notes:

1. Agents created with the Agent Builder require that the entire path from the agent, through the remote monitoring server it connects to, to the hub monitoring server, to the portal server, and the data warehouse be upgraded to V6.2. Agents released after IBM Tivoli Monitoring V6.2 may also require a complete V6.2 path. Check your agent documentation.
2. Installing components or agents in an existing `CANDLEHOME` or installation directory is supported as long as the user ID used to run the installation is always the same. Installing components or agents in an existing `CANDLEHOME` or installation directory using different user IDs is not supported.

Migrated information when upgrading from a previous version

When installing over a previous release (into the same installation directory) the following information is migrated into the new version:

- Windows:
  - Port number and communication protocol settings
  - Situations
- UNIX: situations

Backing up IBM Tivoli Monitoring

Before upgrading, make a backup copy of your current IBM Tivoli Monitoring installation in case you need to revert to it. You should always contact IBM Software Support before attempting to use the files generated by the following procedures to restore the IBM Tivoli Monitoring environment. The support staff can help ensure success of the restoration. Otherwise, errors made during the modification of the Windows registry could lead to a corrupted operating system or other problems.

The following procedures describe valid methods for creating a backup. However, an enterprise typically has a standard process for preserving a backup image of all computer systems, including the IBM Tivoli Monitoring environment. In most cases, the standard process for your enterprise is preferable to the processes described in this section. Use the instructions in this section only when a standard backup process is not available.

You must perform the following instructions when no other activity is running on the system. The user accounts that you use to perform these activities must have **root** or **Administrator** privileges:

- "Backing up a Windows installation"
- "Backing up a UNIX or Linux installation" on page 166

Backing up a Windows installation

Follow these steps to back up a Windows installation. In Step 4 of this procedure, you record the current configuration settings. In Step 9 on page 166, you apply these settings again.

Note: Typically an enterprise has a standard process for preserving a backup image of all computer systems, including the IBM Tivoli Monitoring environment. In most cases, the standard process for your enterprise is preferable to the processes described in this section. Use the instructions in this section only when a standard backup process is not available.

1. Close any browser or desktop clients for the Tivoli Enterprise Portal.
2. Launch the **Manage Tivoli Enterprise Monitoring Services** (`KinConfg.exe`) utility.
3. On the computer where you are backing up an installation, stop the Tivoli Enterprise Portal Server, the Tivoli Enterprise Monitoring Server, the Eclipse Help Server, and all the monitoring agents running on the system.
4. Perform the following steps to record the current configuration settings for IBM Tivoli Monitoring. (For example, you can make a record on a sheet of paper or in a buffer file.)
   a. Right-click in the row of the component that you want to configure.
   b. Select **Reconfigure**.

      Note: You make no changes in the series of windows that are displayed.
   c. Record the settings for the component in the series of configuration windows.

      You record details such as port numbers, host names, protocols for communication, firewall settings, and settings for the data warehouse.
   d. Click **OK** in each window and accept all prompts, *without making changes*.

Note: If you click **Cancel** at any point, the **Reconfigure** process fails to display all configuration windows. You must start the **Reconfigure** process again.

You must *unconfigure* the monitoring environment in the next step. This action restores the Windows registry to a known state that ensures success, if you restore the environment later.

5. Perform the following steps to unconfigure each IBM Tivoli Monitoring component, except the Tivoli Enterprise Portal desktop client. Do not perform the following steps for the desktop client.
   a. Right-click in the row of a component in the **Manage Tivoli Enterprise Monitoring Services** window.
   b. Select **Advanced >> Unconfigure** in the popup menu.

      When you are finished, all components except the Tivoli Enterprise Portal desktop show **No** in **Configured** column of the **Manage Tivoli Enterprise Monitoring Services** window.
6. Use a compression command to compress the contents of the directory where IBM Tivoli Monitoring is installed.
7. Use the appropriate database commands to back up the Tivoli Enterprise Portal Server and Tivoli Data Warehouse databases.

   For more information, see “Backing up your portal server and Tivoli Data Warehouse databases” on page 167.
8. Export the entire Windows registry to a backup file as follows:
   a. Select **Run** in the Windows **Start** menu.
   b. Type **regedit** in the **Open** field.
   c. Click **OK**. The Registry Editor is displayed.
   d. Ensure that the **MyComputer** node at the top of the registry is selected, so that all values in the registry are exported.
   e. Select **Export** in the **File** menu.
   f. Save the copy of the registry to the following path and filename: `C:\WinRegistryBeforeInstall.reg`

   At this time, the backup process is complete.
9. Perform the following steps to return IBM Tivoli Monitoring to its normal configuration and status:
   a. Right-click in the row of each unconfigured component.

      "Unconfigured" components show **No** in the **Configured** column of the **Manage Tivoli Enterprise Monitoring Services** window.
   b. Select **Advanced >> Configure Advanced** in the popup menu.
   c. In the series of dialog boxes that is displayed, enter the original parameter settings that you recorded in Step 4 on page 165.
   d. Click **OK** in each configuration window and accept other prompts to save your changes.

   When you are finished, all components show **Yes** in **Configured** column and **Stopped** in the **Status** column.

You are ready to proceed with the upgrade on this computer.

Always contact IBM Software Support before attempting to use the files generated in this procedure to restore the IBM Tivoli Monitoring environment. The support staff can help ensure success of the restoration. Otherwise, errors made during the modification of the Windows registry could lead to a corrupted operating system.

Backing up a UNIX or Linux installation

Follow these steps to back up a UNIX or Linux installation.

Note: Normally an enterprise has a standard process for preserving a backup image of all computer systems, including the IBM Tivoli Monitoring environment. In most cases, the standard process for your enterprise is preferable to the processes described in this section. Use the instructions in this section only when a standard backup process is not available.

1. Close the Tivoli Enterprise Portal browser and desktop clients.
2. Stop the Tivoli Enterprise Portal Server, the Tivoli Enterprise Monitoring Server, the Eclipse Help Server, and all the monitoring agents running on the system.
3. If the Tivoli Enterprise Portal Server is installed, run the following command:

   ```
   ./itmcmd execute cq "runscript.sh migrate-export.sh"
   ```
4. Use the tar command to compress the contents of *ITM_Install_dir* (the directory where IBM Tivoli Monitoring is installed), using a command that is similar to the following:

   ```
   tar -cvf /tmp/ITM_Install_dir.backup.tar ITM_Install_dir
   ```
5. Add the following files to the tar file created in step 4 above:
   - On AIX:

     ```
     /etc/rc.itm*
     tar -uvf /tmp/ITM_Install_dir.backup.tar /etc/rc.itm*
     ```
   - On HP-UX:

     ```
     /sbin/init.d/ITMAgents*
     tar -uvf /tmp/ITMinstall_dir.backup.tar /etc/init.d/ITMAgents*
     ```
   - On other UNIX or Linux systems:

     ```
     /etc/initd/ITMAgents*
     tar -uvf /tmp/ITM_Install_dir.backup.tar /etc/init.d/ITMAgents*
     ```
6. Use the appropriate database commands to back up the Tivoli Data Warehouse databases.

   For more information, see “Backing up your portal server and Tivoli Data Warehouse databases.”

You are now ready to proceed with the upgrade.

Always contact IBM Software Support before attempting to use the files generated in this procedure to restore the IBM Tivoli Monitoring environment. The support staff can help ensure success of the restoration. Otherwise, errors made during the modification of the Windows registry could lead to a corrupted operating system.

Backing up your portal server and Tivoli Data Warehouse databases

When backing up your IBM Tivoli Monitoring V6.1 environment, you must also back up your portal server database and Tivoli Data Warehouse database. The following sections contains procedures for backup up DB2 for Linux, UNIX, and Windows databases, which are provided as examples. If you use Microsoft SQL Server for your portal server database, or Oracle for your Tivoli Data Warehouse database, use the corresponding commands.

- “Backing up your portal server database”
- “Backing up your Tivoli Data Warehouse database” on page 168

Note: These steps are not necessary if you are updating a Tivoli Monitoring V6.2 environment to a later release of V6.2 (such as V6.2.3).

Backing up your portal server database

DB2 Database for Linux, UNIX, and Windows: Use the following command to backup your portal server database where DB2 Database for Linux, UNIX, and Windows is the relational database management system of choice.

```
db2 backup database yourtepsdatabase to /yourbackuplocation
```

If an existing connection prevents you from backing up the database, use the following commands.

1. `db2 connect to yourtepsdatabase`

2. `db2 quiesce database immediate force connections`
3. `db2 connect reset`
4. `db2 backup database` *`yourtepsdatabase`* `to /`*`yourbackuplocation`*
5. `db2 connect to` *`yourtepsdatabase`*
6. `db2 unquiesce database`
7. `db2 connect reset`

Derby: Use the following procedure to back up your portal server database where the embedded database, Derby, is the relational database management system of choice.

1. Shut down the Tivoli Enterprise Portal Server.
2. To back up the Derby database, you can have the portal server export the IBM Tivoli Monitoring records stored in the Derby database, or you can manually back up the entire Derby database.
   - To export the Tivoli Monitoring data in the Derby database, use the `migrate-export` script, as documented in the *IBM Tivoli Monitoring: Administrator's Guide*.
   - To manually back up the entire Derby database, copy the database directory to a backup location. This database is stored in either of these locations:
     - Windows: *`ITMHOME`*`\CNPSJ\derby\TEPS0`
     - Linux/UNIX: *`ITMHOME`*`/`*`platform`*`/iw/derby/TEPS0`
3. Restart the portal server.

Backing up your Tivoli Data Warehouse database

Use the following command to back up your Tivoli Data Warehouse database where DB2 Database for Linux, UNIX, and Windows is the relational database management system of choice.

```
db2 backup database yourwarehousedatabase to /yourbackuplocation
```

If an existing connection prevents you from backing up the database, use the following commands.

1. `db2 connect to` *`yourwarehousedatabase`*
2. `db2 quiesce database immediate force connections`
3. `db2 connect reset`
4. `db2 backup database` *`yourwarehousedatabase`* `to /`*`yourbackuplocation`*
5. `db2 connect to` *`yourwarehousedatabase`*
6. `db2 unquiesce database`
7. `db2 connect reset`

Sites using the Summarization and Pruning Agent with DB2 for Linux, UNIX, and Windows

Your migrated Tivoli Data Warehouse database requires subsequent checking to ensure it is set up so the Summarization and Pruning Agent can perform multiple system batching. Complete these steps:

1. Back up the database, if necessary.
2. Edit the KSY_DB2_WAREHOUSEMARKER.sql script. If using archive logging, modify it as suggested in the script to avoid the need for extra backup at the end.
3. Execute the script:

   ```
   db2 -tvf KSY_DB2_WAREHOUSEMARKER.sql
   ```

   The code will detect if the table is correctly migrated. If not, a single managed system will be enforced independent of the setting to prevent database deadlocks.

This procedure affects only the WAREHOUSEMARKER table. If you do not complete it, the Summarization and Pruning Agent will do single system batching only.

Upgrading the warehouse

Note:

Complete these steps only if you are upgrading from IBM Tivoli Monitoring V6.1.*x* levels. They do not apply to any level after and including V6.2.0. Running the upgrade scripts again may lead to loss of some data.

Some of the monitoring agents have made changes to the warehouse tables. There are three types of changes, two of these types of changes require performing upgrade procedures before running the 6.2 Warehouse Proxy and Summarization and Pruning Agents. The procedure for the other change can be performed before or after you upgrade IBM Tivoli Monitoring. These procedures are accomplished using product-provided scripts.

Case 1 changes affect the table structure, and Case 3 changes affect the table location. Both Case 1 and Case 3 changes must be completed before running the 6.2 Warehouse Proxy and Summarization and Pruning Agents. These changes and procedures are documented comprehensively in the following user's guide appendix for each monitoring agent that is affected: "Upgrade: upgrading for warehouse summarization."

- Case 1 changes add a new column to a raw table, and assign a default value to the column. If the 6.2 Warehouse Proxy Agent is started before these scripts are run, a NULL default value is assigned, and that default value cannot be changed. The following monitoring agents have Case 1 changes:
  - Monitoring for Databases: DB2 for Linux, UNIX, and Windows Agent
  - Monitoring for Applications: mySAP Agent
  - Monitoring: UNIX OS Agent
  - Monitoring: Windows OS Agent
- Case 2 changes affect how agent data is summarized because of changes in primary key definitions. The Case 2 changes can be done before or after you upgrade IBM Tivoli Monitoring.The following monitoring agents have Case 2 changes:
  - Monitoring for Applications: mySAP Agent
  - Monitoring for Databases: Sybase Server Agent
  - Monitoring: Windows OS Agent
- Case 3 changes are only applicable if you are using DB2 for Linux, UNIX, and Windows as your warehouse database. These changes move a summary table from the 4K table space to the 8K table

space. If the 6.2 Summarization and Pruning Agent is started before these scripts are run, you receive DB2 for Linux, UNIX, and Windows errors, because the row size of the upgraded table is longer than what can fit into the 4K table space. The following monitoring agents have Case 3 changes:

– IBM Tivoli Monitoring for Databases: DB2 for Linux, UNIX, and Windows Agent
– IBM Tivoli Monitoring for Databases: Oracle Agent
– IBM Tivoli Monitoring: UNIX OS Agent

To ensure that you can make the table upgrades required by Case 1 and Case 3 before installing the 6.2 monitoring agents, do the following:

1. Stop the Warehouse Proxy and the Summarization and Pruning Agents prior to upgrading to IBM Tivoli Monitoring V6.2 or V6.2FP1.
2. Change the default startup of these monitoring agents to "Manual" so they do not start automatically during any restart of host computer. This prevents these agents from starting automatically during the upgrade process.
3. Perform the IBM Tivoli Monitoring upgrades.
4. Make the warehouse updates for the monitoring agents with Case 1 and Case 3 changes as described in the user's guide for the affected agent.
5. Re-enable the automatic startup of the Warehouse Proxy and the Summarization and Pruning Agents, and start these agents.

Several monitoring agents made changes to the warehouse collection and summarization characteristics for some agent attribute groups. These changes correct and improve the way warehouse data is summarized, producing more meaningful historical reports. For an explanation of these changes and the implications to your warehouse collection and reporting, see the "Upgrading for warehouse summarization" appendix of the user's guide for your specific monitoring agent.

IBM Tivoli Monitoring V6.2.*x* coexistence and interoperability

The following statements specify the level of interoperability for IBM Tivoli Monitoring V6.2 in support of existing agents and staged upgrade scenarios. Special interoperability notes for Tivoli Monitoring V6.2.1, V6.2.2, and V6.2.3 are provided where appropriate.

Tivoli Enterprise Monitoring Server

- V6.1 remote monitoring servers can connect to V6.2, V6.2.1, V6.2.2, or V6.2.3 hub monitoring servers.
- V6.2, V6.2.1, V6.2.2, or V6.2.3 remote monitoring servers can connect to V6.1 hub monitoring servers as long as no agents require dynamic-affinity support.

  Note: Agents created with the Agent Builder require that the entire path from the agent, through the remote monitoring server to which it connects, to the hub monitoring server, to the portal server and the data warehouse be upgraded to V6.2, V6.2.1, V6.2.2, or V6.2.3. Agents released after IBM Tivoli Monitoring V6.2 may also require the V6.2 path. Check your agent documentation.

  Also, agents built with the V6.2.1.*x* Agent Builder (or previous) cannot be installed into an environment where a System Monitor Agent is also running.
- If your site uses IP.SPIPE for Tivoli Enterprise Monitoring Server communication:
  – A V6.2.2 or V6.2.3 remote monitoring server not running in FIPS mode can connect to a V6.1 hub. Connectivity will default to the Triple DES standard.To enable AES encryption on the hub, customers may set parameter **GSK_V3_CIPHER_SPECS="352F0A"** on the V6.1 monitoring system.
  – A V6.2 or V6.2.1 remote monitoring server can connect to a V6.2.2 or V6.2.3 hub not running in FIPS mode. Connectivity will default to the Triple DES standard. To enable AES encryption, customers may set parameter **GSK_V3_CIPHER_SPECS="352F0A"** on the remote monitoring systems.

  - Agents using pre-V6.2.2 monitoring agent framework can select IP.SPIPE to connect to a V6.2.2 or V6.2.3 monitoring server defined with **FIPS=N**.
  - Agents using pre-V6.2.2 monitoring agent framework cannot select IP.SPIPE to connect to a V6.2.2 or V6.2.3 monitoring server defined with **FIPS=Y**.
  - Agents using V6.2.2 or later monitoring agent framework defined with either **FIPS=N** or **FIPS=Y** can select IP.SPIPE to connect to a V6.2.2 or later monitoring server defined with either **FIPS=N** or **FIPS=Y**.
- A V6.2, V6.2.1, V6.2.2 or V6.2.3 remote monitoring server can connect to a V6.2, V6.2.1, V6.2.2 or V6.2.3 hub monitoring server.
- If you are running a hub monitoring server at release 6.2 or earlier, the tacmd CLI commands for situation groups, situation long names, and adaptive monitoring cannot be used.
- If you are running a monitoring server at release 6.2.2 FP1 or earlier, you cannot use the tacmd CLI commands getFile, putFile, and executeCommand.
- If you are running a monitoring server at a release prior to V6.2.3, you cannot use the tacmd CLI commands for prerequisite checking and setAgentConnection.
- If you are running a hub monitoring server at a release prior to V6.2.3, you cannot use the self-describing agent function and associated tacmd CLI commands. If you are running a remote monitoring server at a release prior to V6.2.3, you cannot use the self-describing agent function for the agents connected to that monitoring server.
- V6.2.x interoperability with OMEGAMON Platform V350/360 remains the same as for IBM Tivoli Monitoring V6.1; see “Upgrading from OMEGAMON Platform V350 and V360” on page 177.

Tivoli Enterprise Portal Server

- The V6.2.3 Tivoli Enterprise Portal Server can interoperate with older hub Tivoli Enterprise Monitoring Servers back to V6.1.
- A Tivoli Enterprise Portal Server upgrade-in-place is supported from IBM Tivoli Monitoring V6.1 portal server to the IBM Tivoli Monitoring V6.2, V6.2.1, V6.2.2, or V6.2.3 portal server on the same platform.
- A Tivoli Enterprise Portal Server upgrade to a new platform is accomplished with database export and import.

  Portal server upgrades to the new release are accomplished with the export/import feature. For information about migrating the database, see the description of the migrate-export and migrate-import scripts in the *IBM Tivoli Monitoring: Administrator's Guide*.
- IBM Tivoli Monitoring V6.2 supports importing an IBM Tivoli Monitoring V6.1 database and then upgrading it by running buildpresentation.bat (for Windows platforms) or InstallPresentation.sh (for UNIX/Linux platforms).
- When upgrading from a prior portal server to the current version, it is important that your system's temporary directory has sufficient space to accommodate the download of the eWAS server code and the Eclipse server code. See the memory and disk space requirements documented in Table 24 on page 154.
- The Tivoli Enterprise Portal clients must be at the same release level as the portal server.

Tivoli Data Warehouse

- All Warehouse Proxy and Summarization and Pruning Agents must be at the same level to avoid inconsistencies in presentation.
- Warehouse Proxy and Summarization and Pruning V6.1, V6.2, V6.2.1, V6.2.2, or V6.2.3 agents can interoperate with monitoring servers at the same or later release level.
- It is recommended that you upgrade the Warehouse Proxy and Summarization and Pruning agents to be at the same release level as the most recent agent monitoring framework release level used by your agents. For example, if you intend to use IBM Tivoli Monitoring V6.2.3 operating system agents, then upgrade the Warehouse Proxy and Summarization and Pruning agents to V6.2.3 as well. This ensures the warehouse agents support all of the history collection functions used by your agents:
  - The V6.2.1 or later Warehouse Proxy Agent and Summarization and Pruning Agent are required for correct functioning of any agent requiring 64-bit integer data support.
  - The V6.2.2 or later Warehouse Proxy Agent and Summarization and Pruning Agent are required if you configure granular history collection for any of your agents using the V6.2.2 or later agent monitoring framework.
  - The V.6.2.3 or later Warehouse Proxy Agent and Summarization and Pruning Agent are required if you configure historical data compression before upload to the warehouse database for any of your agents using V6.2.3 or later agent monitoring framework.

Agents

IBM Tivoli Monitoring V6.2.3 agents (and application agents using the V6.2.3 agent monitoring framework) require a V6.2.3 Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server. In other words, the V6.2.2 (and earlier) monitoring server and portal server do not support V6.2.3 agents.

Back-level agents can connect through up-level monitoring servers. In particular, V6.2.1 agents will work with the V6.2.1, V6.2.2, and V6.2.3 Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

Agents that use dynamic affinity cannot connect or fail over to a pre-V6.2.1 remote monitoring server.

An agent upgrade elevates the user credentials of agents that have been set at deployment time to run with reduced (non-root) credentials. Use one of the following steps to stop the agent and restart it with the appropriate user credentials:

- After stopping the agent, log in (or invoke `su`) as the desired user, and then run the itmcmd command to start the agent in that user's context.
- Edit the startup file, and add the following line to change the default user context for starting agents:

  ```
  /usr/bin/su - dbinstancename-c "itmhome/bin/itmcmd agent -h itmhome
       -o dbinstancename start product_code
  ```

  where:

  product_code
  is the two-character product code for the agent (for example, ud for DB2 for Linux, UNIX, and Windows).

Special interoperability note for IBM Tivoli Monitoring V6.2.1, V.6.2.2, and V6.2.3::

Java Runtime Environment changes were introduced in the V6.2.1, V6.2.2, and V.6.2.3 agents but only to update the service level. No impact is expected.

Tivoli Event Synchronization component

If you have already installed a previous version of Tivoli Event Synchronization, you must upgrade to version 2.3.0.0. If you are using Netcool/OMNIbus, refer to “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746. If you are using Tivoli Enterprise

Console, refer to “Upgrading to Tivoli Event Synchronization version 2.3.0.0” on page 669. The upgrade is required for the event server to correctly parse events coming from the V6.2.3 hub monitoring server and from monitoring agents configured to send private situation events.

Upgrading from IBM Tivoli Monitoring V6.1 or V6.2

The following sections provide information for upgrading from IBM Tivoli Monitoring V6.1 or V6.2 to IBM Tivoli Monitoring V6.2.3:

- “Overview of the upgrade process”
- “Required Java Runtime Environment” on page 177

Overview of the upgrade process

Upgrading from IBM Tivoli Monitoring V6.1 to IBM Tivoli Monitoring V6.2 involves the following steps:

Table 27. Upgrading from IBM Tivoli Monitoring V6.1 or V6.2 to IBM Tivoli Monitoring V6.2.3

Task		Where to find information
Before upgrading your monitoring environment	Review the upgrade planning information to identify what you can and cannot upgrade, as well as the required upgrade order.	“Planning your upgrade” on page 160
	Linux and UNIX sites: If your Tivoli Enterprise Portal Server is running as a non-root process and you plan to upgrade it using a non-root userid, then a special procedure must be performed prior to upgrade.	“Linux and UNIX: Upgrading a portal server running as a non-root process” on page 175
	Stop all components that you are upgrading and change their startup from **Automatic** to **Manual**. Stop Manage Tivoli Enterprise Monitoring Services, and do not start any other Tivoli Management Services components while you complete this upgrade.	“Starting and stopping components” on page 368
	Restart the computer on which you are installing IBM Tivoli Monitoring.	
Upgrading your monitoring environment	Run the IBM Tivoli Monitoring installation program on all components that you want to upgrade. Use your existing installation directory as your IBM Tivoli Monitoring directory.	Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207

Table 27. Upgrading from IBM Tivoli Monitoring V6.1 or V6.2 to IBM Tivoli Monitoring V6.2.3 (continued)

Task		Where to find information
After upgrading your monitoring environment	On platforms other than Windows, you must reconfigure the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server.[2] **Note:** On Windows platforms, the installer manages the reconfiguration process.	Chapter 12, "Configuring IBM Tivoli Monitoring components," on page 363
	On Windows platforms, if you chose upgrade seeding, support for base agents is upgraded automatically, but you must add application support for any other monitoring agents you are upgrading.	"Configuring application support for nonbase monitoring agents" on page 270
	After upgrading your Tivoli Enterprise Monitoring Server, you must add application support to it.[2] This step updates existing situation definitions and installs new situations provided with Tivoli Monitoring V6.2.3. Application support is contained in the **ms** component. • On Windows, you are automatically prompted to add application support to the monitoring server using the kms_upg.sql file. • On Linux or UNIX, you must initiate upgrade reseeding using the **itmcmd support** command and specifying the **ms** component: `./itmcmd support -t` *`tems_name`* `ms` When reseeding completes, recycle your monitoring server: `./itmcmd server stop` *`tems_name`* `./itmcmd server start` *`tems_name`*	"Installing and enabling application support" on page 266

Notes:

1. The installation instructions sometimes reference the default path for the installation of IBM Tivoli Monitoring. If you did not install the product in the default path, you must specify the correct path whenever the upgrade procedures require a path specification.
2. After you upgrade a UNIX monitoring server, you must run the **itmcmd config -S** command to configure the upgraded monitoring server. During this upgrade you are asked if you want to update application support files as well.

   If a silent upgrade is performed by default, all installed support files are updated.
3. You can use the **tacmd updateAgent** command to install an agent update on a specified managed system. For reference information about this command and related commands, see the *IBM Tivoli Monitoring: Command Reference*.
4. You must recycle (stop and restart) the portal server to use upgraded workspaces for your monitoring agents.

Linux and UNIX: Upgrading a portal server running as a non-root process

If the Tivoli Enterprise Portal Server is running as non-root on either Linux or AIX and you plan to upgrade it to IBM Tivoli Monitoring V6.2.3 using a non-root user ID, then you must perform the following procedure prior to upgrading.

On Linux, you need perform only “Step 1: Verify the DB2 Database for Linux, UNIX, and Windows authorizations”; on AIX, perform both “Step 1: Verify the DB2 Database for Linux, UNIX, and Windows authorizations” and “Step 2: Invoke the AIX slibclean command” on page 177.

Step 1: Verify the DB2 Database for Linux, UNIX, and Windows authorizations

In the following example the portal server database is called **TEPS**, the DB2 for Linux, UNIX, and Windows userid is *itmuser*, and the DB2 administrative userid is *db2inst1*. Both Linux and AIX sites must complete this step.

1. Log in as the DB2 administrator:

   ```
   su - db2inst1
   ```

2. Connect to the portal server database using the portal server's DB2 userid:

   ```
   db2 connect to TEPS user itmuser using TEPSpw
   ```

   where *TEPSpw* is the password for userid *itmuser*.

3. Get the authorizations for the portal server's DB2 userid:

   ```
   db2 get authorizations
   ```

   Example output:

   ```
    Administrative Authorizations for Current User

    Direct SYSADM authority                    = NO
    Direct SYSCTRL authority                   = NO
    Direct SYSMAINT authority                  = NO
    Direct DBADM authority                     = NO
    Direct CREATETAB authority                 = NO
    Direct BINDADD authority                   = NO
    Direct CONNECT authority                   = NO
    Direct CREATE_NOT_FENC authority           = NO
    Direct IMPLICIT_SCHEMA authority           = NO
    Direct LOAD authority                      = NO
    Direct QUIESCE_CONNECT authority           = NO
    Direct CREATE_EXTERNAL_ROUTINE authority   = NO
    Direct SYSMON authority                    = NO

    Indirect SYSADM authority                  = NO
    Indirect SYSCTRL authority                 = NO
    Indirect SYSMAINT authority                = NO
    Indirect DBADM authority                   = NO
    Indirect CREATETAB authority               = YES
    Indirect BINDADD authority                 = YES
    Indirect CONNECT authority                 = YES
    Indirect CREATE_NOT_FENC authority         = NO
    Indirect IMPLICIT_SCHEMA authority         = YES
    Indirect LOAD authority                    = NO
    Indirect QUIESCE_CONNECT authority         = NO
    Indirect CREATE_EXTERNAL_ROUTINE authority = NO
    Indirect SYSMON authority                  = NO
   ```

4. Ensure the settings listed below are set to YES, as shown.

   ```
    Direct DBADM authority                     = YES
    Direct CREATETAB authority                 = YES
    Direct BINDADD authority                   = YES
    Direct CONNECT authority                   = YES
    Direct CREATE_NOT_FENC authority           = YES
   ```

```
 Direct IMPLICIT_SCHEMA authority           = YES
 Direct LOAD authority                      = YES
 Direct QUIESCE_CONNECT authority           = YES
 Direct CREATE_EXTERNAL_ROUTINE authority   = YES
```

If they are all set to **YES**, you are finished. Otherwise continue with steps 5 through 7 below.

5. Connect to the Tivoli Enterprise Portal Server database as the DB2 administrator:

```
db2 connect to TEPS user db2inst1 using db2pw
```

   where *db2pw* is the password for userid *db2inst1*.

6. Grant the appropriate authorizations to the portal server's DB2 userid:

```
db2 GRANT DBADM,CREATETAB,BINDADD,CONNECT,CREATE_NOT_FENCED_ROUTINE,IMPLICIT_SCHEMA,LOAD,
    CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT ON DATABASE to user itmuser
```

7. Reconnect to the portal server database using the portal server's DB2 userid, and recheck its authorizations:

```
db2 connect to TEPS user itmuser using TEPSpw
 Database Connection Information

 Database server        = DB2/LINUX 8.2.8
 SQL authorization ID   = ITMUSER
 Local database alias   = TEPS

db2 get authorizations
 Administrative Authorizations for Current User

 Direct SYSADM authority                    = NO
 Direct SYSCTRL authority                   = NO
 Direct SYSMAINT authority                  = NO
 Direct DBADM authority                     = YES
 Direct CREATETAB authority                 = YES
 Direct BINDADD authority                   = YES
 Direct CONNECT authority                   = YES
 Direct CREATE_NOT_FENC authority           = YES
 Direct IMPLICIT_SCHEMA authority           = YES
 Direct LOAD authority                      = YES
 Direct QUIESCE_CONNECT authority           = YES
 Direct CREATE_EXTERNAL_ROUTINE authority   = YES
 Direct SYSMON authority                    = NO

 Indirect SYSADM authority                  = NO
 Indirect SYSCTRL authority                 = NO
 Indirect SYSMAINT authority                = NO
 Indirect DBADM authority                   = NO
 Indirect CREATETAB authority               = YES
 Indirect BINDADD authority                 = YES
 Indirect CONNECT authority                 = YES
 Indirect CREATE_NOT_FENC authority         = NO
 Indirect IMPLICIT_SCHEMA authority         = YES
 Indirect LOAD authority                    = NO
 Indirect QUIESCE_CONNECT authority         = NO
 Indirect CREATE_EXTERNAL_ROUTINE authority = NO
 Indirect SYSMON authority                  = NO
```

   You should now see the authorizations for the following items set to YES:

```
 Direct DBADM authority                     = YES
 Direct CREATETAB authority                 = YES
 Direct BINDADD authority                   = YES
 Direct CONNECT authority                   = YES
 Direct CREATE_NOT_FENC authority           = YES
```

```
    Direct IMPLICIT_SCHEMA authority          = YES
    Direct LOAD authority                     = YES
    Direct QUIESCE_CONNECT authority          = YES
    Direct CREATE_EXTERNAL_ROUTINE authority  = YES
```

Step 2: Invoke the AIX slibclean command

This step applies only to AIX sites, not to Linux sites.

As the *root* user, invoke the AIX `slibclean` command:

```
su -c "/usr/sbin/slibclean"
```

Required Java Runtime Environment

IBM Tivoli Monitoring V6.2.3 FP1 requires Java 6. IBM Tivoli Monitoring agents on the same computer share the same Java environment, and will be automatically upgraded to the latest version when a V6.2.3 FP1 agent is installed.

Upgrading from OMEGAMON Platform V350 and V360

The following sections provide information for upgrading from OMEGAMON Platform 350 or 360 and CandleNet Portal 195 to IBM Tivoli Monitoring 6.2.3:

- “Overview of the upgrade process” on page 178
- “Considerations” on page 179
- “Using existing OMEGAMON and other monitoring agents with IBM Tivoli Monitoring” on page 180

Notes:

1. You cannot upgrade directly to IBM Tivoli Monitoring from a version of OMEGAMON Platform prior to 350 or 360. If you are using an older version of OMEGAMON Platform, you must first upgrade to OMEGAMON Platform 350 or 360 before upgrading to IBM Tivoli Monitoring.
2. There is no direct upgrade path from the OMEGAMON V350 or V360 platform to IBM Tivoli Monitoring V6.2.3. You must first upgrade to either Tivoli Monitoring V6.1 (with any fix pack), V6.2, or V6.2 with fix pack 1, then upgrade to V6.2.3. The recommended upgrade path is OMEGAMON V350/V360 → Tivoli Monitoring V6.1 FP5 → Tivoli Monitoring 6.2.3. Although the IBM Tivoli Monitoring V6.2 images are not included with the V6.2.3 media, you can install the V6.1 fix pack 5 distribution provided with previous releases of the OMEGAMON, NetView, and ITCAM products.

   This two-step upgrade process applies to distributed components only; direct upgrade of z/OS-based Candle Management Servers and monitoring agents continues to be supported using the z/OS Installation and Configuration Tool.

Overview of the upgrade process

Upgrading from Candle OMEGAMON Platform 350 or 360 involves the following steps:

Table 28. Upgrading from OMEGAMON Platform 350 or 360

Task		Where to find information
Before installing IBM Tivoli Monitoring	Review the new terminology. Although the same components exist in IBM Tivoli Monitoring, many have new names.	"Terminology changes" on page 179
	Review the upgrade planning information to identify what you can and cannot upgrade.	"Considerations" on page 179
	Stop all components that you are upgrading and change their startup from **Automatic** to **Manual**. Stop Manage Tivoli Enterprise Monitoring Services, and do not start any other Tivoli Management Services components while you complete this upgrade.	"Starting and stopping components" on page 368
	Restart the computer on which you are installing IBM Tivoli Monitoring.	
Upgrade your monitoring environment to Tivoli Monitoring V6.1 (fix pack 5), included with your OMEGAMON, NetView, or ITCAM media	Run the installation program for IBM Tivoli Monitoring V6.1 (fix pack 5) on all components that you want to upgrade. Use your existing installation directory as your IBM Tivoli Monitoring directory.	Chapter 9, "Installing IBM Tivoli Monitoring," on page 207
After installing IBM Tivoli Monitoring V6.1 (fix pack 5)	For any *existing* OMEGAMON Monitoring Agents that you want to use with the IBM Tivoli Monitoring monitoring server, change their configuration to point to the new monitoring server.	"Using existing OMEGAMON and other monitoring agents with IBM Tivoli Monitoring" on page 180
Upgrade your monitoring environment to Tivoli Monitoring V6.2.x	Run the installation program for IBM Tivoli Monitoring V6.2.x on all components that you want to upgrade. Use your existing installation directory as your IBM Tivoli Monitoring directory.	Chapter 9, "Installing IBM Tivoli Monitoring," on page 207

Notes:

1. After you upgrade a UNIX monitoring server, you must run the **itmcmd config -S** command to configure the upgraded monitoring server and the **itmcmd support** command to update application support files.
2. You must recycle (stop and restart) the portal server to use upgraded workspaces for your monitoring agents.
3. You cannot use the agent deployment function in IBM Tivoli Monitoring to upgrade OMEGAMON agents.

Considerations

Consider the following issues when upgrading from OMEGAMON Platform 350 or 360 and CandleNet Portal 195 to IBM Tivoli Monitoring 6.2.*x*.

Terminology changes

The following terms have changed with the move from Candle OMEGAMON to IBM Tivoli Monitoring:

Table 29. OMEGAMON to IBM Tivoli Monitoring terminology

OMEGAMON term	IBM Tivoli Monitoring term
Candle Management Server (CMS)	Tivoli Enterprise Monitoring Server
CandleNet Portal (CNP)	Tivoli Enterprise Portal
CandleNet Portal Server (CNPS)	Tivoli Enterprise Portal Server
OMEGAMON Monitoring Agent	Tivoli Enterprise Monitoring Agent (monitoring agent)
OMEGAMON Platform	Tivoli Management Services
Manage Candle Services	Manage Tivoli Enterprise Monitoring Services
Event	Situation event
Seeding	Adding application support
OMEGAMON Web Services	Tivoli Monitoring Web Services
Candle Customer Support	IBM Software Support

When to run the upgrade

Run the upgrade to IBM Tivoli Monitoring immediately after you have restarted the computer that you are upgrading. IBM Tivoli Monitoring does not install if the computer has any locked files.

If your OMEGAMON components are currently set to start automatically, change the startup to **Manual** in Manage Tivoli Enterprise Monitoring Services before you restart the components.

Installation directory for upgraded components

When you upgrade an existing OMEGAMON component to the IBM Tivoli Monitoring level, the installation process installs all new files in your existing installation directory (instead of the new default installation directory for IBM Tivoli Monitoring: C:\IBM\ITM on Windows and /opt/IBM/ITM on Linux and UNIX). Your existing files are overwritten.

If you have applied fixes or patches to the OMEGAMON product component, those fixes and patches that were available when IBM Tivoli Monitoring was released are included in the upgraded version.

Configuration settings for upgraded agents

When OMEGAMON Platform V350 or 360 agents are upgraded to IBM Tivoli Monitoring, agent-specific configuration settings regarding the connection to the monitoring server are not maintained. The IBM Tivoli Monitoring installation uses the default settings for all agents (not the one change that you made for one agent). To ensure that your upgraded agents can immediately connect to a monitoring server, determine whether the default settings for all agents point to the new monitoring server prior to upgrading. To change the default settings for all agents, right-click an agent in Manage Candle Services and click **Set defaults for all agents**.

Candle Management Workstation coexistence

If you currently use Candle Management Workstation in your OMEGAMON monitoring environment, you are migrated to IBM Tivoli Monitoring automatically when you migrate the rest of your environment. However, the installed Candle Management Workstation continues to function after the migration, although it is not officially part of IBM Tivoli Monitoring and no new function has been added.

If you use the OMEGAMON XE for CICS 3.1.0 product, you must continue to use Candle Management Workstation to configure workloads for Service Level Analysis. After you have configured the workloads, you can use the Tivoli Enterprise Portal for all other tasks. If you do not currently have Candle Management Workstation (for example, if you are installing OMEGAMON XE for CICS 3.1.0 into an IBM Tivoli Monitoring environment), you must install the Candle Management Workstation that is included with the OMEGAMON XE for CICS 3.1.0 product. Install Candle Management Workstation on a different computer than the Tivoli Enterprise Portal; otherwise the Candle Management Workstation installer attempts to uninstall the Tivoli Enterprise Portal.

Note: As of OMEGAMON XE for CICS 4.1.0, the Candle Management Workstation is no longer required. A new view, CICS SLA, is provided with Tivoli Enterprise Portal V6.2 (and subsequent) that replaces the Service Level Analysis feature of the old Candle Management Workstation.

Additional unsupported OMEGAMON functions

The following functions are no longer supported as part of IBM Tivoli Monitoring:

Table 30. Unsupported OMEGAMON functions

Function	IBM Tivoli Monitoring equivalent function
CandleClone	Agent deployment, as described in Chapter 10, “Deploying monitoring agents across your environment,” on page 325
CandleRemote	
Event emitters	Event forwarding using the Tivoli Enterprise Console event synchronization, as described in Chapter 25, “Setting up event forwarding to Tivoli Enterprise Console,” on page 643
Event adapters	
GUI installation wizard on UNIX	Use the command-line installation option, as described in Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207
Silent installation on UNIX using the multi-platform installation program	Silent installation using custom response files, as described in “Performing a silent installation on a Linux or UNIX computer” on page 793.

CandleNet Portal database

If you use a DB2 for Linux, UNIX, and Windows database for the CandleNet Portal, the database is converted to UTF-8 format during the upgrade. This might take several minutes, depending on the size of your database. If you used the default database password when you created the CandleNet Portal database (TEPS), you are prompted for a new password to comply with the more stringent security provisions in IBM Tivoli Monitoring.

Required Java JRE

The CandleNet Portal requires the Sun Java JRE; however Tivoli Enterprise Portal requires the IBM Java JRE 6. You do not need to install this JRE prior to the upgrade: it is installed automatically when you choose to install an IBM Tivoli Monitoring component that requires it.

Using existing OMEGAMON and other monitoring agents with IBM Tivoli Monitoring

Existing installed OMEGAMON Platform 350 and 360 agents and new IBM Tivoli Composite Application Management and AF/Operator agents can be supported using the Tivoli Enterprise Monitoring Server; however, the following restrictions apply:

- New features in IBM Tivoli Monitoring (such as remote deployment) are *not* supported on OMEGAMON Platform 350 and 360 agents.
- Agents must use one of the following protocols to communicate with the monitoring server:
  - IP.UDP (formerly TCP/IP)
  - IP.PIPE

  - SNA

  The IP.SPIPE protocol is not supported for OMEGAMON XE agents.
- You cannot install an OMEGAMON agent on the same computer as a Tivoli Enterprise Portal Server. If you want to monitor something on the same computer as the portal server, install an IBM Tivoli Monitoring agent, if one is available.
- IBM Tivoli OMEGAMON XE for Messaging V6.0 requires that the hub Tivoli Enterprise Monitoring Server, the Tivoli Enterprise Portal Server, and the Tivoli Enterprise Portal client be at the IBM Tivoli Monitoring V6.2 level.

  After you upgrade the infrastructure components to IBM Tivoli Monitoring V6.2, install application support files for this monitoring agent on the monitoring server, the portal server, and the portal desktop client. To install and enable application support, follow the instructions in the *IBM Tivoli OMEGAMON XE for Messaging Version 6.0 Installation Guide*.
- To enable the Tivoli Data Warehouse to collect data from OMEGAMON Platform 350 and 360 agents (through the Warehouse Proxy), copy the product attribute file to the `itm_installdir/TMAITM6/ATTRLIB` directory on the Warehouse Proxy Agent (where *itm_installdir* is the IBM Tivoli Monitoring installation directory).

For full interoperability between OMEGAMON Platform 350 and 360 and AF/Operator agents and IBM Tivoli Monitoring, you need to install the application support files for these agents on the monitoring server, portal server, and portal desktop client. See the "Agent interoperability" document in the IBM Tivoli Monitoring information center for information about downloading and installing these support files.

Scenario: a rolling product upgrade

Here is an example of a possible scenario for rolling out the upgrade of an installed IBM Tivoli Monitoring environment from one release to a later release.

Configuration

Your existing Tivoli Monitoring environment includes these five distributed servers:
- The primary (acting) hub Tivoli Enterprise Monitoring Server runs on server #1, an AIX 5.3, 64-bit node.
- The secondary (backup) hub monitoring server runs on server #2, a Solaris 10, 32-bit node.
- A remote Tivoli Enterprise Monitoring Server runs on server #3, an RHEL Linux 2.6, 32-bit node.
- A second remote monitoring server runs on server #4, a Windows 2000 node.
- The Tivoli Enterprise Portal Server running on server #5, another RHEL Linux 2.6, 32-bit node, communicates with the acting (primary) hub monitoring server running on server #1. server #5 also runs the Warehouse Proxy Agent and the Summarization and Pruning Agent.

Each system runs an OS agent and the Tivoli Universal Agent. All 14 nodes in this example configuration are connected to one remote monitoring server as its primary and the other as its secondary, to accommodate agent switching when upgrading the remote Tivoli Enterprise Monitoring Servers. Half of the agents point to each remote monitoring server as its primary monitoring server.

Upgrading the Tivoli Monitoring environment

Install a given level of IBM Tivoli Monitoring on the primary and secondary hub monitoring server, on the remote monitoring servers, and on the portal server. Install agents (simple agents, subnode agents, SYSPLEX agents), including the Warehouse Proxy Agent and the Summarization and Pruning Agent, at the same level of IBM Tivoli Monitoring. Configure the Warehouse Proxy and Summarization and Pruning Agents to report directly into both hubs, configured as primary and secondary. You can optionally configure a second Warehouse Proxy Agent, to serve as a backup.

Other agents report to at least one of the remote monitoring servers.

1. Begin historical collection of agent attributes. Assume the historical collection is being done at the agents and not at the Tivoli Enterprise Monitoring Server; otherwise data collection may be interrupted when agents switch from the acting monitoring server to the backup.
2. Cause situations to fire, for example, by varying thresholds. For nodes running the Tivoli Universal Agent, include always-true situations.
3. Ensure the acting (primary) and backup (secondary) monitoring servers are connected by examining their log/trace records and process state for these messages:
   - KQM0001 "FPO started at ..."
   - KQM0003 "FTO connected to ..."
   - KQM0009 "FTO promoted *primary* as the acting HUB"
   - KQM0009 "FTO promoted SITMON**secondary* (Mirror) as the acting HUB"
4. If you have integrated your IBM Tivoli Monitoring events with either Tivoli Enterprise Console or Netcool/OMNIbus (see 641):
   a. First upgrade your event synchronization.
   b. Then restart Tivoli Enterprise Console or OMNIbus, as appropriate.
   c. Finally, restart the monitoring server to which Tivoli Enterprise Console or OMNIbus is connected. This must be the acting monitoring server.
5. Confirm that both the acting and backup hub monitoring servers are operational, so that during this upgrade process, when the acting hub is shut down, the backup will be able to assume the role of acting Tivoli Enterprise Monitoring Server.
6. Upgrade the primary hub Tivoli Enterprise Monitoring Server and all other Tivoli Management Services infrastructure components in that $ITM_HOME installation (in other words, select ALL when asked what to upgrade).
   - If working on a Windows platform, apply support to upgraded agents as part of this upgrade step.
   - If working on a UNIX or Linux platform, apply support for the upgraded infrastructure agents on the primary hub monitoring server:

     ```
     ./itmcmd support -t primary_TEMS_name lz nt ul um ux hd sy
     ```

   Note: All Tivoli Management Services processes running within the same IBM Tivoli Monitoring environment are temporarily shut down when upgrading a monitoring server.
7. Upgrade the secondary hub Tivoli Enterprise Monitoring Server and all other Tivoli Management Services infrastructure components in that $ITM_HOME installation (in other words, select ALL when asked what to upgrade).

   Note: The acting (primary) and backup (secondary) monitoring servers should be installed at the same version level. The best choice is to have the monitoring servers at the same fix pack level as well.
   - If working on a Windows platform, apply support to upgraded agents as part of this upgrade step.
   - If working on a UNIX or Linux platform, apply support for the upgraded infrastructure agents on the secondary hub monitoring server:

     ```
     ./itmcmd support -t secondary_TEMS_name lz nt ul um ux hd sy
     ```
8. Restart both the primary (backup) and secondary (acting) Tivoli Enterprise Monitoring Servers.
9. Upgrade the Tivoli Enterprise Portal Server.
10. Confirm that all remote Tivoli Enterprise Monitoring Servers are operational, so that during this upgrade process, when the acting hub monitoring server is shut down, the backup will be able to assume the role of acting Tivoli Enterprise Monitoring Server.
11. Upgrade each remote monitoring server, one at a time.
    - If working on a Windows platform, apply support to upgraded agents when upgrading the monitoring server.

- If working on a UNIX or Linux platform, immediately after a remote monitoring server is upgraded, apply support for the upgraded infrastructure agents on the same node. Then restart the remote monitoring server to make the upgraded support effective.

12. Upgrade the remaining infrastructure agents.
13. At a time when no history records will be uploaded to it, upgrade the Warehouse Proxy Agent.

Expected results

1. When the primary hub monitoring server is down while completing step 6 on page 182:
   a. Configure the portal server to point to the newly acting hub monitoring server, and verify that expected agents and situations show up on the Tivoli Enterprise Portal client.
   b. Verify that the secondary monitoring server has taken over as acting hub, in other words, that these messages appear in its log:
      - KQM0004 "FTO detected lost parent connection"
      - KQM0009 "FTO promoted *secondary* as the acting HUB"

   c. Verify that the event handler your site is using (either Tivoli Enterprise Console or Netcool/OMNIbus) is still being updated, now by the secondary hub, by causing an event state to change and validating the state of all events on the remote console.
   d. Verify that attribute collection was uninterrupted by examining the history data, by varying attribute values at the agent, and by observing the change in the portal client.
   e. Ensure that the primary hub Tivoli Enterprise Monitoring Server is not restarted by the upgrade process until all remote monitoring servers and directly connected agents have successfully failed over to the secondary hub.
2. After completing step 6 on page 182, verify that the primary hub Tivoli Enterprise Monitoring Server has taken on the role of standby monitoring server: ensure its log contains these messages:
   - KQM0001 "FPO started at ..."
   - KQM0003 "FTO connected to ..."
   - KQM0009 "FTO promoted *secondary* as the acting HUB"
   - KQM0009 "FTO promoted *primary*(Mirror) as the acting HUB"

   The log for the secondary hub monitoring server should contain these messages:
   - KQM0005 "FTO has recovered parent connection ..."
   - KQM0009 "FTO promoted *secondary* as the acting HUB"

   Use the sitpad and taudit tools for assistance. These tools can be downloaded from the IBM Tivoli Integrated Service Management Library Web site at http://www.ibm.com/software/tivoli/opal.
3. When the secondary hub Tivoli Enterprise Monitoring Server is stopped while completing step 7 on page 182:
   a. Configure the Tivoli Enterprise Portal Server to point to the acting hub, the primary.
   b. Verify that the primary monitoring server has taken over as the acting hub; its log should contain these messages:
      - KQM0004 "FTO detected lost parent connection"
      - KQM0009 "FTO promoted *primary* as the acting HUB"

   c. Verify that either Tivoli Enterprise Console or Netcool/OMNIbus is still being updated with event information.
   d. Verify that attribute history is still being collected as in substep 1d.
   e. Ensure that the secondary hub Tivoli Enterprise Monitoring Server is not restarted by the upgrade process until all remote monitoring servers and directly connected agents have successfully failed over to the primary hub.
4. After completing step 7 on page 182, verify that the secondary hub Tivoli Enterprise Monitoring Server is functional by examining the log/trace records of both hub monitoring servers for these messages.

   In the primary hub's log:
   - KQM0005 "FTO has recovered parent connection ..."

- KQM0009 "FTO promoted *primary* as the acting HUB"

In the secondary hub's log:

- KQM0001 "FPO started at ..."
- KQM0003 "FTO connected to ..."
- KQM0009 "FTO promoted *primary* as the acting HUB"
- KQM0009 "FTO promoted *secondary*(Mirror) as the acting HUB"

Use the sitpad and taudit tools for assistance.

5. When the Tivoli Enterprise Monitoring Server is stopped while completing step 11 on page 182 for a given remote monitoring server, verify that either Tivoli Enterprise Console or Netcool/OMNIbus is still being updated, now by the remote monitoring server configured for each agent, and that attribute collection was uninterrupted, as in substep 1d on page 183. Examine the log/trace records of each remote Tivoli Enterprise Monitoring Server to verify agent reporting; use the sitpad and taudit tools for assistance.
6. After completing step 11 on page 182 for a given remote Tivoli Enterprise Monitoring Server, verify that the monitoring server is operational by examining its log/trace records.

   Note: Agents do not automatically switch back to their primary monitoring server unless the secondary server fails or the agents are manually restarted.
7. Ensure the historical collection of attributes is unbroken, in both short-term and long-term history.
8. Ensure situations are reported to Tivoli Enterprise Console and to Netcool/OMNIbus without interruption. Verify that always-true situations are still true after agent switching.
9. Verify that the Tivoli Enterprise Portal client for the acting hub Tivoli Enterprise Monitoring Server shows attribute and situation data at all times.

Notes:

1. Pure events are lost when failover occurs from the primary hub to the secondary and back, as events are not replicated across the two hubs.
2. All situations are restarted during failover, so a situation that was already raised at the primary hub will be raised once again if the situation is still true when failover occurs.
3. A master reset is issued to Tivoli Enterprise Console when the failover occurs, with this event: `"Hotstandby TEMS switched. New Primary TEMS` *`hostname_of_newly_active_TEMS`*`."` Events raised after the failover will be resent.
4. When a failing monitoring server is restarted, Tivoli Enterprise Console receives this event: `"TEMS` *`hostname_of_restarted_TEMS`* `restarted."`

Special instructions for reseeding a Hot Standby monitoring server

Note: This procedure assumes your site needs to keep a Hot Standby Tivoli Enterprise Monitoring Server running at all times. If that is not necessary, it is simpler to shut down the standby monitoring server until after seeding takes place, then start it back up.

If your site has implemented a Hot Standby monitoring server with agent switching (as explained in the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*) and you want to ensure a monitoring server remains active at all times, even while upgrading its application support (remember that upgrading any IBM Tivoli Monitoring process shuts down all local Tivoli Management Services components while the upgrade takes place), follow these steps:

1. Upgrade one of the two hubs, either the active one or the backup. During this process, the other hub monitors your IBM Tivoli Monitoring environment.
2. Upgrade the other hub, during which time the newly upgraded hub monitors your environment.
3. With both upgraded hubs running, one in active mode and the other in backup mode, reseed the active hub without restarting it.
4. Reseed the backup hub.

5. Restart the hub you want to be active, during which time the other hub monitors your IBM Tivoli Monitoring environment.
6. Restart the hub you want to be in backup mode, which puts the other hub in active mode and this hub in backup mode.

Chapter 8. Installing IBM Tivoli Monitoring on one computer

This chapter describes how to deploy a small IBM Tivoli Monitoring environment on a single Windows computer. Installation on one computer might be useful for a test environment, a teaching environment, or for monitoring a small server environment.

The single-computer installation supports between 100 and 200 monitoring agents and can perform minimal historical data collection. See "Sizing your Tivoli Monitoring hardware" on page 46 for further discussion of the capabilities of a small monitoring environment.

The single-computer installation includes the following components:

The hub Tivoli Enterprise Monitoring Server (hub monitoring server)
The Tivoli Enterprise Portal Server (portal server)
The Tivoli Enterprise Portal Client (portal client)
The Tivoli Enterprise Portal Server database (portal server database)
The Tivoli Data Warehouse database (warehouse database)
The Warehouse Proxy Agent
The Summarization and Pruning Agent
The Tivoli Performance Analyzer

The single-computer installation does not include event forwarding to either IBM Tivoli Enterprise Console or IBM Tivoli Netcool/OMNIbus. There is no user authentication.

Prerequisites for the single-computer installation

The software prerequisites for the single-computer installation described in this chapter are as follows:

- A supported Windows operating system
- If your site does not plan to use or cannot use (due to capacity limitations) the embedded Derby database, you will also need one of the following RDBMS servers for the portal server:
  - IBM DB2 Database for Linux, UNIX, and Windows
  - Microsoft SQL Server 2000, 2005, or 2008

  Note: If you plan to use these either of the listed RDBMSes with the portal server, it must already have been installed and be running when you begin this installation. If you plan to use the Derby database, it is not necessary that you preinstall it; this embedded RDBMS is installed, if necessary, with IBM Tivoli Monitoring.

For the Tivoli Data Warehouse, you can also use Microsoft SQL Server 2005. IBM DB2 for Linux, UNIX, and Windows V9.7 fix pack 4 is included with IBM Tivoli Monitoring. This version should be used for new installations to minimize the need to upgrade DB2 for Linux, UNIX, and Windows in the future.

While it is possible to install Tivoli Monitoring on a single Linux or UNIX computer, additional manual configuration is required. When you install Tivoli Monitoring on a Windows computer with IBM DB2 for Linux, UNIX, and Windows or Microsoft SQL Server, much of the configuration required to set up the Tivoli Data Warehouse is automated. (An extra configuration step is required after installation if you use Microsoft SQL Server 2005 for the Tivoli Data Warehouse database.)

In a single-computer installation, the RDBMS server is used for both of the databases required by Tivoli Monitoring:

- The *Tivoli Enterprise Portal Server database* (or *portal server database*) stores user data and information required for graphical presentation on the user interface. The portal server database is created automatically during configuration of the portal server. It is always located on the same computer as the portal server.

- The *Tivoli Data Warehouse database* (also called the *warehouse database* or *data warehouse*) stores historical data for presentation in historical data views. In a single-computer installation, the warehouse database is created on the same relational database management server (RDBMS) used for the portal server database. In larger environments, it is best to create the warehouse database on a different computer from the portal server.

Installation procedure

Complete the following steps to install IBM Tivoli monitoring on one Windows computer:

1. Launch the installation wizard by double-clicking the setup.exe file in the WINDOWS subdirectory of the installation media.
2. Click **Next** on the Welcome window.
3. Click **Accept** to accept the license agreement.
4. Choose the directory where you want to install the product. The default directory is C:\IBM\ITM. Click **Next**.

   Note: If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```
5. Click **Next** to accept the default encryption key and then click **OK** on the popup window to confirm the encryption key.
6. On the Select Features window, select the check boxes for the components you want to install. Select all components for a complete installation on one computer.

   For additional information about these components, press the **Help** button.

   Note: The agent support comprises additional support files for the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and the Tivoli Enterprise Portal desktop client; these are automatically installed whenever you install the agent support. Also, the Eclipse Help Server selection is no longer provided; the Eclipse Help Server is automatically installed whenever you install a Tivoli Enterprise Portal Server.
7. Click **Next**.

   The Agent Deployment window is displayed. This window lists monitoring agents that you can deploy to remote computers. For this installation, do not select any agents for remote deployment.
8. Click **Next**.
9. If the TEPS Desktop and Browser Signon ID and Password window is displayed, enter and confirm the password to be used for logging on to the Tivoli Enterprise Portal. The default logon user ID, **sysadmin**, cannot be changed on this window.

   This password is required only when Security: Validate User has been enabled on the hub monitoring server. This window is not displayed if the **sysadmin** user ID has already been defined in the operating system.
10. Review the installation summary details. The summary identifies the components you are installing. Click **Next** to begin the installation. The status bar indicates the progress of your installation.

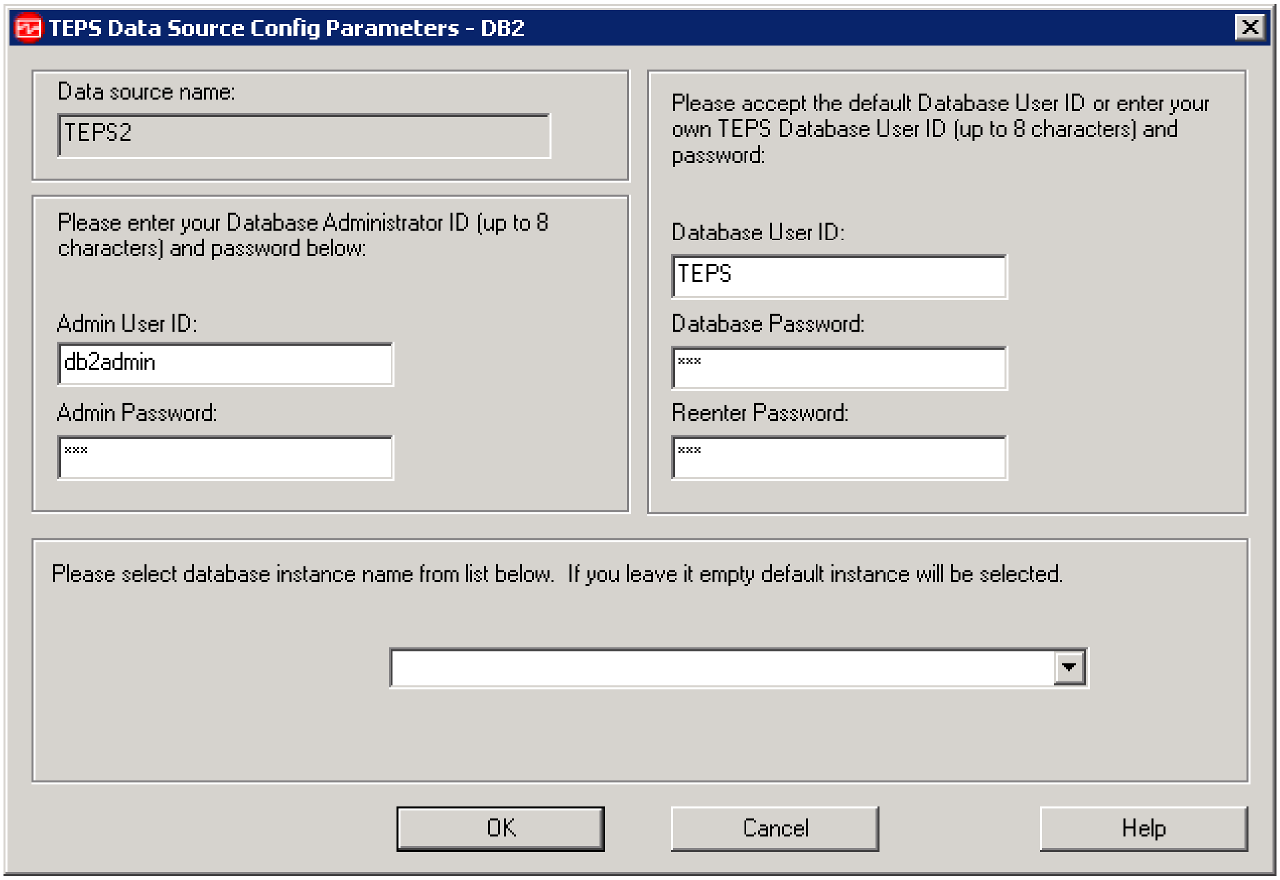
    After the components are installed, a configuration window is displayed.
11. Click **Next** to start configuring all selected components. (Leave all check boxes selected for this installation.)
12. Configure communications for the Tivoli Enterprise Portal Server:
    a. Click **Next** to confirm that you are installing the portal server on this computer. (The host name of this computer is displayed by default.)

b. If no relational database manager can be found in this computer, the embedded Derby RDBMS is used by default. However, if at least one RDBMS product is installed on this computer, a window is displayed for you to choose the RDBMS product you want to use. Choose either **Embedded TEPS database** (that is, Derby), **IBM DB2 Universal Database™ Server**, or **Microsoft SQL Server**, and click **Next**.

c. A window is displayed for you to configure the connection between the portal server and the *portal server database* (TEPS database).

   The installation program uses the information on this window to automatically perform the following tasks:

   - Create the portal server database.
   - Create a database user for the portal server to use to access the database.
   - Configure the ODBC connection between the portal server and the database.



TEPS Data Source Config Parameters - DB2

Data source name: TEPS2

Please enter your Database Administrator ID (up to 8 characters) and password below:

Admin User ID: db2admin

Admin Password: ***

Please accept the default Database User ID or enter your own TEPS Database User ID (up to 8 characters) and password:

Database User ID: TEPS

Database Password: ***

Reenter Password: ***

Please select database instance name from list below. If you leave it empty default instance will be selected.

Figure 32. Configuration window for the portal server database using DB2 for Linux, UNIX, and Windows

Figure 32 shows the configuration window for a portal server database using DB2 for Linux, UNIX, and Windows. The configuration window for a Derby or Microsoft SQL Server database is similar. The fields on the configuration window are described in the following table:

Table 31. Configuration information for the portal server database

Field	DB2 for Linux, UNIX, and Windows default	MS SQL default	Description
Admin User ID	db2admin	sa	The database administrator ID.

Table 31. Configuration information for the portal server database (continued)

Field	DB2 for Linux, UNIX, and Windows default	MS SQL default	Description
Admin Password	(no default)	(no default)	The password for the database administrator ID.
Database User ID	TEPS	TEPS	The login name of the database user that the portal server will use to access the database.
Database Password	(no default)	(no default)	The password for the database login user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Reenter password	(no default)	(no default)	Confirm the password by entering it again.
Select database instance name	(no default)	(no default)	Pull down the drop-down list, and select the appropriate database instance.

d. Optionally change the default values for the administrator ID and database user ID. Enter passwords for the administrator and database user. Click **OK**.

e. Click **OK** on the message that tells you that the portal server configuration was successful.

f. Click **Next** to accept the default Tivoli Data Warehouse user ID and password. (You can change these values in a later step.)

g. In the TEP Server Configuration window, click **OK** to accept the default communications protocol, **IP.PIPE**. IP.PIPE is the protocol that the portal server will use to communicate with the Tivoli Enterprise Monitoring Server.

   A second TEP Server Configuration window is displayed. The **IP.PIPE** area of this window displays the host name of this computer and the default port number, 1918.

h. Click **OK** to accept the default host name and port number.

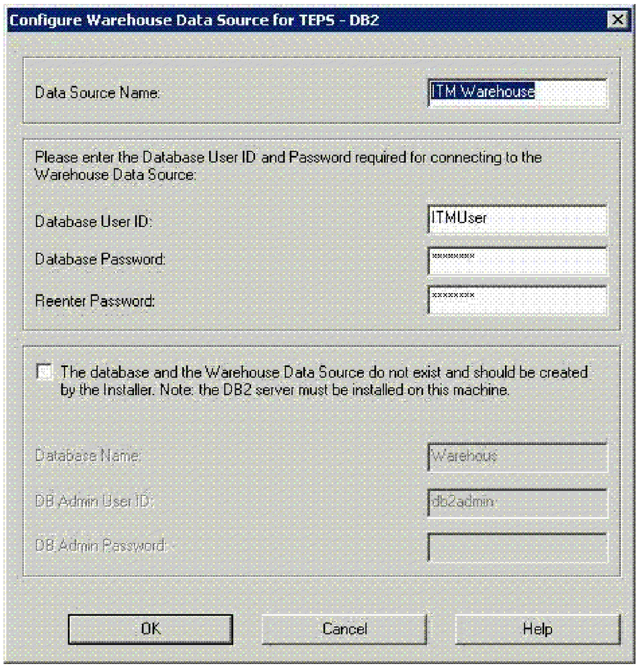
i. Click **Yes** on the message asking if you want to reconfigure the warehouse connection information for the portal server.

j. Select **DB2** or **SQL Server** from the list of RDBMS platforms and click **OK**.

   A window is displayed for you to configure the connection between the portal server and the *Tivoli Data Warehouse database*. (The Tivoli Data Warehouse database is referred to in the title of this window as the data source for the Warehouse Proxy. The Warehouse Proxy Agent sends information collected from monitoring agents to the Tivoli Data Warehouse.)

   The installation program uses the information on this window to automatically perform the following tasks:

   - Create the Tivoli Data Warehouse database.
   - Create a database user (called the *warehouse user*) for the portal server, Warehouse Proxy Agent, and Summarization and Pruning Agent to use to access the warehouse database.
   - Configure the ODBC connection between the portal server and the warehouse database.



Configure Warehouse Data Source for TEPS - DB2

Data Source Name: ITM Warehouse

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

Database User ID: ITMUser

Database Password: ********

Reenter Password: ********

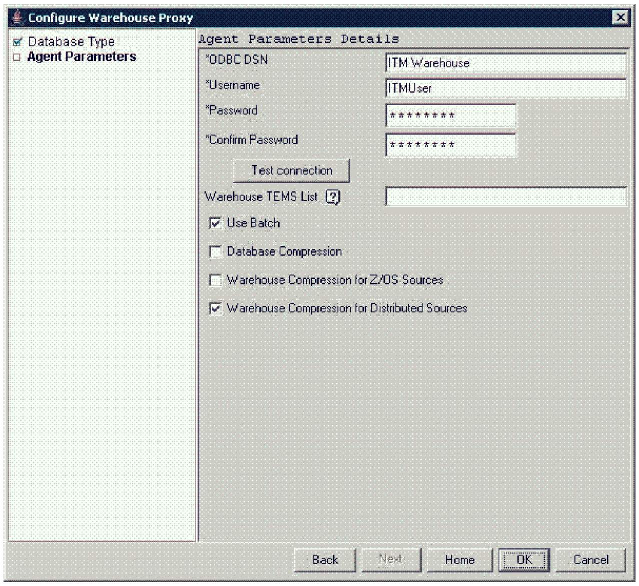
□ The database and the Warehouse Data Source do not exist and should be created by the Installer. Note: the DB2 server must be installed on this machine.

Database Name: Warehous

DB Admin User ID: db2admin

DB Admin Password:

Figure 33. Configuration window for the Tivoli Data Warehouse database using DB2 for Linux, UNIX, and Windows



Configure Warehouse Proxy

☑ Database Type
□ Agent Parameters

Agent Parameters Details

*ODBC DSN: ITM Warehouse

*Username: ITMUser

*Password: ********

*Confirm Password: ********

Warehouse TEMS List:

☑ Use Batch

□ Database Compression

□ Warehouse Compression for Z/OS Sources

☑ Warehouse Compression for Distributed Sources

Figure 34. Configuration window for the Tivoli Data Warehouse database using Microsoft SQL Server

The fields on this window are described in the following table:

Table 32. Configuration information for the Tivoli Data Warehouse database

Field	DB2 for Linux, UNIX, and Windows default	MS SQL default	Description
Data Source Name	ITM Warehouse	ITM Warehouse	The name of the data source. [12j1] [12j2]
Database Name	WAREHOUS	WAREHOUS	The name of the database.
Admin User ID	db2admin	sa	The database administrator ID.
Admin Password	(no default)	(no default)	The password for the database administrator ID.
Database User ID	ITMUser	ITMUser	The login name of the database user that the portal server, Warehouse Proxy Agent, and Summarization and Pruning Agent will use to access the Tivoli Data Warehouse database.
Database Password	itmpswd1	itmpswd1	The password for the database login user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Reenter password	itmpswd1	itmpswd1	Confirm the password by entering it again.

Notes:

1) Warehouse Proxy Agent does not create this DSN - it must already exist. If you are installing Performance Analyzer on the same machine where the Tivoli Enterprise Portal Server is installed, you can use the existing data source created by Tivoli Monitoring. Otherwise, you must create a new System DSN manually, prior to re-configuring Performance Analyzer.

2) On 64-bit versions of Windows, data sources created by the default **ODBC Data Source Administrator** applet available from the **Control Panel** are not available for 32-bit applications. Therefore you must use the 32-bit version of the **ODBC Data Source Administrator** applet from `<WINDOWS>\SysWOW64\odbcad32.exe`.

k. Optionally change the default values on this window. Enter the password of the database administrator. Click **OK**.

l. Click **OK** on the message that tells you that the portal server configuration was successful.

13. If your monitoring server is not local, you must configure communications for the hub Tivoli Enterprise Monitoring Server:

   a. Remove the check mark from **Security: Validate User**; then click **OK** to accept the other defaults on the Tivoli Enterprise Monitoring Server Configuration window:

      - The type of server you are configuring is **Hub**.
      - The default name of the hub monitoring server is HUB_*host_name*, for example, HUB_ITMSERV16.
      - The default communications protocol is **IP.PIPE**. IP.PIPE is the protocol that the monitoring server will use to communicate with the portal server.

        Note: If your site uses the IP.PIPE protocol for communications, be aware of the following limitations:

        – There can be at most 16 IP.PIPE processes per host.

        – IP.PIPE uses one, and only one, physical port per process. Port numbers are allocated using a well-known port allocation algorithm. The first process for a host is assigned port 1918, which is the default.

        – KDC_PORTS is not supported for IP.PIPE.

        If you need to have more than 16 processes per host, use IP.UDP (User Datagram Protocol) for connections*except* the Tivoli Enterprise Portal Server. The use of the User Datagram Protocol is not recommended for the portal server.

      - Do not enable user authentication until you have completed all installation and configuration and verified that the product is working. If you want to enable user authentication, see the *IBM Tivoli Monitoring: Administrator's Guide*.

      The hub Tivoli Enterprise Monitoring Server Configuration window is displayed. The **IP.PIPE** area of this window displays the host name of this computer and the default port number, 1918.

   b. Click **OK** to accept the default host name and port number.

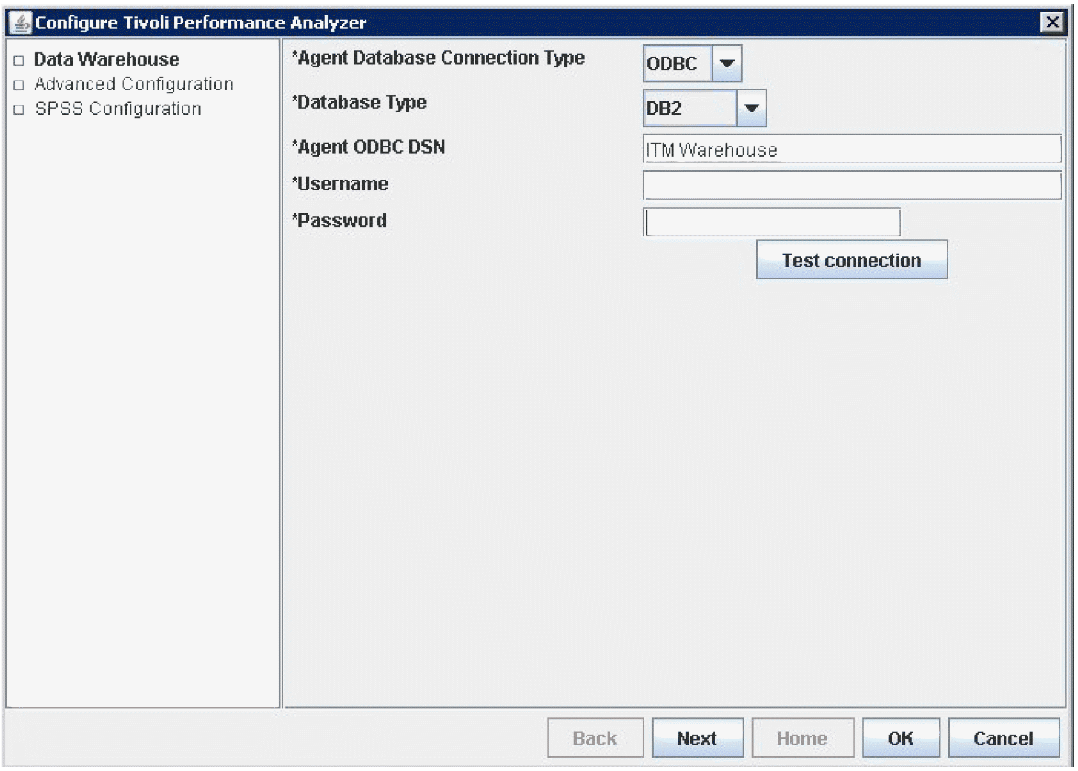
14. If your monitoring server is not local, you must configure the default communication between any IBM Tivoli Monitoring component and the hub Tivoli Enterprise Monitoring Server:

   a. On the configuration window, click **OK** to accept the default communications protocol, **IP.PIPE**.

      A second configuration window is displayed. The **IP.PIPE** area of this window displays the host name of this computer and the default port number, 1918.

   b. Click **OK** to accept the default host name and port number.

15. The **Configure Tivoli Performance Analyzer** window is displayed. Configure the agent for Data Warehouse. Some predefined values will appear depending on whether you choose ODBC or JDBC from the **Agent Database Connection Type** dropdown list.



Configure Tivoli Performance Analyzer

□ Data Warehouse
□ Advanced Configuration
□ SPSS Configuration

*Agent Database Connection Type: ODBC

*Database Type: DB2

*Agent ODBC DSN: ITM Warehouse

*Username:

*Password:

Figure 35. Configure Tivoli Performance Analyzer window - ODBC

If you choose ODBC from the ***Agent Database Connection Type*** dropdown list, you must specify the following:

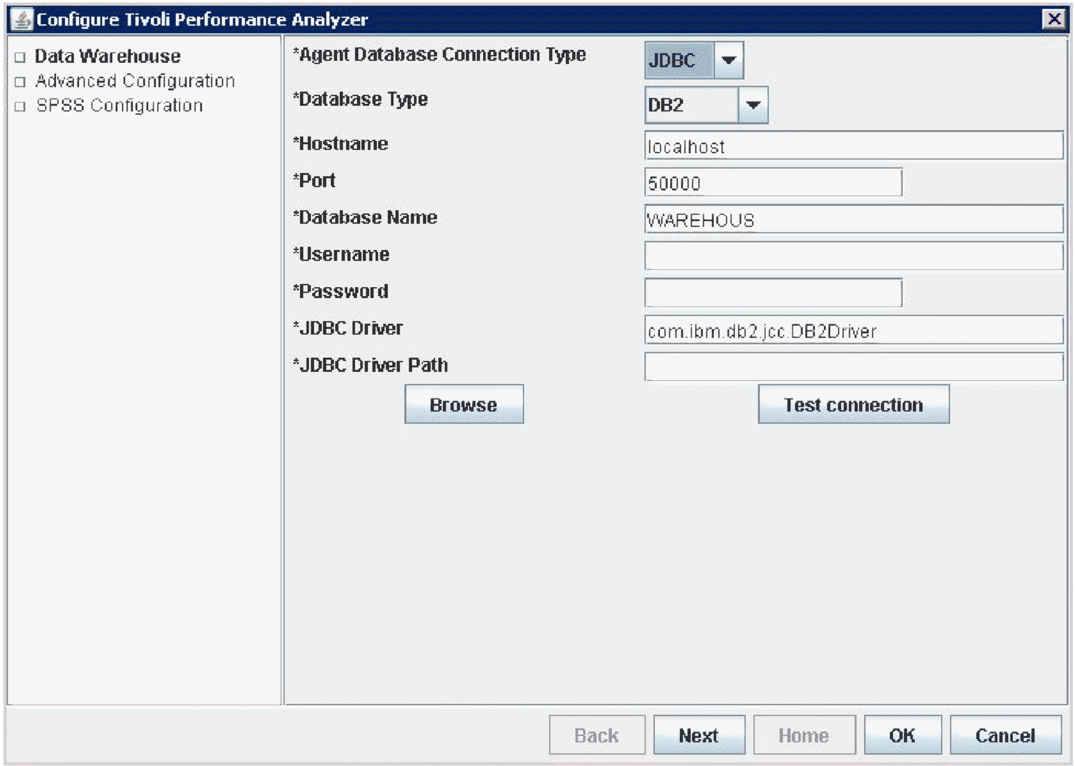
a. Set your **Database Type** to DB2, MSSQL or ORACLE

b. Specify the Data Source Name - **Agent ODBC DSN** (**ITM Warehouse** by default)

   Note:

   - Tivoli Performance Analyzer does not create this DSN - it must already exist. If you are installing Performance Analyzer on the same machine where the Tivoli Enterprise Portal Server is installed, you can use the existing data source created by Tivoli Monitoring. Otherwise, you must create a new System DSN manually, prior to re-configuring Performance Analyzer.
   - On 64-bit versions of Windows, data sources created by the default **ODBC Data Source Administrator** applet available from the **Control Panel** are not available for 32-bit applications. Therefore you must use the 32-bit version of the **ODBC Data Source Administrator** applet from `<WINDOWS>\SysWOW64\odbcad32.exe`.

c. Type the **Username** and **Password**. The entries in these fields are used to connect to the Tivoli Data Warehouse and are the same credentials as those used by the Tivoli Enterprise Portal Server, the Warehouse Proxy Agent and the Summarization and Pruning Agent to communicate with Tivoli Data Warehouse.



*Agent Database Connection Type: JDBC

*Database Type: DB2

*Hostname: localhost

*Port: 50000

*Database Name: WAREHOUS

*Username:

*Password:

*JDBC Driver: com.ibm.db2.jcc.DB2Driver

*JDBC Driver Path:

Figure 36. Configure Tivoli Performance Analyzer window - JDBC

If you choose JDBC from the Agent Database Connection Type dropdown list, you need to specify the following:

a. Set your Database Type to DB2, MSSQL or ORACLE.

b. Provide the **Hostname**. The Hostname is the host where the Tivoli Data Warehouse is installed.

c. Provide the **Port** number.

d. Provide the **Database Name**.

e. Type the **Username** and **Password**.

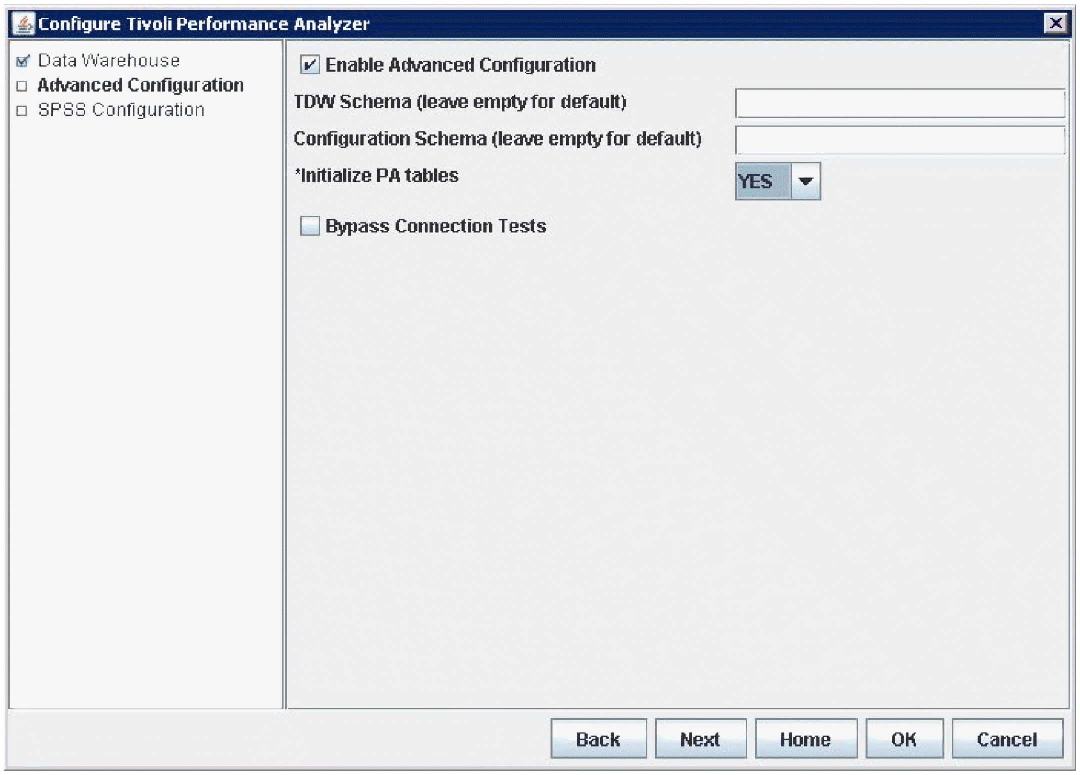
f. Specify the JDBC Driver. The default driver name for DB2 is `com.ibm.db2.jcc.DB2Driver`.

g. Specify the JDBC Driver Path, which should be provided as a list of JAR files with the full path separated by ";".

Note: You can use the **Browse** button in order to specify the path. If you use the **Browse** button a file list is added at the end of the **JDBC Driver Path** text field separated from the existing content by a path separator. The data is used to initialize the database, so it must be provided no matter what Agent Database Connection Type was selected. The driver path should be provided as a list of JAR files with the full path separated by **";"**.

16. Regardless of whether you choose ODBC or JDBC as your **Agent Database Connection Type** you can use the **Test connection** button to check whether the connection can be initiated. Pressing OK button launches more tests. Finishing the installation with incorrect connection settings is impossible unless you use the **Bypass connection tests** option in the **Advanced configuration** window.

17. Click **Next** to proceed to the **Advanced Configuration** window.



☑ Enable Advanced Configuration

TDW Schema (leave empty for default)

Configuration Schema (leave empty for default)

*Initialize PA tables: YES

□ Bypass Connection Tests

Figure 37. Configure Tivoli Performance Analyzer window - Advanced configuration

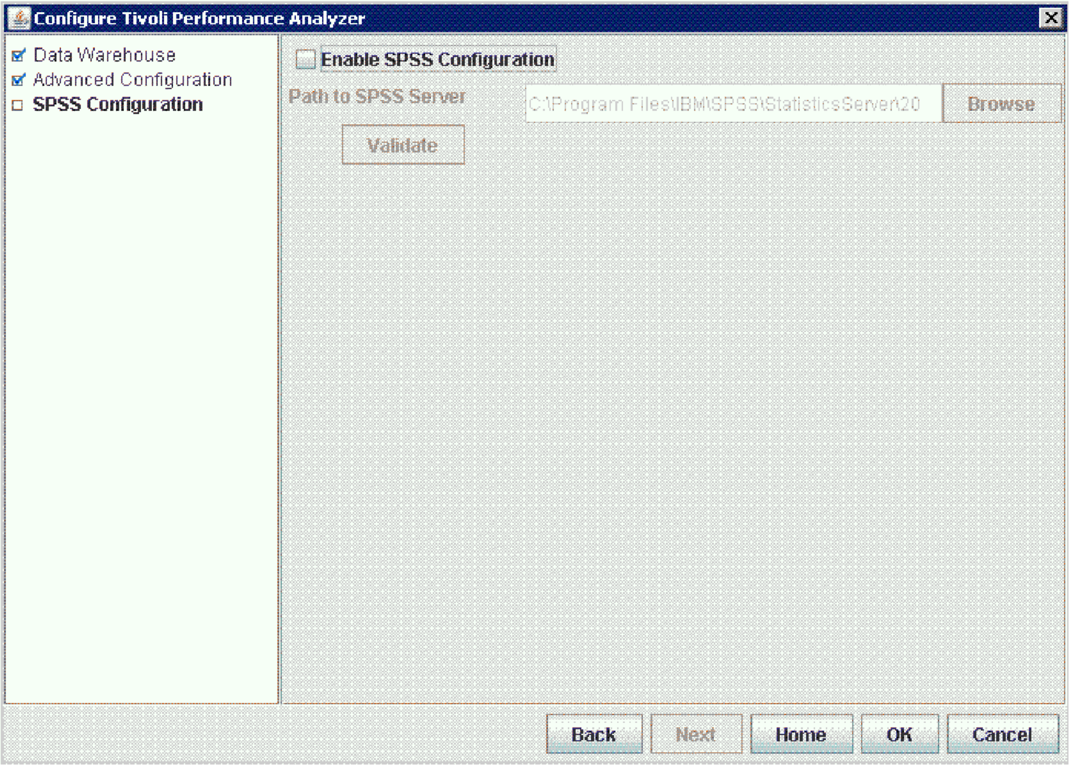
Note: Do not change any of the values displayed in this window unless you are an advanced user. Unless you select **Enable advanced configuration**, all options are greyed out.

- You can enable Advanced Configuration in order to specify **TDW Schema** and **Configuration Schema**.
- You can also choose to have the agent to Initialize PA tables.

  Note: Setting **Initialize PA tables** to **YES** removes and recreates all previously created tables deleting all user tasks and reverting each OS task to its default. If you are upgrading from a previous installation of Tivoli Monitoring, set **Initialize PA tables** to **NO**.
- Use the **Bypass connection tests** option to finish the installation without running connection tests.

18. You can configure Performance Analyzer to use the IBM SPSS Statistics Server to support non-linear trending and forecasting of capacity and performance metrics. To enable SPSS, select **Enable SPSS configuration**.

    Note: The IBM SPSS Statistics Server is a prerequisite for using the non-linear trending feature in Performance Analyzer. SPSS Statistics Server documentation is available at http://publib.boulder.ibm.com/infocenter/spssstat/v20r0m0, or from the Passport Advantage website with part number CI211EN, http://www-01.ibm.com/software/howtobuy/passportadvantage/.


Configure Tivoli Performance Analyzer

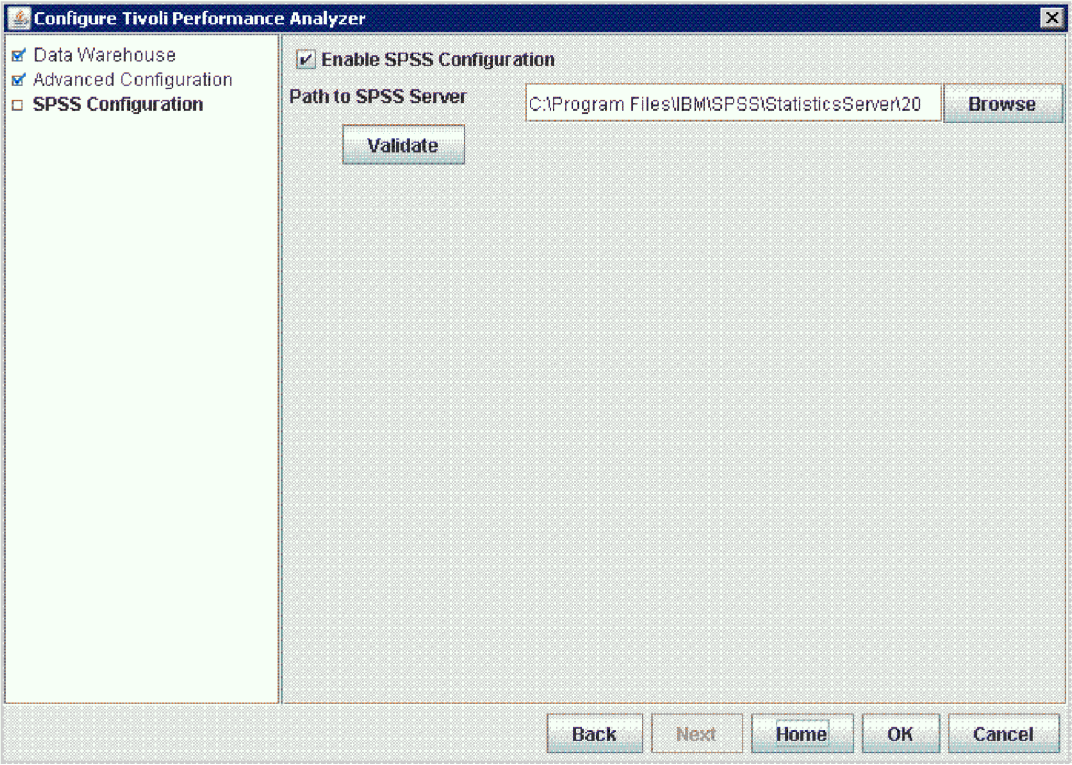
☑ Data Warehouse
☑ Advanced Configuration
□ SPSS Configuration

□ Enable SPSS Configuration

Path to SPSS Server: C:\Program Files\IBM\SPSS\StatisticsServer\20

Figure 38. Configure Tivoli Performance Analyzer SPSS Configuration

19. Browse to the location of your SPSS Statistics Server. Click **Validate** to confirm the path you entered is correct.

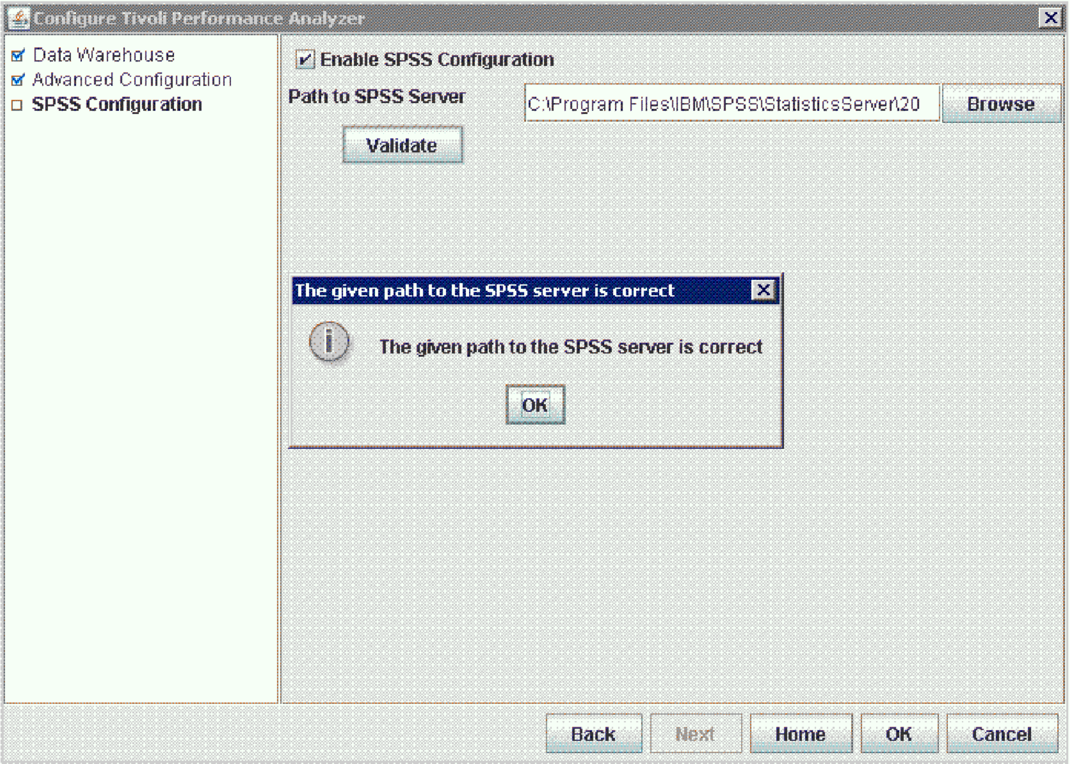


☑ Enable SPSS Configuration

Path to SPSS Server: C:\Program Files\IBM\SPSS\StatisticsServer\20

Figure 39. Configure Tivoli Performance Analyzer Enable SPSS Configuration

20. Click **OK**.



☑ Enable SPSS Configuration

Path to SPSS Server: C:\Program Files\IBM\SPSS\StatisticsServer\20

The given path to the SPSS server is correct

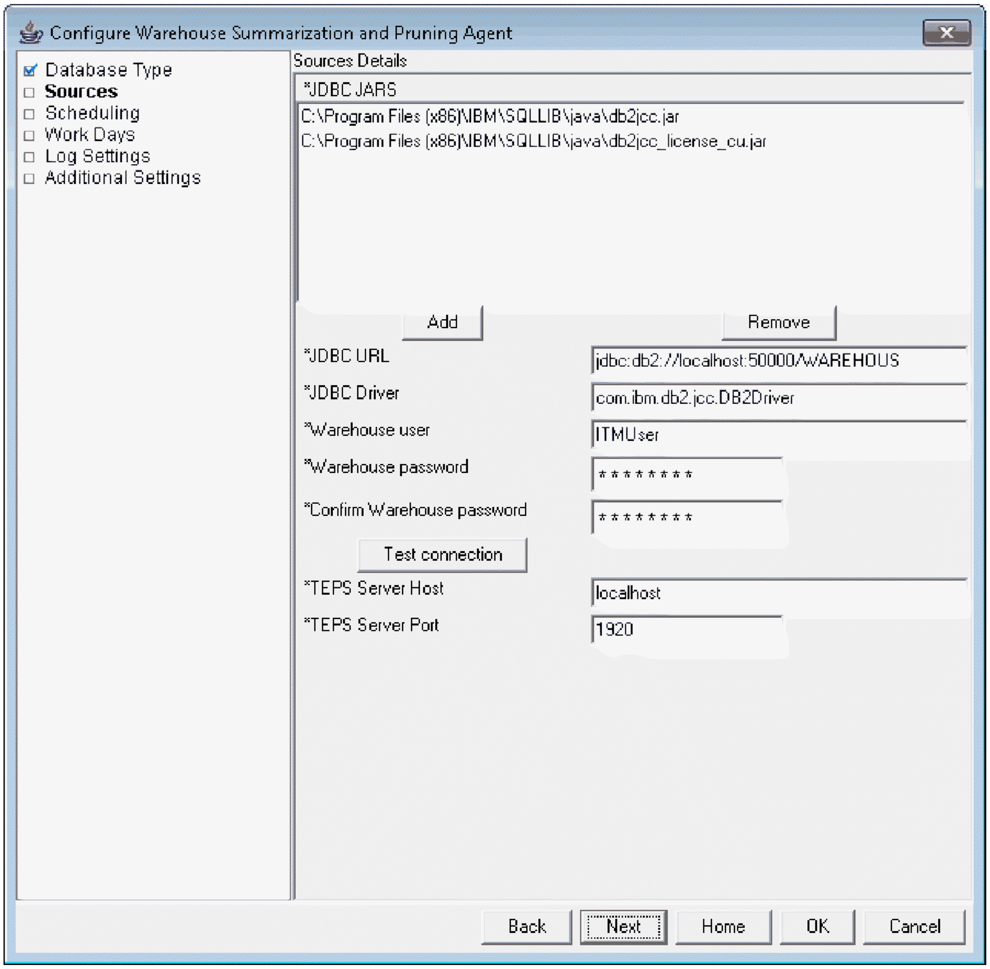
The given path to the SPSS server is correct

Figure 40. Configure Tivoli Performance Analyzer

21. Click **OK**. The system configures and starts the agent.
22. Click **Yes** on the message asking if you want to configure the Summarization and Pruning Agent.

    A multi-tabbed configuration window is displayed with the **Sources** tab at the front.

    Figure 41 on page 199 shows the configuration window for a Summarization and Pruning Agent on Windows (values displayed are for a DB2 for Linux, UNIX, and Windows warehouse database). The configuration window for a Summarization and Pruning Agent on Linux or UNIX is similar.



*JDBC JARS
C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar
C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar

*JDBC URL: jdbc:db2://localhost:50000/WAREHOUS
*JDBC Driver: com.ibm.db2.jcc.DB2Driver
*Warehouse user: ITMUser
*Warehouse password: ********
*Confirm Warehouse password: ********
*TEPS Server Host: localhost
*TEPS Server Port: 1920

Figure 41. Sources pane of Configure Summarization and Pruning Agent window

23. Add the names and directory locations of the JDBC driver JAR files to the **JDBC Drivers** list box:
    a. Click **Add** to display the file browser window. Navigate to the location of the driver files on this computer and select the Type 4 driver files for your database platform. See Table 119 on page 596 for the names and default locations of the driver files to add.
    b. Click **OK** to close the browser window and add the JDBC driver files to the list.

    If you need to delete an entry from the list, select the entry and click **Remove**.
24. The default values for the database platform you selected in the Database Type pane are displayed in the other text fields on the **Sources** pane. Change the default value displayed in the **JDBC URL** field if it is not correct. The following table lists the default Tivoli Data Warehouse URLs for the different database platforms:

Table 33. Tivoli Data Warehouse URLs

Database platform	Warehouse URL
IBM DB2 for Linux, UNIX, and Windows	jdbc:db2://localhost:60000/WAREHOUS
Oracle	jdbc:oracle:thin:@localhost:1521:WAREHOUS
Microsoft SQL Server 2000 or SQL Server 2005	jdbc:sqlserver://localhost:1433;databaseName=WAREHOUS

- If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of `localhost`.
- Change the port number if it is different.
- If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name.

25. Verify the JDBC driver name.

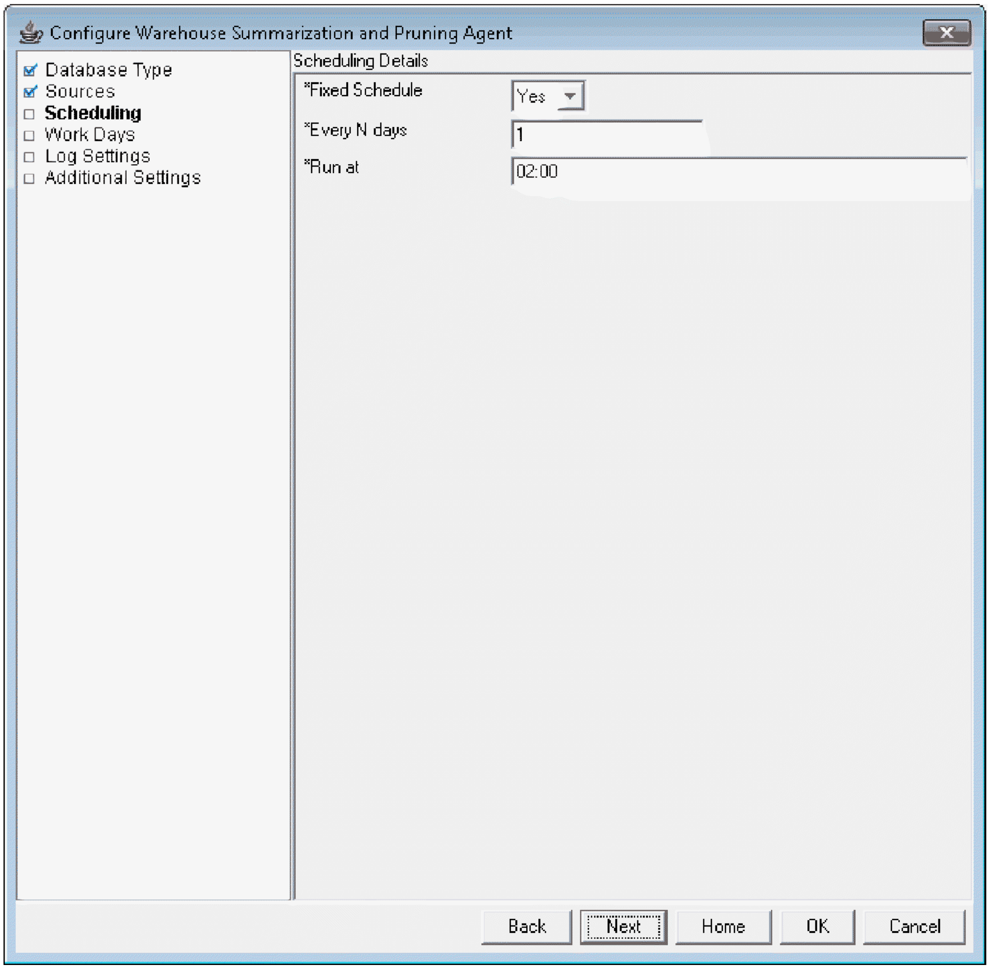
    The following table lists the JDBC Type 4 driver names for each database platform:

Table 34. JDBC driver names

Database platform	JDBC driver name
IBM DB2 for Linux, UNIX, and Windows	com.ibm.db2.jcc.DB2Driver
Oracle	oracle.jdbc.driver.OracleDriver
Microsoft SQL Server	com.microsoft.sqlserver.jdbc.SQLServerDriver **Note:** This is the name of the 2005 SQL Driver. Do not use the SQL Server 2000 JDBC driver, even if the Tivoli Data Warehouse was created in Microsoft SQL 2000. (The name of the 2000 SQL driver was com.microsoft.jdbc.sqlserver.SQLServerDriver. Note the reversal of the string `jdbc.sqlserver`.)

26. If necessary, change the entries in the **Warehouse user** and **Warehouse password** fields to match the user name and password that were created for the Tivoli Data Warehouse. The default user name is `itmuser` and the default password is `itmpswd1`.
27. In the **TEPS Server Host** and **TEPS Server Port** fields, enter the host name of the computer where the Tivoli Enterprise Portal Server is installed and the port number that it uses to communicate with the Summarization and Pruning Agent.

    Note: The default Tivoli Enterprise Portal Server interface port of 15001 is also used after the Summarization and Pruning Agent's initial connection to the portal server over port 1920. Any firewalls between the two need to allow communications on either 15001 or whichever port is defined for any new Tivoli Enterprise Portal Server interface used per the instructions in “Defining a Tivoli Enterprise Portal Server interface on Windows” on page 408.
28. Click **Test connection** to ensure you can communicate with the Tivoli Data Warehouse database.
29. Select the **Scheduling** check box to specify when you want summarization and pruning to take place. You can schedule it to run on a fixed schedule or on a flexible schedule:


Configure Warehouse Summarization and Pruning Agent

- ☑ Database Type
- ☑ Sources
- □ **Scheduling**
- □ Work Days
- □ Log Settings
- □ Additional Settings

Scheduling Details

*Fixed Schedule: Yes

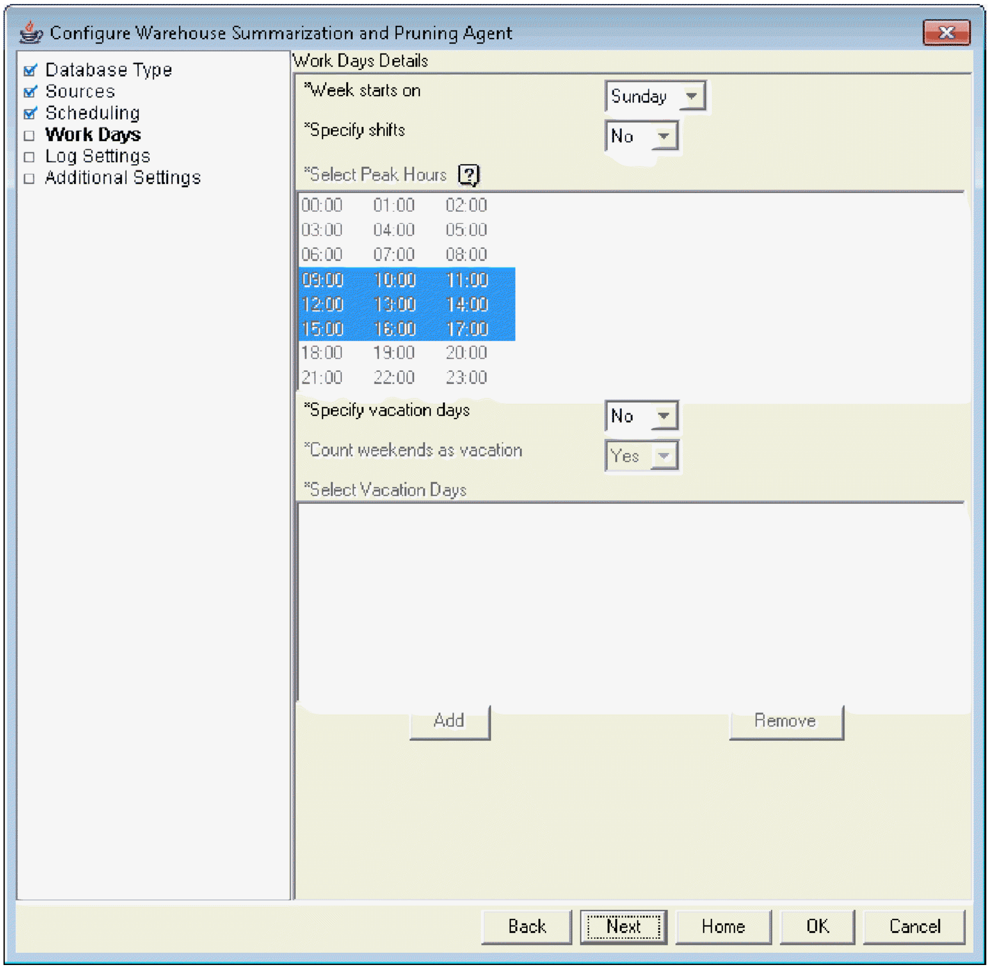
*Every N days: 1

*Run at: 02:00

Figure 42. Scheduling pane of Configure Summarization and Pruning Agent window

Note: If you select Fixed, the Summarization and Pruning Agent does not immediately perform any summarization or pruning when it *starts*. It performs summarization and pruning when it *runs*. It runs according to the schedule you specify on the **Scheduling** pane. If you select Flexible, the Summarization and Pruning Agent runs once immediately after it is started and then at the interval you specified except during any blackout times.

30. Specify shift and vacation settings in the **Work Days** pane:



Configure Warehouse Summarization and Pruning Agent

- ☑ Database Type
- ☑ Sources
- ☑ Scheduling
- □ **Work Days**
- □ Log Settings
- □ Additional Settings

Work Days Details

*Week starts on: Sunday

*Specify shifts: No

*Select Peak Hours

00:00	01:00	02:00
03:00	04:00	05:00
06:00	07:00	08:00
09:00	10:00	11:00
12:00	13:00	14:00
15:00	16:00	17:00
18:00	19:00	20:00
21:00	22:00	23:00

*Specify vacation days: No

*Count weekends as vacation: Yes

*Select Vacation Days

Figure 43. Work Days pane of Summarization and Pruning Agent configuration window

When you enable and configure shifts, IBM Tivoli Monitoring produces three separate summarization reports:

- Summarization for peak shift hours
- Summarization for off-peak shift hours
- Summarization for all hours (peak and off-peak)

Similarly, when you enable and configure vacations, IBM Tivoli Monitoring produces three separate summarization reports:

- Summarization for vacation days
- Summarization for nonvacation days
- Summarization for all days (vacation and nonvacation)

Complete the following steps to enable shifts, vacations, or both:

- Select when the beginning of the week starts.
- To configure shifts:
  a. Select **Yes** in the **Specify shifts** drop-down list.
  b. Optionally change the default settings for peak and off peak hours by selecting hours in the **Select Peak Hours** box.

Note: Changing the shift information after data has been summarized creates an inconsistency in the data. Data that was previously collected is not summarized again to account for the new shift values.

- To configure vacation settings:
  a. Select **Yes** in the **Specify vacation days** drop-down list to enable vacation days.
  b. Select **Yes** in the drop-down list if you want to specify weekends as vacation days.
  c. Select **Add** to add vacation days.
  d. Select the vacation days you want to add from the calendar.

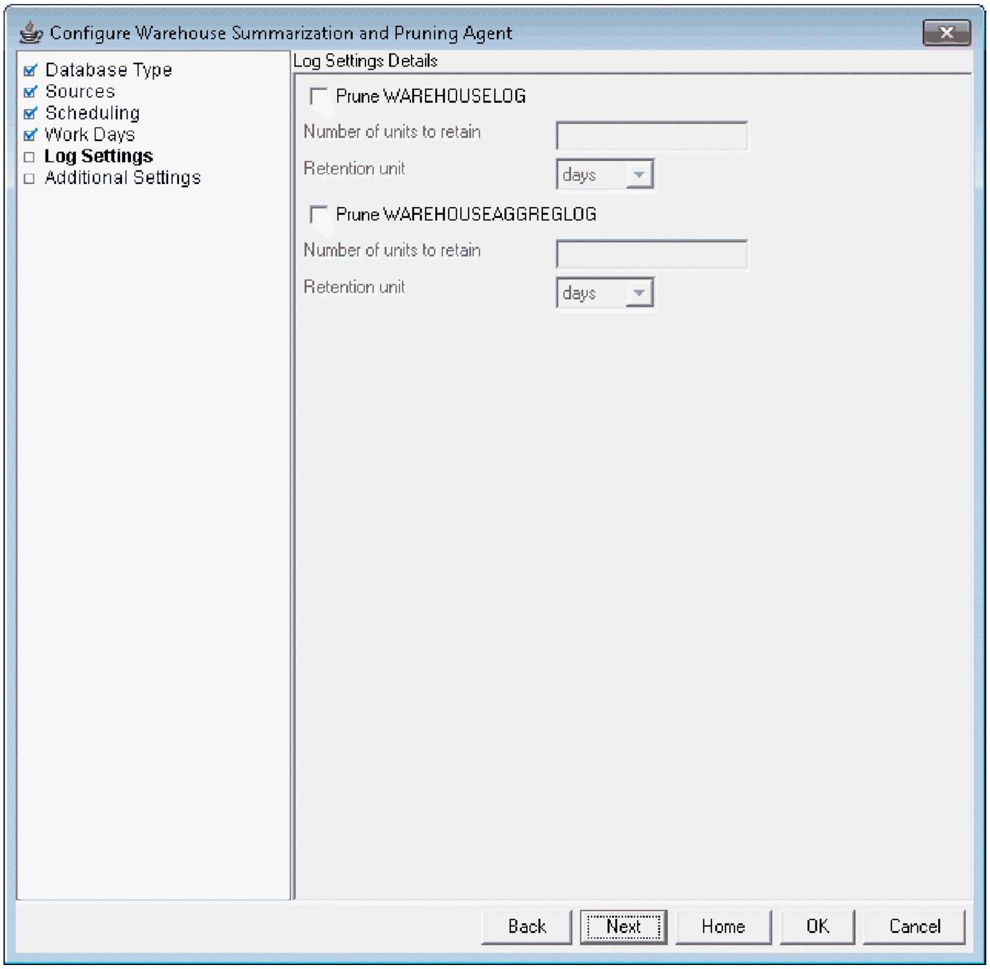
     The days you select are displayed in the list box.

     If you want to delete any days you have previously chosen, select them and click **Delete**.

     Notes:

     1) Add vacation days in the future. Adding vacation days in the past creates an inconsistency in the data. Data that was previously collected is not summarized again to account for vacation days.
     2) Enabling shifts or vacation periods can significantly increase the size of the warehouse database. It will also negatively affect the performance of the Summarization and Pruning Agent.

31. Select the **Log Settings** check box to set the intervals for log pruning:



Configure Warehouse Summarization and Pruning Agent

☑ Database Type
☑ Sources
☑ Scheduling
☑ Work Days
□ **Log Settings**
□ Additional Settings

Log Settings Details

☐ Prune WAREHOUSELOG

Number of units to retain

Retention unit: days

☐ Prune WAREHOUSEAGGREGLOG

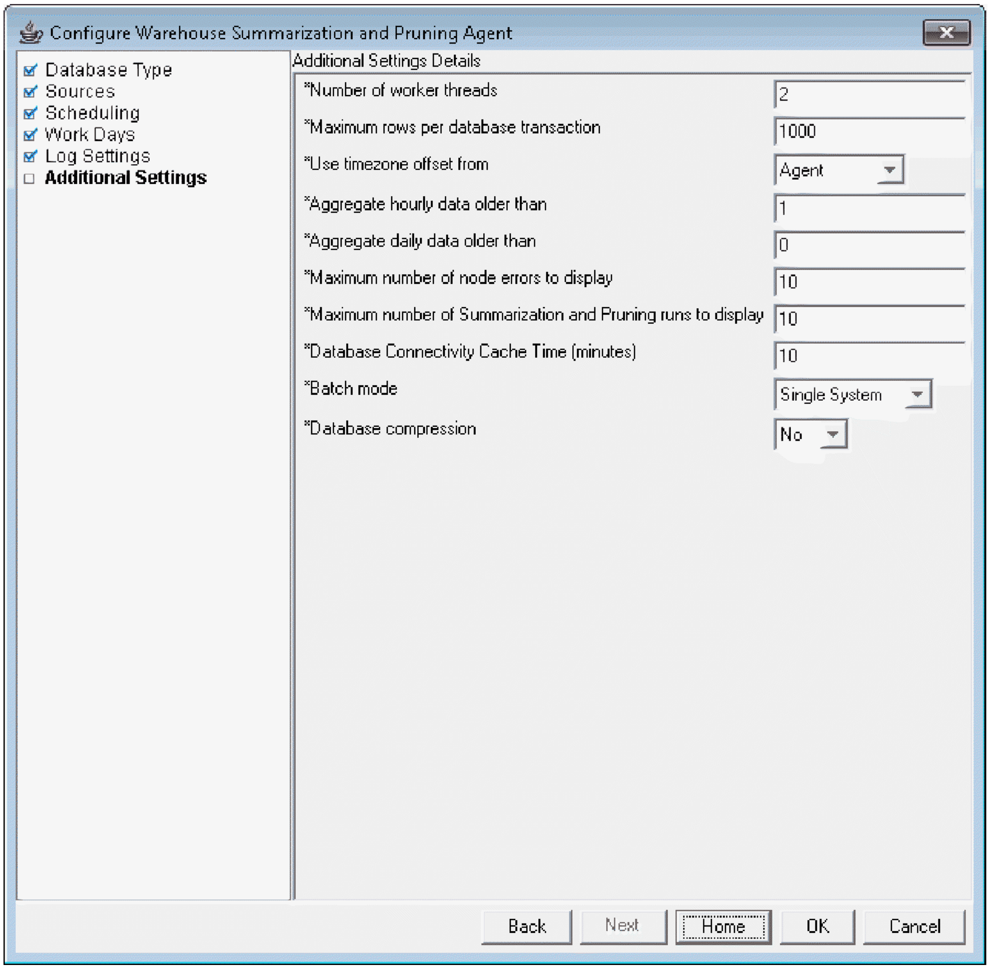
Number of units to retain

Retention unit: days

Figure 44. Log Settings pane of Summarization and Pruning Agent configuration window

- Select Prune WAREHOUSELOG, select the number of units for which data should be kept, and the unit of time (day, month or year).
- Select Prune WAREHOUSEAGGREGLOG, select the number of units for which data should be kept, and the unit of time (day, month or year).

32. Specify additional summarization and pruning settings in the **Additional Settings** pane:



Configure Warehouse Summarization and Pruning Agent

Additional Settings Details

*Number of worker threads: 2

*Maximum rows per database transaction: 1000

*Use timezone offset from: Agent

*Aggregate hourly data older than: 1

*Aggregate daily data older than: 0

*Maximum number of node errors to display: 10

*Maximum number of Summarization and Pruning runs to display: 10

*Database Connectivity Cache Time (minutes): 10

*Batch mode: Single System

*Database compression: No

Figure 45. Additional Settings pane of Summarization and Pruning Agent configuration window

a. Specify the number of additional threads you want to use for handling summarization and pruning processing. The number of threads should be 2 * N, where N is the number of processors running the Summarization and Pruning Agent. A higher number of threads can be used, depending on your database configuration and hardware.

b. Specify the maximum rows that can be deleted in a single pruning transaction. Any positive integer is valid. The default value is `1000`. There is no value that indicates you want all rows deleted.

   If you increase the number of threads, you might consider increasing this value if your transaction log allows for it. The effective number of rows deleted per transaction is based on this value divided by the number of worker threads.

c. Indicate a time zone for historical data from the **Use timezone offset from** drop down list.

   This field indicates which time zone to use when a user specifies a time period in a query for monitoring data.

   - Select **Agent** to use the time zone (or time zones) where the monitoring agents are located.

- Select **Warehouse** to use the time zone where the Summarization and Pruning Agent is located. If the Tivoli Data Warehouse and the Summarization and Pruning Agent are in different time zones, the **Warehouse** choice indicates the time zone of the Summarization and Pruning Agent, not the warehouse.

Skip this field if the Summarization and Pruning Agent and the monitoring agents that collect data are all in the same time zone.

d. Specify the age of the data you want summarized in the **Aggregate hourly data older than** and **Aggregate daily data older than** fields. The default value is `1` for hourly data and `0` for daily data.

e. The **Maximum number of node errors to display** refers to the node error table in the Summarization and Pruning workspace. It determines the maximum number of rows that workspace is to save and display.

f. The **Maximum number of summarization and pruning runs to display** refers to the Summarization and Pruning Run table in the Summarization and Pruning workspace. It determines the maximum number of rows that workspace is to save and display.

   Maximum number of Summarization and Pruning runs to display and Maximum number of node errors to display together determine the number of rows shown in the Summarization and Pruning overall run table and Errors table respectively. There is a minimum value of 10 for each. These equate to keywords KSY_SUMMARIZATION_UNITS and KSY_NODE_ERROR_UNITS in file `KSYENV/sy.ini`.

g. The **Database Connectivity Cache Time** determines how long after a positive check for connectivity that the result will be cached. Longer times may result in inaccurate results in the workspace; however, it saves processing time.

   Database Connectivity Cache Time records the number of minutes to cache the database connectivity for MOSWOS reporting purposes. The minimum value is 5 minutes. This equates to keyword KSY_CACHE_MINS in file `KSYENV/sy.ini`.

h. **Batch mode** determines if data from different managed systems are used in the same database batch; this setting also improves performance.

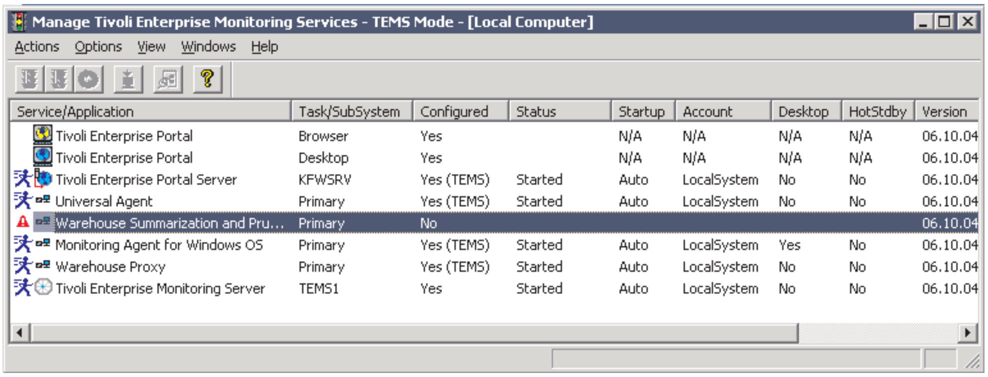
   Batch mode controls the batching method used by the Summarization and Pruning Agent. A value of Single Managed System (0) means that data should only be batched for the same system, whereas a value of Multiple Managed System (1) means that data from multiple systems can be batched together; this can lead to higher performance at potentially bigger transaction sizes. The default value is Single Managed System (0). This equates to keyword KSY_BATCH_MODE in file `KSYENV/sy.ini`.

i. Specify if you want to turn **Database compression** on or off.

To change these values, you can either use the Summarization and Pruning configuration window's **Additional settings** tab or update these parameters directly in file `KSYENV/sy.ini`.

33. Save your settings and close the window. Click **Save** to save your settings. Click **Close** to close the configuration window.
34. Click **Finish** to complete the installation.

When the installation is complete, the Manage Tivoli Enterprise Monitoring Services window is displayed. This window is used for starting, stopping, and configuring installed components. This window can be displayed at any time from the Windows Start menu by clicking **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**.


Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]
Actions Options View Windows Help

Service/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A	N/A	N/A	06.10.04
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	N/A	06.10.04
Tivoli Enterprise Portal Server	KFWSRV	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.04
Universal Agent	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.04
Warehouse Summarization and Pru...	Primary	No						06.10.04
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	No	06.10.04
Warehouse Proxy	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.04
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Started	Auto	LocalSystem	No	No	06.10.04

Figure 46. Manage Tivoli Enterprise Monitoring Services window

The symbol next to a component indicates its current state:

🏃 A blue running figure indicates the component is started.

● A green check mark indicates that the component is configured and can be started.

⚠ A red exclamation mark indicates that the component needs to be configured before it can be started.

Postinstallation procedures

Complete the following tasks after you finish the installation procedure:

- Start the Eclipse Help Server and the Tivoli Enterprise Portal Server, if not started. Right-click the component in the Manage Tivoli Enterprise Monitoring Services window and select **Start**.

  To log on to the Tivoli Enterprise Portal using the desktop client, double-click the Tivoli Enterprise Portal icon on your Windows desktop. Use the **sysadmin** password that you specified in Step 9 on page 188. (For more information about logging on to the Tivoli Enterprise Portal, see “Starting the Tivoli Enterprise Portal client” on page 320.)
- Configure the Summarization and Pruning Agent.

  The installation procedure automatically starts and configures all the monitoring agents that you installed except for the Summarization and Pruning Agent. The Summarization and Pruning Agent is not configured and started during installation to give you an opportunity to configure history collection in advance for all installed monitoring agents, a task that must be performed prior to starting the Summarization and Pruning Agent for the first time.

  To configure and start the Summarization and Pruning Agent, see the following procedures:
  - “Configuring the Summarization and Pruning Agent (JDBC connection)” on page 595
  - “Starting the Summarization and Pruning Agent” on page 608
- If you are using Microsoft SQL Server 2005 for the Tivoli Data Warehouse database, perform the following additional steps:
  - Create a schema with the same name (and owner) as the database user ID for accessing the Tivoli Data Warehouse. (The default user ID is `ITMUser`.) Change the default schema from `dbo` to this database user ID.

    You specified the database user ID on the Configure SQL Data Source for Warehouse Proxy window during installation. See Step 12j of the installation procedure.
  - Ensure that the database is set up to support inbound network TCP/IP connections.
- Enable authentication of user credentials.

  Enable user authentication through either the portal server (for LDAP authentication and single sign-on capability) or the monitoring server (for LDAP or local registry authentication and for SOAP Server commands). See the *IBM Tivoli Monitoring: Administrator's Guide*.

Chapter 9. Installing IBM Tivoli Monitoring

You install the Tivoli Monitoring distributed components and base agents using the Base DVDs or DVD images. The installation and initial configuration of your environment consists of the steps described in Table 35.

Before you begin, take note of the following information concerning the installation procedures in this chapter:

- The installation procedures provide information for installing a single component (such as the monitoring server) on one computer. If you want to install multiple components (such as the monitoring server and the portal server) on the same computer and you want to install them simultaneously, the actual steps might vary. See the individual sections for required configuration information during your installation.
- The installer now validates configuration values for hub, remote, and standby monitoring servers. During configuration, the installer may prompt you to verify values that it considers suspect.
- The installation procedures in this chapter contain instructions for new installations. Follow the same procedures for upgrading or updating your installation. An *upgrade* is an installation that replaces a previous release or fix pack level of the product or component with a later release or fix pack level. An *update* is a modification to an existing installation at the same release or fix pack level.

  For more information on fix pack installation, see “Installing product maintenance” on page 324.
- If your Tivoli Enterprise Portal Server is configured to a DB2 database, and you are upgrading your installation from a previous release, DB2 must be started before you choose to upgrade your Tivoli Enterprise Portal Server.
- See Appendix A, “Installation worksheets,” on page 777 for a set of worksheets that you can use to gather the information required for installation.
- The following sections contain Windows, Linux, and UNIX procedures for installing the various components. Use the procedure that best applies to your environment layout. For example, you can install a monitoring server on UNIX, a portal server on Linux, and a portal desktop client on Windows. To install a monitoring server or monitoring agents on a z/OS system, see the appropriate z/OS documentation as described in Table 35.
- If your site uses the IP.PIPE protocol for communications, be aware of the following limitations:
  - There can be at most 16 IP.PIPE processes per host.
  - IP.PIPE uses one, and only one, physical port per process. Port numbers are allocated using a well-known port allocation algorithm. The first process for a host is assigned port 1918, which is the default.
  - KDC_PORTS is not supported for IP.PIPE.

  If you need to have more than 16 processes per host, use IP.UDP (User Datagram Protocol) for connections between IBM Tivoli Monitoring components.

Table 35. IBM Tivoli Monitoring high-level installation steps

Step	Where to find information
Install the hub Tivoli Enterprise Monitoring Server.	• To install a hub monitoring server on a Windows, Linux, or UNIX system, see “Installing and configuring the hub Tivoli Enterprise Monitoring Server” on page 208. • To install a hub monitoring server on a z/OS system, see the IBM Tivoli Monitoring publication entitled *Configuring Tivoli Enterprise Monitoring Server on z/OS*.

Table 35. IBM Tivoli Monitoring high-level installation steps (continued)

Step	Where to find information
Install any remote monitoring servers.	• To install a remote monitoring server on a Windows, Linux, or UNIX system, see "Installing and configuring the remote monitoring servers" on page 221. • To install a remote monitoring server on a z/OS system, see the IBM Tivoli Monitoring publication entitled *Configuring Tivoli Enterprise Monitoring Server on z/OS*.
Install the Tivoli Enterprise Portal Server.	"Installing the Tivoli Enterprise Portal Server" on page 228
Install monitoring agents.	• To install monitoring agents on Windows, Linux, or UNIX systems, see "Installing monitoring agents" on page 253. • To install monitoring agents on a z/OS system, see the configuration guides for your z/OS agent product. If you plan to use the remote deployment function in IBM Tivoli Monitoring to install distributed monitoring agents across your environment, see Chapter 10, "Deploying monitoring agents across your environment," on page 325 for the required steps. (You cannot use the remote deployment function to install z/OS monitoring agents.) If you used self-describing agents to install your product support, see "Self-describing agent installation" on page 347 to ensure that your environment is prepared for self-describing agents.
Install non-agent bundles (optional).	"Working with non-agent bundles" on page 345
Install the portal desktop client (optional). If you want to use only the browser client, you do not need to install the desktop client.	"Installing the Tivoli Enterprise Portal desktop client" on page 263
Install required application support on the monitoring server, portal server, and portal desktop client.	"Installing and enabling application support" on page 266
Install the language packs for all languages other than English.	"Installing language packs" on page 292
Configure the clients, browser and Java runtime environment.	"Configuring clients, browsers, and JREs" on page 296
Specify what browser to use to display the online help.	"Specifying the browser used for online help" on page 317
Start the Tivoli Enterprise Portal to verify your installation.	"Starting the Tivoli Enterprise Portal client" on page 320
Use Web Start to download the run the desktop client.	"Using Web Start to download and run the desktop client" on page 321
Enable user authentication.	See the *IBM Tivoli Monitoring: Administrator's Guide*

Installing and configuring the hub Tivoli Enterprise Monitoring Server

The following sections provide detailed information for installing and initially configuring the hub monitoring server:

- "Windows: Installing the hub monitoring server" on page 209
- "Linux or UNIX: Installing the hub monitoring server" on page 213

Note: IBM Tivoli Monitoring does not support multiple hub monitoring servers on a single Windows, Linux, or UNIX computer or LPAR.

Windows: Installing the hub monitoring server

Use the following steps to install the hub monitoring server on a Windows computer:

1. Launch the installation wizard by double-clicking the setup.exe file on the Base Infrastructure DVD or DVD image.

   Note: If you are running Windows 2003 or Windows XP and have security set to check the software publisher of applications, you might receive an error stating that the setup.exe file is from an unknown publisher. Click **Run** to disregard this error message.
2. Click **Next** on the Welcome window.

   Note: If you have another IBM Tivoli Monitoring component already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 6.
3. Click **Accept** to accept the license agreement.
4. Choose the directory where you want to install the product. The default directory is C:\IBM\ITM. Click **Next**.
5. Type a 32-character encryption key. You can use the default key.

   Notes:

   a. Do not use any of the following characters in your key:

&	ampersand
|	pipe
'	single quote
=	equal sign
$	dollar sign

   In addition, do not specify double-byte (DBCS) characters.

   b. Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

   Click **Next** and then click **OK** to confirm the encryption key.
6. On the Select Features window, select the check box for **Tivoli Enterprise Monitoring Server**.

   When you select the **Tivoli Enterprise Monitoring Server** check box, all of the check boxes in the attached subtree are automatically selected. The support check boxes in the subtree are for installing application support files for monitoring agents to the monitoring server. You can leave all of the support check boxes selected so you do not need to reconfigure application support as new agent types are added to your environment, but adding support for many agents can extend the installation time considerably. If you do not intend to install a particular type of agent, remove the check mark from the selection box. For detailed information about application support, see “Installing and enabling application support” on page 266.

   For additional information about these components, press the **Help** button.

   Notes:

   a. If you have purchased monitoring agents that run on z/OS, but have not purchased IBM Tivoli Monitoring as a separate product, expand the **Tivoli Enterprise Monitoring Server** node. Clear the check boxes in the subtree for the base agents except for the Warehouse Proxy and Summarization and Pruning Agents. (The *base* monitoring agents are included with the base IBM Tivoli Monitoring installation package: see “Installing monitoring agents” on page 253.)

b. If you are updating an existing installation (you selected **Modify** on the Welcome window), all check boxes on the Select Features window reflect your choices during the initial installation. Clearing a check box has the effect of *uninstalling* the component. Clear a check box only if you want to remove a component.

c. If you have decided to use self-describing agents to install product application support, you do not have to select the support check box for this product.

7. Click **Next** to display the Agent Deployment window.

   The Agent Deployment window lists monitoring agents on this installation image that you can add to the agent depot. The agent depot contains agents that you can deploy to remote computers. For information about how to deploy agents in the agent depot to remote computers, see Chapter 10, "Deploying monitoring agents across your environment," on page 325.

   For additional information about agent deployment, press the **Help** button.

   Note: By default, the agent depot is located in the `itm_installdir/CMS/depot` directory on Windows. If you want to use a different directory, create the directory (if it does not exist) and specify the directory using the DEPOTHOME key in the KBBENV file.

   Select the agents, if any, that you want to add to the agent depot. (You can add agents to the agent depot at a later time by updating your installation.) Click **Next**.

8. If no IBM Tivoli Monitoring component has been previously installed on this computer, a window is displayed for you to select a program folder for the Windows Start menu. Select a program folder, and click **Next**. The default program folder name is IBM Tivoli Monitoring.

9. If the TEPS Desktop and Browser Signon ID and Password window is displayed, enter and confirm the password to be used for logging on to the Tivoli Enterprise Portal. The default logon user ID, **sysadmin**, cannot be changed on this window.

   This password is required only when **Security: Validate Users** is enabled on the hub monitoring server. This window is not displayed if the **sysadmin** user ID has already been defined in the operating system.

10. Review the installation summary details. The summary identifies the components you are installing. Click **Next** to begin the installation.

    After the components are installed, a configuration window (called the Setup Type window) is displayed.

11. Clear the check boxes for any components that have already been installed and configured (at the current release level) on this computer, unless you want to modify the configuration. Click **Next** to start configuring all selected components.

12. Configure the Tivoli Enterprise Monitoring Server:

    a. Select the type of monitoring server you are configuring: **Hub** or **Remote**. For this procedure, select **Hub**.

    b. Verify that the name of this monitoring server is correct in the **TEMS Name** field. If it is not, change it.

       The default name is HUB_*host_name*, for example, HUB_itmserv16. *This name must be unique in the enterprise*.

    c. Identify the communications protocol for the monitoring server. You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify up to three methods for communication. If the method you identify as Protocol 1 fails, Protocol 2 is used as a backup. If Protocol 2 fails, Protocol 3 is used as a backup.

       Do not select any of the other options on this window (for example **Address Translation**, **Tivoli Event Integration Facility** or the option to configure **Hot Standby**). You can configure these options after installation is complete.

       Important: Remove the check mark from the box for **Security: Validate Users**. Use the procedures in the *IBM Tivoli Monitoring: Administrator's Guide* to configure security after installation is complete. If you decide to leave security enabled, and you want to use LDAP to authenticate users instead of the hub security system, use the information in the *IBM Tivoli Monitoring: Administrator's Guide* to complete the configuration.

d. Click **OK**.

e. Complete the following fields for the communications protocol for the monitoring server.

Table 36. Communications protocol settings for the hub monitoring server

Field	Description
IP.UDP Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port # or Port Pools	The listening port for the hub monitoring server. The default port is 1918.
IP.PIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default number is 1918.
IP.SPIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port number	The listening port for the hub monitoring server. The default value is 3660.
SNA Settings	
Network Name	The SNA network identifier for your location.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is CANCTDCS.
TP Name	The transaction program name for the monitoring server.

f. If you are certain that you have typed the values for all of these fields with *exactly* the correct casing (upper and lower cases), you can select **Use case as typed**. However, because IBM Tivoli Monitoring is case-sensitive, consider selecting **Convert to upper case** to reduce the chance of user error.

g. Click **OK** to continue.

For additional information about the monitoring server's configuration parameters, press the **Help** button.

13. If you selected **Tivoli Event Integration Facility**, provide the host name and port number for the TEC event server or Netcool/OMNIbus EIF probe to which you want to forward events and click **OK**.

    The default port number for the TEC event server is 5529 and the default port number for the Netcool/OMNIbus EIF Probe Version 10 or later is 9998. However, if you used Tivoli Business Service Manager V4.2.1 to install the EIF probe then the probe's default port number is 5530.

    Note: Before configuring the Tivoli Event Integration Facility support, verify that you have performed other prerequisite steps to integrate with the event server. See Chapter 25, “Setting up event forwarding to Tivoli Enterprise Console,” on page 643 or Chapter 26, “Setting up event forwarding to Netcool/OMNIbus,” on page 675 for details.

14. Enable application support on the monitoring server.

In Step 6 on page 209, you selected the base monitoring agents for which you wanted to install application support files on the monitoring server. In this step, you activate the application support through a process known as *seeding* the monitoring server.

Note:

- If you are running in a Hot Standby environment, shut down your Hot Standby (that is, mirror) monitoring server before completing this procedure. You may restart the Hot Standby monitoring server only after you have seeded the hub server.
- If you are using self-describing agent capability to install monitoring server support, you might be able to skip this step. This depends on the method of seeding you want to use for a given self-describing agent product. If automatic self-describing agent seeding is required, then you can skip this step. If you want to manually seed the self-describing agent installed product, then you must perform this step. For more information, see "Configuring self-describing agent seeding" on page 217.

a. Specify the location of the monitoring server to which to add application support. You have two choices:

   - **On this computer**
   - **On a different computer**

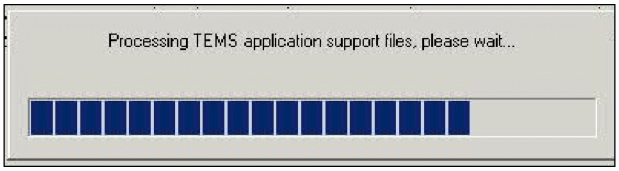
   Click **OK**.

   For additional information about these parameters, press the **Help** button.

b. Click **OK** on the Select the application support to add to the TEMS window.

   This window lists the monitoring agents that you selected in Step 6 on page 209. Click **OK** to begin seeding the monitoring server (using the SQL files listed on this window).

   This process can take up to 20 minutes. As the seeding process completes, a progress bar is displayed, showing the progress of seeding, in turn, the application support for the agents you selected.



Processing TEMS application support files, please wait...

Figure 47. Progress bar for application seeding

   Once seeding completes, if support could not be added, a window is displayed showing all seeding results.

c. Click **Next** on the message that provides results for the process of adding application support (see Figure 66 on page 275). A return code of 0 (rc: 0) indicates that the process succeeded.

   Note: If the Application Support Addition Complete window is not displayed after 20 minutes, look in the IBM\ITM\CNPS\Logs\seedk*pc*.log files (where *pc* is the two-character product code for each monitoring agent) for diagnostic messages that help you determine the cause of the problem. For a list of product codes, see Appendix D, "IBM Tivoli product, platform, and component codes," on page 815.

15. Configure the communication between any IBM Tivoli Monitoring component and the hub monitoring server:

    a. Specify the default values for IBM Tivoli Monitoring components to use when they communicate with the monitoring server.

       1) If agents must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.

2) Identify the type of protocol that the agents use to communicate with the hub monitoring server. You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify up to three methods for communication. If the method you identify as Protocol 1 fails, Protocol 2 is used as a backup. If Protocol 2 fails, Protocol 3 is used as a backup.

   Click **OK**.

b. Complete the communication protocol fields for the monitoring server. See Table 36 on page 211 for definitions of these fields. Click **OK**.

For additional information about these parameters, press the **Help** button.

16. Click **Finish** to complete the installation.
17. Click **Finish** on the Maintenance Complete window if you are updating an existing installation.

The Manage Tivoli Enterprise Monitoring Services utility is opened. (This might take a few minutes.) You can start, stop, and configure IBM Tivoli Monitoring components with this utility.

Linux or UNIX: Installing the hub monitoring server

Use the following steps to install and configure the hub monitoring server on a Linux or UNIX computer.

Table 37. Steps for installing a hub monitoring server on a Linux or UNIX computer

Steps	Where to find information
Install the hub monitoring server.	"Installing the monitoring server"
Configure the hub monitoring server.	"Configuring the hub monitoring server" on page 215
Add application support to the hub monitoring server.	"Adding application support to the hub monitoring server" on page 218

Installing the monitoring server

Use the following steps to install the monitoring server on a Linux or UNIX computer:

1. In the directory where you extracted the installation files, run the following command:

   ```
   ./install.sh
   ```
2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default (/opt/IBM/ITM). If you want to use a different installation directory, type the full path to that directory and press Enter.

   Note: You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.
3. If the directory you specified does not exist, you are asked whether to create it. Type `1` to create this directory.
4. The following prompt is displayed:

   ```
   Select one of the following:
   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   3) Install TEMS support for remote seeding
   4) Exit install.
   Please enter a valid number:
   ```

   Type **1** to start the installation and press Enter.

   The end user license agreement is displayed. Press Enter to read through the agreement.
5. Type **1** to accept the agreement and press Enter.

   If you are upgrading IBM Tivoli Monitoring from global GSKit enabled to local GSKit enabled, the following text is displayed:

```
Local GSKit has been installed for IBM Tivoli Monitoring. The global GSKit installation
will no longer be used by IBM Tivoli Monitoring. It will not be uninstalled automatically,
because it might be used by other applications. You may uninstall it manually, if IBM
Tivoli Monitoring is the only application utilizing it.
```

6. Type a 32-character encryption key and press Enter. If you want to use the default key, press Enter without typing any characters.

   Notes:

   a. Do not use any of the following characters in your key:

&	ampersand
|	pipe
'	single quote
=	equal sign
$	dollar sign

   In addition, do not specify double-byte (DBCS) characters.

   b. Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

   Note: If you are upgrading IBM Tivoli Monitoring instead of a first-time installation, you will be prompted by the installer to upgrade the common prerequisite components for agents and servers.

   The product packages available for this operating system and component support categories are listed.

7. Type the number that corresponds to your operating environment. For a new installation, type **1** `IBM Tivoli Monitoring components for this operating system` installs all components for your operating system. Options 2 - 4 install application support to the server components listed. Select **5** `Other operating systems` to install components for an operating system other than the one detected.
8. Type **4** to install the monitoring server for your current operating system. Press Enter.

   A list of the components to install is displayed.
9. Type **1** to confirm the installation.

   The installation begins.
10. For first-time installations only, when prompted type a name for your monitoring server. For example: HUB_*hostname*. Do not use the fully qualified host name. Press Enter.

    Note: When installing the monitoring server, the application support available on the installation media for the monitoring server is installed automatically.
11. When you finish installing the monitoring server, you are asked if you want to add application support to it.

    Note: If you are using self-describing agent capability to install monitoring server support, you might be able to skip this step. This depends on the method of seeding you want to use for a given self-describing agent product. If automatic self-describing agent seeding is required, then you can skip this step. If you want to manually seed the self-describing agent installed product, then you must perform this step. For more information, see “Configuring self-describing agent seeding” on page 217.
12. After all of the components are installed, you are asked whether you want to install components for a different operating system. Type **2** and press Enter.
13. Installation is complete. If your IBM Tivoli Monitoring environment is not already secured you will be asked at this point if you want to secure it. If your IBM Tivoli Monitoring environment is already secured this question is skipped. The product installation process creates the majority of directories and files with world write permissions. IBM Tivoli Monitoring provides the secureMain utility to help you keep the monitoring environment secure. You can secure your installation now, or manually

execute the secureMain utility later. For more information, see Appendix G, "Securing your IBM Tivoli Monitoring installation on Linux or UNIX," on page 851.

The next step is to configure your monitoring server.

Configuring the hub monitoring server

Use the following steps to configure the hub monitoring server:

Note: If you will be configuring user security, you should use root login ID to configure.

1. At the command-line, change to the /opt/IBM/ITM/bin directory (or the directory where you installed IBM Tivoli Monitoring).
2. Run the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

   where *tems_name* is the name of your monitoring server (for example, HUB_itmdev17).
3. Press Enter to indicate that this is a hub monitoring server (indicated by the *LOCAL default).
4. Press Enter to accept the default host name for the monitoring server. This should be the host name for your computer. If it is not, type the correct host name and then press Enter.
5. Enter the type of protocol to use for communication with the monitoring server. You have four choices: ip.udp, ip.pipe, ip.spipe, or sna. Press Enter to use the default communications protocol (IP.PIPE).
6. If you want to set up a backup protocol, enter that protocol and press Enter. If you do not want to use backup protocol, press Enter without specifying a protocol.
7. Depending on the type of protocol you specified, provide the following information when prompted:

Table 38. UNIX monitoring server protocols and values

Protocol	Value	Definition
IP.UDP	IP Port Number	The port number for the monitoring server. The default is 1918.
SNA	Net Name	The SNA network identifier for your location.
	LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
	Log Mode	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
IP.PIPE	IP.PIPE Port Number	The port number for the monitoring server. The default is 1918.
IP.SPIPE	IP.SPIPE Port Number	The port number for the monitoring server. The default is 3660.

8. Press Enter to *not* specify the name of the KDC_PARTITION. You can configure the partition file at a later time, as described in Appendix C, "Firewalls," on page 799.
9. Press Enter when prompted for the path and name of the KDC_PARTITION.
10. If you want to use Configuration Auditing, press Enter. If you do not want to use this feature, type `2` and press Enter.
11. Press Enter to accept the default setting for the Hot Standby feature (none).

    For best results, wait until after you have fully deployed your environment to configure the Hot Standby feature for your monitoring server. See the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems* for information about configuring Hot Standby.
12. Press Enter to accept the default value for the Optional Primary Network Name (none).
13. Press Enter for the default **Security: Validate User** setting (NO).

    Important: Do *not* change this to set **Security: Validate User**. You can configure security after configuring the monitoring server, as described in the *IBM Tivoli Monitoring: Administrator's Guide*.

14. If you want to forward situation events to either IBM Tivoli Enterprise Console (TEC) or the IBM Tivoli Netcool/OMNIbus console, type `1` and press Enter to enable the Tivoli Event Integration Facility. Complete the following additional steps:
    a. For EIF Server, type the hostname of the TEC event server or the hostname of the Netcool/OMNIbus EIF probe and press Enter.
    b. For EIF Port, type the EIF reception port number for the TEC event server or the Netcool/OMNIbus EIF probe and press Enter.

       The default port number for the Tivoli Enterprise Console event server is 5529 and the default port number for the Netcool/OMNIbus EIF Probe Version 10 or later is 9998. However, if you used Tivoli Business Service Manager V4.2.1 to install the EIF probe then the probe's default port number is 5530.

    Note: Before configuring the Tivoli Event Integration Facility support, verify that you have performed other prerequisite steps to integrate with the event server. See Chapter 25, "Setting up event forwarding to Tivoli Enterprise Console," on page 643 or Chapter 26, "Setting up event forwarding to Netcool/OMNIbus," on page 675 for details.
15. To disable Workflow Policy Activity event forwarding, type **1** and press Enter. Otherwise, press Enter to accept the default value (2=NO). See the note in "Event integration with Tivoli Enterprise Console" on page 644 for more information.
16. Type **6** to accept the default SOAP configuration and exit the configuration.

    Note: You can configure any SOAP information at a later time. See Chapter 16, "Configuring IBM Tivoli Monitoring Web Services (the SOAP Server)," on page 415 for information.

The monitoring server is now configured.

A configuration file is generated in the `*install_dir*/config` directory with the format `*host_name*_ms_*tems_name*.config` (for example, `itmdev17_ms_HUBitmdev17.config`).

Enabling self-describing agent capability at the hub monitoring server

This new feature in V6.2.3 integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure. The self-describing agent feature makes it possible for new or updated Tivoli Monitoring agents to become operational after installation, without having to perform additional product support installation steps. Self-describing agents apply version updates to other components automatically without the need to recycle your hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or remote Tivoli Enterprise Monitoring Server.

Self-describing agent requirements: To take advantage of self-describing agent capability, your Tivoli Management Services server components must be at Version 6.2.3 or higher.

By default, the self-describing agent capability is disabled at the hub monitoring server. Self-describing agent environment variables are enabled by default for remote monitoring servers, portal servers, and self-describing agent enabled agents. However, these components disable self-describing agent capability if connected to a hub monitoring server that has the self-describing agent capability disabled.

Enabling the self-describing agent capability at the hub monitoring server controls the capability across all components. You can disable the capability individually at a remote monitoring server, portal server, or agent, as described in "Enabling or disabling self-describing agent capability" on page 348. However, the best practice is to control the self-describing agent capability from the hub monitoring server by using the **`KMS_SDA`** environment variable.

After installing and configuring the hub monitoring server, start the hub monitoring server if not already running in order to specify the default self-describing agent seeding options. Set the **`KMS_SDA`** environment variable to enable self-describing agent capability. Then recycle the hub monitoring server.

Beginning with V6.2.3 fix pack 1, self-describing agent application support installation is supported in Hot-Standby (FTO) configuration. The **KMS_SDA** settings should match on the primary and standby monitoring servers. An error message will report a discrepancy if the primary and standby monitoring servers have different **KMS_SDA** settings.

Configuring self-describing agent seeding: Two types of installation scenarios are available: a pristine installation and an upgrade installation. A product installation is considered a *pristine installation* if you have no existing monitoring configuration definitions, specifically situation definitions already configured in the Tivoli Enterprise Monitoring Server for that product. A product installation is considered an *upgrade installation* if you have configured monitoring definitions, specifically situation definitions configured for the self-describing agent of the installed product.

For both the pristine installation and upgrade installation, use the **tacmd editSdaOptions** command to configure how the self-describing agent feature seeds product definitions.

Configuring seeding for a pristine installation: Configure one of the following values to define how distribution definitions are applied during the pristine installation of the specified product:

NEW Use the configuration choice of **NEW** to add all product distribution definitions to the Tivoli Enterprise Monitoring Server during a pristine installation.

NONE Use the configuration choice of **NONE** to prevent product distribution definitions from being added to the Tivoli Enterprise Monitoring Server during a pristine installation.

ALL Use the configuration choice of **ALL** to add all product distribution definitions to the Tivoli Enterprise Monitoring Server during a pristine installation.

DISABLE
Use the configuration choice of **DISABLE** to prevent all product monitoring definitions from being added to the Tivoli Enterprise Monitoring Server during a pristine installation.

Configuring seeding for an upgrade installation: Configure one of the following values to define how distribution definitions are applied during the upgrade installation of the product identified:

NEW Use the configuration choice of **NEW** to prevent any product distribution definitions from being added to the Tivoli Enterprise Monitoring Server during a upgrade installation.

NONE Use the configuration choice of **NONE** to prevent any product distribution definitions from being added to the Tivoli Enterprise Monitoring Server during a upgrade installation.

ALL Use the configuration choice of **ALL** to add all product distribution definitions to the Tivoli Enterprise Monitoring Server during a upgrade installation.

DISABLE
Use the configuration choice of **DISABLE** to prevent all product monitoring definitions from being added to the Tivoli Enterprise Monitoring Server during an upgrade installation.

Default installation seeding values: If no seeding values are configured the following default seeding values are used:

- The default installation value for a pristine installation is **ALL**.
- The default installation value for an upgrade installation is **NEW**.

If only a single installation value is provided to the `editSdaOptions` command, the value for the unspecified option is set to the default value, that is, ALL for a pristine installation and NEW for an upgrade installation.

Example of using the tacmd listsdaoptions command when no record is found:

```
tacmd listsdaoptions

KUILS0101I: No records found on the server.

This means default SDA seeding will occur: ALL for pristine install, NEW for upgrade install
```

Example of using the tacmd editsdaoptions command to set default behaviors:

```
tacmd editSdaOptions -t DEFAULT -o SEEDING=DISABLE

No configuration options were found for the specified type. The following options will be created:
PRODUCT INSTALL SEED UPGRADE SEED
DEFAULT DISABLE DISABLE

Are you sure you want to update the selected options? Type Y for yes. Type N for no.
Y

KUIES0150I: The selected SDA configuration options records were successfully updated.
```

Example of using the tacmd editsdaoptions to set options for a specific agent:

```
tacmd editSdaOptions -t NT LZ UX -o INSTALL_SEED=ALL UPGRADE_SEED=NEW

No configuration options were found for the specified type. The following options will be created:
PRODUCT INSTALL SEED UPGRADE SEED
LZ ALL NEW
NT ALL NEW
UX ALL NEW

Are you sure you want to update the selected options? Type Y for yes. Type N for no.
Y

KUIES0150I: The selected SDA configuration options records were successfully updated.
```

Example of using the tacmd listsdaoptions to display existing records:

```
tacmd listsdaoptions
PRODUCT INSTALL SEED UPGRADE SEED
DEFAULT DISABLE DISABLE
LZ ALL NEW
NT ALL NEW
UX ALL NEW
```

For more information on these commands, see the *IBM Tivoli Monitoring: Command Reference*.

Complete the following steps to enable self-describing agent capability:

- On Windows systems:
  1. In the Manage Tivoli Enterprise Monitoring Services application, right-click **Tivoli Enterprise Monitoring Server** and select **Advanced** → **Edit ENV file**. The component environment variable file is displayed.
  2. Change the variable `KMS_SDA=N` to **`KMS_SDA=Y`**.
  3. Save and close the file.
  4. Recycle the **Tivoli Enterprise Monitoring Server** to have your changes take effect.
- On Linux and UNIX systems:
  1. Change to the `<install_dir>/config` directory and open the coordinating file for the monitoring server: `<hostname>_ms_<tems_name>.config`.
  2. Change the variable `KMS_SDA=N` to **`KMS_SDA=Y`**.
  3. Save and close the file.
  4. Recycle the **Tivoli Enterprise Monitoring Server** to have your changes take effect.

Adding application support to the hub monitoring server

All monitoring agents require that application support files be installed on the monitoring servers (hub and remote), portal server, and portal desktop clients in your environment. Application support files contain the information required for agent-specific workspaces, helps, predefined situations, and other data.

When you installed the monitoring server on Linux or UNIX, following the instructions in "Installing the monitoring server" on page 213, the application support files for all base monitoring agents and other supported agents were automatically installed on the monitoring server. This is different to installing the monitoring server on Windows, in that the Linux or UNIX installation always automatically installs the application support files for monitoring agents. This might affect self-describing agents and you should always check the monitoring server and portal server to ensure that self-describing agent products are installed as expected.

The *base* monitoring agents are the monitoring agents included on the IBM Tivoli Monitoring installation media. Now you must enable the application support for those agents. The process of enabling application support is also referred to as *activating* or *adding* application support, and in the case of the monitoring server, as *seeding* the monitoring server.

Follow the instructions in this section to add application support for agents to a monitoring server on Linux or UNIX. For detailed information about application support, and for instructions to install and enable application support for nonbase agents, see "Installing and enabling application support" on page 266 and "Configuring application support for nonbase monitoring agents" on page 270.

Note: If you are running in a Hot Standby environment, shut down your Hot Standby (that is, mirror) monitoring server before completing this procedure. You may restart the Hot Standby monitoring server only after you have seeded the hub server.

Use one of the following procedures to add application support for Base monitoring agents to a monitoring server:

Command-line procedure: Complete the following steps to enable application support on the monitoring server for base monitoring agents, using the Linux or UNIX command-line:

1. Start the monitoring server by running the following command:
   ```
   ./itmcmd server start tems_name
   ```
2. Run the following command to add the application support:
   ```
   ./itmcmd support -t tems_name pc pc pc
   ```
   where *tems_name* is the name of the monitoring server (for example, HUB_itmserv16) and *pc* is the product code for each agent for which you want to enable application support.

   By default, support seeding is skipped if application support is seeded in self-describing mode. To force support seeding over the self-describing mode seeding status, you can pass the additional parameter [-d] to the `./itmcmd support` command.

   To view the product codes for the applications installed on this computer, run the following command:
   ```
   ./cinfo
   ```
   Type `1` when prompted to display the product codes for the components installed on this computer. See your product documentation for the product code for other agents.

   Activate support only for the monitoring agents for which you installed support. For example, if you installed support for the DB2 agent, you would enter the following command:
   ```
   ./itmcmd support -t tems_name ud
   ```
3. Stop the monitoring server by running the following command:
   ```
   ./itmcmd server stop tems_name
   ```
4. Restart the monitoring server by running the following command:
   ```
   ./itmcmd server start tems_name
   ```

GUI procedure: This section describes how to use the Manage Tivoli Enterprise Monitoring Services window on a Linux Intel or UNIX computer to enable application support on a monitoring server that is located on the local computer. You can use this procedure as an alternative to the **itmcmd support**

command. (This command applies only to monitoring servers that are installed on the local computer. To enable application support on a monitoring server that is located on a remote computer, see “Configuring application support on a nonlocal monitoring server from a Linux or UNIX system” on page 285.)

This procedure assumes that you have installed the support files on this computer, and that X Windows is enabled on this computer.

Complete the following steps to enable application support from the Manage Tivoli Enterprise Monitoring Services window on the local Linux or UNIX monitoring server:

1. Log on to the computer where the Tivoli Enterprise Portal Server is installed.
2. Start the Manage Tivoli Enterprise Monitoring Services utility:
   a. Change to the bin directory:
      ```
      cd install_dir/bin
      ```
   b. Run the following command using the parameters described in Table 39:
      ```
      ./itmcmd manage [-h ITMinstall_dir]
      ```

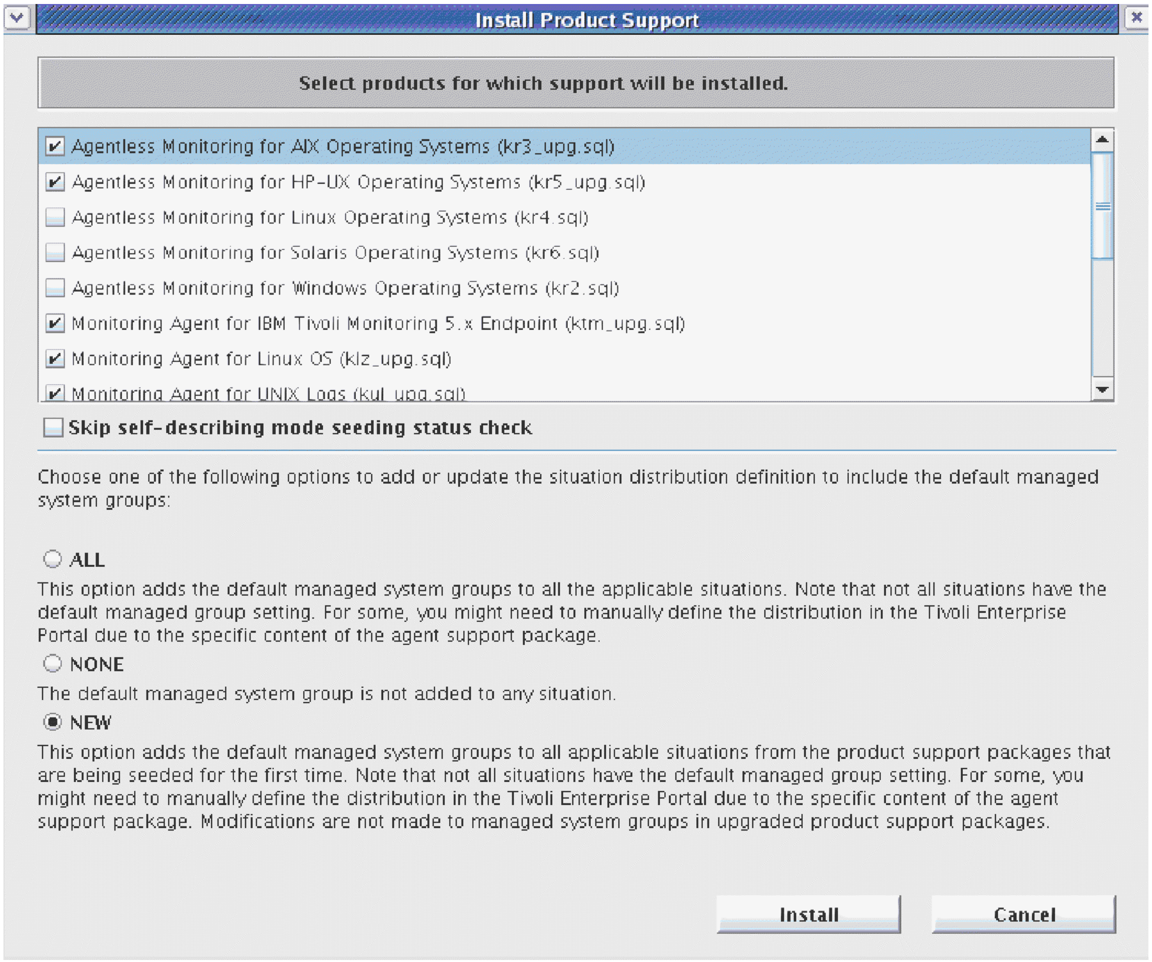
      where:

Table 39. Parameters for the itmcmd manage command

-h	(optional) An option used to specify the installation directory.
ITMinstall_dir	The directory where the monitoring server is installed. The default installation directory is /opt/IBM/ITM.

   The Manage Tivoli Enterprise Monitoring Services window is displayed.
3. Start the monitoring server if it is not already started: Right-click **Tivoli Enterprise Monitoring Server** and click **Start**.
4. Right-click **Tivoli Enterprise Monitoring Server**, and select one of the following options:
   - To enable all application support packages installed on this computer, click **Quick (all available support)**.
   - To select which application support packages you want to enable, click **Advanced . . .**
5. If you selected the **Advanced** option, the Install Product Support window is displayed. For each product-specific support package the installer checks if the Tivoli Enterprise Monitoring Server database was previously seeded with product-specific support in the self-describing mode. If so, the selected support file is excluded from the Tivoli Enterprise Monitoring Server seeding process. If you want to overwrite the support that was seeded in self-describing mode, you can select the option to **Skip self-describing mode seeding status check**.

   Select the application support packages you want to install and click **Install**.



Install Product Support

Select products for which support will be installed.

- ☑ Agentless Monitoring for AIX Operating Systems (kr3_upg.sql)
- ☑ Agentless Monitoring for HP-UX Operating Systems (kr5_upg.sql)
- □ Agentless Monitoring for Linux Operating Systems (kr4.sql)
- □ Agentless Monitoring for Solaris Operating Systems (kr6.sql)
- □ Agentless Monitoring for Windows Operating Systems (kr2.sql)
- ☑ Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint (ktm_upg.sql)
- ☑ Monitoring Agent for Linux OS (klz_upg.sql)
- ☑ Monitoring Agent for UNIX Logs (kul_upg.sql)

□ Skip self-describing mode seeding status check

Choose one of the following options to add or update the situation distribution definition to include the default managed system groups:

○ ALL

This option adds the default managed system groups to all the applicable situations. Note that not all situations have the default managed group setting. For some, you might need to manually define the distribution in the Tivoli Enterprise Portal due to the specific content of the agent support package.

○ NONE

The default managed system group is not added to any situation.

● NEW

This option adds the default managed system groups to all applicable situations from the product support packages that are being seeded for the first time. Note that not all situations have the default managed group setting. For some, you might need to manually define the distribution in the Tivoli Enterprise Portal due to the specific content of the agent support package. Modifications are not made to managed system groups in upgraded product support packages.

Figure 48. Install Product Support window

6. Stop and restart the monitoring server:
   a. Right-click **Tivoli Enterprise Monitoring Server** and click **Stop**.
   b. Right-click **Tivoli Enterprise Monitoring Server** and click **Start**.

Installing and configuring the remote monitoring servers

The following sections provide detailed information for installing and configuring a remote monitoring server:

- "Windows: Installing a remote monitoring server"
- "Linux or UNIX: Installing a remote monitoring server" on page 225

Windows: Installing a remote monitoring server

Use the following steps to install a remote monitoring server on a Windows computer:

1. Launch the installation wizard by double-clicking the setup.exe file in the Infrastructure DVD or DVD image.

   Note: If you are running Windows 2003 or Windows XP and have security set to check the software publisher of applications, you might receive an error stating that the setup.exe file is from an unknown publisher. Click **Run** to disregard this error message.
2. Click **Next** on the Welcome window.

Note: If you have another IBM Tivoli Monitoring component already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 6.

3. Click **Accept** to accept the license agreement.
4. Choose the directory where you want to install the product. The default directory is C:\IBM\ITM. Click **Next**.

   Note: If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```
5. Type a 32-character encryption key. You can use the default key.

   Notes:

   a. Do not use any of the following characters in your key:

&	ampersand
|	pipe
'	single quote
=	equal sign
$	dollar sign

   In addition, do not specify double-byte (DBCS) characters.

   b. Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

   Click **Next** and then click **OK** to confirm the encryption key.
6. On the Select Features window, select the check box for **Tivoli Enterprise Monitoring Server**.

   When you select the **Tivoli Enterprise Monitoring Server** checkbox, all of the check boxes in the attached subtree are automatically selected. These check boxes are for installing application support files for base monitoring agents and other to the monitoring server. (The *base* monitoring agents are included with the base IBM Tivoli Monitoring installation package.) If you leave all of the application support check boxes selected, you do not need to reconfigure application support as new agent types are added to your environment. However, installing support for many agents at a time can increase the installation time and you may still have to add support for an agent later if it has been updated. For detailed information about application support, see "Installing and enabling application support" on page 266.

   Notes:

   a. If you have purchased monitoring agents that run on z/OS, but have not purchased IBM Tivoli Monitoring as a separate product, expand the **Tivoli Enterprise Monitoring Server** node. Clear all check boxes in the subtree except the check boxes labeled **Tivoli Enterprise Monitoring Server** and **Summarization and Pruning Agent**.

   b. If you are updating an existing installation (you selected **Modify** on the Welcome window), all check boxes on the Select Features window reflect your choices during the initial installation. Clearing a check box has the effect of *uninstalling* the component. Clear a check box only if you want to remove a component.
7. If you want to install any agents on this remote monitoring server, expand **Tivoli Enterprise Monitoring Agents** and select the agents you want to install.
8. Click **Next** to display the Agent Deployment window.

   The Agent Deployment window lists monitoring agents on this installation image that you can add to the agent depot. The agent depot contains agents that you can deploy to remote computers. For information about how to deploy agents in the agent depot to remote computers, see Chapter 10, "Deploying monitoring agents across your environment," on page 325.

Note: By default, the agent depot is located in the *itm_installdir*/CMS/depot directory on Windows. If you want to use a different directory, create the directory (if it does not exist) and specify the directory using the DEPOTHOME key in the KBBENV file.

Select the agents, if any, that you want to add to the agent depot. (You can add agents to the agent depot at a later time by updating your installation.) Click **Next**.

9. If no IBM Tivoli Monitoring component has been previously installed on this computer, a window is displayed for you to select a program folder for the Windows Start menu. Select a program folder and click **Next**. The default program folder name is IBM Tivoli Monitoring.
10. If the TEPS Desktop and Browser Signon ID and Password window is displayed, enter and confirm the password to be used for logging on to the Tivoli Enterprise Portal. The default logon user ID, **sysadmin**, cannot be changed on this window. The logon password must match the password that you specified for **sysadmin** when you configured the hub monitoring server.

    This window is not displayed if the **sysadmin** user ID has already been defined in the operating system.
11. Review the installation summary details. The summary identifies the components you are installing. Click **Next** to begin the installation.

    After the components are installed, a configuration window (called the Setup Type window) is displayed.
12. Clear the check boxes for any components that have already been installed and configured (at the current release level) on this computer, unless you want to modify the configuration. Click **Next** to start configuring all selected components.
13. Configure the Tivoli Enterprise Monitoring Server:
    a. Select the type of monitoring server you are configuring: **Hub** or **Remote**. For this procedure, select **Remote**.
    b. Verify that the name of this monitoring server is correct in the **TEMS Name** field. If it is not, change it.

       The default name is REMOTE_*host_name*, for example, REMOTE_itmserv16. *This name must be unique in the enterprise*.
    c. Identify the communications protocol for the monitoring server. You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify up to three methods for communication. If the method you identify as Protocol 1 fails, Protocol 2 is used as a backup. If Protocol 2 fails, Protocol 3 is used as a backup.
    d. Click **OK**.
    e. Complete the following fields for the communications protocol for the monitoring server.

Table 40. Remote monitoring server communications protocol settings

Field	Description
IP.UDP Settings: Primary Hub TEMS	
Host name or IP Address	The host name or IP address for the hub monitoring server.
Port # or Port Pools	The listening port for the hub monitoring server. The default port is 1918.
IP.PIPE Settings: Primary Hub TEMS	
Host name or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default value is 1918.
IP.SPIPE Settings: Primary Hub TEMS	
Host name or IP Address	The host name or IP address for the hub monitoring server.

Table 40. Remote monitoring server communications protocol settings (continued)

Field	Description
Port Number	The listening port for the monitoring server. The default value is 3660.
SNA Settings: Remote TEMS	
Local LU Alias	The LU alias.
TP Name	The transaction program name for this monitoring server.
SNA Settings: Primary Hub TEMS	
Network Name	The SNA network identifier for your location.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
TP Name	The transaction program name for the monitoring server.

f. If you are certain that you have typed the values for all of these fields with *exactly* the correct casing (upper and lower cases), you can select **Use case as typed**. However, because IBM Tivoli Monitoring is case-sensitive, consider selecting **Convert to upper case** to reduce the chance of user error.

g. Click **OK** to continue.

14. Enable application support on the monitoring server.

In Step 6 on page 222, you selected the base monitoring agents for which you wanted to install application support files on the monitoring server. In this step, you activate the application support through a process known as *seeding* the monitoring server.

a. Specify the location of the monitoring server to which to add application support. You have two choices:

- **On this computer**
- **On a different computer**

Click **OK**.

For additional information about these parameters, press the **Help** button.

b. Click **OK** on the Select the application support to add to the TEMS window.

This window lists the monitoring agents that you selected in Step 6 on page 222. Click **OK** to begin seeding the monitoring server (using the SQL files listed on this window).

This process can take up to 20 minutes. As the seeding process completes, a progress bar is displayed, showing the progress of seeding, in turn, the application support for the agents you selected. Once seeding completes, if support could not be added, a window is displayed showing all seeding results.

c. Click **Next** on the message that provides results for the process of adding application support (see Figure 66 on page 275). A return code of 0 (rc: 0) indicates that the process succeeded.

Note: If the Application Support Addition Complete window is not displayed after 20 minutes, look in the IBM\ITM\CNPS\Logs\seedk*pc*.log files (where *pc* is the two-character product code for each monitoring agent) for diagnostic messages that help you determine the cause of the problem. For a list of product codes, see Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815.

15. Configure the communication between any IBM Tivoli Monitoring component and the monitoring server:

a. Specify the default values for IBM Tivoli Monitoring components to use when they communicate with the monitoring server.

   1) If agents must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.

   2) Identify the type of protocol that the agents use to communicate with the hub monitoring server. You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify up to three methods for communication. If the method you identify as Protocol 1 fails, Protocol 2 is used as a backup. If Protocol 2 fails, Protocol 3 is used as a backup.

   Click **OK**.

b. Complete the communication protocol fields for the monitoring server. See Table 40 on page 223 for definitions of these fields. Click **OK**.

For additional information about these parameters, press the **Help** button.

16. Click **Finish** to complete the installation.
17. Click **Finish** on the Maintenance Complete window if you are updating an existing installation.

Note: IBM Tivoli Monitoring does not support multiple remote monitoring servers on the same Windows computer.

Linux or UNIX: Installing a remote monitoring server

Use the following steps to install and configure the remote monitoring server on a Linux or UNIX computer.

Table 41. Steps for installing a remote monitoring server on a Linux or UNIX computer

Steps	Where to find information
Install the remote monitoring server. Use the same instructions as for installing the hub monitoring server.	"Installing the monitoring server" on page 213
Configure the remote monitoring server.	"Configuring the remote monitoring server"
Add application support to the remote monitoring server. Use the same instructions as for adding application support to the hub monitoring server.	"Adding application support to the hub monitoring server" on page 218

Note: Under both Linux and UNIX, IBM Tivoli Monitoring (version 6.1 fix pack 6 and subsequent) supports multiple remote monitoring servers on the same LPAR or computer (however, it does not support multiple *hub* monitoring servers on the same computer). Note that each instance of a remote monitoring server must have its own network interface card and its own unique IP address; in addition, each monitoring server must be installed on its own disk. These limitations isolate each remote monitoring server instance so you can service each one independently: you can upgrade a remote Tivoli Enterprise Monitoring Server without affecting another server's code base and shared libraries.

Configuring the remote monitoring server

Use the following steps to configure the remote monitoring server:

1. At the command-line change to the /opt/IBM/ITM/bin directory (or the directory where you installed IBM Tivoli Monitoring).
2. Run the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

   where *tems_name* is the name of your monitoring server (for example, remote_itmdev17).
3. Type `remote` to indicate that this is a remote monitoring server.
4. Press Enter to accept the default host name for the hub monitoring server. This should be the host name for hub computer. If it is not, type the correct host name and then press Enter.

5. Enter the type of protocol to use for communication with the monitoring server. You have four choices: ip.udp, ip.pipe, ip.spipe, or sna. Press Enter to use the default communications protocol (IP.PIPE).
6. If you want to set up a backup protocol, enter that protocol and press Enter. If you do not want to use backup protocol, press Enter without specifying a protocol.
7. Depending on the type of protocol you specified, provide the following information when prompted:

Table 42. UNIX monitoring server protocols and values

Protocol	Value	Definition
IP.UDP	IP Port Number	The port number for the monitoring server. The default is 1918.
SNA	Net Name	The SNA network identifier for your location.
	LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
	Log Mode	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
IP.PIPE	IP.PIPE Port Number	The port number for the monitoring server. The default is 1918.
IP.SPIPE	IP.SPIPE Port Number	The port number for the monitoring server. The default is 3660.

8. Press Enter to *not* specify the name of the KDC_PARTITION.

   Note: You can configure the partition file at a later time, as described in Appendix C, “Firewalls,” on page 799.
9. Press Enter when prompted for the path and name of the KDC_PARTITION.
10. If you want to use Configuration Auditing, press Enter. If you do not want to use this feature, type `n` and press Enter.
11. Press Enter to accept the default setting for the Hot Standby feature (none).

    For best results, wait until after you have fully deployed your environment to configure the Hot Standby feature for your monitoring server. See the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems* for information about configuring Hot Standby.
12. Press Enter to accept the default value for the Optional Primary Network Name (none).
13. Press Enter for the default **Security: Validate User** setting (NO).

    Note: This option is valid only for a hub monitoring server.
14. For Tivoli Event Integration, type `2` and press Enter.

    Note: This option is valid only for a hub monitoring server.
15. At the prompt asking if you want to disable Workflow Policy/Tivoli Emitter Agent event forwarding, press Enter to accept the default (2=NO).

    Note: This option is valid only for a hub monitoring server.
16. At the prompt asking if you want to configure the SOAP hubs, press Enter to save the default settings and exit the installer.

    Note: This option is valid only for a hub monitoring server.

The monitoring server is now configured.

A configuration file is generated in the *install_dir*/config directory with the format *host_name*_ms_*tems_name*.config (for example, itmdev17_ms_HUBitmdev17.config).

Remote copy and seeding of the application support from the hub monitoring server to distributed monitoring servers

If your requirements meet the following conditions, go to "Remote seeding from the hub monitoring server":

- You will not be using cinfo to report on installed application support.
- You will not be using itmcmd support to perform local seeding on other monitoring servers.
- You will be using itmcmd manage to perform remote seeding from the hub monitoring server.

If your requirements meet the following conditions, go to "Local seeding on distributed monitoring servers" on page 228:

- You will be using cinfo to report on installed application support.
- You will be using itmcmd support to perform local seeding on other monitoring servers.
- You will not be using itmcmd manage to perform remote seeding from the hub monitoring server.

Remote seeding from the hub monitoring server: The following conditions apply for remote seeding from the hub monitoring server:

- Do not use the *FTP Catalog and Attribute files* option on the Manage Tivoli Enterprise Monitoring Services window. This option is only for sending files by FTP to z/OS systems.
- You must perform this task manually using a utility of your choice, such as FTP, SFTP, SCP, and so on.
- Replace all instances of <TEMSNAME> in these instructions with the name of the monitoring server on your distributed system.

You must copy the required files from the hub monitoring server to the other monitoring servers:

1. Copy the CAT and ATR files:
   a. Copy the CAT files from $CANDLEHOME/tables/ciCATrsq/RKDSCATL on the hub monitoring server to $CANDLEHOME/tables/ciCATrsq/RKDSCATL on the other monitoring server system.
   b. Copy the CAT files from $CANDLEHOME/tables/ciCATrsq/RKDSCATL on the other monitoring server system to $CANDLEHOME/tables/<monitoring serverNAME>/RKDSCATL on the same monitoring server system.
   c. Copy the ATR files from $CANDLEHOME/tables/ciCATrsq/ATTRLIB on the hub monitoring server to $CANDLEHOME/tables/ciCATrsq/ATTRLIB on the other monitoring server system.
   d. Copy the ATR files from $CANDLEHOME/tables/ciCATrsq/ATTRLIB on the other monitoring server system to $CANDLEHOME/tables/<monitoring serverNAME>/ATTRLIB on the same monitoring server system.
2. If you use Tivoli Enterprise Console servers, copy the .map and .baroc files:
   a. Copy the baroc and map files from $CANDLEHOME/tables/ciCATrsq/TECLIB on the hub monitoring server to $CANDLEHOME/tables/ciCATrsq/TECLIB on the other monitoring server system.
   b. Copy the baroc and map files from $CANDLEHOME/tables/ciCATrsq/TECLIB on the other monitoring server system to $CANDLEHOME/tables/<monitoring serverNAME>/TECLIB on the same monitoring server system.
3. Use the Manage Tivoli Enterprise Monitoring Services window to perform the seeding. On Linux/UNIX systems, run the itmcmd manage command to display the Manage Tivoli Enterprise Monitoring Services window:
   a. In the Manage Tivoli Enterprise Monitoring Services window, select the Tivoli Enterprise Monitoring Server.
   b. Click **Actions** → **Install Product Support**.
   c. A pop-up window is displayed with the options **On this computer** and **On a different computer**. Click **On a different computer** and then click **OK**.

d. Click **OK** on the pop-up window after you have ensured that the monitoring server on the other machine is running.

e. The **Non-Resident monitoring server Connection** window is displayed. Complete the information in this window and click **OK**.

f. A second **Non-Resident monitoring server Connection** window is displayed. Enter the information required and click **OK**.

g. If all of the connection information is correct a list of products available for seeding is displayed. Select the desired products from the list and execute the seeding.

h. Remember to recycle the monitoring server on the system that has been updated.

Local seeding on distributed monitoring servers: The following conditions apply for local seeding on distributed monitoring servers:

- Do not use the *FTP Catalog and Attribute files* option on the Manage Tivoli Enterprise Monitoring Services window. This option is only for sending files by FTP to z/OS systems.
- You must perform this task manually using a utility of your choice, such as FTP, SFTP, SCP, and so on.
- Replace all instances of `<monitoring serverNAME>` in these instructions with the name of the monitoring server on your distributed system.

You must copy the required files from the hub monitoring server to the other monitoring servers:

1. Copy the CAT and ATR files:
   a. Copy the CAT files from `$CANDLEHOME/tables/ciCATrsq/RKDSCATL` on the hub monitoring server to `$CANDLEHOME/tables/ciCATrsq/RKDSCATL` on the other monitoring server system.
   b. Copy the ATR files from `$CANDLEHOME/tables/ciCATrsq/ATTRLIB` on the hub monitoring server to `$CANDLEHOME/tables/ciCATrsq/ATTRLIB` on the other monitoring server system.
2. If you use Tivoli Enterprise Console servers, copy the .map and .baroc files:
   - Copy the baroc and map files from `$CANDLEHOME/tables/ciCATrsq/TECLIB` on the hub monitoring server to `$CANDLEHOME/tables/ciCATrsq/TECLIB` on the other monitoring server system.
3. Copy the `*tms.ver` files to be able to use `cinfo` and `itmcmd` support.
   - Copy the `*tms.ver` files from `$CANDLEHOME/registry` on the hub monitoring server to `$CANDLEHOME/registry` on the other monitoring server system.
4. Perform local seeding on the other monitoring server system:
   a. Log in to the system on which the other monitoring server resides.
   b. Change directory to the `$CANDLEHOME/bin` directory.
   c. Run the following command:

   ```
   ./itmcmd support -t <monitoring serverNAME> pc
   ```

   where *pc* can be a space delimited list to seed for multiple products in one run.

   d. Remember to recycle the monitoring server on the system that has been updated.

Installing the Tivoli Enterprise Portal Server

You can install the Tivoli Enterprise Portal Server on Windows, Linux or AIX. Follow the instructions for your operating system:

- "Windows: Installing the portal server" on page 229
- "Linux or AIX: Installing the portal server" on page 238

Notes:

1. When you install or upgrade the Tivoli Enterprise Portal Server, the Tivoli Enterprise Services User Interface Extensions software is automatically installed in the same directory. The portal server extensions are required for some products that use the Tivoli Enterprise Portal (for example, IBM Tivoli Composite Application Manager for Service Oriented Architecture).

The Tivoli Enterprise Services User Interface Extensions software is supported on the same operating systems as the Tivoli Enterprise Portal Server.

2. If you run the **tacmd listsystems** command prior to the CQ agent coming online, the CQ agent version number might show up as **06.23.00.xx**, while all other agent version numbers are **06.23.00.00**. The seventh and eighth digits in the IBM Tivoli Monitoring version string are only utilized by the remote deployment capability. Since remote deploy is not used for the Tivoli Enterprise Portal Server agent, you do not have to worry about the 'xx' digits.

Windows: Installing the portal server

The installation procedure for a portal server on Windows includes steps for configuring the connection between the portal server and the following components:

- The hub monitoring server
- The portal server database
- The Tivoli Data Warehouse database

Notes:

1. If you have not set up the Tivoli Data Warehouse, complete this procedure but click **No** when asked if you want to configure the connection to the data warehouse. You can reconfigure the connection after you set up the warehouse. See Step 18 on page 233 for more information.
2. The Windows userid you use when creating the portal server database must be a **local** ID with Administrator privileges. In other words, it cannot be a domain ID.

Complete the following steps to install the Tivoli Enterprise Portal Server and portal client on a Windows computer:

1. Launch the installation wizard by double-clicking the setup.exe file in the Infrastructure DVD or DVD image.
2. Click **Next** on the Welcome window.
3. Read and accept the software license agreement by clicking **Accept**.
4. Specify the directory where you want to install the portal server software and accompanying files. The default location is C:\IBM\ITM. Click **Next**.

   Note: If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```

5. Type an encryption key to use. This key should be the same as what was used during the installation of the hub monitoring server to which this portal server will connect. Click **Next** and then **OK** to confirm the encryption key.
6. On the Select Features window, select **Tivoli Enterprise Portal Server** from the list of components to install.

   When you select the **Tivoli Enterprise Portal Server** check box, all of the check boxes in the attached subtree are automatically selected. The support check boxes in the subtree are for installing application support files for base monitoring agents to the monitoring server. (The *base* monitoring agents are included with the base IBM Tivoli Monitoring installation package.) It is best to leave all of the support check boxes selected so you do not need to reconfigure application support as new agent types are added to your environment. For detailed information about application support, see “Installing and enabling application support” on page 266.

   Notes:

   a. If you have purchased monitoring agents that run on z/OS, but have not purchased IBM Tivoli Monitoring as a separate product, expand the **Tivoli Enterprise Portal Server** node. Clear all check boxes in the subtree except the check boxes labeled **Tivoli Enterprise Portal Server** and, optionally, **TEC GUI Integration**. (See Step 7a.)

   b. If you are updating an existing installation, all check boxes on the Select Features window reflect your choices during the initial installation. If your installation media contains a newer version of a previously installed component, the component will be upgraded unless you clear the check box for that component.

   c. The Eclipse Help Server is automatically selected when you select the Tivoli Enterprise Portal Server.

7. Optionally select the following additional components to install:

   a. If you want to view events on the IBM Tivoli Enterprise Console event server through the Tivoli Enterprise Portal, expand **Tivoli Enterprise Portal Server** and ensure that **TEC GUI Integration** is selected.

   b. If you want to install a portal desktop client on this computer, select **Tivoli Enterprise Portal Desktop Client**.

      When you select the **Tivoli Enterprise Portal Desktop Client** check box, all of the check boxes in the attached subtree are automatically selected. These check boxes are for installing application support files for base monitoring agents to the portal desktop client. Leave these check boxes selected as you did for the portal server in Step 6 on page 229.

      Note: If you have purchased monitoring agents that run on z/OS, but have not purchased IBM Tivoli Monitoring as a separate product, expand the **Tivoli Enterprise Portal Desktop Client** node. Clear all check boxes in the subtree except the check boxes labeled **Tivoli Enterprise Portal Desktop Client** and, optionally, **TEC GUI Integration**.

8. Click **Next**.

9. If a monitoring server is not installed on this computer, go to Step 10.

   If you are installing the portal server on a computer that already has a monitoring server installed, the Agent Deployment window is displayed.

   The Agent Deployment window lists monitoring agents on this installation image that you can add to the agent depot. The agent depot contains agents that you can deploy to remote computers. For information about how to deploy agents in the agent depot to remote computers, see Chapter 10, "Deploying monitoring agents across your environment," on page 325.

   Note: By default, the agent depot is located in the *itm_installdir*/CMS/depot directory on Windows. If you want to use a different directory, create the directory (if it does not exist) and specify the directory using the DEPOTHOME key in the KBBENV file.

   Select the agents, if any, that you want to add to the agent depot. (You can add agents to the agent depot at a later time by updating your installation.) Click **Next**.

10. If no IBM Tivoli Monitoring component has been previously installed on this computer, a window is displayed for you to select a program folder for the Windows Start menu. Select a program folder and click **Next**. The default program folder name is IBM Tivoli Monitoring.

11. If you are installing the Tivoli Enterprise Monitoring Server at the same time as the Tivoli Enterprise Portal Server, you are asked to set a sign on password for the Tivoli Enterprise Portal desktop and browser clients.

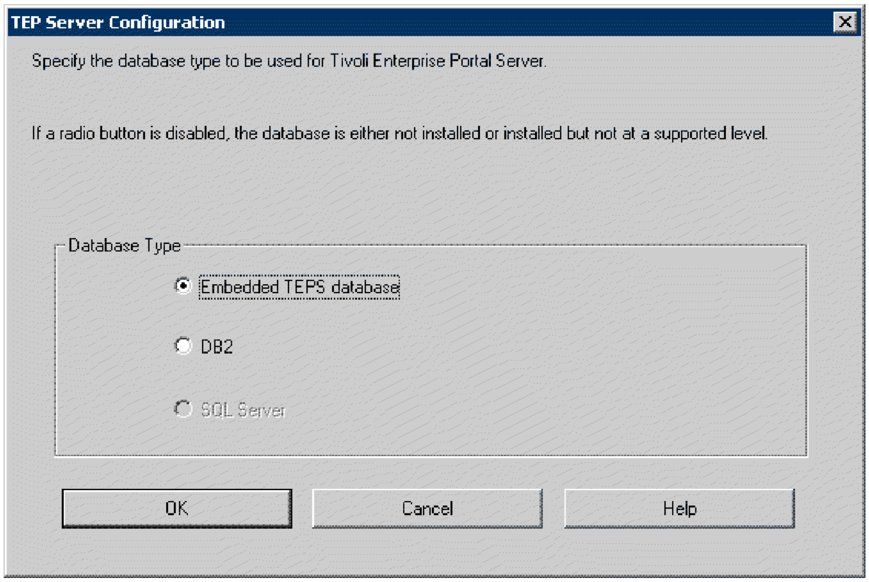
12. Review the installation summary details. The summary identifies what you are installing and where you chose to install it. Click **Next** to start the installation.

    After installation is complete, a configuration window (called the Setup Type window) is displayed.

13. Clear the check boxes for any components that have already been installed and configured (at the current release level) on this computer, unless you want to modify the configuration. (For example, clear the check box for the Tivoli Enterprise Monitoring Server if it has already been installed and configured on this computer.) Click **Next** to start configuring all selected components.

14. You are asked to choose the database management system you want to use for your portal server database, Derby, DB2 Database for Linux, UNIX, and Windows, or Microsoft SQL Server, as shown in

Figure 49. Note that if a particular Database Type is uninstalled on this computer or if it installed but not at the necessary release level, the radio button is grayed out.



TEP Server Configuration

Specify the database type to be used for Tivoli Enterprise Portal Server.

If a radio button is disabled, the database is either not installed or installed but not at a supported level.

Database Type

● Embedded TEPS database

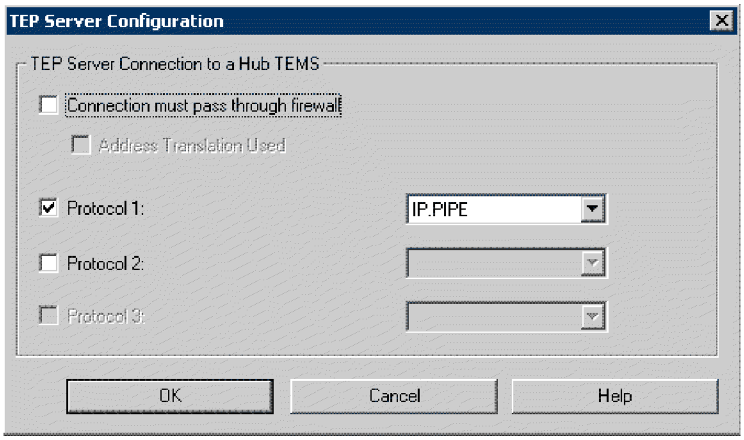
○ DB2

○ SQL Server

Figure 49. The Select Database for Tivoli Enterprise Portal window

15. Select the communications protocol for the connection between the portal server and the hub monitoring server. You can choose from the following protocol options:
    - IP.PIPE
    - IP.SPIPE
    - SNA
    - IP.UDP

    You can select up to three protocols and specify if the connection between the portal server and the hub monitoring server passes through a firewall. Click **OK**.



TEP Server Configuration

TEP Server Connection to a Hub TEMS

□ Connection must pass through firewall

□ Address Translation Used

☑ Protocol 1: IP.PIPE

□ Protocol 2:

□ Protocol 3:

Figure 50. Communications protocol between the portal server and the hub monitoring server

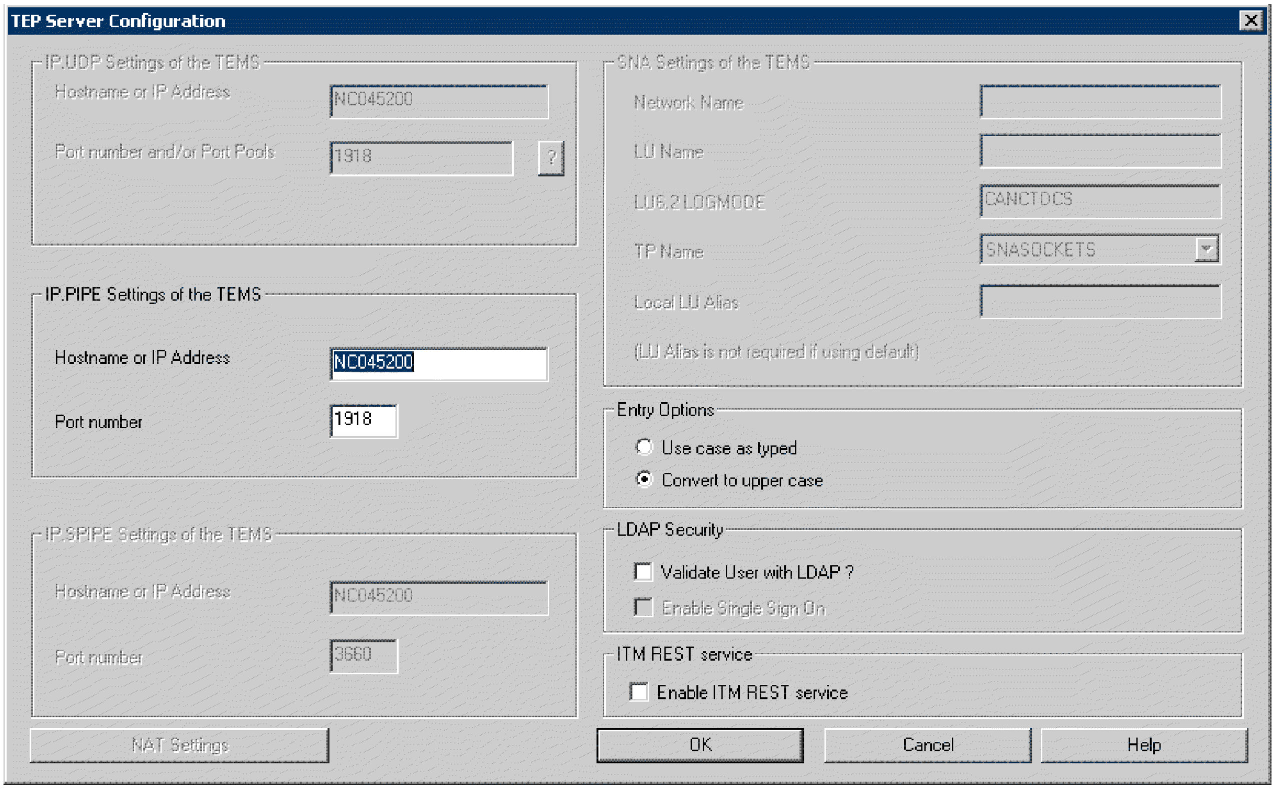
16. Configure communications for the Tivoli Enterprise Portal Server:
    a. Type the host name of the computer where you are installing the portal server. (The host name of this computer is displayed by default.)

b. Select **Validate user with LDAP** if you want to specify a distinguished name for a base entry in the LDAP repository. The distinguished name uniquely identifies this set of entries in the realm, instead of using the default value `o=ITMSSOEntry`. During configuration enter a value for the parameter `LDAP DN Base Entry`.

   Specifying a value here means you can avoid the scenario, when reconfiguring IBM Tivoli Monitoring to use single sign-on, of manually changing the value in the eWAS console to match all the other servers participating in the single sign-on realm.

   Note: The LDAP server must be started for this configuration step.

c. Click **OK**.



TEP Server Configuration

IP.UDP Settings of the TEMS

Hostname or IP Address: NC045200

Port number and/or Port Pools: 1918

IP.PIPE Settings of the TEMS

Hostname or IP Address: NC045200

Port number: 1918

IP.SPIPE Settings of the TEMS

Hostname or IP Address: NC045200

Port number: 3660

SNA Settings of the TEMS

Network Name:

LU Name:

LU6.2 LOGMODE: CANCTDCS

TP Name: SNASOCKETS

Local LU Alias:

(LU Alias is not required if using default)

Entry Options

○ Use case as typed

● Convert to upper case

LDAP Security

□ Validate User with LDAP ?

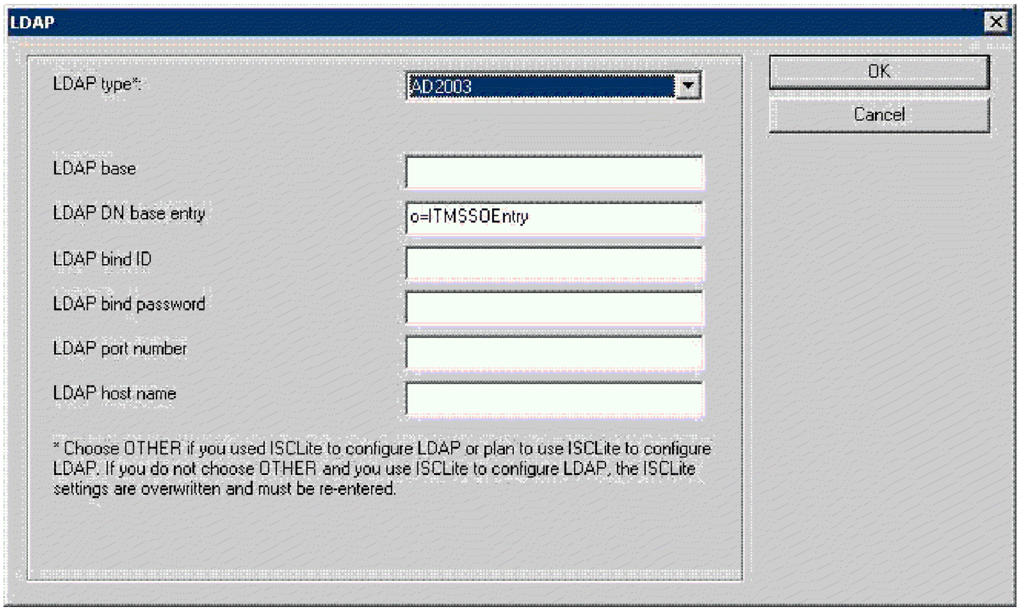
□ Enable Single Sign On

ITM REST service

□ Enable ITM REST service

Figure 51. Tivoli Enterprise Portal Server Configuration

17. If you selected **Validate User with LDAP** in step 16b, enter the LDAP configuration details in the fields provided and click **OK**.


LDAP

LDAP type*: AD2003

LDAP base

LDAP DN base entry: o=ITMSSOEntry

LDAP bind ID

LDAP bind password

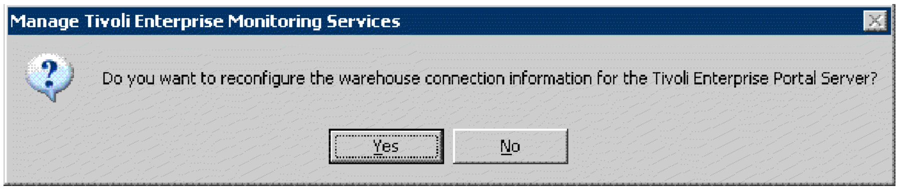
LDAP port number

LDAP host name

* Choose OTHER if you used ISCLite to configure LDAP or plan to use ISCLite to configure LDAP. If you do not choose OTHER and you use ISCLite to configure LDAP, the ISCLite settings are overwritten and must be re-entered.

Figure 52. LDAP configuration

18. A message is displayed asking if you want to reconfigure the warehouse connection for the portal server.


Manage Tivoli Enterprise Monitoring Services

Do you want to reconfigure the warehouse connection information for the Tivoli Enterprise Portal Server?

Figure 53. Reconfigure warehouse connection information for the portal server

Do one of the following:

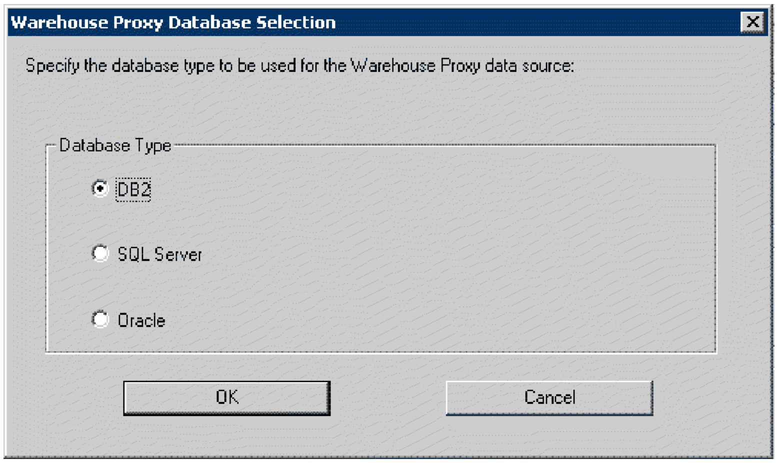
- Click **No** if you have not set up a Tivoli Data Warehouse.

  Follow the instructions later in this book for implementing a Tivoli Data Warehouse solution, beginning with Chapter 18, “Tivoli Data Warehouse solutions,” on page 465. These instructions will direct you to reconfigure the connection between the portal server and the warehouse database after you have completed all preliminary setup tasks.

  If you select **No**, go to Step 25 on page 238.
- Click **Yes** if you have completed the tasks for setting up a Tivoli Data Warehouse and want to configure the connection between the portal server and the Tivoli Data Warehouse database at this time. (You can choose to configure the connection later.) The prerequisite tasks and the information you need to configure the connection for each database type are described in the following steps.

For additional information about warehouse configuration, press the **Help** button.

19. Specify the database type to be used for the Warehouse Proxy data source.



Warehouse Proxy Database Selection

Specify the database type to be used for the Warehouse Proxy data source:

Database Type

● DB2

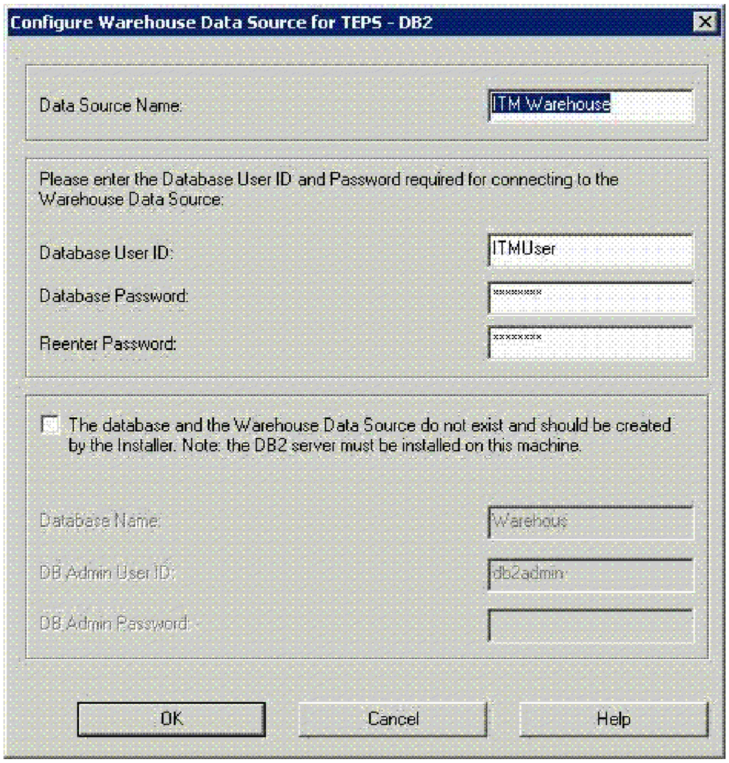
○ SQL Server

○ Oracle

Figure 54. Warehouse Proxy Database Selection

If you select Oracle without having installed an Oracle ODBC driver an error message is displayed saying *Oracle ODBC driver not installed on this machine* and you cannot continue with the configuration. You can obtain Oracle ODBC drivers from the following Web site: http://www.oracle.com/technology/software/tech/windows/odbc/htdocs/utilsoft.html

20. A window is displayed for you to configure the connection between the portal server and the *portal server database* (TEPS database). There is check box on the configuration window: **The database and the Warehouse Data Source do not exist and should be created by the Installer**. If you select this check box the installation program uses the information on this window to automatically perform the following tasks:
    - Create the portal server database.
    - Create a database user for the portal server to use to access the database.
    - Configure the ODBC connection between the portal server and the database.



Configure Warehouse Data Source for TEPS - DB2

Data Source Name: ITM Warehouse

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

Database User ID: ITMUser

Database Password: *********

Reenter Password: ********

□ The database and the Warehouse Data Source do not exist and should be created by the Installer. Note: the DB2 server must be installed on this machine.

Database Name: Warehous

DB Admin User ID: db2admin

DB Admin Password:

Figure 55. Configuration window for the portal server database using DB2 for Linux, UNIX, and Windows

Figure 55 shows the configuration window for a portal server database using DB2 for Linux, UNIX, and Windows. The configuration window for a Microsoft SQL Server database is similar. The fields on the configuration window are described in the following table:

Table 43. Configuration information for the portal server database

Field	DB2 for Linux, UNIX, and Windows default	MS SQL default	Description
Data Source Name	ITM Warehouse	ITM Warehouse	The name of the data source.
Database User ID	ITMUser	ITMUser	The name of the Windows OS user that the portal server will use to access the Tivoli Data Warehouse database.
Database Password	itmpswd1	itmpswd1	The password for the Windows user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Reenter Password	itmpswd1	itmpswd1	Confirm the password by entering it again.

21. To configure a connection between the portal server and a DB2 for Linux, UNIX, and Windows data warehouse, complete the following steps:
    a. Verify that you have completed the following tasks:
       - Created a warehouse database using DB2 for Linux, UNIX, and Windows
       - Created a warehouse user on the computer where you created the warehouse database.

         The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.

    - Activated the UNIX listeners on the computer where the warehouse database is located if the warehouse database is installed on UNIX systems.
    - Installed a DB2 for Linux, UNIX, and Windows client on the portal server if the data warehouse is remote and the portal server database does not use DB2 for Linux, UNIX, and Windows.
    - Cataloged the warehouse database on the computer where you are installing the portal server if the warehouse database is remote from the portal server.

    These tasks are described in Chapter 20, "Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows," on page 489.
  b. Gather the following information: data source name, database name, database administrator ID and password, warehouse user ID and password. The warehouse user ID is the one declared in the configuration panel of the Warehouse Proxy Agent. This user ID serves as the first part of the name of all the tables created in the Warehouse database. If you do not declare the same user ID when configuring the Tivoli Enterprise Portal server you will not be able to see the Warehouse tables with the portal client even if they exist in the database.
  c. Complete the Configuring a Windows portal server (ODBC connection) procedure, starting from Step 4 on page 510.

  For additional information about these parameters, press the **Help** button.

22. To configure a connection between the portal server and a Microsoft SQL Server data warehouse, complete the following steps:
  a. Verify that you have completed the following tasks:
    - Created a warehouse database using Microsoft SQL Server.
    - Created a warehouse user on the computer where you created the warehouse database.

      The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.
    - Installed a Microsoft SQL Server client on the portal server if the data warehouse is remote and the portal server database does not use Microsoft SQL Server.
    - Configured a remote client connection on the computer where you are installing the portal server if the warehouse database is remote from the portal server.

    These tasks are described in Chapter 22, "Tivoli Data Warehouse solution using Microsoft SQL Server," on page 547.
  b. Gather the following information: data source name, database name, database administrator ID and password, warehouse user ID and password. The warehouse user ID is the one declared in the configuration panel of the Warehouse Proxy Agent. This user ID serves as the first part of the name of all the tables created in the Warehouse database. If you do not declare the same user ID when configuring the Tivoli Enterprise Portal server you will not be able to see the Warehouse tables with the portal client even if they exist in the database.
  c. Complete the procedure Configuring the portal server (ODBC connection), starting from Step 4 on page 563.

  For additional information about these parameters, press the **Help** button.

23. To configure a connection between the portal server and an Oracle data warehouse, complete the following steps:
  a. Verify that you have completed the following tasks:
    - Created a warehouse database using Oracle
    - Created a warehouse user on the computer where you created the warehouse database.

      The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.
    - Activated the Oracle listener on the computer where the warehouse database is located
    - Installed an Oracle client on the portal server

- Created a TNS Service Name on the computer where you are installing the portal server if the warehouse database is remote from the portal server.

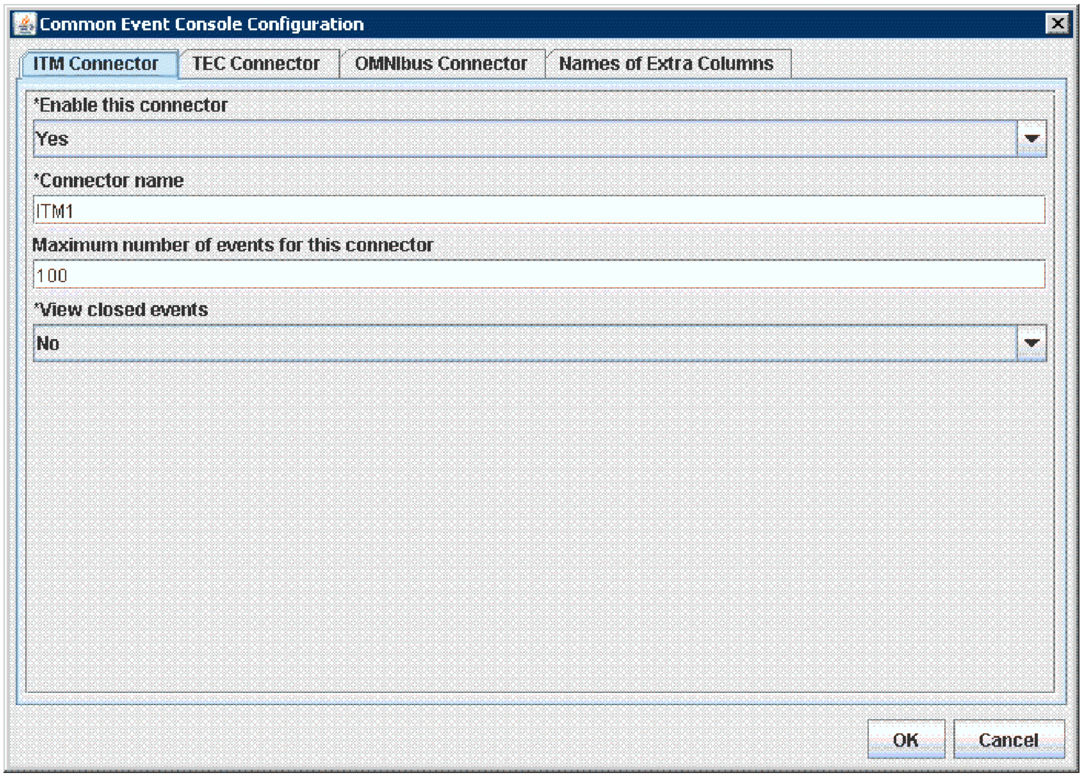
These tasks are described in Chapter 23, “Tivoli Data Warehouse solution using Oracle,” on page 571.

b. Gather the following information: the data source name, database name, database administrator ID and password, and the warehouse user ID and password. The warehouse user ID is the one declared in the configuration panel of the Warehouse Proxy Agent. This user ID serves as the first part of the name of all the tables created in the Warehouse database. If you do not declare the same user ID when configuring the Tivoli Enterprise Portal server you will not be able to see the Warehouse tables with the portal client even if they exist in the database.

c. Complete the procedure Configuring a Windows portal server (ODBC connection), starting from Step 4 on page 586.

For additional information about these parameters, press the **Help** button.

24. Configure the default connector for the Common Event Console.



*Enable this connector: Yes

*Connector name: ITM1

Maximum number of events for this connector: 100

*View closed events: No

Figure 56. Common Event Console Configuration window

The default connector retrieves situation events reported to Tivoli Enterprise Monitoring Servers for display in the Common Event Console. You can configure connectors for other event management systems after you have completed the product installation. For configuration instructions, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Click **OK** to accept the default values or specify values for the following fields, then click **OK**:

`Enable this connector`

Select `Yes` to enable the connector to collect and display situation events in the Common Event Console, or `No` to configure the connector without enabling it. The connector is enabled by default.

Connector name
The name that is to be displayed in the Common Event Console for this connector. The default name is ITM1.

Maximum number of events for this connector
The maximum number of events that are to be available in the Common Event Console for this connector. The default value is 100 events.

View closed events
Select `No` to display only active events in the Common Event Console for this connector. Select `Yes` to view both active and closed events. By default, only active events are displayed.

25. Configure the default communication between any monitoring agents installed on this computer and the hub Tivoli Enterprise Monitoring Server:
    a. Click **OK** to accept the default communications protocol.
    b. Ensure that the host name and port number of the Tivoli Enterprise Monitoring Server are correct. Click **OK**.

    For additional information about these parameters, press the **Help** button.
26. Click **Finish** to complete the installation.

Linux or AIX: Installing the portal server

Complete the procedures in this section to install and configure the Tivoli Enterprise Portal Server and portal client on a Linux or AIX computer.

Important: Run these installation and configuration procedures as either the root user or as the DB2 for Linux, UNIX, and Windows administrator. If you are configuring the Tivoli Enterprise Portal Server using the DB2 for Linux, UNIX, and Windows administrator ID, then:

- The configuration ID must be the same as the ID used to install the Tivoli Enterprise Portal Server.
- The configuration ID must be the same as the ID specified during the configuration dialog for the DB2 Admin ID.
- The configuration ID must have the proper DB2 for Linux, UNIX, and Windows authority/rights to attach to the DB2 Instance Name specified during the configuration dialog.
- The ID specified during the configuration dialog for the DB2 User ID must be an already existing ID. You may not allow the configuration dialog to attempt to create the user. For the GUI dialog, ensure that the check box for **Create the user** is cleared. For the CLI dialog, ensure that the response for Create New User is NO. After you have installed and configured the portal server, you can use a different user to run the portal server, as long as that user has access to the binaries used by the portal server.

  Note: Security policies for newly created users usually auto-expire the password after the first use and require the user to set a new password as part of the initial logon process. Until this is done, the installer always fails because the user password is expired and must be reset. Before running the installer, the user must invoke a process like ssh or telnet to log onto the target user ID and set the password appropriately.

Table 44. Steps for installing a portal server on a Linux or AIX computer

Steps	Where to find information
Install the portal server.	“Installing the portal server on Linux or AIX” on page 239
Configure the portal server.	“Configuring the portal server on Linux or AIX: command-line procedure” on page 240 –OR– “Configuring the portal server on Linux or AIX: GUI procedure” on page 244
Start the portal server.	“Starting the portal server” on page 252

Prerequisites for users installing on Linux on zSeries

If you plan to install the Tivoli Enterprise Portal Server on Linux for zSeries, make sure you complete these steps before beginning your installation:

- Ensure your Linux environment provides at least 10 gigabytes of free disk space.
- Turn off the Just-In-Time Java compiler.
- Users of RHEL version 5: set security-enhanced Linux (SELinux) to **permissive**.

Installing the portal server on Linux or AIX

Complete the following steps to install the portal server:

1. In the directory where you extracted the installation files, run the following command:

   ```
   ./install.sh
   ```

2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`) or type the full path to a different directory.

   Notes:

   a. If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special   or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```

   b. You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.

3. If the installation directory does not already exist, you are asked if you want to create it. Type `y` to create this directory and press Enter.

4. The following prompt is displayed:

   ```
   Select one of the following:
   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   3) Install TEMS support for remote seeding
   4) Exit install.

   Please enter a valid number:
   ```

   Type `1` to start the installation and display the software license agreement.

5. Press Enter to read through the agreement.

6. Type `1` to accept the agreement and press Enter.

7. Enter a 32-character encryption key or press Enter to accept the default key. This key should be the one used during the installation of the monitoring server to which this portal server will connect.

   A numbered list of available operating systems is displayed.

8. Type `4` to install the portal server for your current operating system. Press Enter.

   A message is displayed indicating that the Tivoli Enterprise Portal Server is about to be installed.

   Note: The Eclipse Help Server is automatically installed when you install the Tivoli Enterprise Portal Server.

9. Type `1` to confirm the installation.

   The installation begins.

10. After the Tivoli Enterprise Portal Server is installed, you are asked whether you want to install additional products or product support packages. Type `1` and press Enter.

    The installer presents a numbered list of products and application support packages.

11. Install the required application support packages.

    All monitoring agents require that application support files be installed on the monitoring servers (hub and remote), portal server, and portal desktop clients in your environment. Application support files contain the information required for agent-specific workspaces, helps, predefined situations, and other data.

    This step installs the application support files for base monitoring agents. The *base* monitoring agents are included with the base IBM Tivoli Monitoring installation package. For detailed information about application support, see "Installing and enabling application support" on page 266.

    When you entered 1 in the preceding step, the installer presents a numbered list of items, including the following application support packages:

    ```
    Tivoli Enterprise Portal Browser Client support
    Tivoli Enterprise Portal Server support
    ```

    Note: The Tivoli Enterprise Portal Browser Client support package is portal server code that supports the browser clients. You must install the browser client support package on the computer where you install the portal server if you want to connect to it using a browser client.

    Complete the following steps to install the portal server and browser client support packages for the base monitoring agents:

    a. Type the number that corresponds to `Tivoli Enterprise Portal Browser Client support` and press Enter.

       A numbered list of base monitoring agents is displayed.

    b. Type the numbers that correspond to the base monitoring agents for which you want to install the application support package, or type the number that corresponds to `All of the above`. Type the numbers on the same line separated by spaces or commas (,). Press Enter.

       It is best to select all of the base monitoring agents (`All of the above`) so you do not need to reconfigure application support as new agent types are added to your environment.

    c. Type `1` to confirm the installation and press Enter.

       The installation begins.

    d. After the support package is installed, you are asked whether you want to install additional products or product support packages. Enter `1` and repeat the preceding steps for the `Tivoli Enterprise Portal Server support` package.

    Note: This step installs the application support files. However, you must *enable* the application support by configuring the portal server. The next two sections show you how to configure the portal server.

12. After you are finished installing the portal server and browser client packages, you are asked whether you want to install additional products or product support packages. Type `2` and press Enter.

13. If your IBM Tivoli Monitoring environment is not already secured you will be asked at this point if you want to secure it. If your IBM Tivoli Monitoring environment is already secured this question is skipped. The product installation process creates the majority of directories and files with world write permissions. IBM Tivoli Monitoring provides the secureMain utility to help you keep the monitoring environment secure. You can secure your installation now, or manually execute the secureMain utility later. For more information, see Appendix G, "Securing your IBM Tivoli Monitoring installation on Linux or UNIX," on page 851.

Configuring the portal server on Linux or AIX: command-line procedure

Configure the portal server using either the command-line procedure in this section or the GUI procedure in "Configuring the portal server on Linux or AIX: GUI procedure" on page 244.

Either of these configuration procedures accomplishes the following tasks:

- Automatically enables application support on the portal server for the base monitoring agents. (See Step 11 of the installation procedure.)

- Includes steps for configuring the connection between the portal server and the following components:
  - The hub monitoring server
  - The portal server database
  - The Tivoli Data Warehouse database

Note: If you have not set up the Tivoli Data Warehouse, complete this procedure but accept the default values at the prompts for configuring the connection to the data warehouse. You can reconfigure the connection after you set up the warehouse. See Step 9 on page 242 for more information.

Complete the following steps to configure the Tivoli Enterprise Portal Server from the command-line on Linux or AIX:

1. Log on to the computer where the Tivoli Enterprise Portal Server is installed.
2. At the command-line change to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the directory where you installed the product.
3. Run the following command to start configuring the Tivoli Enterprise Portal Server:

   ```
   ./itmcmd config -A cq
   ```

   where `cq` is the product code for the portal server.
4. Edit the ITM Connector settings.

   The ITM Connector retrieves situation events reported to Tivoli Enterprise Monitoring Servers for display in the Common Event Console. If you do not enable the connector, you will not see any events in the Common Event Console. You can configure connectors for other event management systems after you have completed the product installation. For configuration instructions, see the *IBM Tivoli Monitoring: Administrator's Guide*.

   To enable the connector:

   a. Press Enter to accept the default value for the following prompt:

   ```
   Edit 'ITM Connector' settings? [ 1=Yes, 2=No ] (default is: 1):
   ```

   b. Press Enter to enable the connector.

   c. Press Enter to accept the default name of ITM1 or type your preferred name and press Enter. This is the name that is to be displayed in the Common Event Console for this connector.

   d. Press Enter to accept the default number of events (100) that are to be available in the Common Event Console for this connector, or type the number of events you would like to see displayed and press Enter.

   e. Type `2` and press Enter to display only active events in the Common Event Console for this connector. Type `1` and press Enter to view both active and closed events. By default, only active events are displayed.

   f. Type 2 and press Enter to skip defining data for extra columns in the Common Event Console.

   When you define a Tivoli Enterprise Console or Tivoli Netcool/OMNIbus connector, you can define the information that is to be mapped to each of these customizable columns. See the *IBM Tivoli Monitoring: Administrator's Guide* for information on configuring these connectors.
5. Press Enter when you are asked if the agent connects to a monitoring server. (Although the prompt refers to an *agent*, this command is used to configure the portal server.)
6. Configure the connection between the portal server and the hub monitoring server:

   a. Type the host name for the hub monitoring server and press Enter.

   b. Type the protocol that the hub monitoring server uses to communicate with the portal server. You have the following choices: IP.PIPE, IP.SPIPE, IP.UDP, or SNA. The IP.UDP and SNA protocols have been deprecated in favour of IP.PIPE and IP.SPIPE. You should avoid using the IP.UDP and SNA protocols even though they are available configuration options.

   c. If you want to set up a backup protocol, enter that protocol and press Enter. If you do not want to use a backup protocol, press Enter without specifying a protocol.

d. Depending on the type of protocol you specified, provide the following information as described in Table 45 when prompted:

Table 45. Hub monitoring server protocols and values

Protocol	Value	Definition
IP.UDP (deprecated, use IP.PIPE or IP.SPIPE instead)	IP Port Number	The port number for the monitoring server. The default is 1918.
SNA (deprecated, use IP.PIPE or IP.SPIPE instead)	Net Name	The SNA network identifier for your location.
	LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
	Log Mode	The name of the LU6.2 LOGMODE. The default value is `CANCTDCS`.
IP.PIPE	IP.PIPE Port Number	The port number for the monitoring server. The default is 1918.
IP.SPIPE	IP.SPIPE Port Number	The port number for the monitoring server. The default is 3660.

e. Press Enter when you are asked if you want to configure the connection to a secondary monitoring server. The default value is `none`.

f. Press Enter to accept the default value for the Optional Primary Network Name (none).

g. Press Enter to accept the default setting for SSL between the portal server and clients (`N`). By default, SSL is disabled. To enable SSL, type `1` and press Enter.

7. Configure the connection between the portal server and the portal server database:

   a. Type `1` if your site is using the embedded Derby database, `2` if you're using DB2 for Linux, UNIX, and Windows.

   b. Type the DB2 for Linux, UNIX, and Windows instance name. The default value is `db2inst1`. Press Enter.

   c. Type the DB2 for Linux, UNIX, and Windows administrator ID. The default value is `db2inst1`. Press Enter.

   Note: The DB2 for Linux, UNIX, and Windows administrator account was created during DB2 for Linux, UNIX, and Windows installation.

   d. Type the password for the DB2 for Linux, UNIX, and Windows administrator ID, and press Enter.

   e. Confirm the password for the DB2 for Linux, UNIX, and Windows administrator ID by typing it again. Press Enter.

8. If you are configuring DB2 for Linux, UNIX, and Windows for the portal server (instead of the embedded Derby database), complete the following parameters as well:

   a. Type the name of the portal server database. The default value is `TEPS`. Press Enter.

   b. Type the login name of the database user that the portal server will use to access the database. The default value is `itmuser`. Press Enter.

   c. Type the password for the database user and press Enter.

   d. Confirm the password for the database user by typing it again. Press Enter.

   e. You are asked if you want to create the DB2 for Linux, UNIX, and Windows login user if it does not exist. Type `1` and press Enter.

9. You are asked for the database parameters for either DB2 for Linux, UNIX, and Windows or Oracle for the Tivoli Data Warehouse. Enter `D` for DB2 for Linux, UNIX, and Windows, `J` for Oracle (JDBC). DB2 for Linux, UNIX, and Windows is the default.

Note: This prompt and all remaining prompts ask for information to configure the connection between the portal server and the Tivoli Data Warehouse database. If you have not set up a Tivoli Data Warehouse, accept the default values at these prompts. Follow the instructions later in this book for implementing a Tivoli Data Warehouse solution, beginning with Chapter 18, "Tivoli Data Warehouse solutions," on page 465. These instructions will direct you to reconfigure the connection between the portal server and the warehouse database after you have completed all preliminary setup tasks.

10. Do one of the following:
    - If you have not set up a Tivoli Data Warehouse:
      a. Press Enter to accept the DB2 for Linux, UNIX, and Windows default (even if you are going to create the Tivoli Data Warehouse using Oracle).
      b. Press Enter at all remaining prompts to accept the default values.

      or
    - Configure a connection between the portal server and a DB2 for Linux, UNIX, and Windows warehouse database.

      Before you configure the connection, verify that you have already completed the following tasks:
      – Created a warehouse database using DB2 for Linux, UNIX, and Windows
      – Created a warehouse user on the computer where you created the warehouse database.

        The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.
      – Cataloged the warehouse database on the computer where you are installing the portal server if the warehouse database is remote from the portal server
      – Activated the UNIX listeners on the computer where the warehouse database is located if the warehouse database is installed on UNIX.

      These tasks are described in Chapter 20, "Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows."

      To perform the steps for configuring the connection between the portal server and the DB2 for Linux, UNIX, and Windows data warehouse, you will need the name of the warehouse database and the user ID and password of the warehouse user. Press Enter after answering each prompt:
      a. Press Enter to accept the DB2 for Linux, UNIX, and Windows default.
      b. Enter the name of the Tivoli Data Warehouse database. The default value is `WAREHOUS`.
      c. Enter the user ID of the warehouse user. The default value is `itmuser`.

         Note: The warehouse user ID must be the one declared in the configuration panel of the Warehouse Proxy Agent. This user ID serves as the first part of the name of all the tables created in the Warehouse database. If you do not declare the same user ID when configuring the Tivoli Enterprise Portal server you will not be able to see the Warehouse tables with the portal client even if they exist in the database.
      d. Enter the password of the warehouse user.
      e. Confirm the password by entering it again.

      –OR–
    - Configure a connection between the portal server and an Oracle warehouse database.

      Before you configure the connection, verify that you have already completed the following tasks:
      – Created a warehouse database using Oracle
      – Created a warehouse user on the computer where you created the warehouse database.

        The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.
      – Activated the Oracle listener on the computer where the warehouse database is located
      – Installed an Oracle JDBC Type 4 driver on the portal server.

These tasks are described in Chapter 23, “Tivoli Data Warehouse solution using Oracle.”

To perform the steps for configuring the connection between the portal server and the Oracle data warehouse, you will need:

– The name of the warehouse database and the user ID and password of the warehouse user

– The location, name, and URL of the JDBC driver

Press Enter after answering each prompt:

a. Enter `oracle` at the prompt asking if you are using DB2 for Linux, UNIX, and Windows or Oracle for the Warehouse.

b. Enter the name of the Tivoli Data Warehouse database. The default value is `WAREHOUS`.

c. Enter the user ID of the warehouse user. The default value is `itmuser`.

   Note: The warehouse user ID must be the one declared in the configuration panel of the Warehouse Proxy Agent. This user ID serves as the first part of the name of all the tables created in the Warehouse database. If you do not declare the same user ID when configuring the Tivoli Enterprise Portal server you will not be able to see the Warehouse tables with the portal client even if they exist in the database.

d. Enter the password of the warehouse user.

e. Confirm the password by entering it again.

f. Enter the full path name of the Oracle Type 4 JDBC driver JAR file as follows:

```
oracleinstalldir/jdbc/lib/ojdbc14.jar
```

   where *oracleinstalldir* is the directory location of the JDBC driver JAR file on this computer.

g. Enter the following JDBC driver name:

```
oracle.jdbc.driver.OracleDriver
```

h. Enter the JDBC driver URL. This is the Oracle-defined URL that identifies the locally or remotely installed Oracle instance used for the Tivoli Data Warehouse. The following entry is an example:

```
jdbc:oracle:thin:@localhost:1521:WAREHOUS
```

   – If the warehouse database is located on a remote computer, replace `localhost` with the host name of the remote computer.

   – Change the default port number (`1521`) and Tivoli Data Warehouse name (`WAREHOUS`) if they are different.

i. Enter any user-defined attributes that are used to customize the behavior of the driver connection. Use semi-colons (;) to delimit the attributes. Press Enter to finish the configuration.

11. Configure LDAP Security. By default, the **LDAP Security: Validate User with LDAP** is enabled (1=Yes, 2=No). Press Enter if you want to specify a distinguished name for a base entry in the LDAP repository. The distinguished name uniquely identifies this set of entries in the realm, instead of using the default value `o=ITMSSOEntry`. During configuration enter a value for the parameter `LDAP DN Base Entry`. Specifying a value here means you can avoid the scenario, when reconfiguring IBM Tivoli Monitoring to use single sign-on, of manually changing the value in the eWAS console to match all the other servers participating in the single sign-on realm.

    Note: The LDAP server must be started for this configuration step.

A message is displayed telling you that InstallPresentation is running, and then a message telling you that the installation has completed.

Configuring the portal server on Linux or AIX: GUI procedure

Configure the portal server using either the GUI procedure in this section or the command-line procedure in “Configuring the portal server on Linux or AIX: command-line procedure” on page 240.

Either of these configuration procedures accomplishes the following tasks:

- Automatically enables application support on the portal server for the base monitoring agents.
- Includes steps for configuring the connection between the portal server and the following components:
  - The hub monitoring server
  - The portal server database
  - The Tivoli Data Warehouse database

Note: If you have not set up the Tivoli Data Warehouse, complete this procedure but accept the default values at the prompts for configuring the connection to the data warehouse. You can reconfigure the connection after you set up the warehouse. See Step 9 on page 242 for more information.

Complete the following steps to configure the Tivoli Enterprise Portal Server from the Tivoli Enterprise Monitoring Services window on Linux or AIX:

1. Log on to the computer where the Tivoli Enterprise Portal Server is installed.
2. Start the Manage Tivoli Enterprise Monitoring Services utility:
   a. Change to the bin directory:
      ```
      cd install_dir/bin
      ```
   b. Run the following command using the parameters described in Table 46:
      ```
      ./itmcmd manage [-h ITMinstall_dir]
      ```
      where:

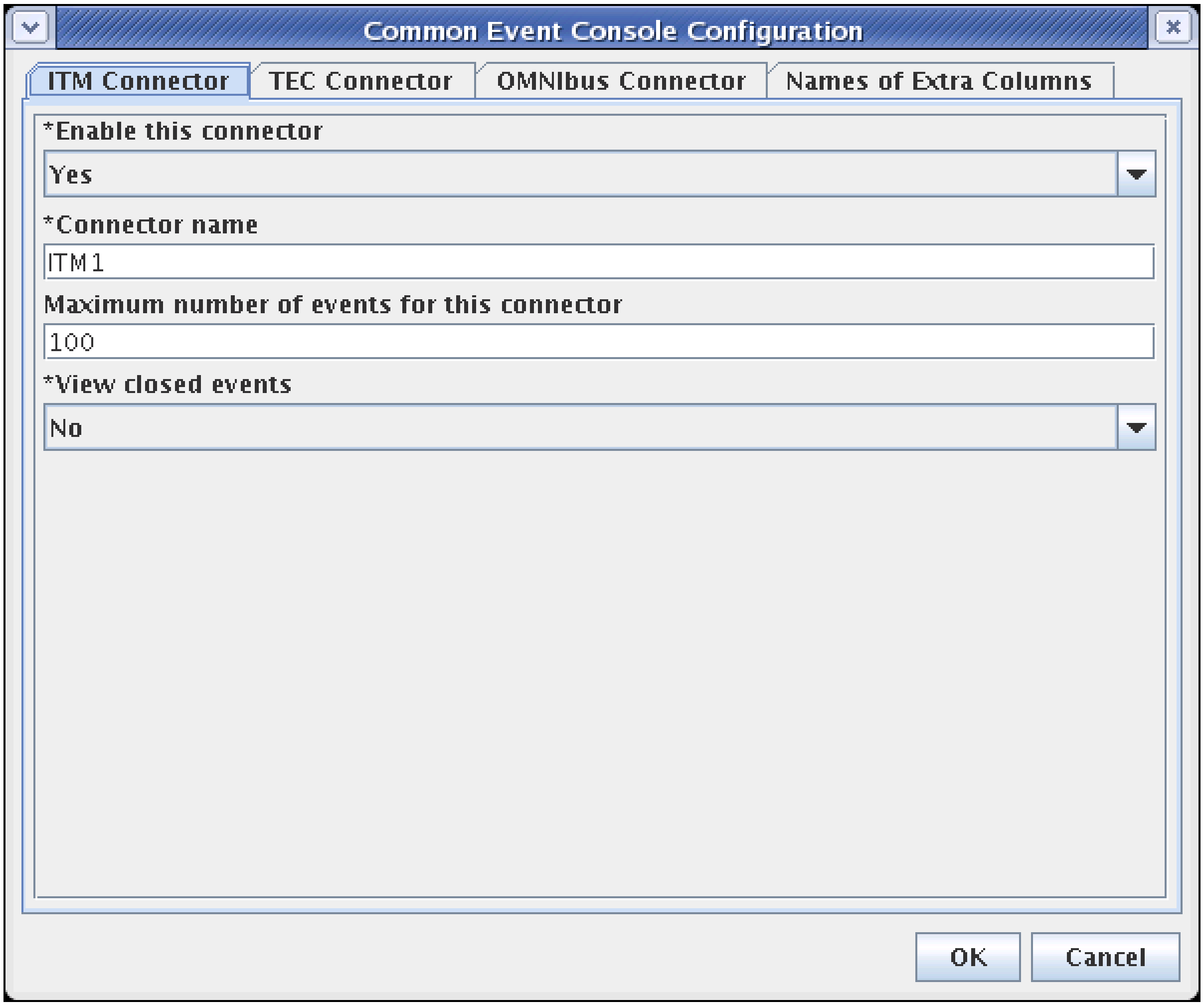
Table 46. Parameters for the itmcmd manage command

-h	(optional) An option used to specify the installation directory.
ITMinstall_dir	The directory where the portal server is installed. The default installation directory is /opt/IBM/ITM.

   The Manage Tivoli Enterprise Monitoring Services window is displayed.

3. Right-click **Tivoli Enterprise Portal Server** and click **Configure**.

   The Common Event Console Configuration window is displayed.



*Enable this connector

Yes

*Connector name

ITM1

Maximum number of events for this connector

100

*View closed events

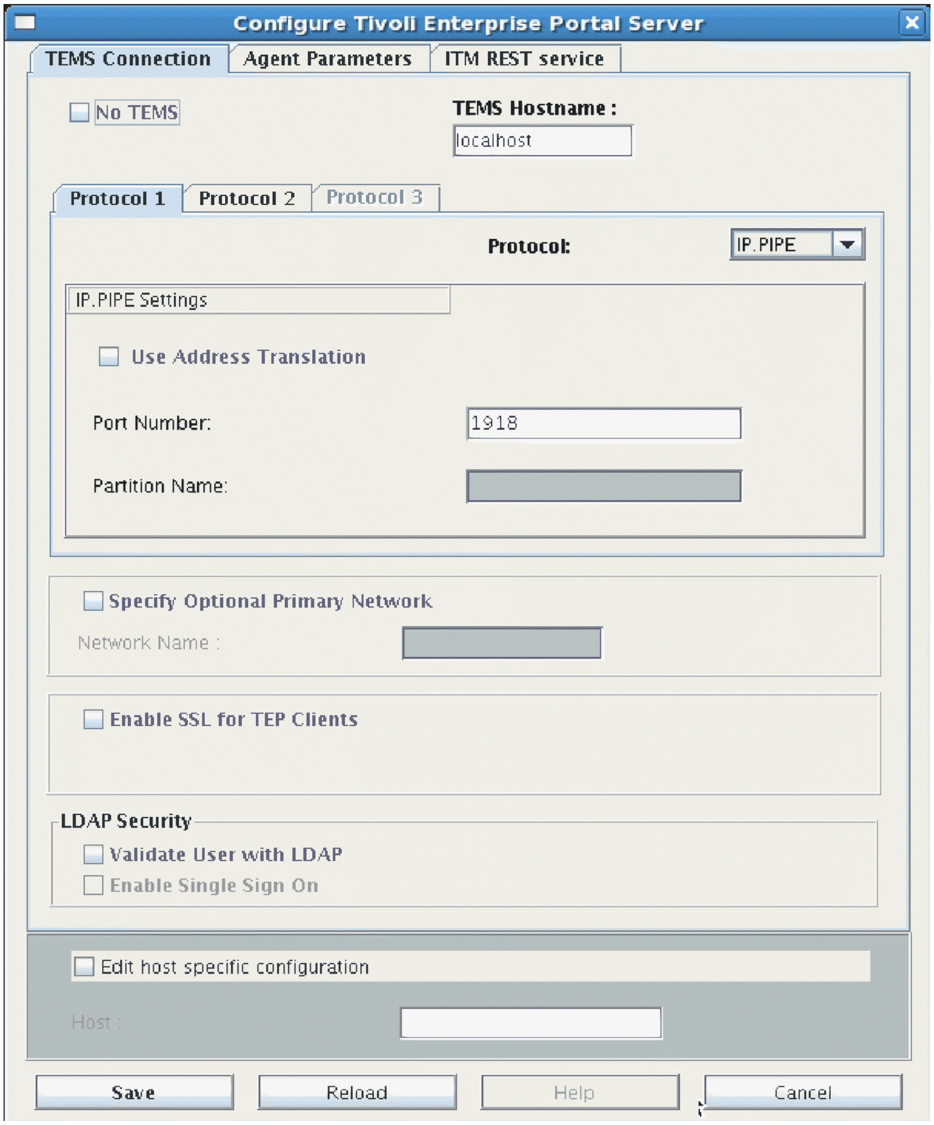
No

Figure 57. Common Event Console Configuration window

4. Click **OK** to accept the default values for the ITM Connector, or specify your preferred values and then click OK.

   The ITM Connector retrieves situation events reported to Tivoli Enterprise Monitoring Servers for display in the Common Event Console. You can configure connectors for other event management systems after you have completed the product installation. For configuration instructions, see the *IBM Tivoli Monitoring: Administrator's Guide*.

   The Configure Tivoli Enterprise Portal Server window is displayed.



□ No TEMS

TEMS Hostname : localhost

Protocol: IP.PIPE

□ Use Address Translation

Port Number: 1918

Partition Name:

□ Specify Optional Primary Network

Network Name :

□ Enable SSL for TEP Clients

□ Validate User with LDAP

□ Enable Single Sign On

□ Edit host specific configuration

Host :

Figure 58. Registering the portal server with the Tivoli Enterprise Monitoring Server

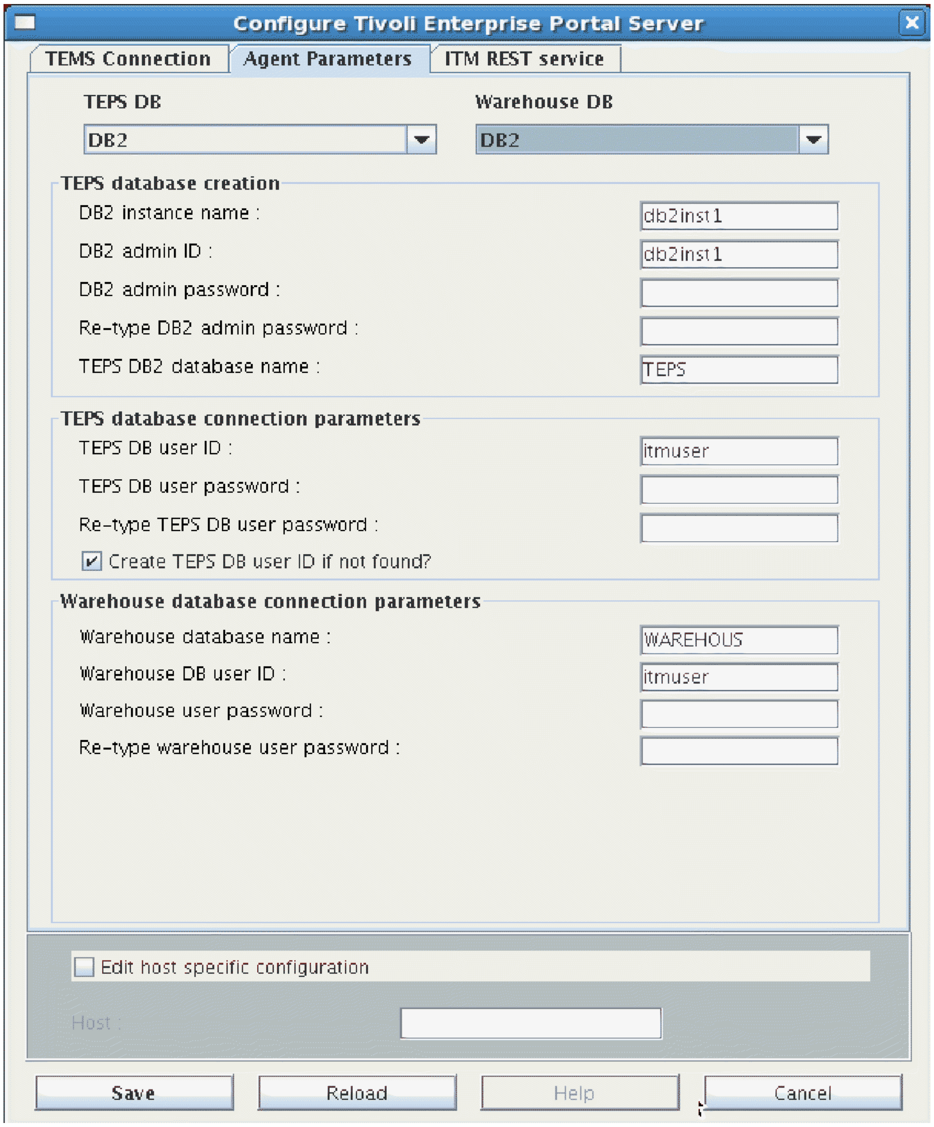
5. On the TEMS Connection page, enter information about the Tivoli Enterprise Monitoring Server to which the Tivoli Enterprise Portal Server connects:
   a. Enter the host name of the monitoring server in the **TEMS Hostname** field. (If the field is not active, clear the **No TEMS** check box.)
   b. Select the communications protocol that the monitoring server uses from the **Protocol** drop-down list.
      - If you select SNA, enter information in the **Net Name**, **LU Name**, and **LOG Mode** fields.
      - If you select IP.UDP, IP.PIPE or IP.SPIPE, enter the port number of the monitoring server in the **Port Number** field.

      For information about these fields, see Table 45 on page 242.
   c. Select **Validate user with LDAP** if you want to specify a distinguished name for a base entry in the LDAP repository. The distinguished name uniquely identifies this set of entries in the realm, instead of using the default value `o=ITMSSOEntry`. During configuration enter a value for the parameter `LDAP DN Base Entry`.

Specifying a value here means you can avoid the scenario, when reconfiguring IBM Tivoli Monitoring to use single sign-on, of manually changing the value in the eWAS console to match all the other servers participating in the single sign-on realm.

Note: The LDAP server must be started for this configuration step.

6. Click the **Agent Parameters** tab.



TEPS DB: DB2

Warehouse DB: DB2

TEPS database creation

- DB2 instance name : db2inst1
- DB2 admin ID : db2inst1
- DB2 admin password :
- Re-type DB2 admin password :
- TEPS DB2 database name : TEPS

TEPS database connection parameters

- TEPS DB user ID : itmuser
- TEPS DB user password :
- Re-type TEPS DB user password :
- ☑ Create TEPS DB user ID if not found?

Warehouse database connection parameters

- Warehouse database name : WAREHOUS
- Warehouse DB user ID : itmuser
- Warehouse user password :
- Re-type warehouse user password :

☐ Edit host specific configuration

Host :

Figure 59. Configuring database connections for the portal server

7. Configure the connection between the portal server and the portal server database by entering information in the fields described in the following table:

Table 47. Configuration information for the Tivoli Enterprise Portal Server database

Field	Default value	Description
DB2 instance name	db2inst1	The DB2 for Linux, UNIX, and Windows instance name. Not required if the embedded Derby database is used for the portal server and Oracle is selected for the warehouse database.

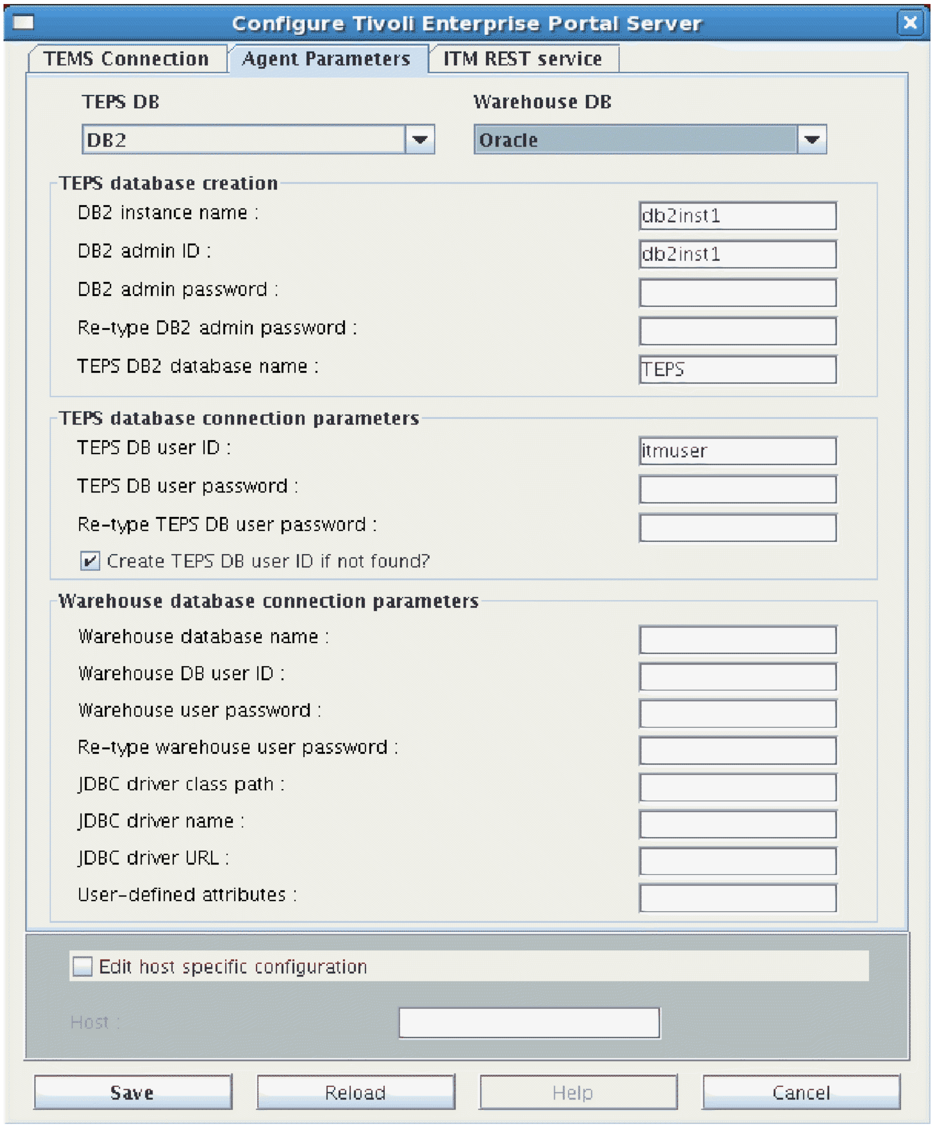
Table 47. Configuration information for the Tivoli Enterprise Portal Server database (continued)

Field	Default value	Description
DB2 admin ID	db2inst1	The DB2 for Linux, UNIX, and Windows administrator ID. The DB2 for Linux, UNIX, and Windows administrator account was created during DB2 for Linux, UNIX, and Windows installation. Not required if the embedded Derby database is used for the portal server and Oracle is selected for the warehouse database.
DB2 admin password	(no default)	The password for the DB2 for Linux, UNIX, and Windows administrator ID. Not required if the embedded Derby database is used for the portal server and Oracle is selected for the warehouse database.
Re-type DB2 admin password	(no default)	The password for the DB2 for Linux, UNIX, and Windows administrator ID. Not required if the embedded Derby database is used for the portal server and Oracle is selected for the warehouse database.
TEPS DB2 database name	TEPS	The Tivoli Enterprise Portal Server database name. Disabled if the embedded Derby database is chosen for the portal server database.
TEPS DB user ID	itmuser	The login name of the warehouse user that the portal server will use to access the database. Disabled if the embedded Derby database is chosen for the portal server database.
TEPS DB user password	(no default)	The password for the warehouse user ID. Disabled if the embedded Derby database is chosen for the portal server database.
Re-type TEPS DB user password	(no default)	The password for the warehouse user ID. Disabled if the embedded Derby database is chosen for the portal server database.
Create TEPS DB user ID if not found?	yes	This check box is selected by default. If the database login account (user ID and password) that you specified in the preceding fields does not exist, it is created. Disabled if the embedded Derby database is chosen for the portal server database.

8. Optionally configure the connection between the portal server and the Tivoli Data Warehouse database.

   Note: If you have not set up a Tivoli Data Warehouse, accept the default values for these fields. Follow the instructions later in this book for implementing a Tivoli Data Warehouse solution, beginning with Chapter 18, "Tivoli Data Warehouse solutions," on page 465. These instructions will direct you to reconfigure the connection between the portal server and the warehouse database after you have completed all preliminary setup tasks.

   The bottom section of the **Agent Parameters** tab contains fields for configuring the connection between the portal server and a Tivoli Data Warehouse database using DB2 for Linux, UNIX, and Windows or Oracle. (See Figure 60.)



Configure Tivoli Enterprise Portal Server

TEMS Connection | Agent Parameters | ITM REST service

TEPS DB: DB2

Warehouse DB: Oracle

TEPS database creation

DB2 instance name :	db2inst1
DB2 admin ID :	db2inst1
DB2 admin password :	
Re-type DB2 admin password :	
TEPS DB2 database name :	TEPS

TEPS database connection parameters

TEPS DB user ID :	itmuser
TEPS DB user password :	
Re-type TEPS DB user password :	

☑ Create TEPS DB user ID if not found?

Warehouse database connection parameters

Warehouse database name :	
Warehouse DB user ID :	
Warehouse user password :	
Re-type warehouse user password :	
JDBC driver class path :	
JDBC driver name :	
JDBC driver URL :	
User-defined attributes :	

☐ Edit host specific configuration

Host :

Figure 60. Configuration information for the Tivoli Data Warehouse using an Oracle database

Do one of the following:

- If you have not set up the Tivoli Data Warehouse, click **Save** to save your settings and close the window.

  –OR–
- Configure a connection between the portal server and a DB2 for Linux, UNIX, and Windows warehouse database.

  Before you configure the connection, verify that you have already completed the following tasks:
  - Created a warehouse database using DB2 for Linux, UNIX, and Windows
  - Created a warehouse user on the computer where you created the warehouse database.

    The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.
  - Cataloged the warehouse database on the computer where you are installing the portal server if the warehouse database is remote from the portal server

  - Activated the UNIX listeners on the computer where the warehouse database is located if the warehouse database is installed on UNIX.

  These tasks are described in Chapter 20, “Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows.”

  To configure the connection between the portal server and the DB2 for Linux, UNIX, and Windows data warehouse, you will need the name of the warehouse database and the user ID and password of the warehouse user.

  To perform the configuration, complete the procedure Configuring a Linux or AIX portal server (DB2 for Linux, UNIX, and Windows CLI connection), starting from Step 4 on page 511.

  –OR–

- Configure a connection between the portal server and an Oracle warehouse database.

  Before you configure the connection, verify that you have already completed the following tasks:

  - Created a warehouse database using Oracle
  - Created a warehouse user on the computer where you created the warehouse database.

    The *warehouse user* is the user account (user ID and password) used by the portal server and other warehousing components to access the warehouse database.
  - Activated the Oracle listener on the computer where the warehouse database is located
  - Installed an Oracle JDBC Type 4 driver on the portal server.

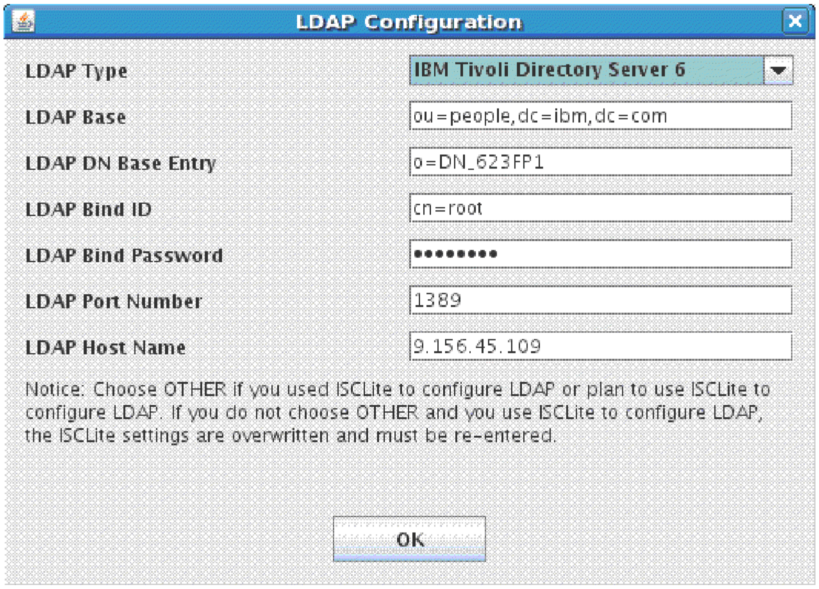
  These tasks are described in Chapter 23, “Tivoli Data Warehouse solution using Oracle.”

  To configure the connection between the portal server and the Oracle data warehouse, you will need:

  - The name of the warehouse database and the user ID and password of the warehouse user
  - The location, name, and URL of the JDBC driver

  To perform the configuration, complete the procedure Configuring a Linux or AIX portal server (JDBC connection), starting from Step 4 on page 587.

9. When you are done configuring the portal server, click **Save** to save your settings and close the window.
10. If you selected **Validate User with LDAP** in step 5c on page 247, enter the LDAP configuration details in the fields provided and click **OK**.



LDAP Type: IBM Tivoli Directory Server 6

LDAP Base: ou=people,dc=ibm,dc=com

LDAP DN Base Entry: o=DN_623FP1

LDAP Bind ID: cn=root

LDAP Bind Password: •••••••••

LDAP Port Number: 1389

LDAP Host Name: 9.156.45.109

Notice: Choose OTHER if you used ISCLite to configure LDAP or plan to use ISCLite to configure LDAP. If you do not choose OTHER and you use ISCLite to configure LDAP, the ISCLite settings are overwritten and must be re-entered.

Figure 61. LDAP Configuration

Starting the portal server

From the bin directory of /opt/IBM/ITM (or where you installed IBM Tivoli Monitoring), run the following command to start the portal server:

```
./itmcmd agent start cq
```

Upgrading a 32-bit portal server to 64 bit

The installation process does not convert an existing 32-bit Tivoli Enterprise Portal Server running on Linux or UNIX to 64 bit (although it does convert an existing 32-bit portal server from a prior release to a 32-bit portal server for the current release). The following procedure allows you to manually perform this conversion from 32 bit to 64 bit.

1. Using the product media for the current release, install and configure the 32-bit Tivoli Enterprise Portal Server for IBM Tivoli Monitoring V6.2.*x*.

   Before proceeding to step 2, ensure that the portal server is either completely initialized or completely stopped.

2. Invoke tepsBackup from the /*ITM_installdir*/bin directory:

   ```
   cd /ITM_installdir/bin
   ./tepsBackup
   ```

   where:

   ITM_installdir
   is the root location of your IBM Tivoli Monitoring V6.2.*x* environment.

   This process creates a compressed tar file with default and customized application data in the /*ITM_installdir*/tmp directory.

3. Uninstall the 32-bit portal server. If asked to completely remove the Tivoli Monitoring installation directories, answer **no**; otherwise the backup tar file will be lost.

4. Again using the product media for the current release, install and configure the 64-bit Tivoli Enterprise Portal Server.

   Before proceeding to step 5, ensure that the portal server is either completely initialized or completely stopped.

5. Invoke tepsRestore from the /*ITM_installdir*/bin directory:

```
cd /ITM_installdir/bin
./tepsRestore
```

Installing monitoring agents

This section describes how to install distributed monitoring agents. A *distributed* monitoring agent is one that is installed on a distributed (not z/OS) operating system. These instructions also apply to the Warehouse Proxy and Summarization and Pruning Agents. For instructions on how to install a z/OS monitoring agent, see the agent documentation for your z/OS agent product.

To install a distributed monitoring agent, use the appropriate installation media:

- Use the IBM Tivoli Monitoring Base Agent DVD or DVD image to install the agents in the following list
  - IBM Tivoli Monitoring 5.x Endpoint
  - Linux OS
  - UNIX Logs
  - UNIX OS
  - IBM Tivoli Universal Agent
  - Windows OS
- You can also use the IBM Tivoli Monitoring Base Agent DVD to install application support for any IBM Tivoli Monitoring V6.x-based distributed monitoring agent you intend to install. Installing and configuring application support on base components (monitoring servers, portal server, desktop clients) when you install them means that you do not have to stop and restart those components later when you install the agent. However, if the monitoring agents you install have been updated since the DVD was issued, you must install the application support files from the monitoring agent product CD.
- Use the agent product CDs to install distributed monitoring agents that are delivered separately from the IBM Tivoli Monitoring base installation package. For example, use the IBM Tivoli Monitoring for Databases product CDs to install a monitoring agent for DB2 Database for Linux, UNIX, and Windows or for Oracle. Depending on the agent that you are installing, there might be additional configuration steps required. See the agent documentation for more information.
- New in release 6.2.3 are a set of best practice situations for the UNIX OS agent, Linux OS agent Windows OS agent, and UNIX Logs agent. These situations are automatically installed, but not automatically distributed, during installation to the default managed system list or managed system. The best practice situations can be recognized by the `_BP_` string in the situation name.

All monitoring agents require that agent-specific application support be installed on the monitoring servers, portal server, and portal desktop clients. See “Installing and enabling application support” on page 266 for information.

The following sections provide instructions for installing a monitoring agent:

- “Windows: Installing a monitoring agent”
- “Linux or UNIX: Installing a monitoring agent” on page 259

Windows: Installing a monitoring agent

Use the following steps to install a distributed monitoring agent on a Windows computer:

1. If you are installing on either an x86-32 CPU or an x86-64 CPU, launch the installation wizard by double-clicking the `setup.exe` file in the `\WINDOWS` subdirectory on the installation media. However, if you are installing on an Itanium CPU, launch the wizard by double-clicking on the `setup.exe` file in the `\WIA64` directory.

   Use either the IBM Tivoli Monitoring Base Agent DVD or a distributed agent product CD for Windows. Do *not* use a Data Files for z/OS CD.
2. Click **Next** on the Welcome window.

Note: If you have another IBM Tivoli Monitoring component already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 6.

3. Read and accept the software license agreement by clicking **Accept**.
4. Choose the directory where you want to install the product. Click **Next**.

   Notes:

   a. This step applies only to those agents that you install from the IBM Tivoli Monitoring installation image. Agents installed from the agent installation image do not have this step. If you are installing an agent from an agent installation image, skip to step 6.

   b. If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```
5. Type a 32-character encryption key. This key must be the same as the key that was used during the installation of the monitoring server to which this monitoring agent connects.

   Click **Next** and then click **OK** to confirm the encryption key.
6. On the Select Features window, expand **Tivoli Enterprise Monitoring Agents**.
7. Select the names of the agents that you want to install and click **Next**.

   Notes:

   a. Agents available on the IBM Tivoli Monitoring installation image include the Warehouse Proxy Agent and the Summarization and Pruning Agent. Before you install these agents, follow the instructions later in this book for setting up a Tivoli Data Warehouse solution, beginning with Chapter 18, “Tivoli Data Warehouse solutions,” on page 465. On Windows Itanium CPUs, only the Windows agent, the Tivoli Universal Agent, and the agent framework are available for installation.

   b. If both of these conditions are met:

   - The components you selected for installation would result in a mixed 32-bit and 64-bit environment.
   - No Agent Compatibility Package (see “Installing the Agent Compatibility (AC) component” on page 256) can be found on the installation DVD, or the package found is not at the appropriate level.

   then this error message is displayed:

   ```
   The Agent Compatibility Package in version ver_num is required but unavailable.
   ```
8. If a monitoring server is not installed on this computer, go to Step 9.

   If you are installing monitoring agents on a computer that already has a monitoring server installed, the Agent Deployment window is displayed.

   The Agent Deployment window lists monitoring agents on this installation image that you can add to the agent depot. The agent depot contains agents that you can deploy to remote computers. For information about how to deploy agents in the agent depot to remote computers, see Chapter 10, “Deploying monitoring agents across your environment,” on page 325.

   Note: By default, the agent depot is located in the *itm_installdir*`\CMS\depot` directory on Windows. If you want to use a different directory, create the directory (if it does not exist), and specify the directory using the DEPOTHOME key in the KBBENV file.

   Select the agents, if any, that you want to add to the agent depot. (You can add agents to the agent depot at a later time by updating your installation.) IBM strongly recommends you also add the agent compatibility (AC) package at this time. Click **Next**.
9. This step applies only to agents that you install from the IBM Tivoli Monitoring installation image. If you are installing agents from an agent product installation image, go to Step 10 on page 255.

If no IBM Tivoli Monitoring component has been previously installed on this computer, a window is displayed for you to select a program folder for the Windows Start menu. Select a program folder and click **Next**. The default program folder name is IBM Tivoli Monitoring.

10. Review the installation summary details. The summary identifies what you are installing and where you chose to install it. Click **Next** to start the installation.

    After installation is complete, a configuration window (called the Setup Type window) is displayed.

11. Clear the check boxes for any components that have already been installed and configured (at the current release level) on this computer, unless you want to modify the configuration. (For example, clear the check box for the Tivoli Enterprise Monitoring Server if it has already been installed and configured on this computer.) Click **Next** to start configuring all selected components.

12. Define the communications between the monitoring agents and the monitoring server:

    a. If the agents must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.

    b. Identify the type of protocol that the agents use to communicate with the monitoring server.

       You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify up to three methods for communication. If the method you have identified as Protocol 1 fails, Protocol 2 is used. If Protocol 2 fails, Protocol 3 is used. When you configure the protocol for the Warehouse Proxy Agent, configure the same protocols used by the application agents and by the hub monitoring. If the proxy agent does not have the same protocol as the hub monitoring server, it cannot register with the hub. If the proxy does not have the same protocol as the application agents, then the application agents cannot communicate with the proxy when they to create a route to it.

       Note: Do not select **Optional Secondary TEMS Connection**. You can set up the failover support for agents after installation, as described in the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*.

    c. Click **OK**. A second configuration window is displayed.

    d. Complete the fields shown in the following tableTable 48 for the protocol that you specified.

Table 48. Communications protocol settings

Field	Description
IP.UDP Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port # or Port Pools	The listening port for the hub monitoring server. The default number is 1918.
IP.PIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default number is 1918.
IP.SPIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port number	The listening port for the hub monitoring server. The default value is 3660.
SNA Settings	
Network Name	The SNA network identifier for your location.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.

Table 48. Communications protocol settings (continued)

Field	Description
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
TP Name	The transaction program name for the monitoring server.
Local LU Alias	The LU alias.

e. Click **OK** to exit the Configuration Defaults for Connecting to a TEMS window.

For additional information about these parameters, press the **Help** button.

13. Click **Finish** to complete the installation.
14. Click **Finish** on the Maintenance Complete window if you are updating an existing installation.
15. Open the Manage Tivoli Enterprise Monitoring Services utility (if it does not open automatically) to see if the monitoring agents that you installed have been configured and started. If **Yes** is displayed in the **Configured** column, the agent has been configured and started during the installation process.
16. If the value in the **Configured** column is blank and **Template** is displayed in the **Task/Subsystem** column, right-click the Template agent and complete the following steps:
    a. Click **Configure Using Defaults**.
    b. Complete any windows requiring information by using the agent-specific configuration settings in the user's guide for your agent.

       Note: Do not type non-ASCII characters on any of these windows. Typing characters from other character sets has unpredictable results.
    c. Repeat this step as necessary to create monitoring agent instances for each application instance you want to monitor.

Installing the Agent Compatibility (AC) component

With the introduction of a native 64-bit OS agent for Windows at IBM Tivoli Monitoring V6.2.2 fix pack 1, it is likely that mixed environments of both 32-bit and 64-bit Tivoli Monitoring agents will be found on the same machine. To provide a runtime environment capable of supporting such mixed 32- and 64-bit executables, as well as to provide backward compatibility between native 64-bit agents and existing 32-bit agents, a new component, the 32/64 Bit Agent Compatibility Package (component code AC), was also introduced at IBM Tivoli Monitoring V6.2.2 fix pack 1. The AC package comprises the following pieces:

- 32-bit and 64-bit agent frameworks
- 32-bit and 64-bit versions of the IBM Global Security Toolkit (GSKit), component KGS
- The Tivoli Monitoring configuration utilities, component KGL

Note: The AC component is a Windows component only. There is no equivalent for Linux or UNIX systems.

When to install the AC component: You need to install the AC component in these situations:

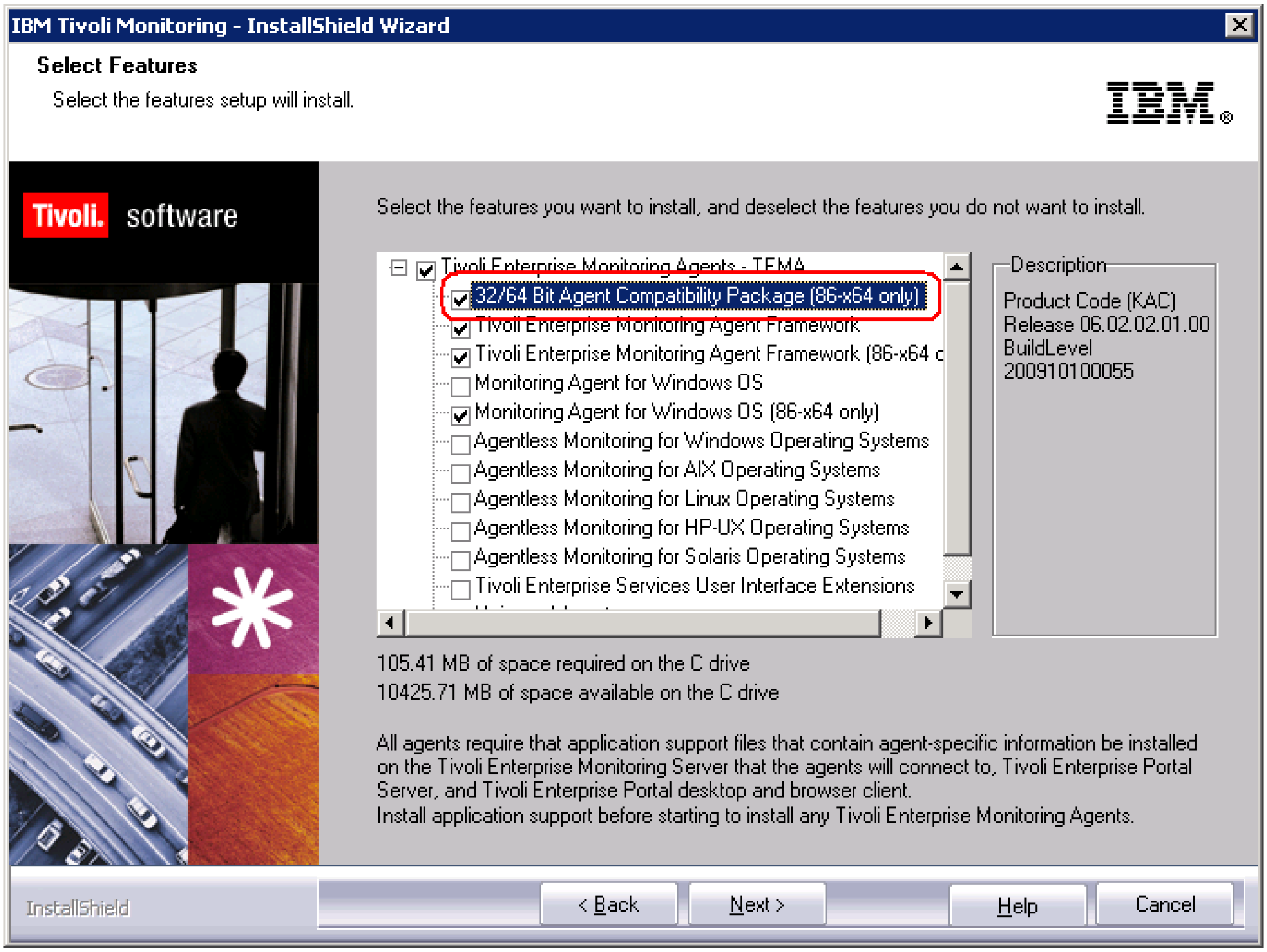
- When adding a native 64-bit agent into preexisting IBM Tivoli Monitoring installations comprising only 32-bit components.
- When adding a 32-bit agent into preexisting Tivoli Monitoring installations comprising only 64-bit components.
- The Tivoli Monitoring installer has detected a situation where proceeding without the AC component would result in incompatibility either between the 32-bit and the 64-bit agent frameworks or between the 32-bit and the 64-bit GSKit libraries.

The IBM Tivoli Monitoring installer will automatically detect any of the above situations and then stop the installation with this error:

```
The version ver_num or newer of the Agent Compatibility Package must be installed in order to ensure
the correct cooperation between 32bit and 64bit components. Exiting installation.
```

When not to install the AC component: The AC component can be installed with any 32-bit or 64-bit agent where a mixed 32/64 bit environment is anticipated. There is no need to install the AC component if only 32-bit or only 64-bit IBM Tivoli Monitoring agents are planned for this machine.

Installing the AC component using the Windows GUI: The AC component can be found on the Agents DVD. IBM recommends you install the AC component at the same version as the Windows OS agent (component code NT). Since the AC is a standard Tivoli Monitoring component, it can be installed using the standard interactive IBM Tivoli Monitoring installer by selecting the **32/64 Bit Agent Compatibility Package** feature for installation, as shown in Figure 62.



IBM Tivoli Monitoring - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- ☑ Tivoli Enterprise Monitoring Agents - TEMA
  - ☑ 32/64 Bit Agent Compatibility Package (86-x64 only)
  - ☑ Tivoli Enterprise Monitoring Agent Framework
  - ☑ Tivoli Enterprise Monitoring Agent Framework (86-x64 c
  - ☐ Monitoring Agent for Windows OS
  - ☑ Monitoring Agent for Windows OS (86-x64 only)
  - ☐ Agentless Monitoring for Windows Operating Systems
  - ☐ Agentless Monitoring for AIX Operating Systems
  - ☐ Agentless Monitoring for Linux Operating Systems
  - ☐ Agentless Monitoring for HP-UX Operating Systems
  - ☐ Agentless Monitoring for Solaris Operating Systems
  - ☐ Tivoli Enterprise Services User Interface Extensions

Description

Product Code (KAC)
Release 06.02.02.01.00
BuildLevel
200910100055

105.41 MB of space required on the C drive
10425.71 MB of space available on the C drive

All agents require that application support files that contain agent-specific information be installed on the Tivoli Enterprise Monitoring Server that the agents will connect to, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop and browser client.
Install application support before starting to install any Tivoli Enterprise Monitoring Agents.

Figure 62. Installing the Agent Compatibility Package (component code AC)

Remotely deploying the AC components: There are special considerations for deploying native 64-bit agents as well as the AC component. When deploying an agent to a Windows machine running an x86-64 CPU, checks are automatically made to verify whether the AC component is required. If the checks report the AC component is needed and it is available in the depot, it is sent automatically with no action required on your part. However, if the checks report the AC component is needed but the component is *not* available in the depot, an error is reported, and the deployment request fails. Therefore, it is highly recommended that you populate the deployment depot with the AC component.

Sample `tacmd addBundles` command to add the AC component to the deploy depot:

```
tacmd addBundles -i C:\ITM_6.2.3FP1_Agents_Image\WINDOWS\Deploy -t ac
```

For more details on managing the agent depot, see Chapter 10, “Deploying monitoring agents across your environment,” on page 325.

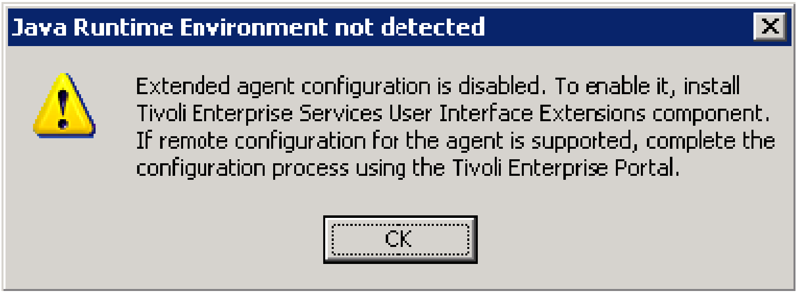
Note: Once you have added the AC bundle to the remote-deployment depot, it is listed among the available packages in the Tivoli Enterprise Portal. Thus your interactive users can select it when invoking the **tacmd AddSystem** command for a particular 64-bit node. However, if they do this, your users will receive this error:

```
KFWITM291E an Agent Configuration schema not found
```

When it becomes necessary to invoke the remote deployment of the Agent Compatibility package, instead use the CLI.

Installing the Embedded Java Runtime and the User Interface Extensions

On nodes where only the Windows OS agent has been installed, either locally or remotely, the Embedded Java Runtime and the Tivoli Enterprise Services User Interface Extensions (the KUE component) are not also installed by default. If you later decide to install an Agent Builder agent on this node, you may not be able to reconfigure this factory agent, and you may receive the error shown in Figure 63.



Java Runtime Environment not detected

Extended agent configuration is disabled. To enable it, install Tivoli Enterprise Services User Interface Extensions component. If remote configuration for the agent is supported, complete the configuration process using the Tivoli Enterprise Portal.

Figure 63. Java Runtime Environment Not Detected error

This error also may occur when you attempt to run a tacmd CLI command on nodes where either the Embedded Java Runtime or the User Interface Extensions are unavailable.

If this happens, you must install the Embedded Java Runtime and the KUE component. Complete one of the following procedures, depending on the installation method you're following:

local GUI installation
Select **Tivoli Enterprise Services User Interface Extensions** from the list of features to install.

local silent installation
Uncomment this line in the silent response file:

```
;KUEWICMA=Tivoli Enterprise Services User Interface Extensions
```

remote installation
First ensure that component UE has been added to the server depot. Once the UE component is available, from the command-line, perform a **tacmd addsystem** command, specifying `-t UE` as the component to install.

Once you have completed one of the above procedures, the Embedded Java Runtime and the Tivoli Enterprise Services User Interface Extensions are installed and can be accessed on this node.

Linux or UNIX: Installing a monitoring agent

Use the following steps to install and configure a distributed monitoring agent on a Linux or UNIX computer.

Table 49. Steps for installing a monitoring agent on Linux or UNIX

Steps	Where to find information
Install the monitoring agent.	"Installing the monitoring agent"
Configure the monitoring agent.	"Configuring the monitoring agent" on page 260 Some agents require additional, agent-specific configuration parameters. See the agent documentation for the specific agents that you are configuring.
Change the file permissions for files on the computer where you installed the agent.	"Postinstallation steps for nonroot installations" on page 261
Start the monitoring agent.	"Starting the monitoring agents" on page 262

Installing the monitoring agent

Use the following steps to install a monitoring agent on a Linux or UNIX computer:

1. In the directory where you extracted the installation files, run the following command:
   ```
   ./install.sh
   ```
2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (/opt/IBM/ITM) or type the full path to a different directory.

   Note: You must not specify the path of the directory containing ./install.sh as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.
3. If the installation directory does not already exist, you are asked if you want to create it. Type y to create this directory and press Enter.
4. The following prompt is displayed:
   ```
   Select one of the following:
   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   3) Install TEMS support for remote seeding
   4) Exit install.

   Please enter a valid number:
   ```
   Note: This prompt might vary depending on the installation image from which you are installing.

   Type 1 to start the installation and display the software license agreement.
5. Press Enter to read through the agreement.
6. Type 1 to accept the agreement and press Enter.
7. Type a 32 character encryption key and press Enter. This key should be the same as the key that was used during the installation of the monitoring server to which this monitoring agent connects.

   Note: This step applies only to those agents that you install from the IBM Tivoli Monitoring installation image. Agents installed from the agent installation image do not need to provide the encryption key.

   A numbered list of available operating systems is displayed.
8. Type 1 to install the IBM Tivoli Monitoring support for your current operating system. Press Enter.

   A numbered list of available components is displayed.

9. Type the number that corresponds to the monitoring agent or agents that you want to install. If you want to install more than one agent, use a comma (,) or a space to separate the numbers for each agent. Press Enter.

   Note: Before you install the Warehouse Proxy Agent or Summarization and Pruning Agent, follow the instructions later in this book for setting up a Tivoli Data Warehouse solution, beginning with Chapter 18, “Tivoli Data Warehouse solutions,” on page 465.

   A list of the components to be installed is displayed.
10. Type `1` to confirm the installation.

    The installation begins.
11. After all of the components are installed, you are asked whether you want to install additional products or product support packages. Type `2` and press Enter.
12. If your IBM Tivoli Monitoring environment is not already secured you will be asked at this point if you want to secure it. If your IBM Tivoli Monitoring environment is already secured this question is skipped. The product installation process creates the majority of directories and files with world write permissions. IBM Tivoli Monitoring provides the secureMain utility to help you keep the monitoring environment secure. You can secure your installation now, or manually execute the secureMain utility later. For more information, see Appendix G, “Securing your IBM Tivoli Monitoring installation on Linux or UNIX,” on page 851.

Continue with “Configuring the monitoring agent.”

Configuring the monitoring agent

Use the following steps to configure your monitoring agent:

1. Run the following command:

   ```
   ./itmcmd config -A pc
   ```

   where *pc* is the product code for your agent. For the UNIX agent, use the product code "ux"; for Linux, use "lz". See Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815 for a list of agent product codes.
2. Press Enter when you are asked if the agent connects to a monitoring server.
3. Type the host name for the monitoring server.
4. Type the protocol that you want to use to communicate with the monitoring server. You have four choices: ip.udp, sna, ip.spipe, or ip.pipe. Press Enter to accept the default protocol (IP.PIPE).
5. If you want to set up a backup protocol, enter that protocol and press Enter. If you do not want to use backup protocol, press Enter without specifying a protocol.
6. Depending on the type of protocol you specified, provide the following information when prompted:

Table 50. UNIX monitoring server protocols and values

Protocol	Value	Definition
IP.UDP	IP Port Number	The port number for the monitoring server. The default is 1918.
SNA	Net Name	The SNA network identifier for your location.
	LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
	Log Mode	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
IP.PIPE	IP.PIPE Port Number	The port number for the monitoring server. The default is 1918.

Table 50. UNIX monitoring server protocols and values (continued)

Protocol	Value	Definition
IP.SPIPE	IP.SPIPE Port Number	The port number for the monitoring server. The default is 3660.

7. Press Enter to *not* specify the name of the KDC_PARTITION.
8. Press Enter when you are asked if you want to configure the connection to a secondary monitoring server. The default value is no.
9. Press Enter to accept the default for the Optional Primary Network Name (none).

You must complete the configuration of the Warehouse Proxy Agent using the Manage Tivoli Enterprise Monitoring Services graphical user interface (GUI), which requires an X11 GUI interface. See the instructions for configuring the Warehouse Proxy for the appropriate Data Warehouse:

- DB2 for Linux, UNIX, and Windows: "Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)" on page 505
- SQL Server: "Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)" on page 558
- Oracle: "Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)" on page 581

If you used a non-root user to install a monitoring agent on a UNIX computer, the file permissions are initially set to a low level. Use the procedure in "Postinstallation steps for nonroot installations" to change these file permissions:

Postinstallation steps for nonroot installations

If you used a non-root user to install a monitoring agent on a UNIX computer, complete the following procedure:

1. Log in to the computer as the root user.
2. Create a new group (such as *itmusers*) to own all of the files in the IBM Tivoli Monitoring installation directory. For an AIX computer, use the `mkgroup` command. For Linux, Solaris, and HP-UX computers, use the `groupadd` command. See the documentation for your operating system for the complete syntax information.
3. Add both the root and the nonroot user to the new group *itmusers*. For an AIX computer, use the `chgrpmem` command. For Linux, Solaris, and HP-UX computers, use the `usermod` command. See the documentation for your operating system for the complete syntax information.
4. Run the following commands and verify that both the root and the nonroot user are in the *itmusers* group:

   ```
   groups root
   groups nonroot
   ```
5. Run the following command to set the permissions and group ownership of CANDLEHOME:

   ```
   CANDLEHOME/bin/secureMain -g itmusers lock
   ```
6. Run the following command to update the system boot scripts:

   ```
   CANDLEHOME/bin/UpdateAutoRun.sh
   ```
7. Start the agent as described in "Starting the monitoring agents" on page 262.

Note:

- Whenever you update the agent, you must log in to the system as the root user and run the secureMain step again. If the agent is running:
  1. Stop the agent.
  2. Run secureMain as described in Step 5.
  3. Restart the agent.

- If you installed IBM Tivoli Monitoring using a nonroot user and did not run secureMain as described in Step 5 on page 261, you must run secureMain before using the **tacmd updateagent** command to update the agent. If you do not run secureMain first, the update process will not complete.
- On AIX systems you should invoke the AIX `slibclean` command:

  ```
  su -c "/usr/sbin/slibclean"
  ```

Starting the monitoring agents

You can either start all agents running on a computer or start individual agents by using the product codes.

To start all monitoring agents, run the following command:

```
./itmcmd agent start all
```

To start specific agents, run the following command:

```
./itmcmd agent start pc pc pc
```

where *pc* is the product code for the agent that you want to start. See Appendix D, "IBM Tivoli product, platform, and component codes," on page 815 for a list of agent product codes.

To start multi-instance agents (agents that may run more than one instance on a computer, like Seibel or Domino® agents), run the following command:

```
./itmcmd agent -o instance_name start pc
```

where *pc* is the product code for the agent that you want to start and *instance_name* is the name that uniquely identifies the instance you want to start. See Appendix D, "IBM Tivoli product, platform, and component codes," on page 815 for a list of agent product codes.

Populating the data warehouse's ManagedSystem table

If your site runs the Tivoli Data Warehouse, each time you install one or more monitoring agents, you must update the warehouse's ManagedSystem table; the `populate_agents.sql` script is provided for this purpose.

- If your site uses DB2 Database for Linux, UNIX, and Windows to manage its Tivoli Data Warehouse, call this stored procedure:

  ```
  db2 call ITMUSER.POPULATE_OSAGENTS();
  ```

- If your site uses Microsoft SQL Server to manage its Tivoli Data Warehouse, run this script at the MS SQL command-line:

  ```
  sqlcmd -i populate_agents.sql [-U my_username -P my_password] [-H my_host])
  ```

- If your site uses Oracle to manage its Tivoli Data Warehouse, start this procedure:

  ```
  POPULATE_OSAGENTS('ITMUSER');
  ```

Where:

ITMUSER
Is the database user for Tivoli Data Warehouse.

my_username
Is the user ID for the Tivoli Data Warehouse (in most cases the same as ITMUSER).

my_password
Is the password for the user.

my_host
Is the optional computer name.

Installing the Tivoli Enterprise Portal desktop client

There are two methods of deploying the desktop client: You can install the desktop client from the installation media and run and maintain it on the local system. You can also use IBM Web Start for Java to download and run the desktop client from the Tivoli Enterprise Portal Server. See “Tivoli Enterprise Portal client” on page 39 for a discussion of the advantages and disadvantages of each type of portal server client available with IBM Tivoli Monitoring.

This section describes how to install the desktop client from the installation media on Windows and Linux computers. When you install the desktop client from the installation media, you must also install support for all the applications that you will be using.

- “Windows: Installing the desktop client”
- “Linux: Installing the desktop client” on page 264

See “Using Web Start to download and run the desktop client” on page 321 for instructions on downloading and running the desktop client from the portal server.

Windows: Installing the desktop client

Complete the following steps to install the Tivoli Enterprise Portal desktop client from the Base Infrastructure DVD or DVD image.

1. On the computer where you want to install the desktop client, start the installation wizard by launching the setup.exe file in the \WINDOWS subdirectory on the installation media.
2. Click **Next** on the Welcome window.

   Note: If you have another IBM Tivoli Monitoring component already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 6.
3. Read and accept the software license agreement by clicking **Accept**.
4. Specify the directory where you want to install the portal desktop client software and accompanying files. The default location is C:\IBM\ITM. Click **Next**.

   Note: If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```
5. Type an encryption key to use. This key must be the same as what was used during the installation of the hub monitoring server to which the client will connect. Click **Next** and then **OK** to confirm the encryption key.
6. On the Select Features window, select **Tivoli Enterprise Portal Desktop Client** from the list of components to install.

   When you select the **Tivoli Enterprise Portal Desktop Client** check box, all of the check boxes in the attached subtree are automatically selected. The support check boxes in the subtree are for installing application support files for distributed monitoring agents to the portal desktop client. It is best to leave all of the support check boxes selected so you do not need to reconfigure application support as new agent types are added to your environment. However, if you purchased an OMEGAMON XE monitoring product and you did not separately purchase IBM Tivoli Monitoring, you should not install support for the distributed operating system monitoring agents nor the Tivoli Universal Agent. If you are installing from the Base DVD, you may see application support for components that are not supported on the operating system. For detailed information about application support, see “Installing and enabling application support” on page 266.

Note: If you are updating an existing installation (you selected **Modify** on the Welcome window), all check boxes on the Select Features window reflect your choices during the initial installation. Clearing a check box has the effect of *uninstalling* the component. Clear a check box only if you want to remove a component.

7. If you want to view IBM Tivoli Enterprise Console events through the Tivoli Enterprise Portal, expand **Tivoli Enterprise Portal Desktop Client** and ensure that **TEC GUI Integration** is selected.
8. Click **Next**.
9. If a monitoring server is not installed on this computer, go to Step 10.

   If you are installing the desktop client on a computer that already has a monitoring server installed, the Agent Deployment window is displayed.

   The Agent Deployment window lists monitoring agents on this installation image that you can add to the agent depot. The agent depot contains agents that you can deploy to remote computers. For information about how to deploy agents in the agent depot to remote computers, see Chapter 10, “Deploying monitoring agents across your environment,” on page 325.

   Note: By default, the agent depot is located in the *itm_installdir*\CMS\depot directory on Windows. If you want to use a different directory, create the directory (if it does not exist) and specify the directory using the DEPOTHOME key in the KBBENV file.

   Select the agents, if any, that you want to add to the agent depot. (You can add agents to the agent depot at a later time by updating your installation.) Click **Next**.
10. If no IBM Tivoli Monitoring component has been previously installed on this computer, a window is displayed for you to select a program folder for the Windows Start menu. Select a program folder and click **Next**. The default program folder name is IBM Tivoli Monitoring.
11. Review the installation summary details. The summary identifies what you are installing and where you chose to install it. Click **Next** to start the installation.

    After installation is complete, a configuration window (called the Setup Type window) is displayed.
12. Clear the check boxes for any components that have already been installed and configured (at the current release level) on this computer, unless you want to modify the configuration. (For example, clear the check box for the Tivoli Enterprise Monitoring Server if it has already been installed and configured on this computer.) If the desktop client is being installed on the same host as the portal server but as part of a separate installation process, you must reconfigure the portal server.

    Click **Next** to start configuring all selected components.
13. Type the host name of the portal server and click **OK**.
14. Click **Finish** to complete the installation.

Linux: Installing the desktop client

Use the following steps to install the Tivoli Enterprise Portal desktop client:

1. In the directory where you extracted the installation files, run the following command:

   ```
   ./install.sh
   ```
2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`) or type the full path to a different directory.

   Note: You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.
3. If the installation directory does not already exist, you are asked if you want to create it. Type `1` to create this directory and press Enter.
4. The following prompt is displayed:

   ```
   Select one of the following:
   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   ```

```
3) Install TEMS support for remote seeding
4) Exit install.

Please enter a valid number:
```

Type `1` to start the installation and display the software license agreement.

5. Press Enter to read through the agreement.
6. Type `1` to accept the agreement and press Enter.
7. Type an 32-character encryption key to use and press Enter. This key should be the same key as that used during the installation of the portal server to which the client will connect.

   A numbered list of available operating systems is displayed.
8. Type `3` to install the desktop client support for your current operating system. Press Enter.

   A message is displayed indicating that the Tivoli Enterprise Portal Desktop Client is about to be installed.
9. Type `1` to confirm the installation.

   The installation begins.
10. After the portal desktop client is installed, you are asked whether you want to install additional products or product support packages. Type `1` and press Enter.

    A numbered list is displayed, including the following application support package:

    ```
    Tivoli Enterprise Portal Desktop Client support
    ```
11. Install the application support package for the portal desktop client.

    All monitoring agents require that application support files be installed on the monitoring servers (hub and remote), portal server, and portal desktop clients in your environment. Application support files contain the information required for agent-specific workspaces, helps, predefined situations, and other data.

    This step installs the application support files for base monitoring agents. The *base* monitoring agents are included with the base IBM Tivoli Monitoring installation package. For detailed information about application support, see “Installing and enabling application support” on page 266.

    a. Type the number that corresponds to `Tivoli Enterprise Portal Desktop Client support` and press Enter.

       A numbered list of base monitoring agents is displayed.

    b. Type the numbers of the base monitoring agents for which you want to install application support, or type the number that corresponds to `All of the above`. Type the numbers on the same line separated by spaces or commas. Press Enter.

       It is best to select all of the base monitoring agents (`All of the above`) so you do not need to reconfigure application support as new agent types are added to your environment.

    c. Type `1` to confirm the installation and press Enter.

       The installation begins.

       Note: This step installs the application support files. However, you must *enable* the application support by configuring the portal desktop client. The next sections shows you how to configure the portal desktop client.
12. After application support for the monitoring agents is installed, you are asked whether you want to install additional products or product support packages. Type `2` and press Enter.

The next step is to configure the desktop client. Use the instructions in “Linux: Configuring the desktop client.”

Linux: Configuring the desktop client

Complete the following steps to configure the desktop client if you installed the client from the IBM Tivoli Monitoring installation media. You do not need to complete this procedure if you obtained the desktop client by using IBM Web Start for Java to download it from the Tivoli Enterprise Portal Server.

1. At the command-line /opt/IBM/ITM/bin directory (or the /bin subdirectory where you installed the product), run the following command:

   ```
   ./itmcmd config -A cj
   ```
2. Press Enter to use the default instance name.
3. Type the host name for the portal server and press Enter.
4. Press Enter when you are asked if you want to use HTTP Proxy support. The default value is no.
5. Start the desktop client:

   ```
   /itmcmd agent start cj
   ```

Installing and enabling application support

Before you can view data collected by monitoring agents, you must install and enable application support for those agents. Application support files provide agent-specific information for workspaces, helps, situations, templates, and other data. Application support for a monitoring agent includes two types of files:

- **SQL files** are required for adding product-provided situations, templates, and policies to the Enterprise Information Base (EIB) tables maintained by the hub monitoring server. These SQL files are also called *seed data*, and installing them on a monitoring server is also called *seeding* the monitoring server.
- **Catalog and attribute (cat and atr) files** are required for presenting workspaces, online help, and expert advice for the agent in Tivoli Enterprise Portal.

All monitoring agents require that application support be configured on all instances of the following infrastructure components:

- Tivoli Enterprise Monitoring Server (both hub and remote monitoring servers)
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal desktop client, if the desktop client was installed from the installation media.

  You do not need to configure application support for desktop clients downloaded from the Tivoli Enterprise Portal Server using IBM Web Start for Java.

Note: New in V6.2.3 is the self-describing agent capability that integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure. The self-describing agent feature makes it possible for new or updated IBM Tivoli Monitoring agents to become operational after installation, without having to perform additional product support installation steps. for more information see "Self-describing agent installation" on page 347.

Application support for monitoring agents is installed independently of where and when the monitoring agents themselves are installed:

- Install application support for a particular type of monitoring agent on the monitoring servers, portal server, and portal desktop clients. Install agents of that type on any managed system in the environment that is compatible with the agent.
- Install application support for a type of monitoring agent *before* or *after* any monitoring agents of that type are installed. After you install application support for a particular type of monitoring agent, you can add any number of agents of that type to your environment without having to install application support again.

For example, you can install application support for a Linux OS monitoring agent (the agent *type*) to a Windows monitoring server (using the IBM Tivoli Monitoring installation media for Windows). Later, you can install any number of Linux OS monitoring agents to Linux computers in your environment (using the IBM Tivoli Monitoring installation media for Linux).

Important:

1. If you are installing a non-OS agent remotely through the Tivoli Enterprise Portal (see "Deploying non-OS agents" on page 331), application support for the agent must be installed on the Tivoli Enterprise Portal Server *before* the agent is deployed.
2. When you install application support for an IBM Tivoli Composite Application Manager agent on the V6.2.2 fix pack 2 (or subsequent) version of the Tivoli Enterprise Monitoring Server, do not select for installation the File Transfer Enablement component. If you do, the File Transfer Enablement component shipped with the V6.2.2 fix pack 2 (or subsequent) monitoring server will be replaced, and the **tacmd getfile**, **tacmd putfile**, and **tacmd executecommand** commands will become inoperative.

Because application support for some IBM Tivoli Monitoring 6.x-based distributed agents is included on the Base Infrastructure DVD, you may see application support files that do not apply to all systems.

Configuring application support is a two-step process:

1. Installing the application support files (from installation media).
2. Enabling the application support (sometimes referred to as *adding* or *activating* the application support).

On the portal server and portal desktop clients, application support is enabled when the component is configured. On monitoring servers, application support is enabled by *seeding* the database with agent-specific information.

The procedures for configuring application support differ by operating system, as summarized in Table 51. On Windows, both installation and enablement of application support are accomplished during the installation of the monitoring servers, portal server, and desktop clients. On Linux or UNIX, this two-step process is more visible, with the enablement step done separately from the installation.

Table 51. Procedures for installing and enabling application support

Operating system	Monitoring servers	Portal server	Desktop clients[1]
Windows	Install and enable application support using installation media.	Install and enable application support using installation media.	Install and enable application support using installation media.
Linux or UNIX	1. Install application support files from installation media. 2. *Seed* the monitoring server using: • **itmcmd support**[2] command, OR • Manage Tivoli Enterprise Monitoring Services window	1. Install application support files from installation media. 2. *Configure* the portal server using: • **itmcmd config** command, OR • Manage Tivoli Enterprise Monitoring Services window	1. Install application support files from installation media. 2. *Configure* the desktop client using: • **itmcmd config** command, OR • Manage Tivoli Enterprise Monitoring Services window
z/OS	This book does not describe configuring application support for a monitoring server installed on a z/OS system. See *Configuring the Tivoli Enterprise Monitoring Server on z/OS* for information and instructions. The portal server and desktop client are not supported on z/OS. If you have a monitoring server on z/OS, use the procedures in this book to configure application support on the portal server and desktop clients.		

Table 51. Procedures for installing and enabling application support (continued)

Operating system	Monitoring servers	Portal server	Desktop clients[1]

1. You need to configure application support for desktop clients that are installed from the installation media. You do not need to configure application support for desktop clients that are installed by using IBM Java Web Start to download the client from the Tivoli Enterprise Portal Server.
2. You can seed a nonlocal monitoring server, even if one is not installed on the local computer, by installing the support using option `3, Install TEMS support for remote seeding`, then using Manage Tivoli Enterprise Monitoring Services to seed the nonlocal monitoring server. You cannot use **itmcmd support** to seed a nonlocal monitoring server.
3. There is no way to uninstall the application support files laid down without uninstalling the Tivoli Enterprise Monitoring Server.

Selecting the correct support media

Application support for IBM Tivoli Monitoring V6.2.3 distributed agents is found on the following Base DVDs:

- *IBM Tivoli Monitoring V6.2.3 Infrastructure DVD* contains the Tivoli Enterprise Monitoring Server and its application support, the Tivoli Enterprise Portal Server and its application support, and the Tivoli Enterprise Portal desktop and browser clients together with their application support, along with the Warehouse Proxy Agent, the Summarization and Pruning Agent, and the Tivoli Performance Analyzer. This DVD is platform-specific: Windows, Linux, or UNIX.
- *IBM Tivoli Monitoring V6.2.3 Agents DVD* contains the monitoring agents listed in Table 52. This DVD is platform-nonspecific: the same DVD is used for installing on Windows, Linux, and UNIX.

The Infrastructure DVD contains application support files for the monitoring server (hub and remote), the portal server and its client, the Warehouse Proxy Agent, the Summarization and Pruning Agent; the Agents DVD contains the application support files for the monitoring agents. To install application support for all IBM Tivoli Monitoring components and agents on distributed nodes, use the procedures documented in this chapter for installing the monitoring servers, the Tivoli Enterprise Portal, and the portal desktop client. See "Configuring application support for agents on the Base DVDs" on page 269 for more information. For instructions on installing application support for the monitoring agents on a monitoring server on z/OS, see *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

Table 52. Product support on the Infrastructure and Agent DVDs

Product Code	Product Name
R3	agentless monitor for AIX operating systems
R5	agentless monitor for HP-UX operating systems
R4	agentless monitor for Linux operating systems
R6	agentless monitor for Solaris operating systems
R2	agentless monitor for Windows operating systems
LZ	monitoring agent for Linux OS
UL	monitoring agent for UNIX Logs
UX	monitoring agent for UNIX OS
NT	monitoring agent for Windows OS
A4	monitoring agent for i5/OS
SY	Summarization and Pruning Agent
HD	Warehouse Proxy Agent
UM	IBM Tivoli Universal Agent

Table 52. Product support on the Infrastructure and Agent DVDs (continued)

Product Code	Product Name
PA	IBM Tivoli Performance Analyzer
P0	Tivoli Performance Analyzer Domain for DB2
P3	Tivoli Performance Analyzer Domain for OS agent
P4	Tivoli Performance Analyzer Domain for Oracle
P6	Tivoli Performance Analyzer Domain for System P
PI	Tivoli Performance Analyzer Domain for ITCAM RT
PU	Tivoli Performance Analyzer Domain for VMware

The Agent DVD delivered with IBM Tivoli Monitoring V6.2.3 must be used instead of the agent installation CDs if you are installing support for the agents listed in the following tables on the component specified. The installer on the agent installation CDs does not support these components on these platforms. To install application support for these agents on distributed components, see “Configuring application support for nonbase monitoring agents” on page 270. To install application support for the monitoring agents on a z/OS monitoring server, see *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

For all other distributed agents, install application support from the product installation CDs. See “Configuring application support for nonbase monitoring agents” on page 270 for instructions.

Configuring application support for agents on the Base DVDs

You install and configure application support for the agents on the Base DVDs when you install and configure distributed monitoring servers, the portal server, and the portal desktop clients using the instructions in this chapter.

- When you install a monitoring server on Linux or UNIX, support files for all agent types on the Base are installed automatically. When you install the portal server and portal desktop client, you can specify which agent types to support.
- When you install a monitoring server, portal server, or portal desktop client on Windows, you can select the Base monitoring agents for which you want to install application support on the Select Features window. By default, all the agents are selected.

In both cases, it is best to install and enable application support for all agent types on the Base DVDs during the initial installation to avoid having to reconfigure application support on all monitoring servers, the portal server, and portal desktop clients as new agent types are added to your environment.

If you need to reconfigure application support, restart the installation from the IBM Tivoli Monitoring installation media. Update the installation by installing application support for additional Base monitoring agents. When you update an installation, keep in mind the following considerations:

- Following an installation update on Linux or UNIX, you must reseed the monitoring server and reconfigure the portal server and desktop clients.
- When you update an installation on Windows, proceed through the installation as you did for the initial installation except as follows:
  - All check boxes on the Select Features window reflect your choices during the initial installation. Do not clear any check boxes. Clearing a check box has the effect of *uninstalling* the component.

    For example, if you are updating the installation to add application support for the i5/OS agent, select only the application support check box for the i5/OS agent. Do not clear any application support check boxes that are already checked unless you want to remove support for those agents.
  - You do not need to reconfigure any components (such as the monitoring server or portal server) that have already been configured.

- You must update the installation on all computers where a monitoring server, portal server, or desktop client is installed.

Before seeding actually begins, as part of the installation process, you will be asked if you want to use the seeding files that are included with the installation media or if you want to use the ones you customized during a previous installation.

Important: Do not install agent application support for the current release on a Tivoli Enterprise Monitoring Server for a prior release (for example, do not install V6.2.3 agent application support on a V6.2 monitoring server). Doing so can cause agents to fail.

When the application support for an agent is updated on the Tivoli Enterprise Portal Server, the current agent DLA template file is renamed to use the `.bak` extension and the latest version of the template file is installed. If the agent DLA template file was edited to remove the IP address filtering for private network addresses, you must update the new version of the agent DLA template file to contain the edits that are in the `.bak` version of the template file. For more information on editing an agent DLA template to remove private network address filtering, see *Using the Tivoli Management Services Discovery Library Adapter* in the *IBM Tivoli Monitoring: Administrator's Guide*.

Configuring application support for nonbase monitoring agents

The term *nonbase* is used here to refer to distributed monitoring agents for which support is included on the current Agents DVD, and to other products such as IBM Tivoli Monitoring for Applications, OMEGAMON XE monitoring agents, and IBM Tivoli Composite Application Manager monitoring agents that provide their own support files.

Some monitoring agents that run on z/OS or z/VM are packaged with their own CD that contains the data files for adding application support to distributed components. Other monitoring agents that run on z/OS are packaged with a CD that contains data files for a number of agents. If in doubt, see the configuration guide for each of your monitoring agents to find the exact name of the CD to use.

The following table shows you which installation media to use and where to find instructions for installing application support, according to the type of agent (distributed or z/OS) and whether the agent reports to a distributed or z/OS monitoring server.

Table 53. Installation media and instructions for installing application support for nonbase monitoring agents

Agent	Monitoring server	Installation media	Instructions for installing application support
Distributed	Distributed	*IBM Tivoli Monitoring V6.2: Agents DVD* Agent product installation CDs	Follow the instructions in this section to install application support on the monitoring server, portal server, and desktop client.
Distributed	z/OS	*IBM Tivoli Monitoring V6.2: Agents DVD* Agent product installation CDs	• Follow the instructions in this section to install application support on the portal server and desktop clients. • Follow the instructions in *Configuring the Tivoli Enterprise Monitoring Server on z/OS* to install application support on the monitoring servers.
z/OS	Distributed	Data Files CD	Follow the instructions in this section to install application support on the monitoring server, portal server, and desktop client.

Table 53. Installation media and instructions for installing application support for nonbase monitoring agents (continued)

Agent	Monitoring server	Installation media	Instructions for installing application support
z/OS	z/OS	Data Files CD	• Follow the instructions in this section to install application support on the portal server and desktop clients. • Follow the instructions in *Configuring the Tivoli Enterprise Monitoring Server on z/OS* to install application support on the monitoring servers.

Use the instructions in the following sections to install application support for nonbase distributed or z/OS monitoring agents on the distributed monitoring servers, portal server, and portal desktop clients in your environment:

- "Installing application support on monitoring servers"
- "Installing application support on the Tivoli Enterprise Portal Server" on page 277
- "Installing application support on the Tivoli Enterprise Portal desktop client" on page 280

Each of these sections provides information for installing application support files to a single component, such as the monitoring server. If you have multiple components on the same computer (such as the monitoring server and the portal server), combine steps from the individual sections to install application support to all components.

When the application support for an agent is updated on the Tivoli Enterprise Portal Server, the current agent DLA template file is renamed to use the `.bak` extension and the latest version of the template file is installed. If the agent DLA template file was edited to remove the IP address filtering for private network addresses, you must update the new version of the agent DLA template file to contain the edits that are in the `.bak` version of the template file. For more information on editing an agent DLA template to remove private network address filtering, see *Using the Tivoli Management Services Discovery Library Adapter* in the *IBM Tivoli Monitoring: Administrator's Guide*.

Installing application support on monitoring servers

Use the following procedures to install application support for nonbase monitoring agents on distributed monitoring servers (hub or remote) in your environment:

- "Windows: Installing application support on a monitoring server"
- "Linux or UNIX: Installing application support on a monitoring server" on page 275

Windows: Installing application support on a monitoring server: Complete the following steps to install application support for monitoring agents on Windows monitoring servers.

Notes:

1. The monitoring server is stopped during this process.
2. If you are running in a Hot Standby environment, shut down your Hot Standby (that is, mirror) monitoring server before completing this procedure. Restart the Hot Standby monitoring server only after you have seeded the hub server.

1. In the \WINDOWS subdirectory on the agent product CD (for distributed products) or data files CD (for z/OS products), double-click the setup.exe file to launch the installation.
2. Click **Next** on the Welcome window.

   Note: If a monitoring agent is already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 6.

3. On the Install Prerequisites window, read the information about the required levels of IBM Global Security Toolkit (GSKit) and IBM Java.

   The check box for each prerequisite is cleared if the correct level of the software is already installed. Otherwise, the check box is selected to indicated that the software is to be installed. If you are installing support from the data files CD for z/OS agent products, you might be prompted to install Sun Java Runtime Environment (JRE) 1.4.2, even if you have already installed IBM JRE 1.5 with the distributed components of Tivoli Management Services. The two versions can coexist, and installation of application support for some monitoring agents requires Sun Java 1.4.2. You might also see a message indicating that you can decline the installation of JRE 1.4.2 and that accepting installation of JRE 1.4.2 results in uninstallation of other Java versions. Ignore this message, because you cannot proceed without accepting the installation of Sun Java 1.4.2, and accepting the installation does not uninstall IBM Java 1.5.

4. Click **Next**. The prerequisite software is installed if necessary.

   If the installation program installs IBM GSKit or IBM JRE, you might be prompted to restart the computer when the installation is complete. If so, you receive an `abort` message with a **Severe error** heading. This is normal and does not indicate a problem.

   If you are prompted to reboot, do the following:

   a. Click **OK** on the window prompting you to reboot.

   b. Click **No** on the window asking whether you want to view the abort log.

   c. Restart the computer.

   d. Restart the installation program.

5. Click **Accept** to accept the license agreement.

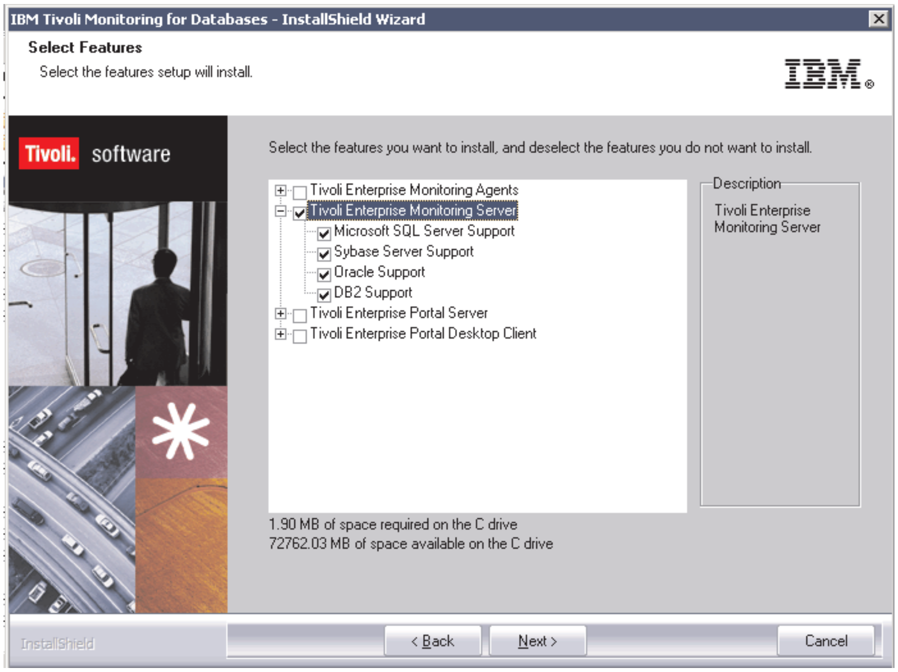
6. If you see a message regarding installed versions being newer than the agent installation, click **OK** to ignore this message.

7. Select the application support packages that you want to install:

   a. On the Select Features window, select **Tivoli Enterprise Monitoring Server**.

   b. Expand the **Tivoli Enterprise Monitoring Server** node to display a list of application support packages that you can install on the monitoring server.

      The following example shows the application support packages available with the IBM Tivoli Monitoring for Databases product:


IBM Tivoli Monitoring for Databases - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- □ Tivoli Enterprise Monitoring Agents
- ☑ Tivoli Enterprise Monitoring Server
  - ☑ Microsoft SQL Server Support
  - ☑ Sybase Server Support
  - ☑ Oracle Support
  - ☑ DB2 Support
- □ Tivoli Enterprise Portal Server
- □ Tivoli Enterprise Portal Desktop Client

Description: Tivoli Enterprise Monitoring Server

1.90 MB of space required on the C drive
72762.03 MB of space available on the C drive

Figure 64. IBM Tivoli Monitoring for Databases: application support packages

Initially, all application support packages are selected.

c. Clear the check boxes for application support packages that you do not want to install.

Note: If you are updating an existing installation (you selected **Modify** on the Welcome window), all check boxes on the Select Features window reflect your choices during the initial installation. Clearing a check box has the effect of *uninstalling* the component (for example, the Tivoli Enterprise Monitoring Server) or product package. Clear a checkbox for an application support package only if you want to remove the application support.

d. If you have other components installed on the same computer, such as the desktop client, also select those components to install the component-specific application support.

e. Click **Next**.

8. (*Distributed agents only*) If you want to add the agent to the deployment depot, select the agent and click **Next**.

   This step does not occur for z/OS agents. z/OS agents cannot be added to the deployment depot.

9. On the Start Copying Files window, read the list of actions to be performed and click **Next** to start the installation.

   The application support packages that you selected are installed.

10. On the Setup Type window, do the following:

    a. Select **Install application support for a local/remote Tivoli Enterprise Monitoring Server**.

    b. Optionally, select the check box for launching the Manage Tivoli Enterprise Monitoring Services window. (If selected, this window is displayed when the installation procedure is finished.)

    c. Clear the check boxes for any components that you have already installed on this computer, such as the monitoring server.

    d. Click **Next**.

11. On the two Tivoli Enterprise Monitoring Server configuration windows that are displayed, make sure the information is correct and click either **Next** or **OK**.

12. Enable application support on the monitoring server:

In Step 7 on page 272, you selected the application support packages that you wanted to install on the monitoring server. In this step, you activate the application support through a process known as *seeding* the monitoring server.

a. Specify the location of the monitoring server to which to add application support. You have two choices:

   - **On this computer**
   - **On a different computer**

   Click **OK**.

   For additional information about these parameters, press the **Help** button.

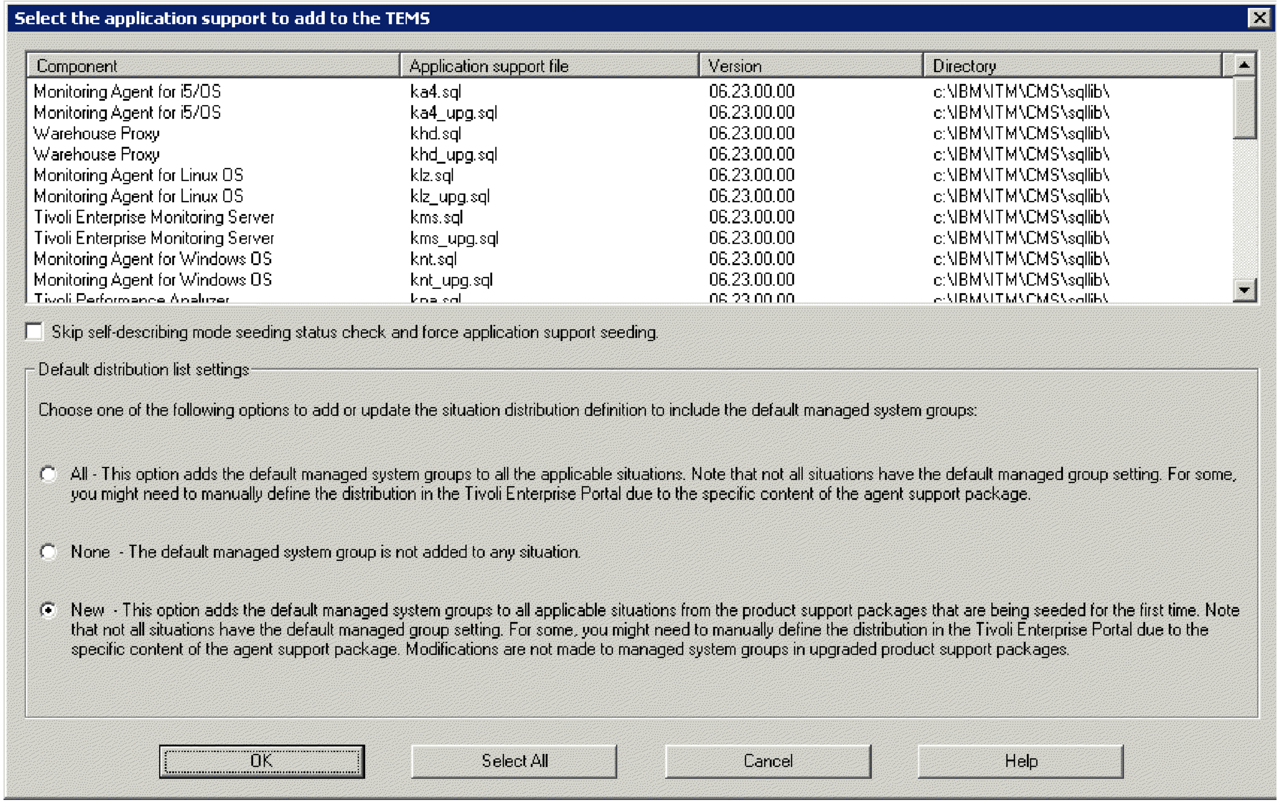
b. If you are updating a hub Tivoli Enterprise Monitoring Server, you are asked to choose whether you want to add the default managed system list when you process the application-support files:

   All Add the default managed system groups to all applicable situations.

   New Add the default managed system groups to all applicable situations from the product support packages being seeded for the first time. Modifications are not made to managed system groups in previously upgraded product support packages.

   None The default managed system group is not added to any situation.

   Note: Not all situations support the default managed group setting. For some, you might need to manually define the distribution using the Tivoli Enterprise Portal due to the specific content of the agent support package.



Select the application support to add to the TEMS

Component	Application support file	Version	Directory
Monitoring Agent for i5/OS	ka4.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Monitoring Agent for i5/OS	ka4_upg.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Warehouse Proxy	khd.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Warehouse Proxy	khd_upg.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Monitoring Agent for Linux OS	klz.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Monitoring Agent for Linux OS	klz_upg.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Tivoli Enterprise Monitoring Server	kms.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Tivoli Enterprise Monitoring Server	kms_upg.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Monitoring Agent for Windows OS	knt.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Monitoring Agent for Windows OS	knt_upg.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\
Tivoli Performance Analyzer	kpa.sql	06.23.00.00	c:\IBM\ITM\CMS\sqllib\

□ Skip self-describing mode seeding status check and force application support seeding.

Default distribution list settings

Choose one of the following options to add or update the situation distribution definition to include the default managed system groups:

○ All - This option adds the default managed system groups to all the applicable situations. Note that not all situations have the default managed group setting. For some, you might need to manually define the distribution in the Tivoli Enterprise Portal due to the specific content of the agent support package.

○ None - The default managed system group is not added to any situation.

● New - This option adds the default managed system groups to all applicable situations from the product support packages that are being seeded for the first time. Note that not all situations have the default managed group setting. For some, you might need to manually define the distribution in the Tivoli Enterprise Portal due to the specific content of the agent support package. Modifications are not made to managed system groups in upgraded product support packages.

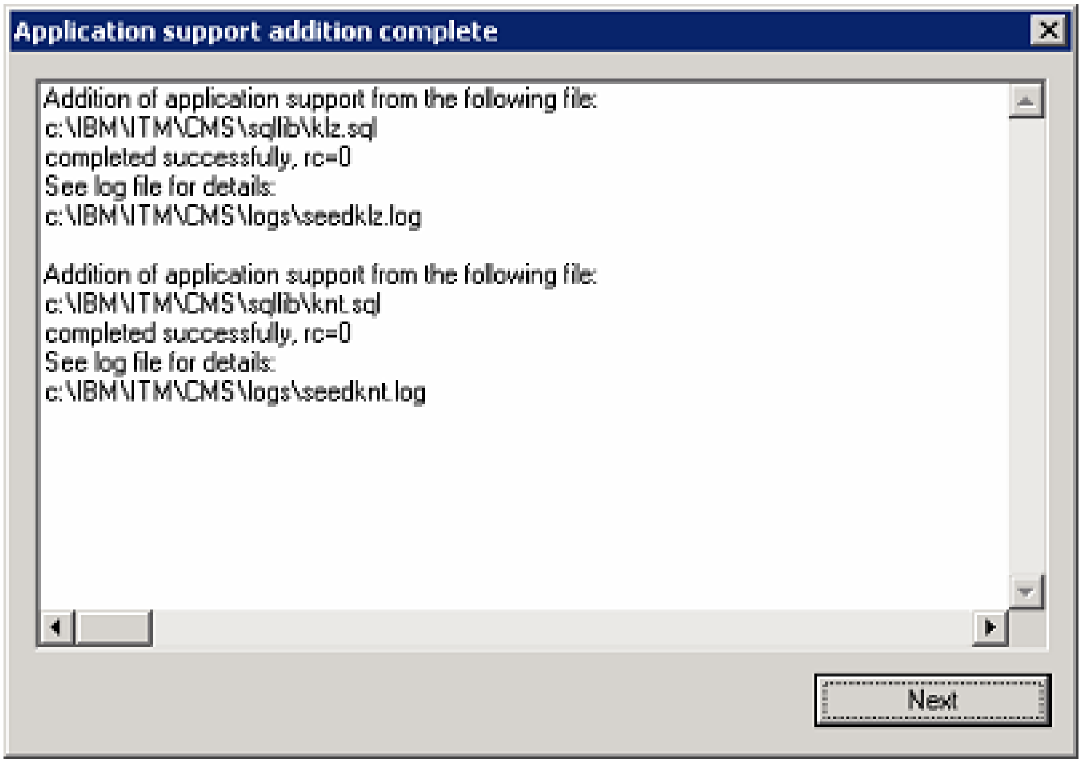
Figure 65. The Select the Application Support to Add to the TEMS window

c. For each product-specific support package the installer checks if the Tivoli Enterprise Monitoring Server database was previously seeded with product-specific support in the self-describing mode. If so, the selected support file is excluded from the Tivoli Enterprise Monitoring Server seeding process. If you want to overwrite the support that was seeded in self-describing mode, you can select the option to **Skip self-describing mode seeding status check**.

d. Click **OK** on the **Select the Application Support to Add to the TEMS** window.

This window lists the application support packages that you selected in Step 7 on page 272. Click **OK** to begin seeding the monitoring server (using the SQL files listed on this window). This process can take up to 20 minutes.

e. Click **Next** on the message that provides results for the process of adding application support (see Figure 66).



```
Addition of application support from the following file:
c:\IBM\ITM\CMS\sqllib\klz.sql
completed successfully, rc=0
See log file for details:
c:\IBM\ITM\CMS\logs\seedklz.log

Addition of application support from the following file:
c:\IBM\ITM\CMS\sqllib\knt.sql
completed successfully, rc=0
See log file for details:
c:\IBM\ITM\CMS\logs\seedknt.log
```

Figure 66. Application Support Addition Complete window

A return code of 0 (`rc=0`) indicates that the process succeeded.

Note: If the Application Support Addition Complete window is not displayed after 20 minutes, look in the IBM\ITM\CNPS\Logs\seedk*pp*.log files (where *pp* is the two-character code for each monitoring agent) for diagnostic messages that help you determine the cause of the problem.

13. Click **Finish** to close the installation wizard.

Linux or UNIX: Installing application support on a monitoring server: Complete the following steps to install application support for monitoring agents on a UNIX or Linux monitoring server:

1. Stop the monitoring server by running the following command:

```
./itmcmd server stop tems_name
```

Note: If you are running in a Hot Standby environment, shut down your Hot Standby (that is, mirror) monitoring server before completing this procedure. You may restart the Hot Standby monitoring server only after you have seeded the hub server.

2. Run the following command from the installation media (the agent product CD for distributed agent products or the data files CD for z/OS agent products):

```
./install.sh
```

3. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`) or type the full path to the installation directory you used.
4. The following prompt is displayed:

   ```
   Select one of the following:

   1) Install products to the local host.
   2) Exit install.

   Please enter a valid number:
   ```

   Enter `1` to start the installation.

   The software license agreement is displayed.
5. Read through the agreement, then type `1` and press Enter to accept it.

   The installer presents a list of installable components for the operating system you are currently running.
6. (*Distributed agent products only, optional*) If you are using a distributed agent product CD, optionally install monitoring agents to this computer. For example, you can install an OS monitoring agent to monitor the operating system. Complete the following steps if you want to install monitoring agents. If you do not want to install monitoring agents, skip to Step 8 on page 277.
   a. Enter `1` to install the IBM Tivoli Monitoring components for your current operating system.
   b. Enter `1` to confirm your selection.

      The installer presents a numbered list of monitoring agents that you can install.
   c. Enter the numbers of the monitoring agents that you want to install or enter the number that corresponds to `All of the above`. Enter more than one number on the same line separated by spaces or commas (,).

      A list of the monitoring agents to be installed is displayed.
   d. Enter `1` to confirm the installation.
   e. After the monitoring agents are installed, the installer asks if you want to install additional products or support packages. Enter `1` and go to Step 8 on page 277.
7. (*z/OS agent products only, required*) If you are using a data files CD, complete the following steps to install required metaprobes. (helper programs) for the monitoring agents that you want to support.
   a. Enter the number of the operating system. The default value is the current operating system.
   b. Enter `y` to confirm your selection.

      The installer presents a numbered list of monitoring agents.
   c. Enter the numbers of the monitoring agents for which you want to install application support, or enter the number that corresponds to `All of the above`. Enter more than one number on the same line separated by spaces or commas (,).

      The installer displays the list of monitoring agents that you selected. For example:

      ```
      The following products will be installed:

        OMEGAMON XE for CICS on z/OS v 4.2.0
        OMEGAMON XE for DB2 PE and PM on z/OS v 4.2.0 and V 5.1.0
        OMEGAMON XE on z/OS  v 4.2.0
      ```

      Note: The prompt (`The following products will be installed`) seems to indicate that the installer is about to install the listed monitoring agents, which is true for distributed agents. However, for z/OS agents, only metaprobes for the monitoring agents are installed. You cannot install z/OS agents on a monitoring server on Linux or UNIX (and you cannot install monitoring agents from a data files CD).
   d. Enter `1` to confirm the installation.

e. After the metaprobes are installed, the installer asks if you want to install additional products or support packages. Enter `y`.

8. Install the application support package for the Tivoli Enterprise Monitoring Server:

   a. Enter the number for `Tivoli Enterprise Monitoring Server support`.

      A list of the monitoring agents for which you can install application support is displayed.

   b. Enter the numbers of the monitoring agents for which you want to install application support, or enter the number that corresponds to `All of the above`. Enter the numbers on the same line separated by spaces or commas (,).

   c. Enter `1` to confirm the installation.

      The installation begins.

9. You are asked if you want to install application support on the Tivoli Enterprise Monitoring Server. If you reply yes, application support is automatically added.

   If you disagree, you can manually add application support later; see "Installing and enabling application support" on page 266.

10. If you are updating a hub Tivoli Enterprise Monitoring Server, you are asked to choose whether you want to add the default managed system groups when you process the application-support files, as shown in Figure 48 on page 221:

    All Add the default managed system groups to all applicable situations.

    New Add the default managed system groups to all applicable situations from the product support packages being seeded for the first time. Modifications are not made to managed system groups in previously upgraded product support packages.

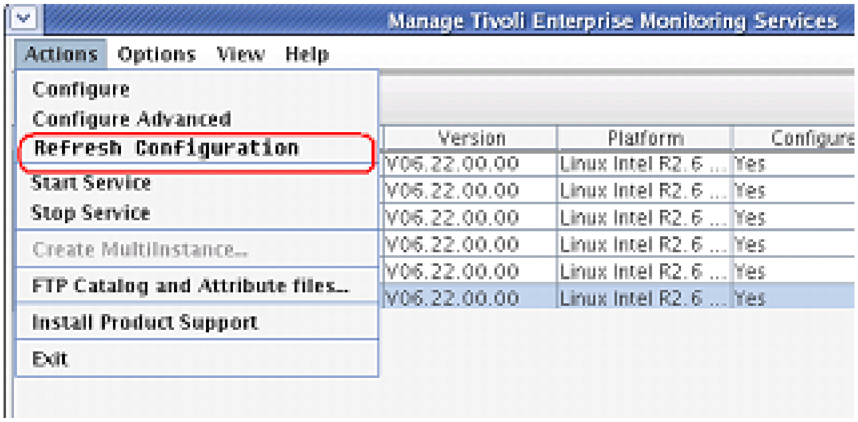
    None The default managed system group is not added to any situation.

    Note: Not all situations support the default managed group setting. For some, you might need to manually define the distribution using the Tivoli Enterprise Portal due to the specific content of the agent support package.

11. After you have added application support for one or more agents, you must refresh the monitoring server configuration:

    a. Start Manage Tivoli Enterprise Monitoring Services.

    b. Pull down the **Actions** menu, and select the **Refresh Configuration** option (see Figure 67).



Version	Platform	Configure
V06.22.00.00	Linux Intel R2.6 ...	Yes
V06.22.00.00	Linux Intel R2.6 ...	Yes
V06.22.00.00	Linux Intel R2.6 ...	Yes
V06.22.00.00	Linux Intel R2.6 ...	Yes
V06.22.00.00	Linux Intel R2.6 ...	Yes
V06.22.00.00	Linux Intel R2.6 ...	Yes

Figure 67. Refresh Configuration menu option

Installing application support on the Tivoli Enterprise Portal Server

Use the following procedures to install application support for nonbase monitoring agents on your portal server:

- "Windows: Installing application support on a portal server" on page 278

- “Linux or AIX: Installing application support on a portal server” on page 279

Windows: Installing application support on a portal server: Complete the following steps to install application support for monitoring agents on a Windows portal server:

1. Stop the portal server:
   a. Open Manage Tivoli Enterprise Monitoring Services.
   b. Right-click **Tivoli Enterprise Portal Server** and click **Stop**.
2. In the /WINDOWS subdirectory on the agent product CD (for distributed products) or data files CD (for z/OS products), double-click the setup.exe file to launch the installation.
3. Click **Next** on the Welcome window.

   Note: If a monitoring agent is already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 7.
4. On the Install Prerequisites window, read the information about the required levels of IBM Global Security Toolkit (GSKit) and IBM Java.

   The check box for each prerequisite is cleared if the correct level of the software is already installed. Otherwise, the check box is selected to indicated that the software is to be installed. If you are installing support from the data files CD for z/OS agent products, you might be prompted to install Sun Java Runtime Environment (JRE) 1.4.2, even if you have already installed IBM JRE 1.5 with the distributed components of Tivoli Management Services. The two versions can coexist, and installation of application support for some monitoring agents requires Sun Java 1.4.2. You might also see a message indicating that you can decline the installation of JRE 1.4.2 and that accepting installation of JRE 1.4.2 results in uninstallation of other Java versions. Ignore this message, because you cannot proceed without accepting the installation of Sun Java 1.4.2, and accepting the installation does not uninstall IBM Java 1.5.
5. Click **Next**. The prerequisite software is installed if necessary.

   If the installation program installs IBM GSKit or IBM JRE, you might be prompted to restart the computer when the installation is complete. If so, you receive an `abort` message with a **Severe error** heading. This is normal and does not indicate a problem.

   If you are prompted to reboot, do the following:
   a. Click **OK** on the window prompting you to reboot.
   b. Click **No** on the window asking whether you want to view the abort log.
   c. Restart the computer.
   d. Restart the installation program.
6. Click **Accept** to accept the license agreement.
7. If you see a message regarding installed versions being newer than the agent installation, click **OK** to ignore this message.
8. Select the application support packages that you want to install:
   a. On the Select Features window, select **Tivoli Enterprise Portal Server**.
   b. Expand the **Tivoli Enterprise Portal Server** node to display a list of application support packages that you can install on the portal server.

      Initially, all application support packages are selected.
   c. Clear the check boxes for application support packages that you do not want to install.

      Note: If you are updating an existing installation (you selected **Modify** on the Welcome window), all check boxes on the Select Features window reflect your choices during the initial installation. Clearing a check box has the effect of *uninstalling* the component (for example, the Tivoli Enterprise Portal Server) or product package. Clear a checkbox for an application support package only if you want to remove the application support.

   d. If you have other components installed on the same computer, such as the desktop client, also select those components to install the component-specific application support.

   e. Click **Next**.

9. On the Start Copying Files window, read the list of actions to be performed and click **Next** to start the installation.

   The application support packages that you selected are installed.

10. On the Setup Type window, clear any components that you have already installed and configured on this computer. Click **Next**.

11. Type the host name for the portal server and click **Next**.

12. Click **Finish** to complete the installation wizard.

13. Restart the portal server.

Linux or AIX: Installing application support on a portal server: Complete the following steps to install application support for monitoring agents on a Linux or AIX portal server:

1. Stop the portal server by running the following command:

   ```
   ./itmcmd agent stop cq
   ```

2. Run the following command from the installation media (the agent product CD for distributed agent products or the data files CD for z/OS agent products):

   ```
   ./install.sh
   ```

3. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`) or enter the full path to the installation directory you used.

4. The following prompt is displayed:

   ```
   Select one of the following:

   1) Install products to the local host.
   2) Exit install.

   Please enter a valid number:
   ```

   Enter `1` to start the installation.

   The software license agreement is displayed.

5. Read through the agreement, then type `1` and press Enter to accept it.

   The installer presents a list of installable products for the operating systems you are currently running.

6. (*Distributed agent products only, optional*) If you are using a distributed agent product CD, optionally install monitoring agents to this computer. For example, you can install an OS monitoring agent to monitor the operating system. Complete the following steps if you want to install monitoring agents. If you do not want to install monitoring agents, skip to Step 7.

   a. Enter `1` for your current operating system.

   b. Enter `1` to confirm your selection.

      The installer presents a numbered list of monitoring agents that you can install.

   c. Enter the numbers of the monitoring agents that you want to install or enter the number that corresponds to `All of the above`. Enter more than one number on the same line separated by spaces or commas (,).

      A list of the monitoring agents to be installed is displayed.

   d. Enter `1` to confirm the installation.

   e. After the monitoring agents are installed, the installer asks if you want to install additional products or support packages. Enter `1`.

7. Install the application support packages for the portal server and browser client. Install application support packages for the portal desktop client if a desktop client is installed on this computer.

The numbered list of items presented by the installer includes the following application support packages. (The numbers might vary from this example.)

```
28) Tivoli Enterprise Portal Browser Client support
29) Tivoli Enterprise Portal Desktop Client support
30) Tivoli Enterprise Portal Server support
```

Note: The Tivoli Enterprise Portal Browser Client support package is portal server code that supports the browser clients. You *must* install the browser client support package on the computer where you install the portal server.

Repeat the following steps for each support package:

a. Enter the number that corresponds to the support package (for example, `28`).

   A numbered list of monitoring agents is displayed.

b. Enter the numbers that correspond to the monitoring agents for which you want to install the application support package, or enter the number that corresponds to `All of the above`. Type the numbers on the same line separated by spaces or commas (,).

c. Enter `1` to confirm the installation.

   The installation begins.

d. After the support package is installed, you are asked whether you want to install additional products or product support packages. Enter `1` to install an additional package and repeat the preceding steps. Enter `2` if you are finished installing support packages.

8. Stop the portal server by running the following command:

   ```
   ./itmcmd agent stop cq
   ```

9. Run the following command to configure the portal server with the new agent information:

   ```
   ./itmcmd config -A cq
   ```

   Complete the configuration as prompted. For information about configuring the portal server, see “Configuring the portal server on Linux or AIX: command-line procedure” on page 240.

10. Restart the portal server by running the following command:

    ```
    ./itmcmd agent start cq
    ```

Installing application support on the Tivoli Enterprise Portal desktop client

Use the following procedures to install application support for nonbase monitoring agents on each computer where you are running a desktop client.

Note: You must install application support on desktop clients that were installed from the installation media. You do not need to install application support on desktop clients that were obtained by using IBM Web Start for Java to download the client from the Tivoli Enterprise Portal Server.

- “Windows: Installing application support on a desktop client”
- “Linux: Installing application support on a desktop client” on page 282

Windows: Installing application support on a desktop client: Complete the following steps to install application support for monitoring agents on a Windows desktop client:

1. Stop the portal desktop client:
   a. Open Manage Tivoli Enterprise Monitoring Services.
   b. Right-click **Tivoli Enterprise Portal Desktop Client** and click **Stop**.
2. In the /WINDOWS subdirectory on the agent product CD (for distributed products) or data files CD (for z/OS products), double-click the setup.exe file to launch the installation.
3. Click **Next** on the Welcome window.

   Note: If a monitoring agent is already installed on this computer, select **Modify** on the Welcome window to indicate that you are updating an existing installation. Click **OK** on the message telling you about preselected items. Then skip to Step 7.

4. On the Install Prerequisites window, read the information about the required levels of IBM Global Security Toolkit (GSKit) and IBM Java.

   The check box for each prerequisite is cleared if the correct level of the software is already installed. Otherwise, the check box is selected to indicated that the software is to be installed. You might be prompted to install Sun Java Runtime Environment (JRE) 1.4.2, even if you have already installed IBM JRE 1.5 with the distributed components of Tivoli Management Services. The two versions can coexist, and installation of application support for some monitoring agents requires Sun Java 1.4.2. You might also see a message indicating that you can decline the installation of JRE 1.4.2 and that accepting installation of JRE 1.4.2 results in uninstallation of other Java versions. Ignore this message, because you cannot proceed without accepting the installation of Sun Java 1.4.2, and accepting the installation does not uninstall IBM Java 1.5.

Tip

If you are installing support from the data files CD for z/OS agent products, you might be prompted to install Java Runtime Environment (JRE) 1.4.2, even if you have already installed IBM JRE 1.5 with the distributed components of Tivoli Management Services. You might also see a message indicating that you can decline the installation of JRE 1.4.2 and that accepting installation of JRE 1.4.2 results in uninstallation of other Java versions. Ignore this message, because you cannot proceed without accepting the installation of Java 1.4.2, and accepting the installation of Java 1.4.2 does not uninstall IBM Java 1.5. The two versions can coexist. However, the most recently installed version of Java becomes the active version, and the distributed components of Tivoli Management Services V6.2.0 require that JRE 1.5 be the active version.

To change the active version back to JRE 1.5 after you complete installation of application support, follow these steps:

a. Open the Windows Control Panel by selecting **Start → Settings → Control Panel**.
b. From the Windows Control Panel, select **IBM Control Panel for Java**.
c. On the Java tab of the Java Control Panel, click the **View** button in the Java Application Runtime Settings section.
d. On the JNLP Runtime Settings window, select version 1.5, and make sure the **Enabled** checkbox is selected.
e. Click **OK** twice to save your settings and exit the Java Control Panel.
f. From the Windows Control Panel, select **Java Plug-in**.
g. On the **Advanced** tab of the Java Plug-in Control Panel, make sure that **JRE 1.5.0** is selected. If you change the setting, click **Apply**.
h. Close the Java Plug-in Control Panel window and the Windows Control Panel.

5. Click **Next** to continue. The prerequisite software is installed if necessary.

   If the installation program installs IBM GSKit or IBM JRE, you might be prompted to restart the computer when the installation is complete. If so, you receive an `abort` message with a **Severe error** heading. This is normal and does not indicate a problem.

   If you are prompted to reboot, do the following:

   a. Click **OK** on the window prompting you to reboot.
   b. Click **No** on the window asking whether you want to view the abort log.
   c. Restart the computer.
   d. Restart the installation program.
6. Read the software license agreement and click **Accept**.
7. If you see a message regarding installed versions being newer than the agent installation, click **OK** to ignore this message.

8. Select the application support packages that you want to install:
   a. On the Select Features window, select **Tivoli Enterprise Portal Desktop Client**.
   b. Expand the **Tivoli Enterprise Portal Desktop Client** node to display a list of application support packages that you can install on the portal server.

      Initially, all application support packages are selected.
   c. Clear the check boxes for application support packages that you do not want to install.

      Note: If you are updating an existing installation (you selected **Modify** on the Welcome window), all check boxes on the Select Features window reflect your choices during the initial installation. Clearing a check box has the effect of *uninstalling* the component or product package. Clear a check box for an application support package only if you want to remove the application support.
   d. Click **Next**.
9. On the Start Copying Files window, read the list of actions to be performed and click **Next** to start the installation.

   The application support packages that you selected are installed.
10. On the Setup Type window, clear any components that you have already installed and configured on this computer. Click **Next**.
11. Type the host name for the portal server and click **Next**.
12. Click **Finish** to complete the installation wizard.

Linux: Installing application support on a desktop client: Complete the following steps to install application support for monitoring agents on a Linux desktop client:

Note: Stop the desktop client before performing this procedure.

1. Stop the desktop client by running the following command:
   ```
   ./itmcmd agent stop cj
   ```
2. Run the following command from the installation media (the agent product CD for distributed agent products or the data files CD for z/OS agent products):
   ```
   ./install.sh
   ```
3. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`) or enter the full path to the installation directory you used.
4. The following prompt is displayed:
   ```
   Select one of the following:

   1) Install products to the local host.
   2) Exit install.

   Please enter a valid number:
   ```
   Enter `1` to start the installation.
5. Read the software license agreement; then type `1` and press Enter to accept it.

   The installer presents a list of installable components for your current operating system.
6. (*Distributed agent products only, optional*) If you are using a distributed agent product CD, optionally install monitoring agents to this computer. For example, you can install an OS monitoring agent to monitor the operating system. Complete the following steps if you want to install monitoring agents. If you do not want to install monitoring agents, skip to Step 7 on page 283.
   a. Enter `1` for your current operating system.
   b. Enter `1` to confirm your selection.

      The installer presents a numbered list of monitoring agents that you can install.

   c. Enter the numbers of the monitoring agents that you want to install or enter the number that corresponds to `All of the above`. Enter more than one number on the same line separated by spaces or commas (,).

      A list of the monitoring agents to be installed is displayed.

   d. Enter `1` to confirm the installation.

   e. After the monitoring agents are installed, the installer asks if you want to install additional products or support packages. Enter `1`.

7. Install the application support package for the portal desktop client:

   a. Enter the number that corresponds to `Tivoli Enterprise Portal Desktop Client support`.

      A numbered list of monitoring agents is displayed.

   b. Enter the numbers that correspond to the monitoring agents for which you want to install the application support package, or enter the number that corresponds to `All of the above`. Type the numbers on the same line separated by spaces or commas (,).

   c. Enter `1` to confirm the installation.

      The installation begins.

8. After application support for all monitoring agents is installed, you are asked whether you want to install additional products or product support packages. Enter `2`.

9. Run the following command to configure the desktop client with the new agent information:

   ```
   ./itmcmd config -A cj
   ```

   Complete the configuration as prompted. For information about configuring the desktop client, see “Linux: Configuring the desktop client” on page 265.

10. Restart the desktop client by running the following command:

    ```
    ./itmcmd agent start cj
    ```

Configuring application support on nonlocal monitoring servers

The following sections contain procedures for installing and enabling application support on a monitoring server (hub or remote) that is located on a remote computer. For example, you might need to install application support on a hub monitoring server on z/OS for a monitoring agent running on Windows or a Linux operating system. Or you might want to install application support on a monitoring server or client on a different computer than the one on which you are installing an agent or a portal server. If you are using a z/OS hub and you want to aggregate and prune historical data for any agent, you must transfer the catalog and attribute files for the Summarization and Pruning Agent to the hub.

The procedures in this section require that either a Tivoli Enterprise Portal Server or a Tivoli Enterprise Monitoring Server be installed on the Windows, Linux, or UNIX computer and configured to communicate with the monitoring server on the nonlocal computer.

- “Configuring application support on a nonlocal monitoring server from a Windows system”
- “Configuring application support on a nonlocal monitoring server from a Linux or UNIX system” on page 285

Configuring application support on a nonlocal monitoring server from a Windows system

This section describes how you can install and enable application support on a nonlocal monitoring server from your local Windows system. The nonlocal monitoring server can be a hub or remote monitoring server installed on a Windows, Linux, UNIX, or z/OS computer. A monitoring server or a portal server must be installed on the local computer and application support for the agent or agents must be installed on it.

You add support to a nonlocal monitoring server by copying the k*pp*.cat and k*pp*.atr files for each from the local computer to the nonlocal computer. If you are adding support to a nonlocal hub monitoring server, you use Manage Tivoli Enterprise Monitoring Services to transfer the k*pp*.sql files for each agent.

Copying the CAT and ATR files to the nonlocal monitoring server: Copy the .cat and .atr files for the agents you want to support from the local Windows monitoring server to the nonlocal monitoring server. If you use ftp, copy the files in ASCII format. The .cat and .atr files are located in the following directories on the local monitoring server:

- .cat files are located in *install_dir*\cms\RKDSCATL
- .atr files are located in *install_dir*\cms\ATTRLIB

Copy the files to the following directories on the remote computer:

Table 54. Locations of CAT and ATR files for the monitoring server

Remote computer on:	File type	Directory
Windows	.cat	*install_dir*\cms\RKDSCATL
	.atr	*install_dir*\cms\ATTRLIB
Linux or UNIX	.cat	*install_dir*/tables/cicatrsq/RKDSCATL
	.atr	*install_dir*/tables/cicatrsq/ATTRLIB

where *install_dir* specifies the IBM Tivoli Monitoring installation directory. The IBM Tivoli Monitoring installation directory is represented by the **%CANDLE_HOME%** (Windows) or **$CANDLEHOME** (Linux and UNIX) environment variable. The default installation directory on Windows is `\IBM\ITM`. The default installation directory on Linux and UNIX is `/opt/IBM/ITM`.

Notes:

1. If you **export** the CANDLEHOME environment variable to your current session, many of the installation and configuration commands do not require that CANDLEHOME be passed to them (usually via the **-h** CLI parameter).
2. If you are adding support to a monitoring server on z/OS, you can use the FTP utility provided with Manage Tivoli Enterprise Monitoring Services. While checking existing files on the z/OS machine, the tool extracts the version line from the content of all files found. In a CAT file, the version line begins with the character **@**. In an ATR file, the version line begins with the comment marks **//**. The version numbers and timestamps that are obtained are used for version comparison. The following rules apply:
   - Files selected for FTP transfer that have a version higher than the corresponding file on z/OS, or have no corresponding file on z/OS, are transferred.
   - Files selected for FTP transfer that have a version lower than the corresponding file on z/OS are not transferred. You are alerted that some files cannot be transferred due to possible backleveling, and those files are excluded from the list of files to be transferred.
   - Files selected for FTP transfer for which it was impossible to determine a version are not transferred. This can be caused by, for example, file access error, or version text not found or incorrect. You are alerted that some files cannot be transferred due to the encountered errors, and those files are excluded from the list of files to be transferred.
   - If all files selected for FTP transfer are excluded from the transfer due to possible backleveling or reading version errors, the transfer is cancelled.
   - If some files selected for FTP transfer are excluded from the transfer due to possible backleveling or reading version errors, you are alerted that some files cannot be transferred and those files are excluded from the list of files to be transferred. The remaining files are transferred after confirmation.

   For more information, see *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.
3. If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```

Adding application support (SQL files) to a nonlocal hub: If you are adding application support to a nonlocal hub monitoring server, you must add the SQL files used by the EIB, in addition to the catalog and attribute files. Use the following procedure to seed the hub with SQL files using the Manage Tivoli Enterprise Monitoring Services utility on your local Windows system:

1. Ensure that the hub monitoring server is started.

   Note: If you are running in a Hot Standby environment, shut down your Hot Standby (that is, mirror) monitoring server before completing this procedure. You may restart the Hot Standby monitoring server only after you have seeded the hub server.
2. In the Manage Tivoli Enterprise Monitoring Services window, select **Actions** → **Advanced** → **Add TEMS application support**.
3. On the Add Application Support to the TEMS window, Select **On a different computer** and click **OK**.
4. When you are prompted to ensure that the hub monitoring server is configured and running, click **OK**.
5. On the Non-Resident TEMS Connection window, provide the hub monitoring server TEMS name (node ID) and select the communication protocol to use in sending the application support SQL files to the hub monitoring server.
6. On the next window, provide any values required by the communication protocol.

   For example, if the protocol is IP.PIPE, you are prompted for the fully qualified TCP/IP host name and port number of the z/OS system where the monitoring server is installed. See for the values you recorded during installation planning.
7. On the Select the Application Support to Add to the TEMS window, select the products for which you want to add application support or click Select All to choose all available products. Click **OK**.

   The SQL application support files are added to the hub monitoring server. This might take several minutes.
8. The Application Support Addition Complete window shown in Figure 66 on page 275 gives you information about the status and location of the application support SQL files. Click Save As if you want to save the information in a text file. Click Close to close the window.

   If the Application Support Addition Complete window is not displayed after 20 minutes, look in the `IBM\ITM\CNPS\Logs\seedk`*`pp`*`.log` files (where *pp* is the two-character code for each monitoring agent) for diagnostic messages that help you determine the cause of the problem.
9. If the monitoring server is not already stopped, stop it.
10. Restart the monitoring server.

Configuring application support on a nonlocal monitoring server from a Linux or UNIX system

This section describes how you can install and enable application support on a nonlocal monitoring server from your local Linux or UNIX system. The nonlocal monitoring server can be a hub or remote monitoring server installed on a Windows, Linux, or UNIX system.

Before you begin:

- As a prerequisite, you must install a monitoring server, a portal server or a monitoring agent on the local Linux or UNIX computer. This step is necessary to make the Manage Tivoli Enterprise Monitoring Services available on the local computer. If you do not have a monitoring server, you must install the support files on the local computer using the procedure described in “Installing application support files on a computer with no monitoring server” on page 289.
- Determine which monitoring agents you want to support on the nonlocal monitoring server. Ensure that the application support files (.cat, .atr, and .sql files) for those agents are available on the local monitoring server:
  - Application support files are located in the following directories on a Linux or UNIX monitoring server:

Table 55. Locations of application support files on a Linux or UNIX monitoring server

File type	Directory
.cat	*install_dir*/tables/cicatrsq/RKDSCATL
.atr	*install_dir*/tables/cicatrsq/ATTRLIB
.sql	*install_dir*/tables/cicatrsq/SQLLIB

- The file names of the application support files have the following format:

  ```
  kpc.ext
  ```

  where *pc* is the product code for the agent and *ext* is the file extension.

  For example, kud.sql is the SQL support file for the DB2 for Linux, UNIX, and Windows monitoring agent. See Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815 for a list of product codes.
- If you cannot find application support files for some agents for which you want to install application support, install the missing files on this computer.
  - To install missing support files for base monitoring agents, follow the installation steps described in “Installing the monitoring server” on page 213.
  - To install missing files for nonbase monitoring agents, follow the installation steps described in “Linux or UNIX: Installing application support on a monitoring server” on page 275.
  - If no monitoring server is installed on this computer, use the procedure in “Installing application support files on a computer with no monitoring server” on page 289.

- Gather the following information about the monitoring server on the remote computer:
  - The host name or IP address
  - The protocol and port number that was specified when the monitoring server was configured

    The monitoring server on the remote computer must be configured to use the IP.UDP, IP.PIPE, or IP.SPIPE communications protocol. This procedure does not support a monitoring server that was configured to use SNA.
- Verify that the monitoring server on the remote computer is running.
- Verify that the hub monitoring server to which this remote server sends its data is running.
- In these instructions, *install_dir* specifies the IBM Tivoli Monitoring installation directory. You can enter either $CANDLEHOME or the name of the directory. The default installation directory on Linux and UNIX is /opt/IBM/ITM.
- If you are running in a Hot Standby environment, shut down your Hot Standby (that is, mirror) monitoring server before completing this procedure. You may restart the Hot Standby monitoring server only after you have seeded the hub server.

Copying the CAT and ATR files to the nonlocal monitoring server: Copy the .cat and .atr files for the agents you want to support from the local Linux or UNIX monitoring server to the nonlocal monitoring server. If you use FTP, copy the files in ASCII format. The .cat and .atr files are located in the following directories on the local monitoring server:

- CAT files are located in *install_dir*/tables/cicatrsq/RKDSCATL
- ATR files are located in *install_dir*/tables/cicatrsq/ATTRLIB

Copy the files to the directory shown in Table 56 on the remote computer:

Table 56. Locations of CAT and ATR files for the monitoring server

Remote computer on:	File type	Directory
Windows	.cat	*install_dir*\cms\RKDSCATL
	.atr	*install_dir*\cms\ATTRLIB

Table 56. Locations of CAT and ATR files for the monitoring server (continued)

Remote computer on:	File type	Directory
Linux or UNIX	.cat	*install_dir*/tables/cicatrsq/RKDSCATL
	.atr	*install_dir*/tables/cicatrsq/ATTRLIB

where *install_dir* specifies the IBM Tivoli Monitoring installation directory. The IBM Tivoli Monitoring installation directory is represented by the %CANDLE_HOME% (Windows) or $CANDLEHOME (Linux and UNIX) environment variable. The default installation directory on Windows is \IBM\ITM. The default installation directory on Linux and UNIX is /opt/IBM/ITM.

Notes:

1. If you are adding support to a monitoring server on z/OS, you can use the FTP utility provided with Manage Tivoli Enterprise Monitoring Services. While checking existing files on the z/OS machine, the tool extracts the version line from the content of all files found. In a CAT file, the version line begins with the character **@**. In an ATR file, the version line begins with the comment marks **//**. The version numbers and timestamps that are obtained are used for version comparison. The following rules apply:
   - Files selected for FTP transfer that have a version higher than the corresponding file on z/OS, or have no corresponding file on z/OS, are transferred.
   - Files selected for FTP transfer that have a version lower than the corresponding file on z/OS are not transferred. You are alerted that some files cannot be transferred due to possible backleveling, and those files are excluded from the list of files to be transferred.
   - Files selected for FTP transfer for which it was impossible to determine a version are not transferred. This can be caused by, for example, file access error, or version text not found or incorrect. You are alerted that some files cannot be transferred due to the encountered errors, and those files are excluded from the list of files to be transferred.
   - If all files selected for FTP transfer are excluded from the transfer due to possible backleveling or reading version errors, the transfer is cancelled.
   - If some files selected for FTP transfer are excluded from the transfer due to possible backleveling or reading version errors, you are alerted that some files cannot be transferred and those files are excluded from the list of files to be transferred. The remaining files are transferred after confirmation.

   For more information, see *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.
2. If you specify an incorrect directory name, you will receive the following error:

   ```
   The IBM Tivoli Monitoring installation directory cannot exceed 80 characters
   or contain non-ASCII, special  or double-byte characters.
   The directory name can contain only these characters:
   "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./".
   ```

If the nonlocal monitoring server to which you are adding application support is a hub, you must also install the SQL files used by the EIB. See “Adding application support (SQL files) to a nonlocal hub.”

Adding application support (SQL files) to a nonlocal hub: To add application support SQL files to a hub monitoring server on a nonlocal system, complete this procedure.

1. Enable the GUI interface.

   Your Linux or UNIX environment might already have a GUI interface enabled. Otherwise, perform the following tasks to enable it:

   a. Enable X11.

   b. Make sure you have access to a native X-term monitor or an X-Window emulator.

   c. If using an X-Window emulator, enable X11 access to the X-Window server (command example: `xhost +`).

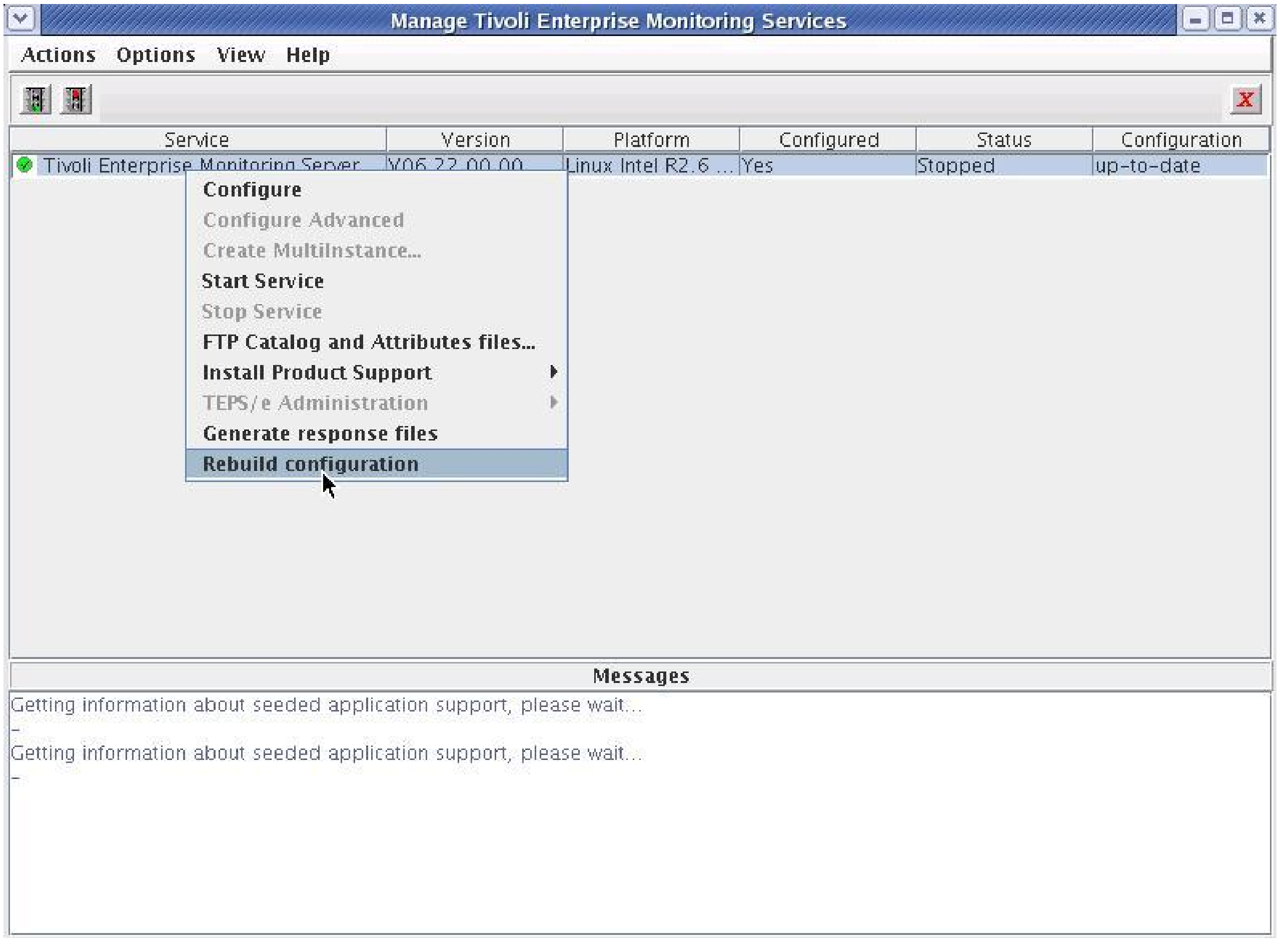
d. If using an X-Window emulator, set the display environment variable to point to the X-Window server:

```
export DISPLAY=pc_ip_address:0
```

2. Ensure that the hub monitoring server is running.
3. To start Manage Tivoli Enterprise Monitoring Services, go to the $CANDLEHOME bin directory (example: `/opt/IBM/ITM/bin`) and run this command:

   ```
   ./itmcmd manage &
   ```

   A GUI window opens for Manage Tivoli Enterprise Monitoring Services.
4. Select **Actions → Install Product Support**.



Service	Version	Platform	Configured	Status	Configuration
Tivoli Enterprise Monitoring Server	V06.22.00.00	Linux Intel R2.6 ...	Yes	Stopped	up-to-date

```
Getting information about seeded application support, please wait...
-
Getting information about seeded application support, please wait...
-
```

Figure 68. Manage Tivoli Enterprise Monitoring Services Install Product Support window

5. On the Add Application Support to the TEMS window, select **On a different computer** and click **OK**.
6. When you are prompted to ensure that the hub monitoring server is configured and running, click **OK**.
7. On the Non-Resident TEMS Connection window, provide the hub monitoring server name (node ID) and select the communication protocol to use in sending the application support SQL files to the hub monitoring server.
8. Select the appropriate communications protocol and click OK.
9. On the next window, supply any values required by the selected communication protocol and click **OK**.
10. On the Install Product Support window, select the monitoring agents for which you want to add application support to the hub monitoring server, and click Install.
11. In Manage Tivoli Enterprise Monitoring Services, look for this message:

```
Remote seeding complete!
```

12. Stop and restart the hub monitoring server.

Installing application support files on a computer with no monitoring server: You can install application support files for a monitoring server on a UNIX or Linux computer where no monitoring server is installed and then use the files to configure support on a nonlocal monitoring server. Use the following procedure to install application support files on computer on which no monitoring server is installed for use on a monitoring server on another computer:

1. Run the following command from the installation media, either the IBM Tivoli Monitoring base media, the agent product CD for distributed agent products or the data files CD for z/OS agent products:

   ```
   ./install.sh
   ```

2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`) or enter the full path to the installation directory you used.

   The following prompt is displayed:

   ```
   Select one of the following:

   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   3) Install TEMS support for remote seeding
   4) Exit install.

   Please enter a valid number:
   ```

3. Type `3` and press Enter.

   The software license agreement is displayed.

4. Read through the software license agreement, then type `1` and press Enter to accept the agreement.

   The installer presents a list of currently installed products, the operating systems for which product packages are available, and component support categories.

5. Type the 32-character encryption key to use and press Enter, or press enter to use the default key. This key must be the same key as that used during the installation of the portal server to which the client will connect.

   The installer presents a list of products for which support packages are available.

6. Enter the numbers of the monitoring agents that you want to install or enter the number that corresponds to `All of the above`. Enter more than one number on the same line separated by spaces or commas (,).

   A list of the monitoring agents to be installed is displayed.

7. Confirm the selection by pressing Enter.

   The installer asks if you want to install additional products or support packages.

8. Type `2` and press Enter.

   The support packages are installed. When installation is complete, you will see the message

   ```
   ... finished postprocessing.
   Installation step complete.
   ```

Installing domain definitions for Tivoli Performance Analyzer

Install the domain definitions to enable performance analytics for a broader set of systems including DB2, Oracle, System p® Series, ITCAM for RT, and VMware. You can install domain definitions in GUI mode or console mode. The domain definitions for the Tivoli Performance Analyzer are found in the CI62LML package, available from the Passport Advantage Web site.

Notes:

1. Tivoli Performance Analyzer is a special agent that works on data collected by other agents. Tivoli Performance Analyzer is shipped with separate packages called domains. These domains bundle

together a set of artifacts that are used to process and present warehoused data collected by other agents. For example, there is Performance Analyzer DB2 domain, Performance Analyzer OS Domain, Performance Analyzer Oracle Domain, and so on.

2. Domain definitions for Tivoli Performance Analyzer must be added in order to complete the product configuration. You must install the domains on the server that has the hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Performance Analyzer installed. In a distributed environment, you must perform domain installation on each server where at least one of these components is installed: hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Performance Analyzer.
3. When run on a host, the domain installer presents options to add domain support for only those components that it finds installed on the server locally, for example Tivoli Enterprise Portal Server, Tivoli Enterprise Portal clients - Browser and Desktop, Tivoli Enterprise Monitoring Server, or Tivoli Performance Analyzer itself. Choose the installed component as appropriate.
4. Due to new domain product codes introduced with Tivoli Monitoring V6.2.3, any historical data collection settings for the Tivoli Performance Analyzer are not maintained when upgrading from V6.2.2 to V6.2.3. The domain installer image includes scripts that create historical data collection settings. You can use these scripts to quickly recreate any historical data collection settings that were lost.

The logs for the domain installation can be found at:

- For Windows: `<ITM DIR>\logs\itpa_domain.log`
- For UNIX: `<ITM DIR>/logs/itpa_domain.log`

Installing domain definitions in GUI mode

Use the GUI to install the domain definitions automatically and check if all the required programs for configuration are in place.

About this task

To install the required domain definitions:

Procedure

1. From the product CD launch `setup.exe`. The installer window opens.
2. Click **Next** on the installer welcome page.
3. In the **License Agreement** window, read the agreement, accept the terms and click **Next**.
4. Select the components for which you want to add domain definitions and click **Next**.

   Note: If you do not mark all the components, there may be a lack of coherence between the selected ones.

   If you select **Tivoli Performance Analyzer Domain support**, only the **active** or **inactive** state of tasks for the selected domain is preserved after reinstallation in the upgrade scenario. All other attributes are set back to **Default**.

   Domains require Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, Tivoli Enterprise Portal Desktop, Tivoli Enterprise Portal Browser and Tivoli Performance Analyzer support files to be installed.
5. Review the settings and click **Next**.
6. Select the configuration type and click **Next**.
7. Enter the host name of the machine where the Tivoli Enterprise Portal Server resides and click **Next**.
8. Choose where to locate the application support for the Tivoli Enterprise Monitoring Server and click **OK**.
9. In the next window, select the application support you want to add to the Tivoli Enterprise Monitoring Server. Choose the component and add or update the situations and click **OK**.

10. In order to finish the installation, check the box to see the installation documentation and click **Finish**.

Results

You have now automatically broadened the Tivoli Performance Analyzer domain support.

What to do next

Important: Restart Tivoli Monitoring components after the installation. The tool shows which components need to be restarted.

Installing the domain definitions in console mode

Use the console mode to install the domain definitions in an interactive manner.

Before you begin

Tivoli Performance Analyzer Agent and its supports must be installed before installing any domain definitions.

About this task

Use the following steps to install the required domain definitions on a UNIX computer:

Procedure

1. In the directory where you extracted the installation files, run the following command:
   ```
   ./install.sh
   ```
2. When prompted for the IBM Tivoli Monitoring home directory, press **Enter** to accept the default (`/opt/IBM/ITM`). If you want to use a different installation directory, type the full path to that directory and press **Enter.**
3. If the directory you specified does not exist, you are asked whether to create it. Type `1` to create this directory.
4. Type `1` in order to choose **Install products to the local host** from the prompt and start installation.
5. The user license agreement is displayed. Press**Enter** to read through the agreement.
6. Type `1` to accept the agreement and press **Enter**.
7. Select a locally installed component for which you want to add domain definitions.

   Note:

   a. If you select **Tivoli Performance Analyzer Domain support**, only the **active** or **inactive** state of tasks for the selected domain is preserved after reinstallation in the upgrade scenario. All other attributes are set back to **Default**.
   b. Domains require Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, Tivoli Enterprise Portal Desktop, Tivoli Enterprise Portal Browser and Tivoli Performance Analyzer support files to be installed.
8. To select more components for which you want to add domain definitions, type **1**, which takes you to step 5. To finish the installation, select **2**.

Results

You have now installed the required domain definitions and exited the console mode.

Attention: In order to display workspaces properly, reconfigure Tivoli Enterprise Portal Server and Tivoli Enterprise Portal Desktop Client. You also must restart Tivoli Performance Analyzer to start calculations for newly installed analytical tasks.

Installing language packs

Language support for products for which application support is provided with IBM Tivoli Monitoring appears on the following media:

- *IBM Tivoli Monitoring V6.2.3 Language Support 1 of 3 DVD*: French, German, Italian, Portuguese Brazilian, Spanish.
- *IBM Tivoli Monitoring V6.2.3 Language Support 2 of 3 DVD*: Japanese, Korean, Simplified Chinese, Traditional Chinese.
- *IBM Tivoli Monitoring V6.2.3 Language Support 3 of 3 DVD*: Czech, Hungarian, Polish, Russian
- *IBM Tivoli Monitoring V6.2.3 DM Upgrade Toolkit Language Support CD*
- *IBM Tivoli Monitoring V6.2.3 ITM 5.1.2 Migration Toolkit Language Support CD*
- *IBM Tivoli Monitoring V6.2.3 Agent Builder Toolkit Language Support CD*
- *IBM Tivoli Monitoring Agents for System P Language Support CD*
- Agent product installation CDs

The IBM Tivoli Monitoring V6.2.3 Language Support DVDs contain the national language versions of the help and presentation files for the components and agents listed in Table 57.

Table 57. Language support included on IBM Tivoli Monitoring V6.2.3 Language Support DVDs

Product code	Component or product name
HD	Warehouse Proxy Agent
SY	Warehouse Summarization and Pruning Agent
NT	Monitoring Agent for Windows OS
LZ	Monitoring Agent for Linux OS
UX	Monitoring Agent for UNIX OS
UL	Monitoring Agent for UNIX Logs
UM	IBM Tivoli Universal Agent
TM	Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint
P5	IBM Tivoli Monitoring for AIX Base Agent
PX	Premium Monitoring Agent for AIX
PH	Base Monitoring Agent for HMC
PK	Base Monitoring Agent for CEC
PV	IBM Tivoli Monitoring for VIOS Base Agent
PA	Tivoli Performance Analyzer
VA	Premium Monitoring Agent for VIOS

Language support for the products found on the IBM Tivoli Monitoring V6.2.3 Tools DVD is available on the following CDs:

- *IBM Tivoli Monitoring V6.2.3: Language Support for Distributed Monitoring Toolkit CD*
- *IBM Tivoli Monitoring V6.2.3: Language Support for Migration Toolkit CD*
- *IBM Tivoli Monitoring V6.2.3: Language Support for Agent Builder CD*

Language support for all other distributed agents, including those agents for which application support is included on the Base DVD, is included with the installation media for the individual agent. For the OMEGAMON XE monitoring agents on z/OS, the installation media for the agents are mainframe tapes, which don't include the language packs; as with the distributed agents, language support is provided on a separate CD or DVD.

Note: Tivoli Monitoring V6.2.3 features self-describing agent capability which integrates the installation of an agent with the dispersal and installation of associated product support files throughout your IBM Tivoli Monitoring infrastructure. Language Pack installation is not supported for products installed using self-describing agent capability. For these products you must install language packs manually, using the steps described in this section.

Install the language packs on any system where you have installed the Tivoli Enterprise Portal or where you have installed a desktop client. (If you download and run a desktop client using Web Start, you do not need to install the language packs on the local system. They are downloaded from the portal server.) *Before you can install a language pack, you must install the component in English*.

Before installing a language pack, first install the component in English. Also ensure that Java Runtime Environment version 1.5 or above is installed and set in the system path. Perform the following steps to install a language pack on any system where you have installed either the Tivoli Enterprise Portal Server or the Tivoli Enterprise Portal desktop client:

1. In the directory where you extracted the language pack installation image, launch the installation program as follows:
   - On Windows, double-click the `lpinstaller.bat` file.
   - On Linux and UNIX, run the following command:

     ```
     ./lpinstaller.sh -c install_dir
     ```

     where:

     `install_dir`
     is the directory where you installed IBM Tivoli Monitoring (usually `/opt/IBM/ITM`).

     To perform a console installation on Linux or UNIX (instead of a GUI installation), add the `-i` console parameter to the above command.
2. Select the language you want installed, and click OK.
3. On the Introduction panel, click Next.
4. On the Select Action Set panel, click Add/Update, and click Next.
5. Select the folder in which the Language Support package files (`win*.jar` and `unix*.jar`) are located, and click Next. The default folder is the directory where the installer is launched.
6. Select the languages that you want to install, and click Next.

   For multiple selections, hold down the Ctrl key.
7. Review the installation summary, and, if correct, click Next.

   The installation's progress is displayed.
8. On the Post Install Message panel, click Next.
9. Click Done once the installation is complete.
10. Reconfigure and restart the Tivoli Enterprise Portal Server and the Eclipse Help Server. See below.

After installing a Tivoli Monitoring V6.2.3 Language Pack, reconfigure the portal server and the desktop client using either the Manage Tivoli Enterprise Monitoring Services utility or the **itmcmd config** command.

Use one of the following methods to reconfigure the affected components:

- Launch Manage Tivoli Enterprise Monitoring Services, right-click the affected component, and select **Reconfigure**. (See "Starting Manage Tivoli Enterprise Monitoring Services" on page 363.)
- Change to the *`install_dir`*`/bin` directory, and enter the following commands:

  ```
  ./itmcmd config -A cq
  ./itmcmd config -A cj
  ```

Accept the default values, which reflect the decisions made when the component was installed or last configured. For instructions on specifying your users' language environment, see the *IBM Tivoli Monitoring: Administrator's Guide*.

After you have reconfigured these components, you need to stop and restart these components:
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal desktop or browser client

For SuSE Linux Enterprise Server (SLES) 10 computers only: On the SLES 10 platform, the Tivoli Enterprise Portal displays corrupted text resources in the Japanese locale. Download the Kochi fonts contained in the kochi-substitute-20030809.tar package from the following Web site: http://sourceforge.jp/projects/efont/files/.

The downloaded tar file includes the truetype fonts (ttf files), which need to be installed on your system. Complete the following steps to install the files:
1. Extract the tar file.
2. Copy the font files (ttf) to X11 font path (for example, `/usr/X11R6/lib/X11/fonts/truetype`).
3. Run **SuSEconfig -module fonts**.

See the following Web site for detailed instructions for installing the additional fonts to SuSE Linux: http://www.suse.de/~mfabian/suse-cjk/installing-fonts.html .

Silent installation of language packs

You can install the language pack using a silent installation method:
1. Copy and paste the following response file template as ITM_LP_slient.txt file.
2. Change the following parameter setting: NLS_PACKAGE_FOLDER; PACKAGE_NAME_LIST; LANG_SELECTION_LIST.
3. Run the following command to silently install the language pack:
   - For Windows systems:

     ```
     lpinstaller.exe -f <path_to_response_file>
     ```
   - For UNIX or Linux systems:

     ```
     lpinstaller.sh -c <candle_home> -f <path_to_response_file>
     ```

   Where `<candle_home>` is the IBM Tivoli Monitoring base directory.

Response file template for language pack silent installation:

```
#     IBM Tivoli Monitoring Language Pack Silent Installation Operation
#
#This is a sample response file for silent installation of the IBM Tivoli Monitoring
#Language Pack Installer. The example used in this file is the IBM Tivoli Monitoring
#language pack. This file was built by the Replay feature of InstallAnywhere.
#The variables in this file were set by Panels, Consoles, or Custom Code.
#-----------------------------------------------------------------------------------------

#-----------------------------------------------------------------------------------------
#To successfully complete a silent installation of the IBM Tivoli Monitoring
Language Pack,complete the following steps:
#1.Copy the ITM_LP_silent.txt to the directory where lpinstaller.bat
#or lpinstaller.sh is located
(IBM Tivoli Monitoring  Language Pack build location).
#
#2.Modify the response file so that it is customized correctly and completely
#for your site.
#Complete all steps listed below in the response file.
#
#3.After customizing the response file, invoke the silent installation using the
#following command:
```

```
#For Windows:
#    lpinstaller.exe -f <path_to_response_file>
#For UNIX and Linux:
#    lpinstaller.sh -c <candle_home> -f <path_to_response_file>
#Note:<candle_home> is the ITM base directory

#-----------------------------------------------------------------------------------------
#Force silent install mode.
#-----------------------------------------------------------------------------------------
INSTALLER_UI=silent

#-----------------------------------------------------------------------------------------
#Run add/update actions.
#-----------------------------------------------------------------------------------------
CHOSEN_INSTALL_SET=ADDUPD_SET

#-----------------------------------------------------------------------------------------
#Choose a folder.
#Modify the following path to the language package driver location:
#Windows path:
# NLS_PACKAGE_FOLDER=C:\\build_machine\\LP\\ITM622_0723\\tvtbuild\\200909011756
#UNIX path:
NLS_PACKAGE_FOLDER=//windows

#-----------------------------------------------------------------------------------------
#List the packages to process; both variables are required.
#Each variable requires that full paths are specified.
#Separate multiple entries with a semicolon (;).
#For Windows:
#          Use the backslash-backslash(\\) as a file separator.
#For Unix and Linux:
#          Use the slash-slash (//) as a file separator.
#-----------------------------------------------------------------------------------------
#Windows path:
#PACKAGE_NAME_LIST=C:\\build_machine\\LP\\ITM622_0723\\tvtbuild\\200909011756\\winDBCS.jar
#UNIX path:
PACKAGE_NAME_LIST=//windows//unixSBCS.jar;//windows//unixDBCS.jar;//windows//unixCER.jar

#-----------------------------------------------------------------------------------------
#List the languages to process.
#Separate multiple entries with semicolons.
#-----------------------------------------------------------------------------------------
LANG_SELECTION_LIST=de;es;it;fr;ja;ko;pt_BR;zh_CN;zh_TW;cs;hu;pl;ru;th
```

Uninstalling a language pack

To uninstall a language pack:

1. In the directory where you extracted the language pack installation image, launch the installation program as follows:
   - On Windows, double-click the `lpinstaller.bat` file.
   - On Linux and UNIX, run the following command:

     ```
     ./lpinstaller.sh -c install_dir
     ```

     where:

     `install_dir`
     is the directory where you installed IBM Tivoli Monitoring (usually `/opt/IBM/ITM`).

     To perform a console installation on Linux or UNIX (instead of a GUI installation), add the `-i console` parameter to the above command.
2. Select the language to be uninstalled, and click OK.
3. On the Introduction panel, click Next.

4. On the Select Action Set panel, click Remove, and click Next.
5. Select the languages that you want to uninstall, and click Next.

   For multiple selections, hold down the Ctrl key.
6. Review the installation summary, and, if correct, click Next.
7. On the Post Install Message panel, click Next.
8. Click Done once the uninstallation is complete.
9. Reconfigure and restart the Tivoli Enterprise Portal Server and the Eclipse Help Server, as described above.

Configuring clients, browsers, and JREs

The configuration required for Tivoli Enterprise Portal clients depends upon the client deployment mode being used, the browser being used, the Java runtime environment (JRE) being used, and the operating system the client is being used on.

The following sections discuss the configuration for clients by mode of deployment:
- “Desktop clients” on page 297
- “Browser clients” on page 297
- “Java Web Start clients” on page 303

The version of IBM JRE required by Tivoli Enterprise Portal clients is installed with Tivoli Management Services components. If you want to run a client on a machine where no other components are installed, you can download the IBM JRE installer from the Tivoli Enterprise Portal Server (see “Installing the IBM JRE” on page 321). The IBM JRE must be installed as the system JVM.

The packaging, installation, and servicing of the Sun JRE is not provided by IBM. The Sun JRE at version 1.6.0.xx must already be installed on the machines where the Tivoli Enterprise Portal client will run. The Sun JRE can be downloaded from the following Web site: `http://www.java.com/getjava`. For help installing the Sun JRE and enabling and configuring its Java plug-in on Linux, visit the following Web site: http://www.java.com/en/download/help/5000010500.xml.

Support for the Sun JRE is a feature of the Tivoli Enterprise Portal client only; installation and use of the IBM JRE is still required for the Tivoli Enterprise Portal Server and some other Tivoli Management Services components.

As of IBM Tivoli Monitoring 6.2.2, the installer no longer modifies the system JRE. Instead it installs a local copy of Java, one that is private to Tivoli Monitoring. This applies to all Java-dependent Tivoli Monitoring components, including those at a pre-6.2.2 level.

This embedded JRE is installed in the `%CANDLE_HOME%\java` directory. The system Java is untouched by the IBM Tivoli Monitoring installer; you can remove it yourself if you desire.

The exceptions are the Tivoli Enterprise Portal Server and the eWAS server, which use the embedded JVM delivered as part of the eWAS component. Also, the browser client and Java Web Start still use the system JRE. For your convenience, the system JRE installation image is still distributed as part of the browser client package.

The desktop client uses the new, embedded JVM.

The IBM Tivoli Monitoring installer does not install an embedded JVM if none of the components selected for installation has a Java dependency.

Desktop clients

You do not need to configure the desktop client if:

- You want the client to use the IBM JRE, or
- The SUN JRE is the only JRE installed on the computer

You must configure the client start-up script with the location of the JRE if:

- Both IBM and Sun JREs are installed on the computer and you want to use the Sun JRE, or
- You have multiple versions of the Sun JRE installed and you want to specify a particular version

On Windows computers, add a user-level environment variable named TEP_JAVA_HOME whose value is the fully qualified directory location of the Sun JRE you want to use. For example, `TEP_JAVA_HOME=C:\Program Files\Java\jre1.6.0_xx\bin\`. On UNIX and Linux computers, define the same environment variable with the fully qualified location of the Sun JRE you want to use for the Tivoli Enterprise Portal desktop client.

Browser clients

If the Firefox browser is used with the Tivoli Enterprise Portal browser client, IBM JRE 1.5 must be used. If you are using Internet Explorer with the browser client, either IBM JRE 1.5, IBM JRE 1.6, or Sun JRE 1.6 can be used.

Configuration for browser clients involves the following steps:

- "Registering the Java plug-in" on page 299

  When the browser client connects to the Tivoli Enterprise Portal Server, it downloads a Java applet. Java applets that run in a browser environment use Java plug-in technology. The applet plug-in must be registered with the browser to establish and initialize an appropriate Java runtime environment for the applet to execute in.

  On Linux and UNIX, the plug-in must be manually registered with the browsers; on Windows, the Java installer automatically registers the associated Java plug-in with both Internet Explorer and Firefox. After installation, a plug-in can be de-registered from a given browser selectively using the Java control panel (see "Removing the Java plug-in on Windows" on page 301).
- "Specifying runtime parameters for the plug-in" on page 300

  Several runtime parameters should be specified before the browser client is started to allocate enough memory for the Tivoli Enterprise Portal applet and decrease the startup time.

If you are using Internet Explorer under Windows, you must complete an additional step to use the Sun JRE:

- "Identifying the version of the Sun JRE the client should use" on page 301

Conflicts in plug-in registration can sometimes occur if both IBM and Sun JREs are installed on Windows. You can resolve such problems by de-registering the plug-in in the browsers:

- "Removing the Java plug-in on Windows" on page 301

Firefox tips

- Unlike Internet Explorer, Firefox does not automatically load the current URL into a new window. If you want to invoke multiple copies of the Tivoli Enterprise Portal browser client, copy the full URL from the first window to the second window. Do not load the initial `http://`*`hostname`*`:1920///cnp/kdh/lib/cnp.html` URL, because that will not work.
- To use the browser's tab support when opening workspaces in the browser client, you may need to configure the browser to force links that otherwise open in new windows to open instead in new tabs. See your browser's instructions for customizing tab preferences. This is true for Internet Explorer version 7 (and subsequent) as well as Mozilla Firefox.

  Reuse of existing tabs is supported only with the Firefox browser. This means if you have a workspace open in an unfocused tab and from your currently focused tab you select the same workspace to open in a new tab, the unfocused tab is refocused, and the workspace is reloaded there. To allow this support:

  1. Enter `about:config` in the address bar, and press Enter.
  2. Find the preference *signed.applets.codebase_principal_support*, and change the value to **true**. (Or double-click the preference name to toggle its value.)
  3. The first time you attempt to reuse an existing tab, the menu shown in Figure 69 on page 299 might display. If so, add a checkmark to the box **Remember this decision**, and press Allow.

  If you do not allow this tab reuse, a new tab is opened each time you open a new workspace.

  Tip: To open a workspace in a new tab using either Firefox or Internet Explorer, press Shift+Ctrl while selecting the workspace. This always opens a new tab whether or not your browser is set to reuse existing tabs.
- If the Firefox process dies unexpectedly on Linux, you may need to also kill the Java process that is spawned when the Java plug-in is started. If you do not kill the Java process, subsequent startups of the Tivoli Enterprise Portal client in a new Firefox browser may be very slow.

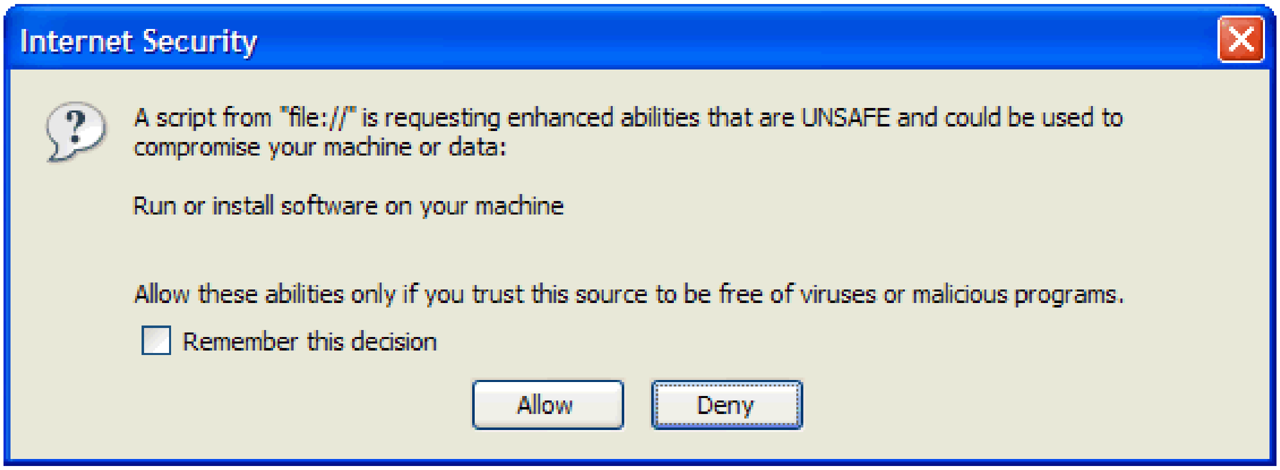
  If you are running the browser client with the IBM JRE, issue the following command to find the errant Java process:

  ```
  ps –aef | grep plugin
  ```

  If you find a process similar to the following, kill it:

  ```
   java -Dmozilla.workaround=true
  -Xbootclasspath/a:/opt/candlehome/7277c/JRE/li6263/lib/javaplugin.jar
  :/opt/candlehome/7277c/JRE/li6263/lib/deploy.jar -Djavaplugin.lib=
  /opt/candlehome/7277c/JRE/li6263/bin/libjavaplugin_jni.so
  -Djavaplugin.nodotversion=150 -Djavaplugin.version=1.5.0
  -DtrustProxy=true -Xverify:remote -Djava.class.path=/opt/candlehome/7277c
  /JRE/li6263/lib/applet sun.plugin.navig.motif.Plugin
  ```

  If running the browser client with the Sun JRE, the errant Java process is called `java_vm`.



A script from "file://" is requesting enhanced abilities that are UNSAFE and could be used to compromise your machine or data:

Run or install software on your machine

Allow these abilities only if you trust this source to be free of viruses or malicious programs.

☐ Remember this decision

Figure 69. Firefox Security Warning

Registering the Java plug-in

Under Linux, you register the Java plug-in by creating a symbolic link to the Java plug-in in the browser `plugins` directory. On Windows, you can use the Java control panel to ensure that the plug-in is associated with the appropriate browser.

To create and test the symbolic link to the plug-in on Linux, take the following steps:

1. Find the path to the Java plug-in file:

   ```
   cd jre_install_dir
   find -name "libjavaplugin_oji.so" -print
   ```

   Note: For the Sun JRE, you may find two plugin files. For example:

   ```
   ./plugin/i386/ns7/libjavaplugin_oji.so
   ./plugin/i386/ns7-gcc29/libjavaplugin_oji.so
   ```

   Use only `ns7-gcc29` if the browser was compiled with gcc2.9.
2. Change to the `plugins` subdirectory under the browser installation directory:

   ```
   cd browser_install_dir/plugins
   ```
3. Create a soft link to the Java plug-in file:

   ```
   ln -s jre_install_dir/java_plugin_file_path
   ```
4. Verify that the soft link has been set up correctly:
   a. Start the browser.
   b. Type `about:plugins` in the Location bar to confirm that the Java plug-in has been correctly installed.

      To show the full path to the plug-in instead of just the file name, type `about:config` in the address bar and press Enter. Find the preference `plugin.expose_full_path` and change the value to `true`. (Double-clicking the preference name toggles the setting.)

On Windows, to verify that the desired plug-in is registered, complete the following steps:

1. Launch the appropriate Control Panel from the Windows Control Panel folder (either IBM Control Panel for Java or Java Control Panel).
2. Select the **Advanced** tab.
3. Expand the branch labeled *APPLET* `tag support`.
4. Ensure that the appropriate browser option box is checked.

5. If you make any changes, press **Apply** and exit the panel.

Specifying runtime parameters for the plug-in

You specify the runtime parameters for the Java applet using the control panel for the appropriate JRE. Use the same user ID from which the browser will be launched to launch the control panel and specify these parameters, or the user-level `deployment.properties` file for the correct user ID will not be updated.

To specify the runtime parameters, take the following steps:

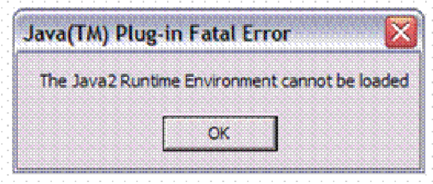
1. Launch the Java control panel:
   - On Windows, launch the IBM Control Panel for Java, or the Java Control Panel.
   - On Linux, Find the Java **ControlPanel** executable under your *jre_install_dir* and launch it. For example:

     ```
     /opt/IBM/ibm-java2-i386-50/jre/bin/ControlPanel
     ```
2. If you have multiple Java versions, verify that you have the correct control panel open by confirming the Java Runtime and that the JRE is in the correct path (for example, `c:\program files\IBM\Java50\jre\bin` for IBM Java on Windows). To verify, click on the **Java(TM)** tab and check the **Location** column for the JRE.
3. Set the Java Runtime Parameters:
   a. Click the **Java** tab.
   b. Click the Java Applet Runtime Settings **View** button.
   c. Click in the **Java Runtime Parameters** field and set the following parameters:
      - **IBM JRE**: `-Xms128m -Xmx256m -Xverify:none -Djava.protocol.handler.pkgs=sun.plugin.net.protocol`
      - **Sun JRE**: `-Xms128m -Xmx256m -Xverify:none`

        The `-Xms128m` specifies the starting size of the Java heap (128 MB) and `-Xmx256m` specifies the maximum size. The `-Xverify:none` parameter disables Java class verification, which can increase startup time.

        The `–Djava.protocol.handler.pkgs` option required *only for the IBM JRE on Linux* due to a problem with the plug-in not caching JAR files. If the parameter is left off, the Tivoli Enterprise Portal applet JAR files will not be cached, making subsequent start ups of the applet slow.
   d. Click **OK**.
4. Confirm that the Temporary Files settings are set to Unlimited:
   a. Click the **General** tab.
   b. Click **Settings**.
   c. Select **Unlimited** for the **Amount of disk space** to use.
   d. Click **OK**.
5. Clear the browser cache:
   a. In the **General** tab, click **Delete Files**.
   b. In the window that opens, select **Downloaded Applets** and click **OK**.

The Sun JRE does not always support the same maximum heap values as the IBM JRE. The true maximum is calculated based on the resources available on the particular machine being configured, and the algorithm that is involved is different between the two JREs. The symptom of a memory problem is that the applet fails to load in the browser, and you receive a message similar to the following:



Java(TM) Plug-in Fatal Error

The Java2 Runtime Environment cannot be loaded

Figure 70. Java memory error message

To resolve this problem, reduce the value of the maximum heap setting in the Java Control panel in 32m or 64m increments until the error goes away. For example, if you start with the recommended value of `-Xmx256m` try reducing it to `-Xmx224m` or `-Xmx192m`. Eventually you will reach a value that is appropriate for the particular machine involved.

Identifying the version of the Sun JRE the client should use

If you are using Internet Explorer for the Tivoli Enterprise Portal browser client, and you want to use the Sun JRE, you must identify to the Tivoli Enterprise Portal Server the version of the JRE you want the client to use.

To identify the version, update the file `jrelevel.js` found in the `\`*`itm_install_dir`*`\CNB` directory (Windows) or *`itm_install_dir`*`/`*`arch`*`/cw` (Linux/UNIX) on the computer where the portal server is installed. Assign a valid value to the following declared variable:

```
var jreLevel = "6.0"
```

The supported values are:

*****	The browser client will use the default JRE for the computer.
1.5.0	The browser client will use the IBM 1.5 JRE (the default after an upgrade to ITM 6.2.3 FP1 or later).
1.6.0	The browser client will use the IBM 1.6 JRE (the default for a pristine install).
6.0	The browser client will use the latest Sun 1.6.0_xx JRE installed.

Removing the Java plug-in on Windows

When either the IBM JRE or the Sun JRE is installed on Windows, the Java installer automatically registers the associated Java plug-in with both Internet Explorer and Firefox. Plug-in registration conflicts sometimes occur on Windows when both JREs are installed. The symptoms usually involve one of the following messages being displayed when you launch the Tivoli Enterprise Portal client:

```
"Java Plug-in detected JRE collision"
```

or

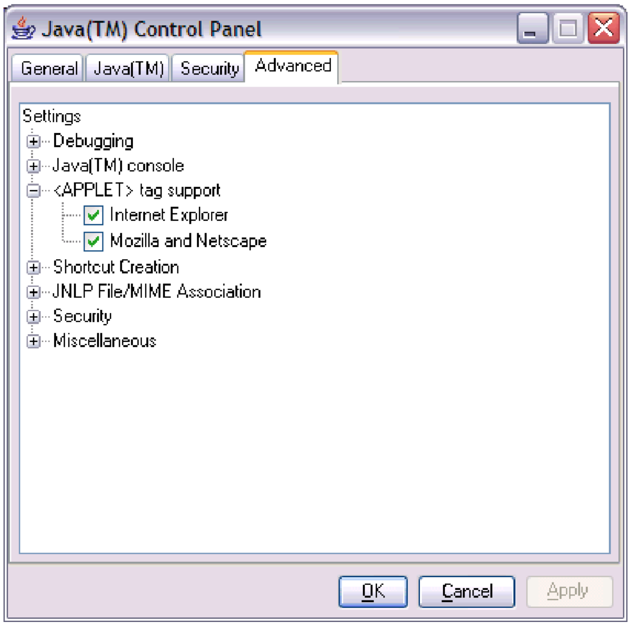
```
"Applet(s) in this HTML page requires a version of java different from the one
the browser is currently using.  In order to run the applet(s) in this HTML page,
a new browser session is required.
Press 'Yes" to start a new browser session.
```

The solution is to de-register the plug-in.

To de-register the Java plug-in take the following steps:

1. Launch the IBM Control Panel for Java or the Sun Java Control Panel from the Windows Control Panel folder.
2. Select the **Advanced** tab.
3. Expand the branch entitled *`APPLET`* `tag support`.

   There should be two entries, one for "Internet Explorer" and one for "Mozilla and Netscape". Normally, after installation of Java, both of these boxes will be checked, meaning that the associated Java plug-ins for those browsers have been registered. Figure 71 on page 302 shows the IBM Control Panel for Java; the Sun version is almost identical.



Settings
- Debugging
- Java(TM) console
- <APPLET> tag support
  - ☑ Internet Explorer
  - ☑ Mozilla and Netscape
- Shortcut Creation
- JNLP File/MIME Association
- Security
- Miscellaneous

Figure 71. Java Control Panel window

If you encounter problems loading Tivoli Enterprise Portal using Firefox, this is one of the first panels you should look at to ensure that the Java plug-in has indeed been registered. Often, a quick way to resolve any registration problems is to simply remove the check mark (if already on), press **Apply**, then check the box again, and again press **Apply**. This action will switch the option off and on so that the plug-in registration is reset and the correct Windows registry entries get re-established.

4. Remove the check mark from the box for the browser or browsers you want to unregister.

Support for Java 6 or higher with browser clients on Windows

With Sun Java versions 1.6.0_10 and higher or IBM Java 6 SR7, a new plug-in architecture was introduced and established as the default plug-in. This support enables an applet-caching feature called **legacy lifecycle** that, coupled with the use of the Java 6 JRE, significantly increases performance of the Tivoli Enterprise Portal browser client after it has been downloaded. Performance measurements using this configuration show that, with legacy lifecycle, performance of the browser client is virtually identical to that of the desktop and Java Web Start deployment modes.

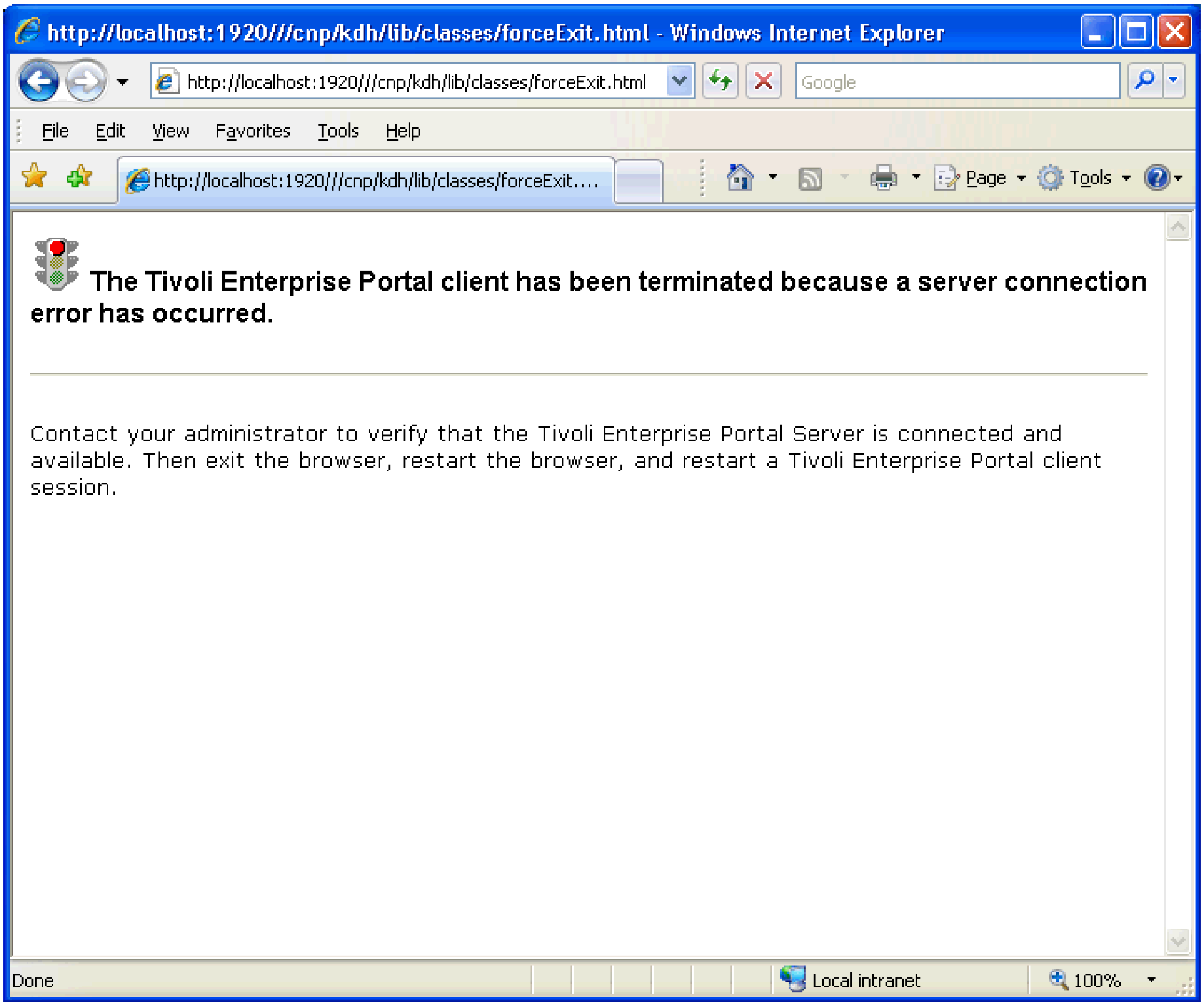
IBM Tivoli Monitoring V6.2.3 Fix Pack 1 and later browser clients automatically run with this new plug-in architecture if the Internet Explorer browser is being used. Since Java 1.6 cannot be used with the Firefox browser, this plug-in architecture is automatically disabled for Firefox browser users. In earlier versions of IBM Tivoli Monitoring, to use the Sun 1.6.0_10 (or higher) JRE, before launching the browser you must enable the **legacy_lifecycle** parameter in the `applet.html` file on the computer where the Tivoli Enterprise Portal Server is installed.

- On Windows, launch Manage Tivoli Enterprise Monitoring Services.
  1. Right-click **Browser Task/SubSystem**, and select **Reconfigure**.
  2. Locate the **legacy_lifecycle** parameter, and double-click the line to bring up the edit dialog.
  3. Change the value to **true**, and check the **In Use** box.
  4. Click OK.
  5. Click OK again to save your changes.
- On Linux and AIX, edit the `applet.html` file in the *itm_home*/*platform*/`cw` branch, where *platform* is a string that represents your current operating system.

1. Locate this line:

   `document.writeln( '<PARAM NAME= "legacy_lifecycle" VALUE="false">' );`
2. Change the parameter value to `true`.
3. Save and close `applet.html`.

If you do not change the configuration correctly, the browser client will terminate, and you may see the server-connection error shown in Figure 72.


http://localhost:1920///cnp/kdh/lib/classes/forceExit.html - Windows Internet Explorer
http://localhost:1920///cnp/kdh/lib/classes/forceExit.html
Google
File
Edit
View
Favorites
Tools
Help
http://localhost:1920///cnp/kdh/lib/classes/forceExit....
Page
Tools
Done
Local intranet
100%

The Tivoli Enterprise Portal client has been terminated because a server connection error has occurred.

Contact your administrator to verify that the Tivoli Enterprise Portal Server is connected and available. Then exit the browser, restart the browser, and restart a Tivoli Enterprise Portal client session.

Figure 72. Server connection error, Tivoli Enterprise Portal browser client

Required maintenance: To use Sun Java 1.6.0_10 or higher with any IBM Tivoli Monitoring portal client (browser, Java Web Start, desktop), the following APAR is required: **IZ41252**. APAR IZ41252 is included in the following maintenance deliveries:

ITM6.2 fix pack 3 and subsequent maintenance
ITM6.2.1 interim fix 3 and subsequent maintenance
ITM6.2.2 (GA release) and subsequent maintenance and releases

Java Web Start clients

No configuration is required if the correct Java Web Start loader is used.

The Tivoli Enterprise Portal client is typically deployed via Java Web Start by entering the following address in the browser's location field:

```
http://teps_host:1920///cnp/kdh/lib/tep.jnlp
```

Under Windows, the last JRE installed on the machine usually controls which Java Web Start loader is associated with the Java Web Start deployment files, which have the extension `.jnlp`. If the Sun JRE was the last JRE installed, the Sun Java Web Start loader and associated JRE will be used for the Tivoli Enterprise Portal client. If both IBM and Sun Java are installed on the same machine and the IBM JRE was installed last, it may be necessary to manually re-associate the `.jnlp` extension with the Sun JRE.

To verify that the loader is associated with the correct JRE:

1. Launch Folder Options from Windows Control Panel folder.
2. Select the **File Types** tab.
3. Find and select the JNLP file type in the list of registered file types.
4. Click the **Advanced** button, select the Launch action, and click the **Edit...** button.
5. If the javaws.exe (the Java Web Start loader) is not associated with the correct JRE installation path, use the **Browse** button to locate and select the correct path.

Note to users of the single sign-on feature: If you have enabled single sign-on and your users start the portal's browser client via Java Web Start, a special URL is required:

```
http://tep_host:15200/LICServletWeb/LICServlet
```

Note that the servlet that supports single sign-on for the enterprise portal's Java Web Start client requires that port **15200** be open on the portal server.

For further information about single sign-on, see either "Support for single sign-on for launch to and from other Tivoli applications" on page 16 or the *IBM Tivoli Monitoring: Administrator's Guide*.

For more information on using and configuring Java Web Start client and setting up its environment, see "Using Web Start to download and run the desktop client" on page 321.

Installing and configuring IBM Java 6

In Tivoli Monitoring V6.2.3 Fix Pack 1, all Java-based Tivoli Monitoring infrastructure components are being upgraded to the Java 6 runtime environment. Although this upgrade is transparent for most Tivoli Monitoring components, you might have to complete additional tasks to upgrade the Tivoli Enterprise Portal component depending on the deployment mode you use for the Tivoli Enterprise Portal.

If you use the Tivoli Enterprise Portal desktop client, the upgrade to Java 6 is transparent because the supporting Java 6 binaries and support files are embedded within the Tivoli Enterprise Portal directory structure, and are updated automatically by the Tivoli Monitoring Installer. However, if you also use the Tivoli Enterprise Portal browser or Web Start client deployments, you must complete additional installation and configuration tasks to successfully upgrade the Tivoli Enterprise Portal to Java 6. You should only upgrade a browser client system to Java 6 if Internet Explorer will be used. This section describes the additional tasks for users of the Tivoli Enterprise Portal browser and the Web Start client.

- "Windows: Installing and configuring IBM Java 6" on page 305
- "Linux: Installing and configuring IBM Java 6" on page 314

Upgrade installation versus pristine installation

If you are upgrading from an earlier release of Tivoli Monitoring to V6.2.3 Fix Pack 1, you already have a correctly installed and configured version of Java available on the system where the Tivoli Enterprise Portal client is used. Supported Java versions for IBM Tivoli Monitoring releases before the 6.2.3 Fix Pack 1 release included IBM Java 5, Oracle/Sun Java 5, and Oracle/Sun Java 6. In this case, the upgrade to IBM Tivoli Monitoring version 6.2.3 Fix Pack 1 does not require any corresponding Java upgrade. You can continue to use the existing version of Java that is working successfully with your current release of the Tivoli Enterprise Portal browser and Web Start clients, after upgrading to IBM Tivoli Monitoring 6.2.3 Fix Pack 1. No other postinstallation configuration is required for the Tivoli Enterprise Portal browser or Web Start client.

However, if you decide after upgrading to IBM Tivoli Monitoring version 6.2.3 Fix Pack 1 that you want to upgrade your installed version of Java to IBM Java 6, you can do so by using much of the same procedure used for a pristine installation of IBM Tivoli Monitoring version 6.2.3 Fix Pack 1. See the following sections for information about manually upgrading your existing version of Java to IBM Java 6:

- "Windows: Upgrading your existing version of Java to IBM Java 6" on page 313
- "Linux: Upgrading your existing version of Java to IBM Java 6" on page 317

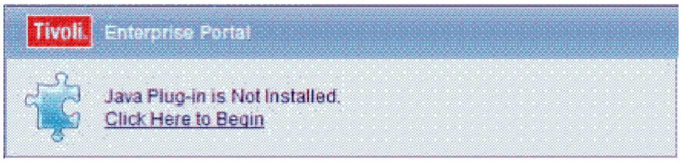
For a pristine installation of IBM Tivoli Monitoring version 6.2.3 Fix Pack 1 on Windows systems, the upgrade to IBM Java 6 is triggered when you first start V6.2.3 Fix Pack 1 on the Tivoli Enterprise Portal browser client. "Windows: Installing and configuring IBM Java 6" describes the installation of IBM Java 6. "Windows: Configuring IBM Java 6" on page 309 describes the required postinstallation configuration of IBM Java 6 for use with the Tivoli Enterprise Portal browser and Web Start clients.

For a pristine installation of IBM Tivoli Monitoring version 6.2.3 Fix Pack 1 on Linux systems, the installation and configuration of IBM Java 1.5 is required to start V6.2.3 Fix Pack 1 on the Tivoli Enterprise Portal browser client. Java 6 must be used with the Web Start Client. "Linux: Installing and configuring IBM Java 6" on page 314 describes the installation of IBM Java 6. "Linux: Configuring IBM Java 6" on page 314 describes the required postinstallation configuration of IBM Java 6 for use with the Tivoli Enterprise Portal Web Start clients.

Note: Only the 32-bit version of Java 6 for Linux is supported at this time. Any attempt to install a 64-bit version of Java to use with the Tivoli Enterprise Portal client will fail. The Tivoli Enterprise Portal is delivered with 32-bit native binaries that are not compatible with a 64-bit Java runtime environment.

Windows: Installing and configuring IBM Java 6

The upgrade to IBM Java 6 is triggered when you first launch the 6.2.3 Fix Pack 1 Tivoli Enterprise Portal browser client after a pristine installation. The Java upgrade requirement is automatically detected and the graphic image and download link in Figure 73 are displayed in your browser page.



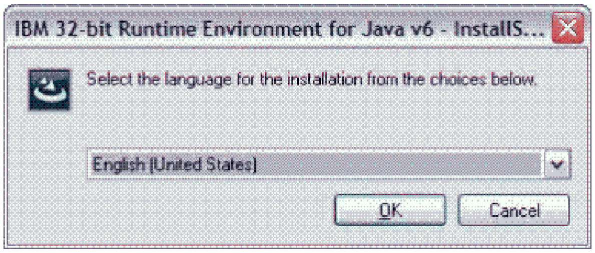
Java Plug-in is Not Installed.
Click Here to Begin

Figure 73. Java Plug-in

Complete the following steps to install IBM Java 6 for use by the Tivoli Enterprise Portal browser and Web Start clients:

1. Click the hyperlink **Click Here to Begin**.
2. A download prompt might be displayed. Click **Run** (or similarly labelled instruction) to download and run the IBM Java 6 installer.

3. After the IBM Java 6 installer has downloaded, you might be prompted again to allow the execution of the installer to proceed. Click **Run** (or similarly labelled instruction) to begin the installation.
4. Select the default language for the installation and click **OK**.



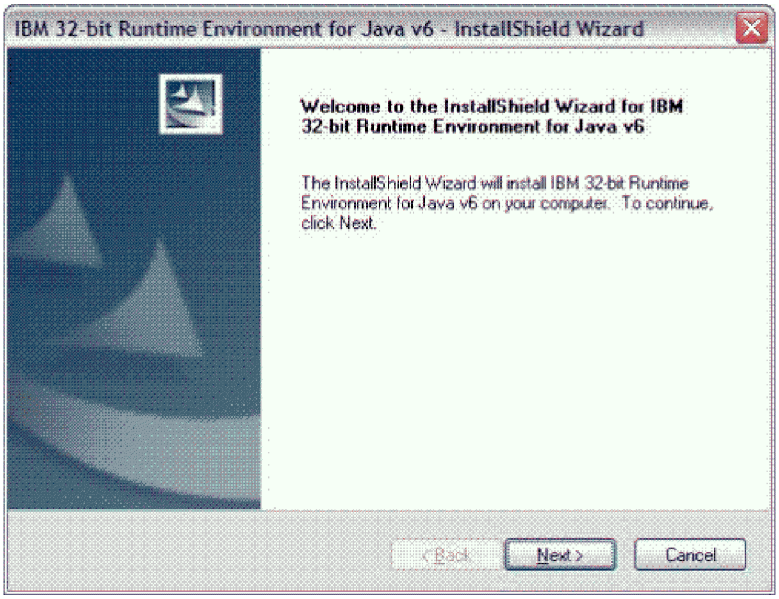
IBM 32-bit Runtime Environment for Java v6 - InstallS...

Select the language for the installation from the choices below.

English (United States)

Figure 74. IBM Java 6 installation default language

5. Click **Next** on the Welcome page.



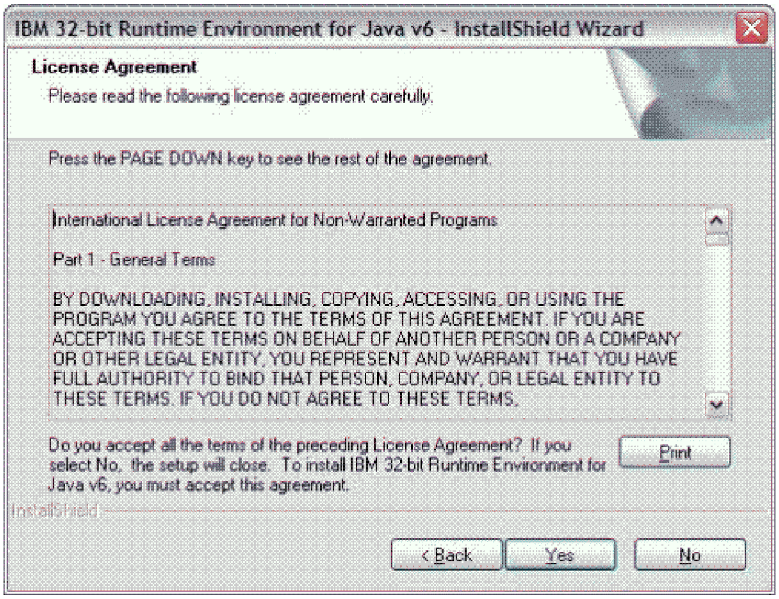
IBM 32-bit Runtime Environment for Java v6 - InstallShield Wizard

Welcome to the InstallShield Wizard for IBM 32-bit Runtime Environment for Java v6

The InstallShield Wizard will install IBM 32-bit Runtime Environment for Java v6 on your computer. To continue, click Next.

Figure 75. IBM Java 6 Welcome page

6. Click **Yes** to accept the license agreement.



IBM 32-bit Runtime Environment for Java v6 - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

International License Agreement for Non-Warranted Programs

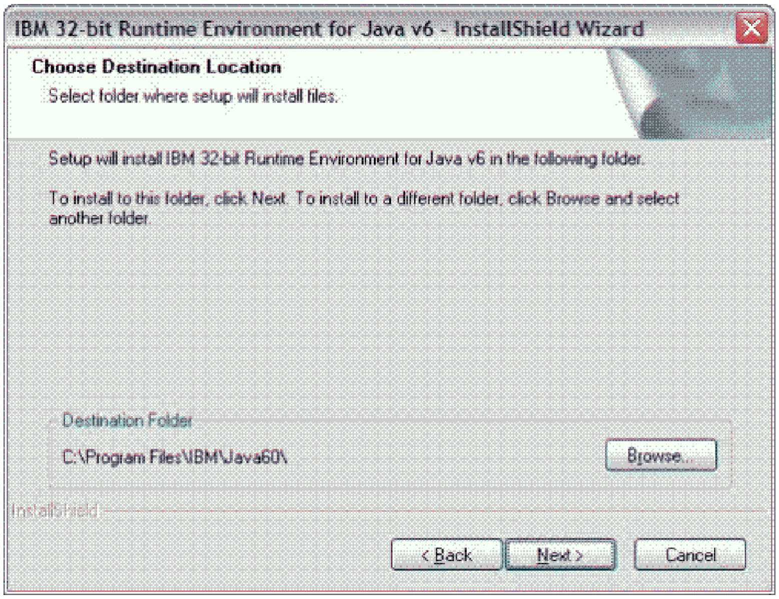
Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

Do you accept all the terms of the preceding License Agreement? If you select No, the setup will close. To install IBM 32-bit Runtime Environment for Java v6, you must accept this agreement.

Figure 76. IBM Java 6 Software License Agreement window

7. Click **Next** to accept the default installation location for IBM Java 6, or click **Browse** to browse to a different target location on your system.



IBM 32-bit Runtime Environment for Java v6 - InstallShield Wizard

Choose Destination Location

Select folder where setup will install files.

Setup will install IBM 32-bit Runtime Environment for Java v6 in the following folder.

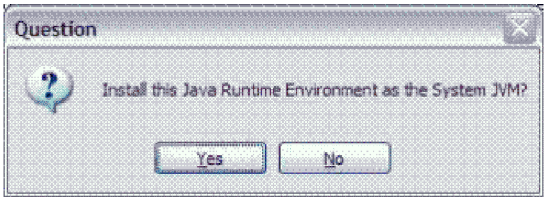
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\Program Files\IBM\Java60\

Figure 77. IBM Java 6 Choose Destination Location

8. Click **Yes** when asked to install IBM Java 6 as the System JVM.

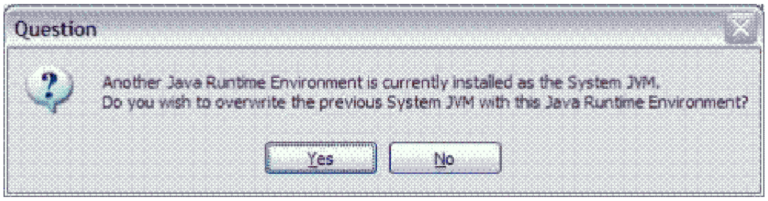


Question

Install this Java Runtime Environment as the System JVM?

Figure 78. Install System JVM

9. If another version of Java is already installed as the System JVM, you are asked if you want to establish IBM Java 6 as the System JVM. Click **Yes**.

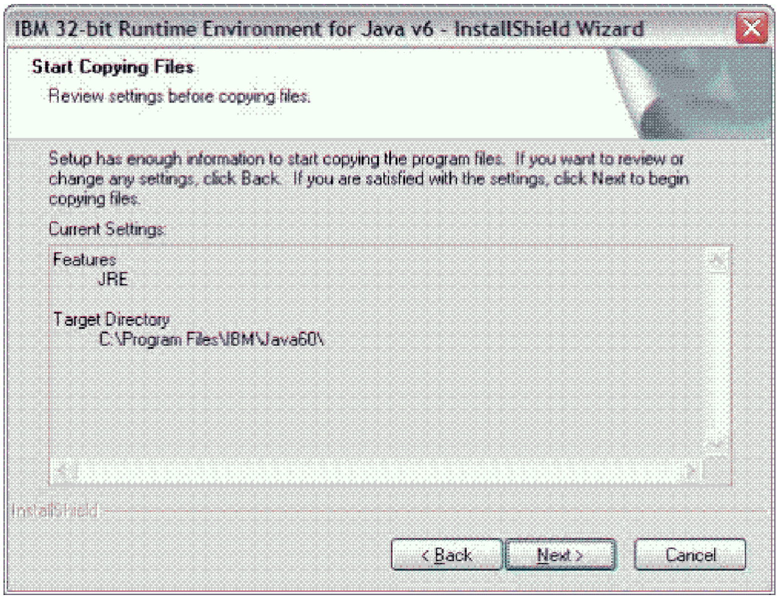


Question

Another Java Runtime Environment is currently installed as the System JVM.
Do you wish to overwrite the previous System JVM with this Java Runtime Environment?

Figure 79. Overwrite previous System JVM

10. A panel lists the installation options you have chosen so far. Click Next to begin the file copy and system update phase of the installation.



IBM 32-bit Runtime Environment for Java v6 - InstallShield Wizard

Start Copying Files

Review settings before copying files.

Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

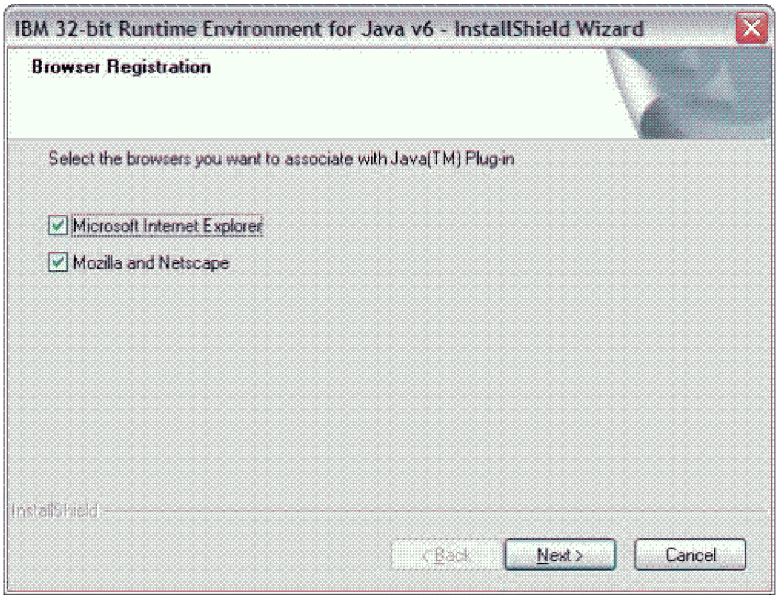
Current Settings:

Features
JRE

Target Directory
C:\Program Files\IBM\Java60\

Figure 80. IBM Java 6 Start Copying Files

11. After all of the support files are copied to your machine, you are be prompted to install and register the Java Plug-in modules used by Microsoft Internet Explorer, Mozilla Firefox, and Netscape. Unless you are an experienced user and have specific reasons for not accepting the defaults here, click **Next** to continue.



IBM 32-bit Runtime Environment for Java v6 - InstallShield Wizard

Browser Registration

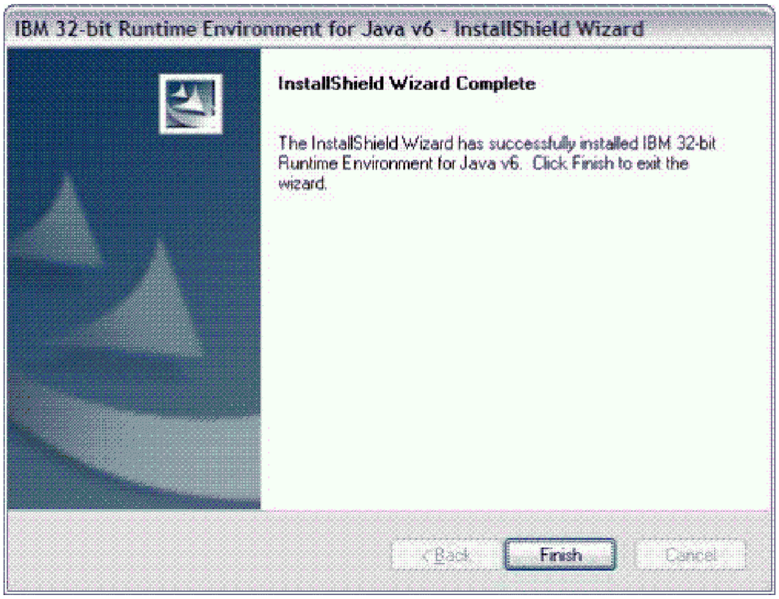
Select the browsers you want to associate with Java(TM) Plug-in

☑ Microsoft Internet Explorer

☑ Mozilla and Netscape

Figure 81. IBM Java 6 Browser Selection

12. After successful registration of the Java Plug-in modules, the Finish panel is displayed. Click **Finish** to complete the installation of IBM Java 6.



IBM 32-bit Runtime Environment for Java v6 - InstallShield Wizard

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed IBM 32-bit Runtime Environment for Java v6. Click Finish to exit the wizard.

Figure 82. IBM Java 6 InstallShield Wizard Complete

You have successfully installed IBM Java 6. "Windows: Configuring IBM Java 6" describes the required postinstallation configuration of IBM Java 6 for use with the Tivoli Enterprise Portal browser and Web Start clients.

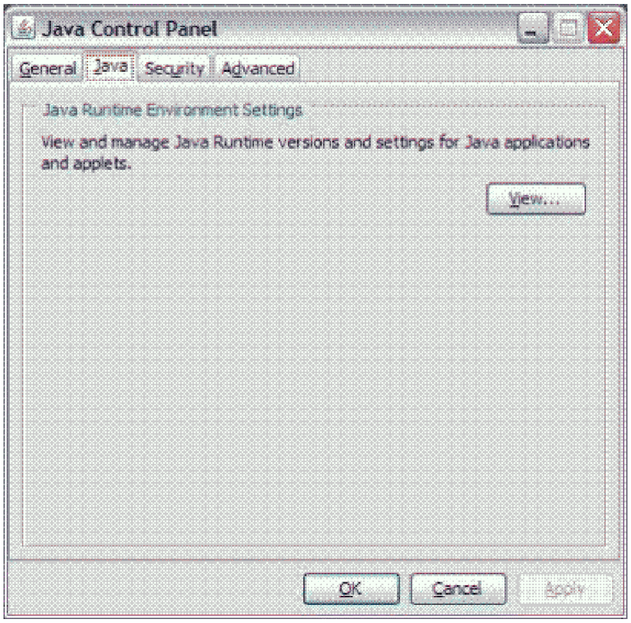
Windows: Configuring IBM Java 6

After installing IBM Java 6 you must configure the JVM to successfully launch the Tivoli Enterprise Portal browser and Web Start clients. Complete the following steps to configure IBM Java 6 for Windows.

1. Open your Windows **Control Panel** folder and locate the IBM Java 6 application icon.

Figure 83. IBM Java 6 application icon

2. To launch the Java Control Panel, double-click the IBM Java 6 application icon.
3. On the Java Control Panel, click the **Java** tab.

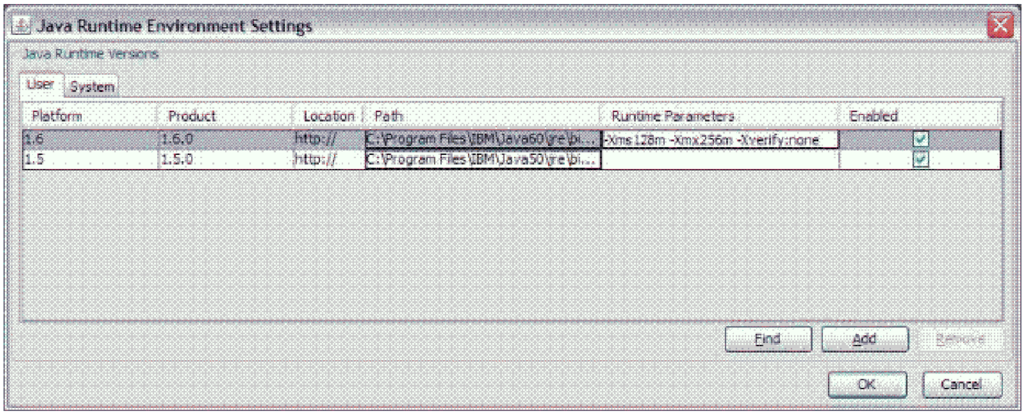


View and manage Java Runtime versions and settings for Java applications and applets.

Figure 84. Java Control Panel

4. Click **View** to list the installed versions of the Java runtime and to configure the Java runtime arguments for the IBM Java 6 environment.
5. The Java 1.6 platform row should be the first (or only) row in the Java Runtime Environment list. Click in the **Runtime Parameters** entry field for this row and type (or copy-and-paste) the following argument string into the parameters field:

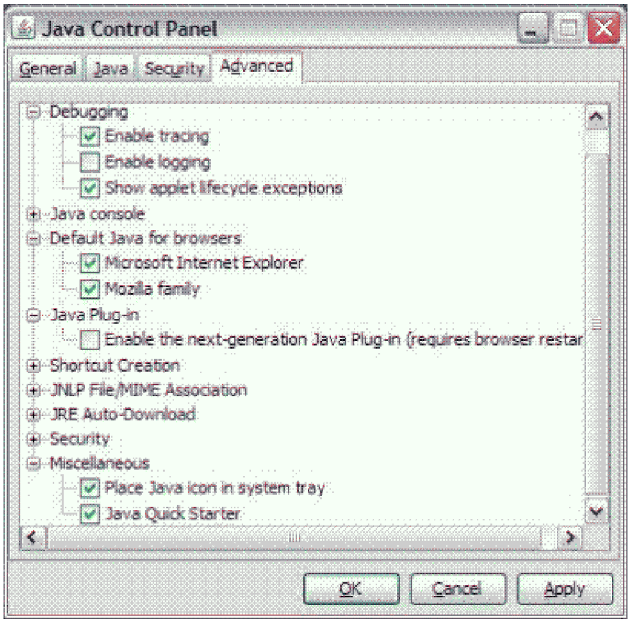
   ```
   -Xms128m -Xmx256m -Xverify:none
   ```



Platform	Product	Location	Path	Runtime Parameters	Enabled
1.6	1.6.0	http://	C:\Program Files\IBM\Java60\jre\bi...	-Xms128m -Xmx256m -Xverify:none	☑
1.5	1.5.0	http://	C:\Program Files\IBM\Java50\jre\bi...		☑

Figure 85. Java Runtime Environment list

6. Press the Enter key to commit the changes to the parameters field, and click **OK** to close the JRE Settings sub-panel.
7. Click **Apply** to save your changes before continuing with the rest of the configuration.
8. Click the **Advanced** tab and expand the branches in the property tree, as shown in Figure 86.



- Debugging
  - ☑ Enable tracing
  - □ Enable logging
  - ☑ Show applet lifecycle exceptions
- Java console
- Default Java for browsers
  - ☑ Microsoft Internet Explorer
  - ☑ Mozilla family
- Java Plug-in
  - □ Enable the next-generation Java Plug-in (requires browser restar
- Shortcut Creation
- JNLP File/MIME Association
- JRE Auto-Download
- Security
- Miscellaneous
  - ☑ Place Java icon in system tray
  - ☑ Java Quick Starter

Figure 86. Java Control Panel

Configure the following properties in the **Advanced** tab:

a. Under the **Debugging** branch, select the following two properties:
   - Enable tracing
   - Show applet lifecycle exceptions

b. Under the **Default Java for browsers** branch, select the following two properties (if not already selected):
   - Microsoft Internet Explorer
   - Mozilla family

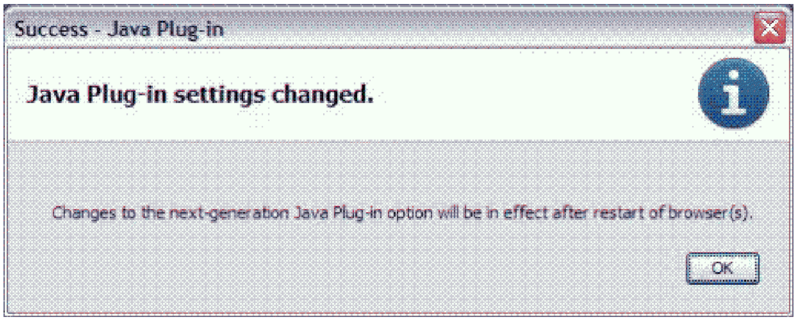
Note: Only one of these properties might be listed, depending on what browsers you have installed on the machine.

c. Under the **Java Plug-in** branch, clear the following property:

- Enable the next-generation Java Plug-in (requires browser restart)

Note: Clearing this property is critical for the successful operation of the Tivoli Enterprise Portal browser client using IBM Java 6.

When this property is changed, the following informational message might be displayed:



Success - Java Plug-in

Java Plug-in settings changed.

Changes to the next-generation Java Plug-in option will be in effect after restart of browser(s).

Figure 87. Java Plug-in settings changed

Click **OK** to continue.

d. Under the **Miscellaneous** branch, select the following two properties:

- Place Java icon in system tray
- Java Quick Starter

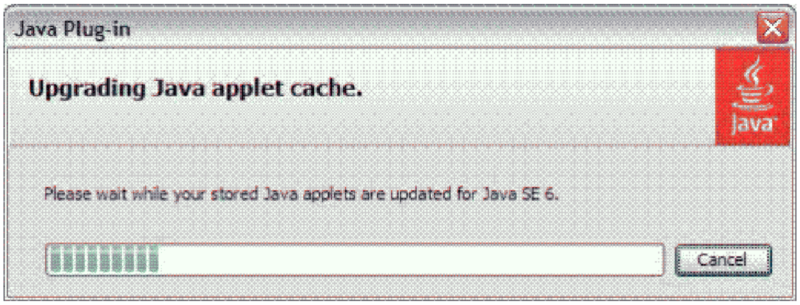
9. Click **Apply** to commit the property changes, and click **OK** to save and close the Java Control Panel.
10. You must recycle the browser for some of the configuration changes to take effect. Close the browser that you are using to launch the Tivoli Enterprise Portal browser or Web Start client.

You have successfully configured IBM Java 6.

Starting the Tivoli Enterprise Portal client

The first time that you start The Tivoli Enterprise Portal browser and Web Start clients, you can expect a longer than normal delay before the Tivoli Enterprise Portal login panel is displayed. The longer delay is caused by the updated IBM Tivoli Monitoring changes you make, as well as the initial population of the IBM Java 6 Java archive file (JAR) cache. Subsequent launches of the Tivoli Enterprise Portal client will be noticeably faster.

If you previously configured the Tivoli Enterprise Portal browser client to operate under IBM Java 5, which might be the case if you are upgrading to IBM Tivoli Monitoring version 6.2.3 Fix Pack 1 from a previous IBM Tivoli Monitoring release, you might see the following progress panel displayed during the initial launch of the client:


Java Plug-in

Upgrading Java applet cache.

Please wait while your stored Java applets are updated for Java SE 6.

Figure 88. Upgrading Java applet cache

This progress panel is normal and is shown only during initial launch of the client.

Windows: Upgrading your existing version of Java to IBM Java 6

If you are upgrading to IBM Tivoli Monitoring 6.2.3 Fix Pack 1 from a previous IBM Tivoli Monitoring release, then as stated previously in this section, no requirement exists to upgrade to IBM Java 6. Your current working Java environment for the Tivoli Enterprise Portal browser and Web Start clients will continue to work after the IBM Tivoli Monitoring upgrade is successfully completed.

However, your Web Start client users can upgrade the Java environment to IBM Java 6 by following the procedure in “Windows: Installing the IBM JRE” on page 321. You can also complete the following steps if you want to upgrade your Java environment to IBM Java 6 for your TEP browser clients if all of your clients are using Internet Explorer:

1. On the machine where the Tivoli Enterprise Portal Server is installed, locate and open the `jrelevel.js` file for editing. The `jrelevel.js` file is located in the `<itm_install_dir>\cnb` directory.
2. Change the value of the `jrelevel` statement to the following value:

   ```
   var jreLevel    = "1.6.0";
   ```

   This value references the use of IBM Java 6, and is used by the Tivoli Enterprise Portal browser client to determine which installed level of Java to use.
3. **Save** the changes you made to this file.
4. Now launch the Tivoli Enterprise Portal browser client from the machine where you normally run the Tivoli Enterprise Portal client, and follow the procedures outlined in the preceding sections: “Windows: Installing and configuring IBM Java 6” on page 305.

Note: Although both IBM Java 5 and Java 6 releases can be installed and co-exist on the same machine, be aware of the following issues if you want to use both of these releases with the Tivoli Enterprise Portal browser client:

1. If you plan on switching from IBM Java 6 back to IBM Java 5 (or vice-versa), make sure you first update the `jrelevel.js` file located on the Tivoli Enterprise Portal Server machine as described in this section. The value to use for IBM Java 5 is the following:

   ```
   var jreLevel    = "1.5.0";
   ```
2. When switching between IBM Java releases, you must reenter and save the Runtime Parameters for the appropriate JRE. Follow step 5 on page 310 in “Windows: Configuring IBM Java 6” on page 309 to update the JRE runtime parameters. Make sure you are entering the parameters in the correct entry field associated with the JRE you are switching over to use, for example 1.5 or 1.6.
3. You cannot launch two or more independent instances of the Tivoli Enterprise Portal browser client under Internet Explorer, from the same machine, using different IBM Java releases. If you must run more than one instance of the Tivoli Enterprise Portal client that use different IBM Java releases, then you can use another deployment mode for the second (or subsequent)

instance. For example, you can run one Tivoli Enterprise Portal instance by using Internet Explorer and IBM Java 5, and a second Tivoli Enterprise Portal instance by using Web Start or desktop deployments running IBM Java 6.

Linux: Installing and configuring IBM Java 6

The IBM Java 6 installation package is provided as an `.rpm` file located on the system where the Tivoli Enterprise Portal Server component is installed. Support for the Tivoli Enterprise Portal browser client on Linux is limited to the use of the Mozilla Firefox browser, releases 3.0.x through 3.5.x at the 32-bit level and Java 1.5. Firefox v3.6 and later releases are not supported by the Tivoli Enterprise Portal browser client.

Complete the following steps to install IBM Java 6 for use with the Tivoli Enterprise Portal Web Start clients in a 32-bit Linux environment:

1. From the system where your Tivoli Enterprise Portal Server is located, download the IBM Java 6 `.rpm` installer package file using the following URL:

   ```
   http://teps_hostname:15200/java/ibm-java-i386-jre-6.0-9.2.i386.rpm
   ```

   Or, if you prefer to use a different method for retrieving this file, you can find this package file in the following directory location on the Tivoli Enterprise Portal Server machine:

   ```
   /<itm_install_dir>/<arch>/cw/java
   ```

   Save the file in a directory of your choosing, for example `/home`.
2. Open a shell prompt, making sure you do so as the **root** user. Make the directory where you saved the IBM Java 6 package file your current working directory.
3. To upgrade your Java environment using the Red Hat Package Manager (RPM) tool, you must first uninstall any previous IBM Java 6 version (assuming it exists on your client machine). You can use the command `rpm -e package_name` to uninstall an existing IBM Java 6 package. You can use the command `rpm -qa` to list all of your installed packages on Linux.
4. To install the IBM Java 6 package, use the following rpm command:

   ```
   rpm -ivh ibm-java-i386-jre-6.0-9.2.i386.rpm
   ```
5. Now create a symbolic link to the plug-in using the following command:

   ```
   ln -s /opt/ibm/java-i386-60/jre/plugin/i386/ns7/libjavaplugin_oji.so
   ```

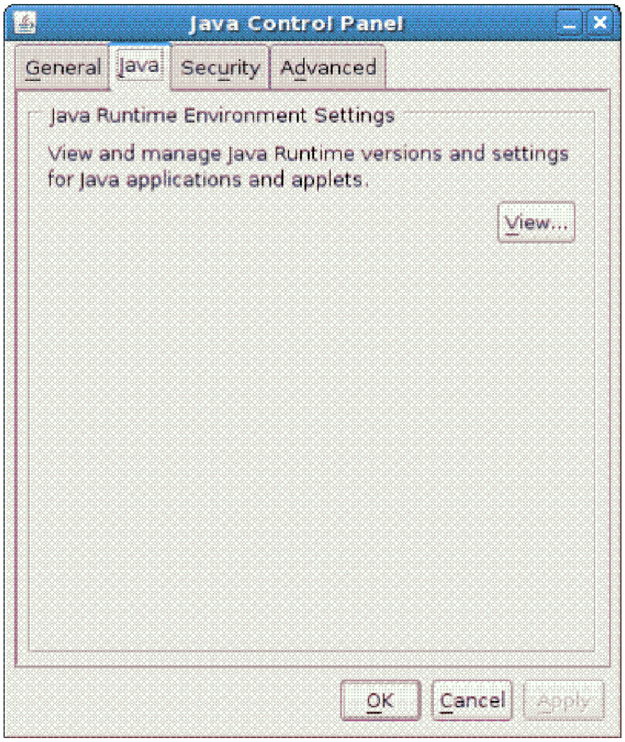
You have successfully installed IBM Java 6. “Linux: Configuring IBM Java 6” describes the required postinstallation configuration of IBM Java 6 for use with the Tivoli Enterprise Portal Web Start clients.

Linux: Configuring IBM Java 6

After installing IBM Java 6 you must configure the JVM to successfully launch the Tivoli Enterprise Portal Web Start clients. Complete the following steps to configure IBM Java 6 for Linux:

1. Open a shell prompt, making sure you do so as the **root** user.
2. Launch the IBM Java 6 Control Panel application:

   ```
   cd /opt/ibm/java-i386-60/jre/bin
   ./ControlPanel
   ```
3. On the Java Control Panel, click the **Java** tab.

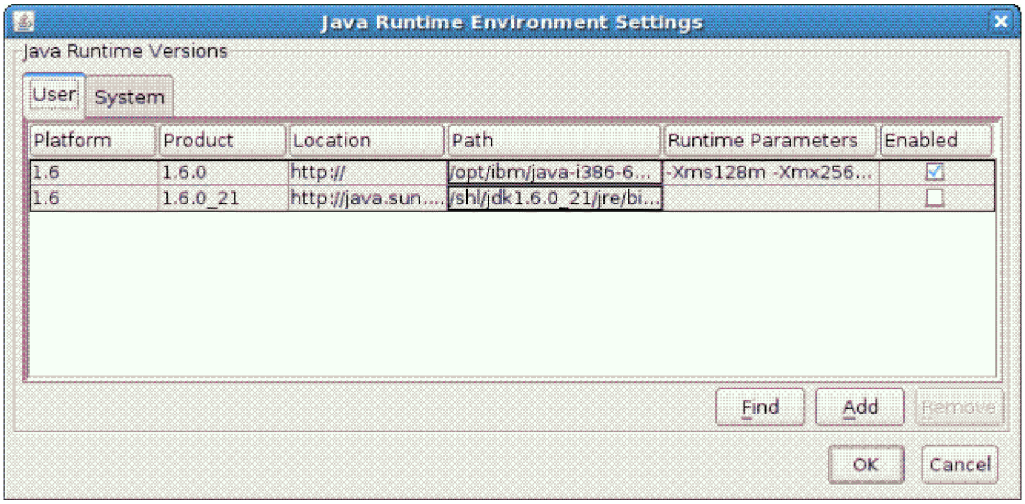


View and manage Java Runtime versions and settings for Java applications and applets.

Figure 89. Java Control Panel

4. On the Java tab, click **View** to see the installed versions of the Java runtime, and to configure the Java runtime arguments for the IBM Java 6 environment.
5. Click in the **Runtime Parameters** entry field associated with the Java 1.6 platform row, choose the one associated with IBM Java Path, and type (or copy-and-paste) the following argument string into the parameters field:

```
-Xms128m -Xmx256m -Xverify:none
-Djava.protocol.handler.pkgs=sun.plugin.net.protocol
```

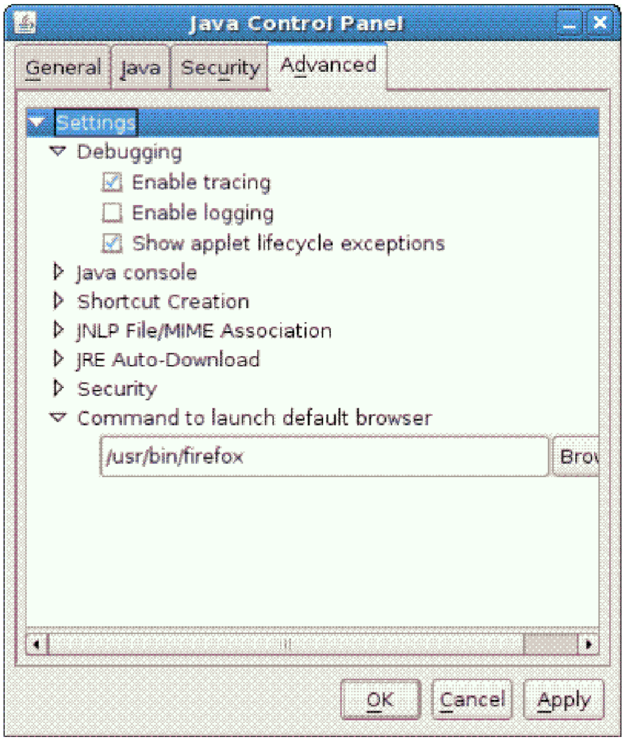


Platform	Product	Location	Path	Runtime Parameters	Enabled
1.6	1.6.0	http://	/opt/ibm/java-i386-6...	-Xms128m -Xmx256...	☑
1.6	1.6.0_21	http://java.sun....	/shl/jdk1.6.0_21/jre/bi...		☐

Figure 90. Java Runtime Environment Settings

6. Press the **Enter** key to commit the changes to the parameters field, and click **OK** to close the JRE Settings sub-panel.

7. The Java Control Panel is displayed, click **Apply** to save your changes before continuing with the rest of the configuration.
8. Click the **Advanced** tab and expand the branches in the property tree, as shown in Figure 91.



Settings
- Debugging
  - ☑ Enable tracing
  - □ Enable logging
  - ☑ Show applet lifecycle exceptions
- Java console
- Shortcut Creation
- JNLP File/MIME Association
- JRE Auto-Download
- Security
- Command to launch default browser
  - /usr/bin/firefox

Figure 91. Java Control Panel

Configure the following properties in the **Advanced** tab:

a. Under the **Debugging** branch, select the following two properties:
   - Enable tracing
   - Show applet lifecycle exceptions

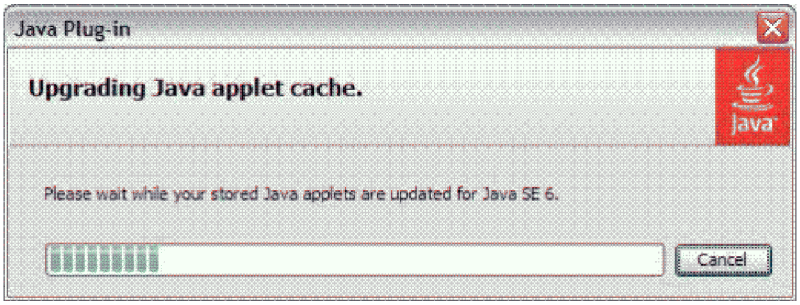
b. Under the **Command to launch default browser** branch, make sure you have the correct path for launching the Firefox browser.

9. Click **Apply** to commit the property changes, and click **OK** to save and close the Java Control Panel.
10. You must recycle the Firefox browser for some of the configuration changes to take effect.

Starting the Tivoli Enterprise Portal client

The first time that you start The Tivoli Enterprise Portal browser and Web Start clients, you can expect a longer than normal delay before the Tivoli Enterprise Portal login panel is displayed. The longer delay is caused by the updated IBM Tivoli Monitoring changes you make, as well as the initial population of the IBM Java 6 Java archive file (JAR) cache. Subsequent launches of the Tivoli Enterprise Portal client will be noticeably faster.

If you previously configured the Tivoli Enterprise Portal browser client to operate under IBM Java 5, as might be the case if you are upgrading to IBM Tivoli Monitoring version 6.2.3 Fix Pack 1 from a previous IBM Tivoli Monitoring release, you might see the following progress panel displayed during the initial launch of the client:



Upgrading Java applet cache.

Please wait while your stored Java applets are updated for Java SE 6.

Figure 92. Upgrading Java applet cache

This progress panel is normal and is shown only during initial launch of the client.

Linux: Upgrading your existing version of Java to IBM Java 6

If you are upgrading to IBM Tivoli Monitoring 6.2.3 Fix Pack 1 from a previous IBM Tivoli Monitoring release, then as stated previously in this section, no requirement exists to upgrade to IBM Java 6. Your current working Java environment for the Tivoli Enterprise Portal browser and Web Start clients will continue to work after the IBM Tivoli Monitoring upgrade is successfully completed.

However, your Web Start client users can upgrade the Java environment to IBM Java 6 by following the procedure in “Linux: Installing the IBM JRE” on page 322.

Note: Although both IBM Java 5 and Java 6 releases can be installed and co-exist on the same machine, be aware of the following issues if you want to use both of these releases with the Tivoli Enterprise Portal browser client:

1. If you plan on switching from IBM Java 6 back to IBM Java 5 (or vice-versa), make sure you first update the `jrelevel.js` file located on the Tivoli Enterprise Portal Server machine as described in this section. The value to use for IBM Java 5 is the following:

   ```
   var jreLevel    = "1.5.0";
   ```

2. When switching between IBM Java releases, you must reenter and save the Runtime Parameters for the appropriate JRE. Follow step 5 on page 315 in “Linux: Configuring IBM Java 6” on page 314 to update the JRE runtime parameters. Make sure you are entering the parameters in the correct entry field associated with the JRE you are switching over to use, for example 1.5 or 1.6.
3. The version of the Java plug-in used by Firefox is based on the registration of the plug-in object. Follow step 5 on page 314 in “Linux: Installing and configuring IBM Java 6” on page 314 to establish a symbolic link to the plug-in object `libjavaplugin_oji.so`, depending on what version of Java you are switching over to use (Java 5 or Java 6).

Specifying the browser used for online help

If you are running the desktop client on Linux, or you want to view the online help with some browser other than Internet Explorer on Windows, you must specify to the portal server the location of the browser you want to use.

- “Windows: Specifying the browser location”
- “UNIX and Linux: Specifying the browser location” on page 318
- “Web Start: Specifying the browser location” on page 319

Windows: Specifying the browser location

Use the Manage Tivoli Enterprise Monitoring Services utility to change the location of the browser that the browser or desktop client uses.

1. Launch Manage Tivoli Enterprise Monitoring Services (**Start → (All) Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**).
2. In the Manage Tivoli Enterprise Monitoring Services window, right-click the browser or desktop client and select **Reconfigure**.

   The Configure the Tivoli Enterprise Portal Browser window is displayed. (If you are configuring the desktop client, the Configure Application Instance window is displayed.)
3. Scroll down in the list of variables until you see the `kjr.browser.default` variable.
4. Double-click `kjr.browser.default`.

   The Edit Tivoli Enterprise Portal Browser Parm window is displayed.
5. In the Value field, type the path and the application name of the alternative browser application. For example:

   ```
   C:\Program Files\Mozilla Firefox\firefox.exe
   ```
6. Check the **In Use** box.
7. Click **OK** to close the editing window and save the change.
8. Click **OK** to close the reconfiguration window.

UNIX and Linux: Specifying the browser location

To change a property such as the location of the Web browser that the Tivoli Enterprise Portal browser client launches in UNIX, update the shell script file or files that are run and the template that is used when the browser client is configured to create the script file or files that are run. You might have to update one or more of the files list in Table 58:

Note: All file paths are relative to your *install_dir* directory where you installed IBM Tivoli Monitoring.

Table 58. File locations for changing application properties for UNIX and Linux

File location	Purpose of file
bin/cnp.sh	The default shell script that launches the Tivoli Enterprise Portal browser client.
bin/cnp_*instance*.sh	The shell script for a specific instance you have created, where *instance* is the name of the instance that launches the Tivoli Enterprise Portal browser client.
platform/cj/original/cnp.sh_template	The template from which the bin/cnp.sh and bin/cnp_*instance*.sh shell scripts are generated during configuration, where *platform* is the code for the operating system platform on which IBM Tivoli Monitoring is installed. For example: *li6243* for Linux 2.4 on a 32-bit Intel CPU). If you change only `bin/cnp.sh` or `bin/cnp_instance.sh` and do not change this template, the next time you configure the client, a new version of the script is created without the changes you made to `bin/cnp.sh` or `bin/cnp_instance.sh`.

To change the location of the Web browser you must change the above file or files to include a new property:

1. Go to the *install_dir*/bin/cnp.sh and edit the cnp.sh shell script.
2. Add your Web browser location to the last line of the file. In the example below, the Web browser location is */opt/foo/bin/launcher*. `-Dkjr.browser.default=/opt/foo/bin/launcher`

   Important: The line is very long and has various options on it, including several other –D options to define other properties. It is very important to add the option in the correct place.

If the last line of your bin/cnp.sh originally looked like the following:

```
${JAVA_HOME}/bin/java -showversion -noverify -classpath ${CLASSPATH}
-Dkjr.trace.mode=LOCAL -Dkjr.trace.file=/opt/IBM/ITM/logs/kcjras1.log
-Dkjr.trace.params=ERROR -DORBtcpNoDelay=true -Dcnp.http.url.host=
-Dvbroker.agent.enableLocator=false
-Dhttp.proxyHost=
-Dhttp.proxyPort=candle.fw.pres.CMWApplet 2>& 1 >> ${LOGFILENAME}.log
```

To set the browser location to */opt/foo/bin/launcher*, change the line to look like the following:

```
${JAVA_HOME}/bin/java -showversion -noverify -classpath ${CLASSPATH}
-Dkjr.browser.default=/opt/foo/bin/launcher
-Dkjr.trace.mode=LOCAL -Dkjr.trace.file=/opt/IBM/ITM/logs/kcjras1.log
-Dkjr.trace.params=ERROR -DORBtcpNoDelay=true -Dcnp.http.url.host=
-Dvbroker.agent.enableLocator=false
-Dhttp.proxyHost=
-Dhttp.proxyPort=candle.fw.pres.CMWApplet 2>& 1 >> ${LOGFILENAME}.log
```

Web Start: Specifying the browser location

Java Web Start deployed applications are described in jnlp deployment files. For IBM Tivoli Monitoring, there is one deployment file that describes the core Tivoli Enterprise Portal framework component and associated JAR files, and one deployment file for each and every Tivoli Enterprise Portal-based monitoring solution that is installed. The core Tivoli Enterprise Portal Server deployment file is named tep.jnlp. The application deployment file is typically called k*xx*_resources.jnlp or k*xx*.jnlp, where *xx* is the application identifier (a product code, such as **nt**, **ux**, or **lz**).

- On a Windows computer where the Tivoli Enterprise Portal Server is installed, the file is located in *itminstall_dir*\CNB (for example, c:\IBM\ITM\CNB).
- On a Linux computer where the Tivoli Enterprise Portal Server is installed, the file is located in *itminstall_dir*/*arch*/cw (for example, /opt/IBM/ITM/li6263/cw).

The deployment file instances are generated whenever the Tivoli Enterprise Portal Server is installed or reconfigured (for example, when adding a new monitoring solution to the environment). The contents of these files are based upon two template deployment files (.jnlpt). The core Tivoli Enterprise Portal template deployment file is called tep.jnlpt. The application template deployment file is named component.jnlpt.

- On a Windows computer where the Tivoli Enterprise Portal is installed, the file is located in *itminstall_dir*\Config (for example: c:\IBM\ITM\Config).
- On a UNIX computer where the Tivoli Enterprise Portal is installed, the file is located in *itminstall_dir*/config (for example, /opt/IBM/ITM/config).

1. To add or modify JVM arguments (such as maximum heap size) or other Tivoli Enterprise Portal-based properties (such as RAS1 trace options), you must edit either the tep.jnlp deployment file or the tep.jnlpt deployment template file. The deployment file is nothing more than XML syntax that describes the Web Start application being deployed. The **<resources>** element is used to define the JVM arguments, the Tivoli Enterprise Portal properties, JAR files, and references to component deployment files.
   - Modify the tep.jnlp file if the change will be temporary (for example, setting a trace option for gathering further diagnostics).
   - Modify the tep.jnlpt template file if the change will be long-term (for example, increasing the maximum heap size to accommodate a larger monitored environment or increased event load).

     If you modify the deployment template file, make sure you then reconfigure the Tivoli Enterprise Portal Server in order to regenerate the instance-level .jnlp deployment files with your changes.

     Note: Changes to tep.jnlpt are not preserved when you migrate to a new release. A backup of the tep.jnlpt file will be made during migration; you must manually compare this backup to the new version installed during migration to determine what changes need to be manually incorporated into the new tep.jnlpt.

2. To specify the location of the browser to use to display the online help, add the following property to the **<resources>** section of the appropriate file:

   ```
   <property name="kjr.browser.default" value="path_where_browser_is_located>"
   ```

   Windows example:

   ```
   <resources os="Windows">
       <jar href="classes/browser-winnt.jar"/>
       <jar href="classes/browser-core-winnt.jar"/>
       <property name="kjr.browser.default" value="C:\Program Files\Internet Explorer\iexplore.exe"/>
   </resources>
   ```

   Linux example:

   ```
   <resources os="Linux">
       <jar href="classes/browser-li.jar"/>
       <jar href="classes/browser-core-li.jar"/>
       <property name="kjr.browser.default" value="/usr/bin/firefox"/>
   </resources>
   ```

   Note:

   For the online help to display on a Linux or Unix computer, the `kjr.browser.default` value must specify Firefox as the default browser.

   `kjr.browser.default` is not the only property you can specify using the **<property>** keyword. You can include any client parameters that are specific to your operating system.

Starting the Tivoli Enterprise Portal client

After you have successfully installed and configured all the components of your IBM Tivoli Monitoring environment, you can verify the installation and configuration by launching the Tivoli Enterprise Portal to view monitoring data. You can access the Tivoli Enterprise Portal using either the desktop client or the browser client.

Your monitoring server and portal server must be running for the portal client to start successfully.

Starting the desktop client

Follow these steps to start the desktop client:

On Windows:

1. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Tivoli Enterprise Portal**.
2. Type your user ID and password in the logon window. The default user ID is **sysadmin**.
3. Click **OK**.

On Linux, run the following command to start the portal desktop client:

```
./itmcmd agent start cj
```

Starting the browser client

Follow these steps to start the browser client:

1. Start the browser.
2. Type the URL for the Tivoli Enterprise Portal into the **Address** field of the browser:

   ```
   http://systemname:1920///cnp/client
   ```

   where the *systemname* is the host name of the computer where the Tivoli Enterprise Portal Server is installed, and 1920 is the port number for the browser client. 1920 is the default port number for the browser client. Your portal server might have a different port number assigned.

3. Click **Yes** on the Warning - Security window.
4. Type your user ID and password in the logon window. The default user ID is **sysadmin**.
5. Click **OK**.

Using Web Start to download and run the desktop client

A desktop client obtained from the Tivoli Enterprise Portal Server through IBM Web Start for Java benefits from centralized administration from the server. Like the browser client, it is automatically configured with the latest updates each time you start the client, and there is no need to configure application support.

Before you use IBM Web Start for Java to download the desktop client from the Tivoli Enterprise Portal Server:

- The Tivoli Enterprise Portal Server must be installed. (See "Installing the Tivoli Enterprise Portal Server" on page 228.)
- IBM 32-bit Runtime Environment for Windows, Java 2, version 5.0 must be installed on the computer to which you want to download the desktop client.

  You can download the IBM JRE installer from the Tivoli Enterprise Portal Server (see "Installing the IBM JRE"). The IBM JRE must be installed as the system JVM.

  If you want to run the desktop client on a system that already has a Tivoli Management Services base component installed (such as a monitoring server or the portal server), there is no need to install the IBM JRE. The correct version of the IBM JRE is installed with the Tivoli Management Services component.

If you run the desktop client using Web Start instead of installing it from the installation media, you must configure the JRE to enable tracing for the desktop client (see "Enabling tracing for the JRE" on page 322).

Installing the IBM JRE

If you intend to download and run the desktop client using Web Start on a computer where no IBM Tivoli Monitoring base component is installed, you must first install IBM Java 6. You download an installer from the computer where the Tivoli Enterprise Portal Server is installed:

- "Windows: Installing the IBM JRE"
- "Linux: Installing the IBM JRE" on page 322

Windows: Installing the IBM JRE

Complete the following steps to download the IBM JRE installer from the Tivoli Enterprise Portal Server and install the JRE on a Windows computer:

1. Start the browser on the computer to which you want to download the installer.
2. Enter the following URL in the **Address** field of the browser:

   `http://TEPS_host_name:1920///cnp/kdh/lib/java/ibm-java6.exe`

   where *TEPS_host_name* is the fully qualified host name of the computer where the portal server is installed (for example, `myteps.itmlab.company.com`).
3. When prompted, save the **java/ibm-java6.exe** file to a directory on your hard drive.
4. Change to the directory where you saved the **java/ibm-java6.exe** file and double-click the file to launch the JRE installer to start the installation program.
5. On the pop-up window, select the language from the drop-down list and click **OK**.
6. Click **Next** on the Welcome page.
7. Click **Yes** to accept the license agreement.
8. Accept the default location for installing the JRE or browse to a different directory. Click **Next**.
9. Click **NO** on the message asking if you want to install this JRE as the system JVM.

   Make Java 1.5 the system JVM only if there are no other JREs installed on the computer.

10. If another JRE is currently installed as the system JVM and you are prompted to overwrite the current system JVM, click **NO**.

    Overwriting the current system JVM may cause applications depending on the current JVM to fail.
11. Click **Next** on the Start Copying Files window to start installing the JRE.
12. On the Browser Registration window, select the browsers that you want the IBM JRE to be associated with. These would normally be the browsers that you want to use with the browser client.
13. Click **Next**.
14. Click **Finish** to complete the installation.

Linux: Installing the IBM JRE

Complete the following steps to download the IBM JRE installer from the Tivoli Enterprise Portal Server and install the JRE on a Linux computer.

1. Start the browser on the computer to which you want to download the installer.
2. Enter the following URL in the **Address** field of the browser:

   ```
   http://teps_hostname:1920///cnp/kdh/lib/java
       /ibm-java-i386-jre-6.0-9.2.i386.rpm
   ```

   where *teps_hostname* is the fully qualified host name of the computer where the portal server is installed (for example, `myteps.itmlab.company.com`).
3. When prompted, save the installer to disk.
4. Change to the directory where you saved the **ibm-java-i386-jre-6.0-9.2.i386.rpm** file and launch the installer to start the installation program using the following command:

   ```
   rpm -ivh ibm-java-i386-jre-6.0-9.2.i386.rpm
   ```

You can also install the JRE without downloading the installer by supplying the URL to the rpm in the command:

```
rpm -ivh http://teps_hostname:1920///cnp/kdh/lib/java
     /ibm-java-i386-jre-6.0-9.2.i386.rpm
```

Enabling tracing for the JRE

Log files are not created for the desktop client launched through Web Start unless you enable tracing for the JRE.

The logs for a desktop client run using Web Start are located in a different place than logs for the browser client and the desktop client installed from the media. On Windows computers, the logs for the Web Start client are located in the `C:\Documents and Settings\Administrator\Application Data\IBM\Java\Deployment\log` directory. On Linux computers, the logs are located in the `.java/deployment` directory of the home directory of the user ID under which the Java JRE was installed. Java Web Start will create a uniquely named trace file for every independent launch of the application. The files are named **javaws***nnnnn***.trace**, where *nnnnn* is an arbitrary five-digit identifier.

Complete the following steps to enable tracing:

1. Launch the IBM Control Panel for Java.
   - On Windows, select **Start → Control Panel**, then double-click IBM Control Panel for Java.

     You must switch to the Classic view to see and select the Control Panel. Alternatively, you can launch the Control Panel by selecting **Start → Run → "C:\Program Files\IBM\Java50\jre\bin\javacpl.exe**".
   - On Linux, change to `install_dir/jre/platform/bin` and run Control Panel:

     ```
     ./Control Panel
     ```
2. Select the **Advanced** tab.
3. Expand the Debugging node in the **Settings** tree and check **Enable Tracing**.
4. Click **OK** to save the setting and close the Java Control Panel.

Downloading and running the desktop client

You can use any of the following three methods to download and run the desktop client using Web Start:

- Entering the URL of the portal server in a browser
- Launching the client from the IBM Java Control Panel
- Entering a URL from the command-line

The first time you launch the desktop client, you are prompted to create a shortcut. After that, you can launch the client using the shortcut.

Entering the URL of the portal server in a browser:

Use the following procedure to launch the desktop client from a browser:

1. Start the browser on the computer on which you want to use the desktop client.
2. Enter the following URL in the **Address** field of the browser:

   http://*TEPS_host_name*:1920///cnp/kdh/lib/tep.jnlp

   where *TEPS_host_name* is the fully qualified host name of the computer where the Tivoli Enterprise Portal Server is installed (for example, `myteps.itmlab.company.com`).
3. Click **Run** on the security message.
4. You are asked if you want to create a shortcut on your desktop for the Tivoli Enterprise Portal. Click **Yes** if you want to create the shortcut.

   The desktop client starts and displays the logon window.

   Note: If IBM Java 1.5 is not the system JVM, you cannot use this shortcut. You must create your own. See "Manually creating a shortcut for the Web Start client" on page 324.
5. Enter the user ID and password to log on to the Tivoli Enterprise Portal or click **Cancel** if you do not want to log on at this time. (The default user ID is **sysadmin**.)

Note: If you set the RAS trace option for the Tivoli Enterprise Portal client as documented in *IBM Tivoli Monitoring: Troubleshooting Guide*, when you recycle the client the kcjras1.log should be created in the location from which the client was launched. On Windows this defaults to \Documents and Settings*userid*\Desktop.

Launching the desktop client from the IBM Java Control Panel:

IBM Java 1.5 introduces a new control panel for managing both Java applets deployed via the Java plug-in, and Java applications deployed via Web Start. Complete the following steps to launch the desktop client using the control panel:

1. Launch the IBM Java Control Panel:
   - Windows: In the Windows control panel, double-click **IBM Java Control Pane**l.

     Note: You must be in the Classic view to see **IBM Java Control Panel**.
   - Linux: Change to `install_dir/jre/platform/bin` directory (the default directory is `/opt/IBM/ITM/jre/platform/bin` and enter `./Control Panel`.
2. On the **General** tab, in the Temporary Internet Files section, click **Settings**. The Temporary Files Settings window is displayed.
3. Click **View Applications**.
4. On the **User** tab, select Tivoli Enterprise Portal, and then click **Launch Online**.

   Web Start downloads and starts the desktop client. When the application is launched, you can close the Control Panel windows.

Launching the desktop client from the command-line:

Complete the following steps to launch the desktop client using Web Start from the command-line:

1. Open a command prompt and change to the directory where Web Start is installed.

   On Windows, the default directory is `C:\Program Files\IBM\Java50\jre\bin`. On Linux, the default directory is *install_dir*/jre/*platform*/bin.
2. Enter the following command:

   ```
   javaws http://TEPS_host_name:1920///cnp/kdh/lib/tep.jnlp (Windows)
   ./javaws http://TEPS_host_name:1920///cnp/kdh/lib/tep.jnlp (Linux)
   ```

   where *TEPS_host_name* is the fully qualified host name of the computer where the Tivoli Enterprise Portal Server is installed (for example, `myteps.itmlab.company.com`).

Web Start downloads and launches the desktop client.

Manually creating a shortcut for the Web Start client

On Windows, the Web Start executable file for the default Java JVM is copied to the Windows\System32 directory. When you let Web Start create a short cut for launching the desktop client, it uses the file in the System32 directory as the target. If the default JVM is not IBM Java 1.5, the shortcut will not launch the desktop client. You must create a shortcut manually.

To create a shortcut to use to launch the desktop client using Web Start:

1. Right-click on the Windows desktop and select **New → Shortcut** from the popup menu.
2. In the Create Shortcut window, type the following path or click **Browse** and navigate to the executable as shown:

   ```
   C:\Program Files\IBM\Java50\jre\bin\javaws.exe
   ```
3. Click **Next** and type a name for the shortcut in the Select a Title for the Program window. For example:

   ```
   ITM Web Start client
   ```
4. Click **Finish**.

   The shortcut appears on your desktop.

Installing product maintenance

The installation procedures in this chapter contain instructions for new installations. Follow the same procedures for upgrading or updating an existing installation. Fix packs (that is, product maintenance) use the same installer as a pristine installation except that in this case, the installer runs only on a system that is already installed and configured. This is true when installing both generally available fix packs and interim fixes.

An *upgrade* is an installation that replaces a previous release or fix pack level of the product or component with a later release or fix pack level. An *update* is a modification to an existing installation at the same release or fix pack level.

Note that, when you upgrade or update an installation, configuration windows for components that are already configured might not be displayed. Skip those steps that do not apply; there is no need to rerun any configuration step unless specifically called for in the fix pack's documentation.

The instructions sometimes reference the default path for the installation of IBM Tivoli Monitoring. If you did not install the product in the default path, you must specify the correct path whenever the upgrade procedures require a path specification.

Chapter 10. Deploying monitoring agents across your environment

IBM Tivoli Monitoring provides the ability to deploy monitoring agents from a central location, the monitoring server. Just as there are two types of monitoring agents, there are two types of agent deployment:

- OS agent deployment from the installation image or using the **tacmd createNode** command
- Non-OS agent (such as the DB2 for Linux, UNIX, and Windows agent) deployment using the Tivoli Enterprise Portal GUI (for other non-OS agents) or the **tacmd addSystem** command

Table 59 describes the steps required to set up and manage remote agent deployment:

Table 59. Remote agent deployment tasks

Goal	Where to find this information
Create and populate the agent deploy depot with installable agent images.	“Populating your agent depot”
View and change the contents of the agent depot.	“Managing your agent depot” on page 328
Use one agent depot for all the monitoring servers in your monitoring environment.	“Sharing an agent depot across your environment” on page 328
Deploy an OS agent.	“Deploying OS agents” on page 329
Deploy a non-OS agent.	“Deploying non-OS agents” on page 331
Deploy a group of agents simultaneously.	“Bulk agent deployment” on page 336

You can also use the remote agent deployment function to configure deployed agents and install maintenance on your agents. For information, see the *IBM Tivoli Monitoring: Administrator's Guide*. See the *IBM Tivoli Monitoring: Command Reference* for commands that you can use to perform these tasks.

Important: Run the **tacmd login** command before executing commands from the tacmd library. This requirement does not apply to the addBundles command. Run the **tacmd logoff** command after you finish using the tacmd command library."

Populating your agent depot

The *agent depot* is an installation directory on the monitoring server from which you deploy agents and maintenance packages across your environment. Before you can deploy any agents from a monitoring server, you must first populate the agent depot with bundles. A *bundle* is the agent installation image and any prerequisites.

When you add a bundle to the agent depot, you need to add the bundle that supports the operating system to which you want to deploy the bundle. For example, if you want to deploy a DB2 for Linux, UNIX, and Windows agent bundle to a computer running HP-UX, add the HP-UX-specific agent bundle to the depot. If your depot directory is on Windows and you want to deploy the DB2 for Linux, UNIX, and Windows agent to HP-UX, load the HP-UX bundle from the DB2 for Linux, UNIX, and Windows agent installation media for HP-UX. (If you are installing from different media for each platform type, for example, Windows, AIX and Solaris, HP-UX, Linux, you need to add the bundle from the specific platform media for the component.)

You can have an agent depot on each monitoring server in your environment or share an agent depot, as described in “Sharing an agent depot across your environment” on page 328. If you choose to have an agent depot for each monitoring server, you can customize the agent depot based on the types of bundles that you want to deploy and manage from that monitoring server. For example, if you have a monitoring server dedicated to monitoring the DB2 for Linux, UNIX, and Windows agents in your environment, populate the depot with DB2-related agent bundles. If you deploy an agent from a remote monitoring server, you must have a agent bundle in the depot available to the monitoring server.

Note: Agent depots cannot be located on a z/OS monitoring server.

There are two methods to populate the agent depot:

- "Populating the agent depot from the installation image"
- "Populating the agent depot with the tacmd addBundles command" on page 327

Populating the agent depot from the installation image

Use the following sections to populate your agent depot from the installation image:

- "Windows: Populating the agent depot during installation"
- "Linux and UNIX: Populating the agent depot during installation"

You can use the installation image to populate the agent depot only when you are populating the depot with bundles for the same operating system as your monitoring server. For example, you can use the installation image to add a bundle for a Windows agent to a Windows monitoring server, but you cannot use the Linux installation image to add a Linux bundle to a Windows monitoring server. If you need to add bundles for operating systems other than that used by your monitoring server, use the **tacmd addBundles** command, as described in "Populating the agent depot with the tacmd addBundles command" on page 327.

Attention: Load only Tivoli-provided product agent bundles into the IBM Tivoli Monitoring deployment depot. User-provided or customized bundles are not supported. Use only Tivoli-provided tacmd commands to process bundles and to execute agent deployments. Manual manipulation of the depot directory structure or the bundles and files within it is not supported and may void your warranty.

Windows: Populating the agent depot during installation

Use the following steps to populate the agent depot during installation:

1. Launch the installation wizard by double-clicking the setup.exe file in the \Windows subdirectory of the installation image.
2. Select **Next** on the Welcome window.
3. Click **Next** on the Select Features window without making any changes.
4. On the Agent Deployment window, select the agents that you want to add to the depot and click **Next**.
5. Review the installation summary and click **Next** to begin the installation.

   After the agents are added to the agent depot, a configuration window (called the Setup Type window) is displayed.
6. Clear all selected components. You have already configured all components on this computer and do not need to reconfigure any now. Click **Next**.
7. Click **Finish** to complete the installation.
8. Click **Finish** on the Maintenance Complete window.

Linux and UNIX: Populating the agent depot during installation

Use the following steps to populate the agent depot from the Linux or UNIX installation image:

1. In the directory where you extracted the installation files, run the following command:

   ```
   ./install.sh
   ```
2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default directory (`/opt/IBM/ITM`). If you want to use a different installation directory, type the full path to that directory and press Enter.

3. If the directory you specified does not exist, you are asked whether to create it. Type y to create this directory.
4. The following prompt is displayed:

   ```
   Select one of the following:
   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   3) Install TEMS support for remote seeding
   4) Exit install.
   Please enter a valid number:
   ```

   Type 2 to start the installation and press Enter.

   The end user license agreement is displayed. Press Enter to read through the agreement.
5. Type 1 to accept the agreement and press Enter.
6. Type the number that corresponds to the agent or agents that you want to add to the agent depot and press Enter. If you are going to add more than one agent, use a comma (,) to separate the numbers.

   To select all available agents, type all.

   You can select multiple agents with consecutive corresponding numbers by typing the first and last numbers for the agents, separated by a hyphen (-). For example, to add all of the agents between 8 and 12, type 8-12.

   To clear an agent that you previously selected, type the number for the agent again.

   Note: Use the following keys to navigate the list of agents:

   U Moves up a line in the list.

   D Moves down a line in the list.

   F Moves forward one page in the list.

   B Moves back one page in the list.
7. When you have specified all the agents that you want to add to the agent depot, type E and press Enter to exit.

Populating the agent depot with the tacmd addBundles command

To populate the agent depot using the **tacmd addBundles** command, run the following command:

```
tacmd addBundles [-i IMAGE_PATH]
                 [-t PRODUCT_CODE]
                 [-p OPERATING_SYSTEM]
                 [-v VERSION]
                 [-n]
                 [-f]
```

For the full syntax, including parameter descriptions, see the *IBM Tivoli Monitoring: Command Reference*.

Examples:

- The following example copies every agent bundle, including its prerequisites, into the agent depot on a UNIX computer from the installation media (CD image) located at /mnt/cdrom/:

  ```
  tacmd addbundles -i /mnt/cdrom/unix
  ```
- The following example copies all agent bundles for the Oracle agent into the agent depot on a UNIX computer from the installation media (CD image) located at /mnt/cdrom/:

  ```
  tacmd addbundles -i /mnt/cdrom/unix -t or
  ```
- The following example copies all agent bundles for the Oracle agent into the agent depot on a Windows computer from the installation media (CD image) located at D:\WINDOWS\Deploy:

  ```
  tacmd addbundles -i D:\WINDOWS\Deploy -t or
  ```

- The following example copies the agent bundle for the Oracle agent that runs on the AIX version 5.1.3 operating system into the agent depot on a UNIX computer from the installation media (CD image) located at /mnt/cdrom/:

  ```
  tacmd addbundles -i /mnt/cdrom/unix -t or -p aix513
  ```

By default, the agent depot is located in the *itm_installdir*/CMS/depot directory on Windows and *itm_installdir*/tables/*tems_name*/depot directory on UNIX. The **tacmd addBundles** command puts the agent bundle in that location unless another location is defined in the monitoring server configuration file for DEPOTHOME.

If you want to change this location, do the following before you run the **tacmd addBundles** command:

1. Open the KBBENV monitoring server configuration file located in the *itm_installdir*\CMS directory on Windows and the *itm_installdir*/tables/*tems_name* directory on Linux and UNIX.
2. Locate the DEPOTHOME variable. If it does not exist, add it to the file.
3. Type the path to the directory that you want to use for the agent depot.
4. Save and close the file.
5. On UNIX or Linux only, add the same variable and location to the kbbenv.ini file located in *itm_installdir*/config/kbbenv.ini.

   If you do not add the variable to the kbbenv.ini file, it will be deleted from the KBBENV file the next time the monitoring server is reconfigured.

Managing your agent depot

Use the following commands to manage your agent depot:

Table 60. Agent depot management commands

Command	Description
tacmd listbundles	Lists the details for one or more bundles available to be added to the local agent depot.
tacmd removebundles	Deletes one or more bundles from the local agent depot.
tacmd viewdepot	Lists the types of bundles available in either the local or remote agent depot.

See the *IBM Tivoli Monitoring: Command Reference* for the full syntax of these commands.

Note: Only Tivoli-provided product agent bundles should be loaded into the IBM Tivoli Monitoring deployment depot. User-provided or customized bundles are not supported. Use only Tivoli provided tacmd commands to process bundles and to execute agent deployments. Manual manipulation of the depot directory structure or the bundles and files within it is not supported and may void your warranty.

Sharing an agent depot across your environment

If your monitoring environment includes multiple monitoring servers (a hub monitoring server and remote monitoring servers), you can put your agent depot in a central location, such as a shared file system, and access the depot from all of the monitoring servers.

After populating your agent depot with either of the methods described in “Populating your agent depot” on page 325, use the following steps to share the agent depot:

1. Open the KBBENV monitoring server configuration file located in the *itm_installdir*\CMS directory on Windows and the *itm_installdir*/tables/*tems_name* directory on Linux and UNIX.

2. Locate the DEPOTHOME variable. By default, the agent depot is located in the *itm_installdir*/CMS/depot directory on Windows and *itm_installdir*/tables/*tems_name*/depot directory on UNIX.
3. Type the path to the shared agent depot for the DEPOTHOME variable.
4. Save and close the file.
5. On UNIX or Linux only, add the same variable and location to the kbbenv.ini file located in *itm_installdir*/config/kbbenv.ini.

   If you do not add the variable to the kbbenv.ini file, it will be deleted from the KBBENV file the next time the monitoring server is reconfigured.

If you are using a Windows monitoring server connecting to a depot on another Windows computer, you must set the service ID for the Windows monitoring server to "Administrator." Also, instead of specifying a mapped drive letter for the path to the depot directory, use the UNC path (such as \\server\share).

Use the following steps to change the service ID:

1. From the Control Panel, double-click **Administrative Tools**.
2. Double-click **Services**.
3. Right-click **Tivoli Enterprise Monitoring Svcs** and click **Properties**.
4. On the **Log On** tab, select **This Account**.
5. Type `Administrator` in the **This Account** field.
6. Type the password for the administrator in the **Password** field. Confirm the password by typing it again in the **Confirm password** field.
7. Click **Enable**.

   If the Administrator user does not have Logon as a service right, you are prompted to add it.

Deploying OS agents

Before you can deploy any non-OS agent, you must first install an OS agent on the computer where you want the non-OS agent to be deployed. In addition to monitoring base OS performance, the OS agent also installs the required infrastructure for remote deployment and maintenance.

Notes:

1. On Windows: IBM Tivoli Monitoring does not support more than one OS agent installation on the same machine. You can not use different directories to install more than one OS agent on the same Windows machine.
2. Ensure that you have populated your agent depot, as described in “Populating your agent depot” on page 325, before attempting to deploy any agents.

You can install the OS agent locally, as described in “Installing monitoring agents” on page 253 or remotely using the **tacmd createNode** command.

The **tacmd createNode** command creates a directory on the target computer called the *node*. The OS and non-OS agents are deployed in this directory. Agent application support is also added at this time (see “Installing and enabling application support” on page 266).

The **tacmd createNode** command uses one of the following protocols to connect to the computers on which you want to install the OS agent:

- Server Message Block (SMB), used primarily for Windows servers
- Secure Shell (SSH), used primarily by UNIX servers, but also available on Windows

  Note: Only SSH version 2 is supported.
- Remote Execution (REXEC), used primarily by UNIX servers, but not very secure
- Remote Shell (RSH), used primarily by UNIX servers, but not very secure

You can specify a protocol to use; if you do not, the **tacmd createNode** command selects the appropriate protocol dynamically.

Requirements for the tacmd createNode command

Before you can use the **tacmd createNode** command to deploy OS agents, ensure the following requirements are met:

- The createNode command no longer has to be executed locally to the Tivoli Enterprise Monitoring Server.
- On both Windows, Linux, and UNIX, you must issue a **tacmd login** command prior to executing the **tacmd createNode** command.
- On Windows, the user ID that you specify using the -u parameter *must* have administrator privileges on the target computer. On UNIX and Linux, you must specify the "root" user ID using the -u parameter and the root password using the -p parameter for the **tacmd createNode** command to execute correctly. No other user ID may be specified.
- Any computer to which you want to deploy the OS agent must have a supported protocol installed.
- Security in your environment must be configured to permit createNode to pass through the firewall, using the protocol that you specify in the command parameters.
- On Windows computers:
  - SMB requires that the default, hidden, and administrative shares be available on the drive being accessed and on the drive that hosts the System temporary directory.
  - SMB signing is not supported when connecting using SMB. The computer to which you are deploying an OS agent cannot require SMB signing.
  - For Windows XP, disable Simple File Sharing. Simple File Sharing requires that all users authenticate with guest privileges, which createNode does not support. To disable Simple File Sharing, perform the following steps:
    1. Open the Windows Explorer.
    2. Click **Tools → Folder Options**.
    3. Click the **View** tab.
    4. Scroll through the list of settings to **Use Simple File Sharing**.
    5. Clear the check box next to **Use Simple File Sharing** and click **OK**.
  - For Windows XP computers with Service Pack 2, disable the Internet Connection Firewall.
  - For Windows XP computers, set Network Access Sharing and Security to "Classic - local users authenticate as themselves." Use the following steps:
    1. From the Control Panel, double-click **Administrative Tools**.
    2. Double-click **Local Security Policy**.
    3. Expand **Local Policies** and click **Security Options**.
    4. Right-click **Network access: Sharing and security for local accounts** and click **Properties**.
    5. Select **Classic - local users authenticate as themselves** from the list and click **OK**.
  - For all Windows computers, enable remote registry administration. (This is enabled by default.)
- On UNIX systems, if you are using the RSH protocol, run the **tacmd createNode** command as root on the monitoring server.
- If you are deploying the OS agent to a UNIX or Linux computer, that computer must have the ksh shell. Only the Korn shell is supported for the execution of the installation and runtime scripts.
- If you are using SSH V2 (for either Windows or UNIX), configure SSH on the target computers to permit the use of password authentication. To permit this, do the following:
  1. Edit the /etc/ss/sshd_config file on the target computer.
  2. Locate the following line:

     ```
     PasswordAuthentication no
     ```

3. Change the `no` to `yes` and save the file.
4. Restart the daemon.

Note: If you are using private key authentication in your environment, you do not need to set SSH to permit password authentication.

For more information see "Remote Execution and Access" on page 112.

Using the tacmd createNode command

To deploy an OS agent from the command-line interface, use **tacmd createNode** command.

For example, the following command deploys the UNIX OS monitoring agent on the server1.ibm.com computer in the /opt/IBM/ITM directory. The installation is done as the root user. The option property `EXECPREREQCHECK=Y` performs a prerequisite check on the agent. For more information, see "Prerequisite Checking for IBM Tivoli Monitoring agents" on page 59.

```
tacmd createNode -h server1.ibm.com -d /opt/IBM/ITM -u root -o EXECPREREQCHECK=Y
```

Important: Unless you specifically indicate otherwise, the agent that you deploy using this command assumes that the monitoring server to which it connects is the monitoring server from which you run the command. The agent also uses the default settings for the communications protocol (IP.PIPE for protocol type and 1918 for the port). To change these default values (especially if you are not using the IP.PIPE protocol), use the following property (specified with the **-p** parameter) when running the command: SERVER=[PROTOCOL://][HOST|IP][:PORT]. For example, `SERVER=IP.PIPE://server1.ibm.com:1918`.

Starting with IBM Tivoli Monitoring V6.2.3, the **tacmd createNode** command contains the optional `–k` group parameter. Passing the `-k` parameter when you remotely deploy an OS agent will execute the secureMain utility at the end of the deployment process to secure your IBM Tivoli Monitoring environment. For more information, see Appendix G, "Securing your IBM Tivoli Monitoring installation on Linux or UNIX," on page 851. For the full syntax, including parameter descriptions, see the *IBM Tivoli Monitoring: Command Reference*.

Deploying non-OS agents

You can deploy non-OS agents through the Tivoli Enterprise Portal or from the command-line.

Notes:

1. The deployment and configuration of agents varies depending on the specific agent. The following procedures provide generic deployment information. For the exact values required for your agent, see the configuration information in the user's guide for the agent.
2. Ensure that you have populated your agent depot, as described in "Populating your agent depot" on page 325, before attempting to deploy any agents.
3. You must have already installed or deployed an OS agent on the computer where you are now deploying the non-OS agent and the agent must be running.
4. You can only deploy the Tivoli Performance Analyzer through the installation media, and not through the Tivoli Enterprise Portal or the command-line.

Deploying through the portal

Before you deploy an agent through the Tivoli Enterprise Portal, application support for that agent must be installed on the portal server (see "Installing and enabling application support" on page 266).

Use the following steps to deploy an agent through the portal GUI:

1. Open the Tivoli Enterprise Portal.
2. In the Navigation tree, navigate to the computer where you want to deploy the agent.

3. Right-click the computer and click **Add Managed System**.
4. Select the agent that you want to deploy and click **OK**.
5. Complete the configuration fields required for the agent. For information about these fields, see the configuration documentation for the agent that you are deploying.
6. Click **Finish**.
7. If the computer where you are deploying the agent already has a version of that agent installed, you can stop the deployment, add a new instance of the agent, if possible, or reconfigure the existing agent.

   A message will tell you when the deployment finishes successfully.

Deploying through the command-line

To deploy non-OS agents from the command-line, use the **tacmd addSystem** command. See the *IBM Tivoli Monitoring: Command Reference* for the full syntax of this command, including parameter descriptions. You can run the **cinfo** command (UNIX) or the **kincinfo -i** command (Windows) to list the product codes for agents installed on the current computer.

For example, the following command deploys the Tivoli Universal Agent (type `um`) to the stone.ibm.com computer and specifies the UA.CONFIG property:

```
tacmd addSystem -t um -n stone.ibm.com:LZ -p UA.CONFIG="file_unix.mdl"
```

Each agent bundle has its own unique configuration parameters that you may need to specify using this command. If you have installed the agent bundle that you want to deploy to the deployment depot, you can view the configuration parameters by running the following command from the monitoring server where that agent bundle is installed:

```
tacmd describeSystemType -t pc -p platform
```

An agent of the same type and platform must be deployed into the depot available to the monitoring server from which the command is run. You can also get more information about agent-specific parameters in the agent user's guide for the agent that you want to deploy.

Note: The tacmd command has been updated to support two new features, the asynchronous remote deployment (which allows you to request the deployment of another remote agent, even if the previously deployed agent has not initialized fully) and the grouping of agents to be remotely deployed. The deployment commands have been modified to accept two new grouping parameters: the –g parameter lets you specify a deployment group and the –b parameter lets you specify a bundle group. For detailed information, including usage examples, see the *IBM Tivoli Monitoring: Command Reference*.

With asynchronous remote deployment, CLI requests are queued, and the tacmd command returns control immediately to the process that invoked them along with a transaction ID that can be used to monitor the status of the request via either the **tacmd getDeployStatus** command or the Deployment Status Summary By Transaction workspace in the Tivoli Enterprise Portal. Asynchronous remote agent deployment applies both to agents started via the tacmd command and to those started using the portal client. Workspace reports regarding asynchronous agent deployment are also available; for information, see the IBM Tivoli Monitoring online help.

A new capability allows you to perform prerequisite checking for agents before carrying out an installation. The two mechanisms available are a manually executed stand-alone prerequisite scanner, or a remote prerequisite scanner facility that extends the capabilities of IBM Tivoli Monitoring's remote deployment component. For more information, see “Prerequisite Checking for IBM Tivoli Monitoring agents” on page 59.

Deploying an instance of the Tivoli Universal Agent

You can use both of the deployment methods described in the previous sections to deploy a Tivoli Universal Agent instance.

Before you can deploy a Tivoli Universal Agent instance, you must create directory named UACONFIG at the top level of the deploy depot, at the same level as the PACKAGES directory. (For example, if you installed your monitoring server at /opt/IBM/ITM, and named it hub_core, then your depot directory is /opt/IBM/ITM/tables/hub_core/depot and that is where you should create the UACONFIG directory.) If you are deploying a Script Data Provider, you must also create a UASCRIPT directory at the same level as the UACONFIG directory.

After you create the directories, you must copy your . mdl files to the UACONFIG directory, and any script metafiles to the UASCRIPT directory. Use the agent depot on the monitoring server to which the Tivoli Universal Agent connects.

To deploy the Tivoli Universal Agent, add the following parameter to the addSystem command:

```
-p UA.CONFIG=metafile.mdl
```

where *metafile* is the name of the .mdl file that you want the Tivoli Universal Agent to use.

To deploy the Tivoli Universal Agent and its referenced script, add the following parameter to the addSystem command:

```
-p UA.CONFIG=metafile.mdl. UA.SCRIPT=script-filename
```

where *metafile* is the name of the .mdl file and *script-filename* is the name of the script that the Tivoli Universal Agent uses.

If a metafile belonging to a nondefault type of Data Provider type is remotely deployed (the four default Data Providers are API, Socket, File, and Script), there is no automatic mechanism to activate the appropriate Data Provider. For example, if you deploy an ODBC metafile to a remote Tivoli Universal Agent and that agent has not already been configured to start the ODBC DP, the Data Provider configuration will not happen automatically as a result of the deployment. You must manually configure the ODBC Data Provider on that Tivoli Universal Agent for the metafile to be usable. For instructions on configuring the Data Provider, see the *IBM Tivoli Universal Agent User's Guide*.

Note to Linux/UNIX users: The instance name is converted to uppercase when the agent registers with the Tivoli Enterprise Monitoring Server. If the name you specify contains lowercase letters, this will cause problems when stopping or starting the agent remotely, as the instance name you specified will not match the actual instance name. Thus, the universal agent, when used on Linux or UNIX, must be named with all uppercase letters for the Tivoli Enterprise Portal to start or stop the agent.

Deploying Netcool/OMNIbus System Service Monitor (SSM) agents

Deploying Netcool/OMNIbus SSM agents is very similar to deploying IBM Tivoli Monitoring OS agents: You first need to populate your agent depot with SSM bundles. Once this is done, you can use the **tacmd createNode** command to deploy the SSM agent; see the *IBM Tivoli Monitoring: Command Reference* for complete information on this command.

To populate the SSM bundles, download the SSM image from the Passport Advantage Web site, http://www-01.ibm.com/software/howtobuy/passportadvantage/, and use the **tacmd addBundles** command:

```
tacmd addBundles  -t ssm  -p li6213 -i c:\SSM40_Bundle
```

This example adds the SSM bundle for the Linux platform to the depot.

Once you have populated the depot with SSM bundles, you can use the **tacmd viewDepot** command to see the bundles. Each SSM bundle consists of the installer files and the corresponding descriptor file.

The tacmd commands for installing, patching, starting, stopping, and configuring the Tivoli Monitoring agents have been expanded to support operations with Netcool SSM agents as well.

Installing an SSM agent

Use the **tacmd createNode** command to install an SSM agent. The mechanism for pushing the SSM image to the agent machine is the same as that used to deploy a Tivoli Monitoring OS agent. Example:

```
tacmd createNode -h smb://achan1.raleigh.ibm.com -t SSM -u achan -d c:\SSMAgent\ssm -p SNMPPORT=16002
```

This example deploys an SSM agent to the Windows machine achan1 using the Server Message Block. The installation directory for the agent is specified by the –d option, and the SNMP port to be used by the SSM agent is specified using the –p SNMPPORT property.

Uninstalling an SSM agent

To uninstall an SSM agent, use the **tacmd removeSystem** command to remove it. Example:

```
tacmd removeSystem -h smb://achan1.raleigh.ibm.com -t SSM -u achan-d c:\SSMAgent\ssm
```

The above command uninstalls the SSM agent on Windows machine achan1.

Installing an SSM patch

Use the **tacmd updateAgent** command to patch an SSM agent.

1. Just like installing an SSM image, first run the **tacmd addBundles** command to add the patch images to the IBM Tivoli Monitoring depot.
2. Once this is done, run the **tacmd updateAgent** command to install the patch on the agent. The transfer of the patch image to the agent is done through the Tivoli Enterprise Monitoring Server's HTTP server.

Here is an example of an HTTP request for agent `http://TEMS_SERVER:1920//ms/kdh/lib/depot/PACKAGES/WINNT\ssm\040000001\ssm40-fix pack1-win32-x86.exe`:

```
tacmd updateAgent -h achan1.raleigh.ibm.com -t SSM -l "fix pack 1"
```

This installs SSM 4.0 agent fix pack 1 to the achan1 agent machine. The agent must be up and running; if not, the request fails.

To start the agent, use the **tacmd startAgent** command, as explained in “Starting an SSM agent.”

Uninstalling an SSM patch

To uninstall an SSM patch, you also use the **tacmd removeSystem** command, with the –l option to remove the patch. Example:

```
tacmd removeSystem -h achan1.raleigh.ibm.com -t SSM -l "fix pack 1" -p SNMPPORT=16002
```

Starting an SSM agent

To start an SSM agent from ITM, use the **tacmd startAgent** command. Example:

```
tacmd startAgent -h achan1.raleigh.ibm.com -p SNMPPORT=16002
```

Stopping an SSM agent

To stop an SSM agent from ITM, use the **tacmd stopAgent** command. Example:

```
tacmd stopAgent -h achan1.raleigh.ibm.com -p SNMPPORT=16002
```

Restarting an SSM agent

To restart an SSM agent from ITM, use the **tacmd restartAgent** command. Example:

```
tacmd restartAgent -h achan1.raleigh.ibm.com -p SNMPPORT=16002
```

Configuring an SSM agent

There are several configuration functions you can perform with SSM agents using the **tacmd configureSystem** command.

- Change the logging file and level on the agent and perform a silent reboot of the agent. Example:

  ```
  tacmd configureSystem -h achan1.raleigh.ibm.com
    -p SNMPPORT=16002 LogFile=test.log LogLevel=debug -r
  ```

- Transfer files, including executable files, to the agent machine. Example:

  ```
  tacmd configureSystem -h achan1.raleigh.ibm.com -p SNMPPORT=16002 -l test1.exe  test2.exe
  ```

- Transfer configuration files to the agent machine. Example:

  ```
  tacmd configureSystem -h achan1.raleigh.ibm.com -p SNMPPORT=16002 -c test1.cfg test2.cfg
  ```

Notes:

1. The files specified in the configfile (–c) option or the filelist (–l) option must be stored in the `ssmconfig` subdirectory of the depot directory.
   - On Windows, if your depot directory is `c:\IBM\ITM\CMS\depot`, you must put the files specified in the configfile option or the filelist option into the `c:\IBM\ITM\CMS\depot\ssmconfig` directory.
   - On Linux or UNIX, if your depot directory is `/opt/IBM/ITM/tables/TEMS_NAME/depot`, put the files into the `/opt/IBM/ITM/tables/TEMS_NAME/depot/ssmconfig` directory.
2. The files are pulled from the agent through the Tivoli Enterprise Monitoring Server's HTTP server. Here is an example of an HTTP request:

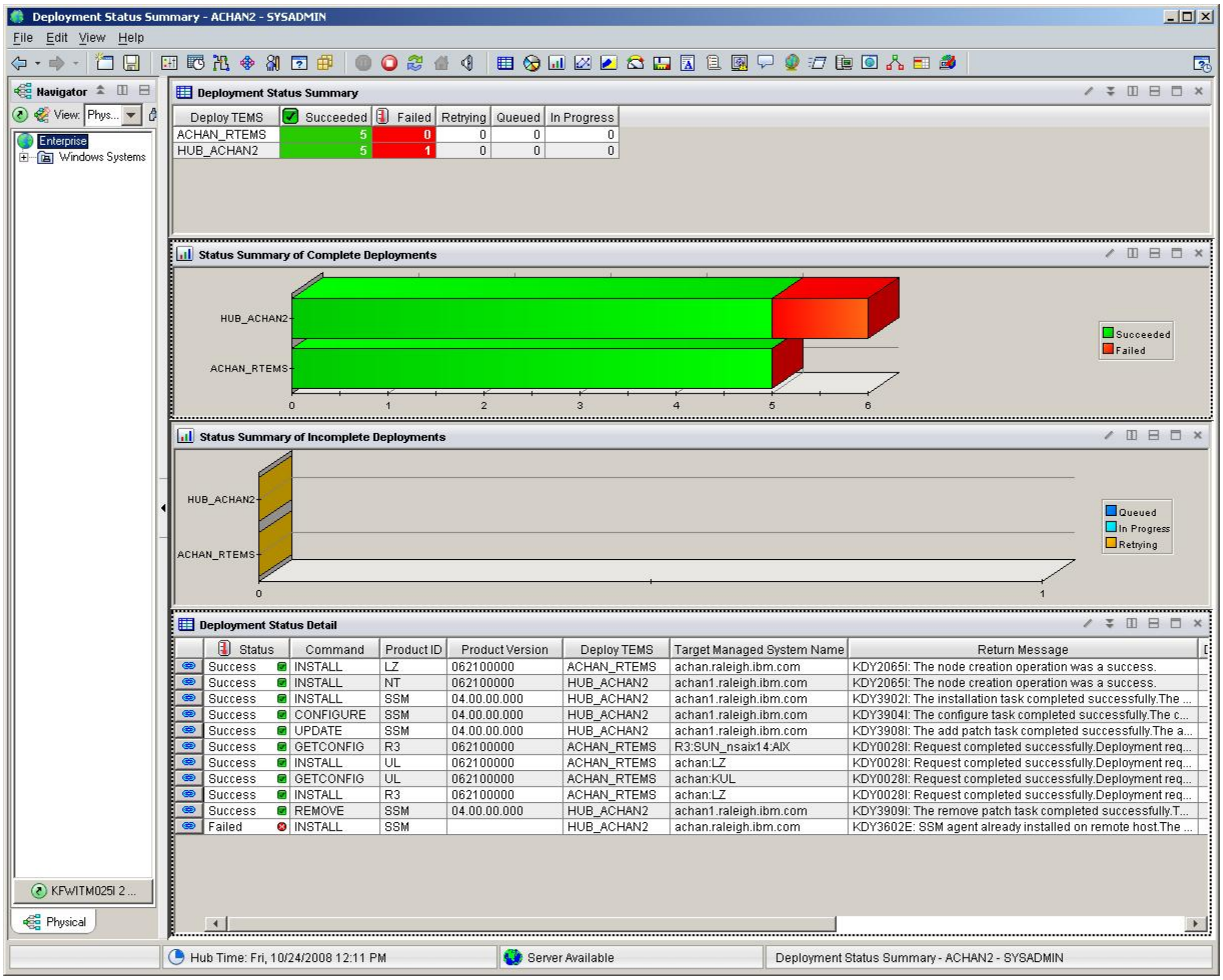
   ```
   http://tems_host_name:1920//ms/kdh/lib/depot/ssmconfig/test1.cfg
   ```

Bulk deployment of NetCool SSM agents

The bulk deployment of SSM agents is very similar to the bulk deployment of IBM Tivoli Monitoring agents. See “Bulk agent deployment” on page 336 for more information.

Query deployment status of Netcool SSM agents

The same tools for checking the status of your IBM Tivoli Monitoring agent deployments also apply to SSM agents; see “Deploy status” on page 337 for more information. Figure 93 on page 336 shows an example using the Tivoli Enterprise Portal's Deployment Status Summary workspace.


Deployment Status Summary - ACHAN2 - SYSADMIN
File Edit View Help
Navigator
View: Phys...
Enterprise
Windows Systems
KFWITM025I 2 ...
Physical
Status Summary of Complete Deployments
HUB_ACHAN2
ACHAN_RTEMS
0
1
2
3
4
5
6
Succeeded
Failed
Status Summary of Incomplete Deployments
HUB_ACHAN2
ACHAN_RTEMS
0
1
Queued
In Progress
Retrying
Hub Time: Fri, 10/24/2008 12:11 PM
Server Available
Deployment Status Summary - ACHAN2 - SYSADMIN

Deployment Status Summary

Deploy TEMS	Succeeded	Failed	Retrying	Queued	In Progress
ACHAN_RTEMS	5	0	0	0	0
HUB_ACHAN2	5	1	0	0	0

Deployment Status Detail

Status	Command	Product ID	Product Version	Deploy TEMS	Target Managed System Name	Return Message
Success	INSTALL	LZ	062100000	ACHAN_RTEMS	achan.raleigh.ibm.com	KDY2065I: The node creation operation was a success.
Success	INSTALL	NT	062100000	HUB_ACHAN2	achan1.raleigh.ibm.com	KDY2065I: The node creation operation was a success.
Success	INSTALL	SSM	04.00.00.000	HUB_ACHAN2	achan1.raleigh.ibm.com	KDY3902I: The installation task completed successfully.The ...
Success	CONFIGURE	SSM	04.00.00.000	HUB_ACHAN2	achan1.raleigh.ibm.com	KDY3904I: The configure task completed successfully.The c...
Success	UPDATE	SSM	04.00.00.000	HUB_ACHAN2	achan1.raleigh.ibm.com	KDY3908I: The add patch task completed successfully.The a...
Success	GETCONFIG	R3	062100000	ACHAN_RTEMS	R3:SUN_nsaix14:AIX	KDY0028I: Request completed successfully.Deployment req...
Success	INSTALL	UL	062100000	ACHAN_RTEMS	achan:LZ	KDY0028I: Request completed successfully.Deployment req...
Success	GETCONFIG	UL	062100000	ACHAN_RTEMS	achan:KUL	KDY0028I: Request completed successfully.Deployment req...
Success	INSTALL	R3	062100000	ACHAN_RTEMS	achan:LZ	KDY0028I: Request completed successfully.Deployment req...
Success	REMOVE	SSM	04.00.00.000	HUB_ACHAN2	achan1.raleigh.ibm.com	KDY3909I: The remove patch task completed successfully.T...
Failed	INSTALL	SSM		HUB_ACHAN2	achan.raleigh.ibm.com	KDY3602E: SSM agent already installed on remote host.The ...

Figure 93. Deployment Status Summary workspace showing the status of SSM deployments

Bulk agent deployment

Bulk deployment of IBM Tivoli Monitoring agents into moderate to large environments is made possible by a set of capabilities and best practices coming together to achieve rapid rollout with ease. The capabilities that enable bulk deployment in larger environments fall into two main categories:

- The components involved in deployment.
- The processing model associated with deployment.

The best practices described in this section are recommended or common ways that you can use these capabilities to successfully deploy agents in larger environments.

Deployment processing model

The processing model used during deployment is critical to the success and timeliness of deployments. The two critical aspects of deployment processing are:

- Asynchronous transaction submission on the front end.
- Parallel transaction handling on the back end.

The asynchronous nature of the IBM Tivoli Monitoring deployment model allows you to submit deployment transactions into the environment quickly without having to wait for the deployment request to complete.

Instead of the deployment command returning after some period of time with a message saying that the deployment completed with a success or a failure, the CLI returns immediately with a transaction identifier (ID). This frees your user environment (be it the Tivoli Enterprise Portal, the command-line, or an automation script) to continue normal processing:

```
C:\>tacmd createnode -g Targets -b Agents

KUICCN022I: Request has been successfully queued to the deploy controller.
The transaction ID is 1224009912765000000000041, use the getDeployStatus CLI to view the status.
```

You can then monitor the status of your deployment requests using the command interface or Tivoli Enterprise Portal workspaces or situations to check for failures.

Once a set of deployment transactions has been submitted into the IBM Tivoli Monitoring infrastructure, they are routed to the appropriate remote Tivoli Enterprise Monitoring Server for processing. Each remote monitoring server can process deployment transactions independently of every other monitoring server, whether remote or hub. This allows for a large degree of parallelism, which increases with each remote monitoring server added to your environment. Additionally, each remote Tivoli Enterprise Monitoring Server can process multiple deployment transactions concurrently, further increasing the level of parallelism applied to bulk deployments.

The default number of concurrent deployments per monitoring server is 10, but this is configurable using the DEPLOYTHREADPOOLSIZE setting in the Tivoli Enterprise Monitoring Server configuration file (KBBENV).

Figure 94 illustrates the processing model for bulk deployment.

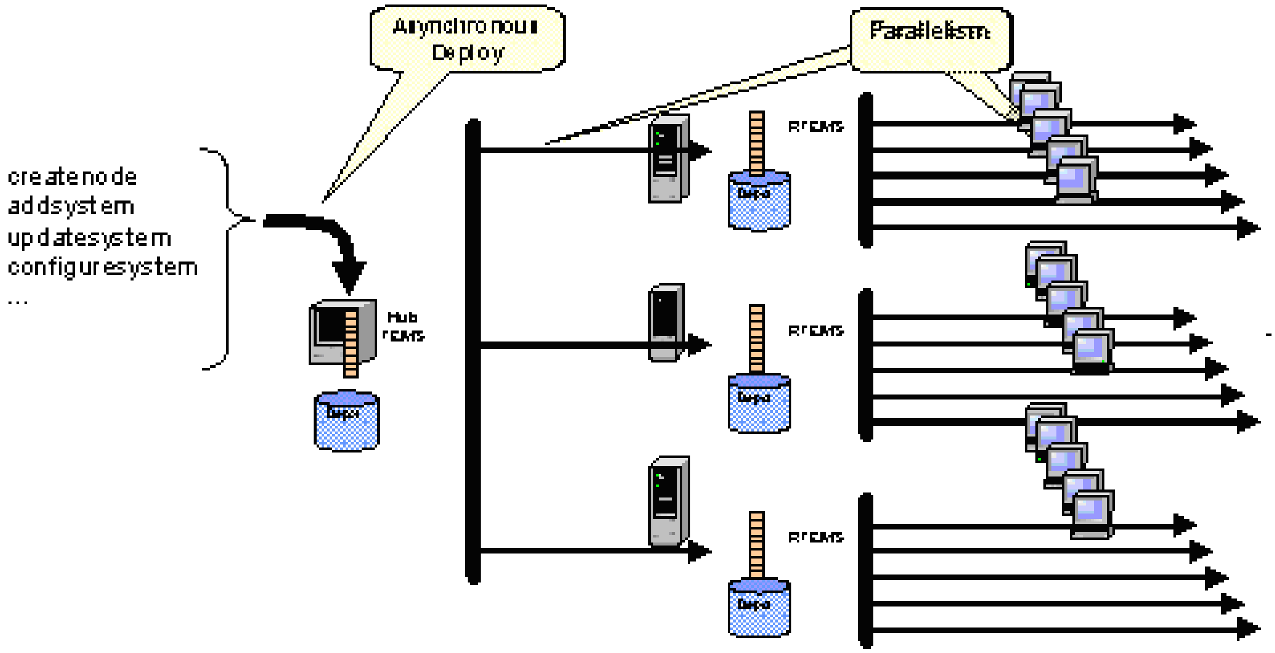


Figure 94. Bulk deployment processing model

Deploy status

Because deployment transactions are asynchronous (that is, they run in the background), monitoring the status of deployments is an important activity to ensure they complete successfully. There are three mechanisms for monitoring deployment status, as follows.

- The **tacmd getDeployStatus** CLI command.

For more information about this command, see the *IBM Tivoli Monitoring: Command Reference*.

- The Tivoli Enterprise Portal's Deployment Status workspaces.
- Deployment attribute groups, and situations created using the Tivoli Enterprise Portal that are based on these groups.

Using these features, you can display and monitor the progress of deployments running in your environment. You can track the status of your deployments either by individual transaction or by group transaction. A group transaction represents a deployment using groups of agents deployed to groups of targets; this is discussed in the succeeding sections. Each transaction possesses a status that indicates the progress the transaction has made in the deployment process. The valid status values for a deployment transaction are listed below.

Status Description

Pending
The transaction is waiting in the queue to begin processing.

In-Progress
The transaction has begun the deployment process but has not yet finished.

Success
The transaction has completed successfully.

Failed Retrying
The transaction experienced a recoverable error during the deployment process and will be restarted periodically at the point of failure for a predefined number of iterations. The number of retry iterations is defined by configurable property ERRORRETRYLIMIT. The time between each successive attempt is defined by the RETRYTIMEINTERVAL property. Both of these properties can be set in the monitoring server's configuration file, KBBENV, and take effect when the monitoring server is restarted. The default value for ERRORRETRYLIMIT is 3 retries, and the default value for RETRYTIMEINTERVAL is 7 minutes. These default values are based on a typical bulk deployment scenario using the createNode command. A smaller RETRYTIMEINTERVAL value (for example, 20 seconds) might be better suited if you perform more bulk upgrade scenarios than deployments.

If an error persists after all retry iterations are exhausted, the transaction moves to Failed status.

Failed The transaction has completed with an unrecoverable failure, or the number of retry attempts has been exhausted. Consult the status message included with the transaction status for more information about the failure.

Deployment Status workspaces

There are six workspaces related to bulk deployment. Three can be accessed directly from the Workspaces menu of the Navigator tree's Enterprise item. The remaining three can be accessed by selecting links. These workspaces are:

- Deploy Depot Package List
- Deployment Status Summary
- Deployment Status Summary by Transaction ID
- Deployment Status Summary by Product
- Deployment Status Summary by Deploy Group
- Installation Logs

For more information about these workspaces, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* or the Tivoli Enterprise Portal online help.

Deployment attribute groups and situations

The hub Tivoli Enterprise Monitoring Server provides two attribute groups that you can use to monitor the status of bulk deployments within your IBM Tivoli Monitoring environment: Deploy Status and Deploy Summary. You can use these attribute groups to create situations that notify you in the event of a failure or exceptional circumstance related to a deployment. For more information about these deployment status attribute groups, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* or the Tivoli Enterprise Portal online help.

IBM Tivoli Monitoring provides two deployment situations that monitor the Deploy Status attribute group, which you can use directly or as samples for creating your own situations. These situations are as follows.

Situation
: **Description**

Deploy_Failed
: Triggers a critical event when a deployment transaction fails.

Deploy_Retrying
: Triggers a minor event when a deployment transaction enters the failed_retry state.

Organizing deployments using groups

There are three primary components that come into play when performing bulk deployments. These are 1) the components being deployed, 2) the recipients (*targets*) of the deployment, and 3) properties that specify, clarify, or alter key details used during the deployment. Planning and organizing these components in a manner that reflects the current condition of your environment as well as the desired outcome for your environment will result in successful bulk deployments.

Organization of deployment targets and deployable components is achieved using *groups*. There are two types of groups used in bulk deployment, as follows.

Type Function

Deploy groups
: Organize deployment targets, which are the systems or agents to which a deployment is targeted.

Bundle groups
: Organize the deployable components that will be used during deployment operations.

In addition to the obvious role these groups play in organizing the participants of deployment, the grouping facilities perform another very important role: Groups and group members can also hold and organize properties that the deployment process uses; this is discussed in the following sections.

Deploy and Bundle groups are created using tacmd commands. For detailed information about creating such groups, see the *IBM Tivoli Monitoring: Command Reference*.

Deploy groups

Deploy groups organize the targets of deployment operations. Deploy targets are specified as members within the deploy group. These deploy targets can represent unmanaged systems, which could be targets of a createNode deployment operation. Deploy targets can also represent managed systems where a Tivoli Monitoring or Netcool System Service Monitor (SSM) agent is already installed. Both of these types of deploy targets can be added as members to deploy groups. This dual representation of targets within deploy groups allows a group to specify:

- Targets of an initial OS agent or SSM deployment using the **tacmd createNode** command.
- Targets of an application agent deployment using the **tacmd addSystem** command.
- Targets of an agent or SSM upgrade using the **tacmd updateAgent** command.
- Targets of an agent or SSM configuration deployment using the **tacmd configureSystem** command.

Once you create the necessary deploy groups to represent logical groupings of systems within your enterprise, these groups can be used repeatedly during the lifecycle of your monitoring infrastructure to deploy, update, start, stop, configure, and eventually remove agents.

Best practices: There are a number of best practices associated with deploy groups. The following are suggested.

- Segment your systems by geographic location. In this case, it is likely that you would want all agents deployed to a deploy group to connect with the same primary Tivoli Enterprise Monitoring Server.
- Segment your systems according to their role within the enterprise. You might create a group of database servers, a group of application servers, a group of Web servers. This allows you to deploy and administer agents to the systems within a group the same way but differently from how you might deploy and administer agents in a different group.
- Segment your systems according to the monitoring infrastructure being used. You might create a set of groups for monitoring using Tivoli Monitoring agents and another group or set of groups for monitoring using SSM agents.
- Segment your systems into smaller groups representing a percentage of total systems. This allows you to perform deployments into more easily consumable sets of activities.

Bundle groups

Bundle groups organize and represent the components that you deploy. These include IBM Tivoli Monitoring agents and Netcool SSM agents. Deployable bundles are added as members to bundle groups using the tacmd command by specifying the agent's product code and optionally the platform and version for the bundle.

To enable bulk deployment of more than one agent at a time, you add multiple members to the bundle group representing multiple agents to deploy. For example, you might create a bundle group to deploy a Windows OS agent and a DB2 application agent by adding a member with product code NT and a member with product code UD. Since one of these members is an OS agent, you use the **tacmd createNode** command to deploy this bundle group.

If you had created a bundle group containing only application agents, such as DB2, Domino, or SAP agents, you deploy them using the **tacmd addSystem** command. Each bundle group member represents a deployment bundle that must be added to the appropriate agent depot using the **tacmd addBundles** command before a bulk deployment command is executed involving the bundle group. To avoid the complexity of managing multiple depots, consider using a shared agent depot; see “Sharing an agent depot across your environment” on page 328.

Best practices: There are a number of best practices associated with bundle groups. The following are suggested.

Grouping
: **Definition**

Grouped by platform type
: A group used to specify one or more agent bundles for a specific platform type.

Grouped by system role
: A group used to specify a standard set of agents being deployed for a specific machine type or role within your enterprise. For example, a group deployed to database servers could contain an OS agent and a database agent.

Grouped for latest update
: A group containing various agents where the version specification for agent member bundles is omitted. This causes initial deployment and upgrades to always deploy the most current available version in the depot. By managing the version of agents in the depot, this type of group always deploys the latest versions to target systems.

Group properties

When using the tacmd command for deployment (createNode, addSystem, etc.), you can use the –p option to specify properties that modify parameters of the deployment or to specify configuration parameters to the agent installation process. When using the command interface with deploy and bundle groups, properties previously specified on the deploy command-line can be attached to the groups themselves or to group members. Attaching properties to groups and group members means that they are available for reuse each time the group is referenced in a deployment command without your needing to specify properties again on the command-line. This creates a very powerful mechanism for defining behaviors around your deployments and reusing them each time you specify a group as part of a deployment.

When creating your deploy and bundle groups for bulk deployment, you can assign properties to these groups. Properties at the group level apply to all members of the group. Additionally, properties can be assigned to individual group members. Any property assigned to a member applies to that member only and overrides a similar property specified at the group level. This allows you to build a hierarchy of properties that is consolidated during deployments to create a complete deployment request. Properties specified on the groups and group members can include any combination of OS agent, application agent, and deployment properties. See the *IBM Tivoli Monitoring: Command Reference* for more information about specific properties.

Consider the following example where you want to deploy the UNIX OS agent to five AIX systems with the following hostnames.

```
aix1.mycompany.com
aix2.mycompany.com
aix3.mycompany.com
aix4.mycompany.com
aix5.mycompany.com
```

To do this, you must provide login credentials. All five systems share the same root userid and password. When creating the group, you can assign the userid and password to the group so it applies to all members. This way, you need specify the login credentials only once.

```
# KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass

# tacmd addgroupmember  -t deploy -g Targets -m aix1.mycompany.com
# tacmd addgroupmember  -t deploy -g Targets -m aix2.mycompany.com
...
```

By attaching the userid and password to the deploy group at the group level, the properties apply to all members without having to repeat them for each member.

Suppose that one of the systems was on the far side of a slow link and required more time to complete a deployment. By adding a property to that member, you can change the behavior associated with that member without affecting the behavior of the others.

```
# tacmd addgroupmember  -t deploy -g Targets -m aix3.mycompany.com -p
#KDYRemote Execution and Access.TIMEOUT=3600
```

But suppose that one of the members had a different root password. By attaching this password to the member for that system, you override the value at the group level and replace it with the member-specific value.

```
# tacmd addgroupmember  -t deploy -g Targets -m aix5.mycompany.com -p
#KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=yourpass
```

Table 61 on page 342 illustrates how the properties from the above example combine to build a useful set of consolidated properties.

Table 61. Interaction between member properties and group properties

Member	Group properties	Member properties	Consolidated properties
aix1	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`		`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`
aix2	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`		`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`
aix3	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`	`KDYRemote Execution and Access .TIMEOUT=3600`	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass KDYRemote Execution and Access.TIMEOUT=3600`
aix4	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`		`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`
aix5	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=mypass`	`KDYRemote Execution and Access .Remote Execution and AccessPASSWORD=yourpass`	`KDYRemote Execution and Access.Remote Execution and AccessUSERNAME=root KDYRemote Execution and Access.Remote Execution and AccessPASSWORD=yourpass`

As you can see in the above example, group and member properties are joined during deployment to create a consolidate set of properties that are used for the deployment. Notice, however, that the deploy group member properties take precedence over and can actually override those at the deploy group level. This is visible with `aix5.mycompany.com` where the member value for KDYRemote Execution and Access.Remote Execution and AccessPASSWORD overrode the group value.

This same property mechanism applies to bundle groups as well. There might be a property that you want to apply to an agent bundle in a group regardless of where they are deployed. In this case you would attach the property to the bundle group. Likewise, you might want to specify a property that applies to a single agent type within a bundle group. In this case you attach the property to the bundle member.

During a deployment operation where you deploy a bundle group to a deploy group, property consolidation occurs across all properties at the group and member level for both groups. Table 62 lays out the precedence of group/ member properties.

Table 62. Property precedence between deploy groups and bundle groups

Precedence	Property location	What goes here
highest	Deploy group members	Properties specific to the individual member target.

Table 62. Property precedence between deploy groups and bundle groups (continued)

Precedence	Property location	What goes here
	Deploy group	Properties common to all targets of the group. For example, if agents deployed to every target member of the group will connect to the same remote monitoring server, then you can specify the server's connection properties here.
	Bundle group member	Properties common to all deployments of the specific member bundle of the group. For example, if the bundle group contains the DB2 agent for Windows and you want all installations of DB2 agents on Windows to use a specific property, specify it here.
lowest	Bundle group	Properties common to all deployments of every bundle within the group, regardless of target.

Best-practice deployment procedures

The following sections provide some best-practice procedures to assist you in planning and implementing your bulk deployment. For more information about the commands listed in the sections below, see the *IBM Tivoli Monitoring: Command Reference*.

Deployment planning and preparation

Employ the following best practices in advance of actual deployments. They represent the setup steps necessary to prepare for actual deployment operations.

1. Add bundles to the deploy depot.

   ```
   # tacmd addbundles -I imagepath ...
   ```

2. Create deploy groups.

   Create a deploy group for any collection of deployment targets. Group properties can be applied to the group during creation or afterward. Use any of the best practices described above in "Deploy groups" on page 339.

   ```
   # tacmd creategroup -t DEPLOY -g DBServers ...
   ```

3. Add deploy group members.

   After creating your deploy groups, add the appropriate systems hostnames as group members (targets) according the best practice chosen for the group. Member properties can be applied during member creation or afterward.

   ```
   # tacmd addgroupmember -t DEPLOY -g DBServers -m hostname  ...
   # ...
   ```

4. Create bundle groups.

   Create a bundle group for any collection of agent bundles you wish to deploy. Group properties can be applied to the group during creation or afterward. Use any of the best practices described above in "Bundle groups" on page 340.

   ```
   # tacmd creategroup -t BUNDLE -g DBAgentBundles ...
   ```

5. Add bundle group members.

   After creating your bundle groups, add the appropriate agent bundles as members to the groups according the best practice chosen for the group. Member properties can be applied during member creation or afterward.

   ```
   # tacmd addgroupmember -t BUNDLE -g DBAgentBundles -m UnixOSAgent -y UX ...
   # tacmd addgroupmember -t BUNDLE -g DBAgentBundles -m DB2Agent -y UD ...
   # ...
   ```

Deployment

The following best practices are employed to initiate and perform actual deployments. These best practices apply regardless of whether the deployment is an initial agent installation, an agent upgrade, or an agent configuration update. In fact, it is important to recognize that the groups created for deployment are applicable to all agent maintenance operations throughout the agent lifecycle.

1. Submit deployment.

   Submitting the deployment differs slightly depending on the type of deployment activity you desire, but they are all very similar.

   - createNode

     Use the **tacmd createNode** command when performing initial deployments of OS and SSM agents. This includes deploying bundle groups that contain OS agents grouped together with application agents.

     ```
     # tacmd createnode -g DBServers -b OS_DBAgentBundles
     ```

   - addSystem

     Use the **tacmd addSystem** command when initially deploying application agents.

     ```
     # tacmd addsystem -g AppServers -b AppAgentBundles
     ```

   - updateAgent

     Use the **tacmd updateAgent** command when performing agent or SSM upgrades.

     ```
     # tacmd updateagent -g DBServers -b LatestAgentfix packs
     # tacmd updateagent -g AppServers -b LatestAppAgentfix packs
     ```

     Note: The `KUIWINNT.dsc` file on Windows systems and the `ui`*`platform`*`.dsc` files (where *platform* is the platform name) on Linux and UNIX systems have been added or updated so that the KUI package (the tacmd commands) can be remotely deployed. Use the following command:

     ```
     tacmd updateAgent -t ui -n node
     ```

     where:

     `node`
       is the IBM Tivoli Monitoring node on which the KUI package is to be remotely deployed.

   - configureSystem

     Use the **tacmd configureSystem** command when performing configuration updates to agents or SSMs.

     ```
     # tacmd configuresystem -g DBServers -b DBServerAgentConfig
     ```

   - removeSystem

     Use the **tacmd removeSystem** command when removing application agents or SSM patches from a system.

     ```
     # tacmd removesystem -g AppServers -b AppAgentBundles
     ```

2. Monitor deployment.

   There are many mechanisms that you can use for monitoring deployment status. These include:

   - Using the CLI.

     The **tacmd getDeployStatus** command returns the complete status of requested deployment transactions.

     ```
     # tacmd getdeploystatus -g group_transactionID
     ```

   - Using portal workspaces.

     The Tivoli Enterprise Portal provides a number of workspaces that display the status of deployments. These provide an easy mechanism to rapidly visualize deployment status summaries as well as detailed status. See the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* or the portal online help for more information.

   - Using automation.

     Automation can be driven in two ways. Situations can be created to indicate deployment failures to initiate corrective activities. Also, automation scripts can use the **tacmd getDeployStatus** command to retrieve and parse status information and initiate corrective activities. Situations and automation scripts can be used in tandem to monitor deployment status.

Regardless of the specific mechanism for monitoring deployment status, the best practice involves the following steps.

a. Check the transaction status to ensure it is queued properly.

b. Wait some period of time for the deployments to proceed.

c. Check the transaction status for successful completions and failures. This involves either using the command interface or the portal interface or checking for situation events that indicate deployment failures.

d. Perform any necessary corrective actions.

e. Repeat starting with step 2b until all transactions have completed.

3. Correct problems.

   When a deployment transaction fails, some corrective action is usually required before reinitiating the deployment. Consult the return message portion of the transaction status to identify the problem and possible resolutions. Rerun the deployment once errors are corrected. For example, when using a deploy group with the addSystem or updateAgent commands, remote deployment may fail to locate the existing managed-system name for some hosts. The message received is **KDY0012E** The target *target_host_name* is incorrect or is offline. The command did not complete because the value for the target is incorrect or the target is offline.

   This message normally indicates that the OS agent is not online. If the agent is, in fact, online, cancel current operations to this node:

```
# tacmd cleardeploystatus  -h hostname
```

   Then issue the operation directly by using the managed system name parameter instead of the deploy group:

```
# tacmd updateAgent  -t product_code -n managed_OS
```

4. Purge status.

   When all deployment transactions have completed or the deployment status is no longer required, clear the transaction's deployment status from the systems.

```
# tacmd clearDeployStatus -g group_transactionID
```

While the steps above describe the complete process of deployment, it is common to perform deployments in small sets of consumable transactions. That is, perform an initial deployment on a smaller group of systems to ensure that there are no issues that would inhibit the successful completion of deployment. Once that deployment is complete, continue with one or more larger deployments using target deploy groups with more members. Work through each deploy group representing some portion of your enterprise until you have completed all deployments for your enterprise.

Working with non-agent bundles

This chapter describes how to deploy, update, and remove non-agent bundles. Traditionally, remote deployment supported remote installation of deployable agents to a connected monitoring server. Now you can use remote deployment to install custom bundles or bundles for other components. This non-agent bundle capability is a way to deploy components that are not required to connect to the monitoring server. This enhancement includes the following:

- The allowable length for a product code for non-agent bundles is restricted to be between 3 to 32 characters in order to distinguish it from the agent product codes.
- The following tacmd commands were enhanced to support product codes up to 32 characters: addBundles, addSystem, getDeployStatus, listBundles, removeSystem, updateAgent, and viewDepot.

Notes:

1. Avoid non-agent bundle product codes that start with K as this name might conflict with IBM Tivoli Monitoring agent bundles.
2. Non-agent bundles do not support the following tacmd commands: **startAgent**, **stopAgent**, **restartAgent**, **viewAgent**, and **configureSystem**.

Deploying a non-agent bundle

Complete the following steps to deploy a non-agent bundle:

1. Run the **tacmd listBundles** command to view the agent bundles.
2. Add the agent bundle to the depot using the **tacmd addBundles** command.
3. Run the **tacmd viewDepot** command to display the contents of the agent depot. Verify the non-agent bundles were added to the depot.
4. Run the **tacmd addSystem** command with appropriate configuration information. If you want to specify an installation path, run this command with the KDY.installPath property set to `CANDLEHOME/`*`productcode`* if you want the location to be within the CANDLEHOME directory, or set to /*productcode* if you do not want the location to be within the CANDLEHOME directory, for example:
   ```
   tacmd addSystem -p KDY.installPath=/IBM/ITM/Uxxx -n managed_os -t Uxxx
   ```
5. Check to ensure the status of transactions is queued or InProgress.

Monitor the workspace or use the **tacmd getDeployStatus** command to verify the result. When the deployment completes successfully, verify that the bundle was installed at the default location if you did not specify the installation path, or to the location that you chose if you did specify an installation path.

Notes:

1. The OS agent must already be installed on the system before deploying the non-agent bundle.
2. The non-agent bundle must exist in the depot of the monitoring server to which the OS agent is connected.
3. For more information about each of these commands, see the table below that links to a description of each command, including syntax and option information, along with examples.

Updating a non-agent bundle

Complete the following steps to update a non-agent bundle:

1. Run the **tacmd listBundles** command to view the agent bundles.
2. Add the agent bundle to the depot using the **tacmd addBundles** command.
3. Run the **tacmd viewDepot** command to display the contents of the agent depot. Verify the non-agent bundles were added to the depot.
4. Run the **tacmd updateAgent** command with appropriate parameters. If the non-agent bundle was installed in a directory other than the default, run this command with the KDY.installPath property set to `CANDLEHOME/`*`productcode`* if the location is within the CANDLEHOME directory, or run this command with the KDY.installPath property set to /*productcode* if the location is not within the CANDLEHOME directory, for example:
   ```
   tacmd updateAgent -p KDY.installPath=/IBM/ITM/Uxxx -n managed_os -t Uxxx
   ```
5. Check to ensure the status of transactions is queued or InProgress.

Monitor the workspace or use the **tacmd getDeployStatus** to verify the result. When the deployment completes successfully, verify that the agent bundles were updated on the target system.

Notes:

1. Ensure that the installation path is the same one that was used when the non-agent bundle was added initially.

2. The non-agent bundle must exist in the depot of the monitoring server to which the OS agent is connected.
3. For more information about each of these commands, see the table below that links to a description of each command, including syntax and option information, along with examples.

Removing a non-agent bundle

1. Run the **tacmd removeSystem** command with appropriate parameters, including the KDY.installPath property if the agent was installed at a user specified location, for example:
   ```
   tacmd removesystem -p KDY.installPath=/IBM/ITM/Uxxx -n managed_os -t Uxxx
   ```
2. Check to ensure the status of transactions is queued or **InProgress**.

Notes:

1. The non-agent bundle must already be in the depot in order to remove it. If it is not, add it to the depot using the **tacmd addBundles** command.
2. Ensure that the installation path is the same one that was used when the non-agent bundle was added initially.
3. The non-agent bundle must exist in the depot of the monitoring server to which the OS agent is connected.

Monitor the workspace or use the **tacmd getDeployStatus** command to verify the result. When the removal completes, verify that the non-agent bundle files are removed from the target system and none of the IBM Tivoli Monitoring system files were inadvertently removed.

Running deployment in a Hot Standby environment

The IBM Tivoli Monitoring Hot Standby capability allows your monitoring environment to continue operating in the event of environmental or operational issues with the primary hub Tivoli Enterprise Monitoring Server (for detailed information about Tivoli Monitoring's Hot Standby feature, see the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*). You should refrain from deploying or updating agents when IBM Tivoli Monitoring is converting to a mirror monitoring server. No agent deployments or remote deployment operations should be executed from a Hot Standby mirror hub, as this may cause your deployment transactions to get stuck in a queued state, and you may not be able to clear them.

Self-describing agent installation

Product support files for IBM Tivoli Monitoring agent applications must be installed at each IBM Tivoli Monitoring server component. The self-describing agent feature combines the installation of an agent with the automatic dispersal and installation of the associated product support files throughout your IBM Tivoli Monitoring infrastructure. The self-describing agent feature must be enabled on the following IBM Tivoli Monitoring components:

- Hub Tivoli Enterprise Monitoring Server.
- Tivoli Enterprise Portal Server.
- Self-describing agent supported agents.
- Any Remote Tivoli Enterprise Monitoring Server the agent is connected to.

Enabling the self-describing agent capability at the hub monitoring server controls the capability across all components. For more information, see “Enabling self-describing agent capability at the hub monitoring server” on page 216.

For information on self-describing agent environment variables, see “Environment variables” on page 826.

Terminal self-describing agent installation errors

In this example, the install record for product code 11 displays a STATE value of **ME**:

```
HUB/RTEMS PRODUCT VERSION  GRPID ID       IDVER    SEEDSTATE  STATE STATUS
RTEMS_LZ  11      06230000 5655  TMS      06230000 Y          ME    1014
```

The STATE=ME record indicates a self-describing agent metadata installation error has occurred on the monitoring server. The monitoring server stops attempting any self-describing agent installation for this product code until some action is taken by the administrator to correct the error. This correction might involve IBM Software Support. In this scenario, you must use the `tacmd deleteappinstallrec` command to clear the self-describing agent error record after you have addressed the problem. For more information about the **tacmd deleteappinstallrec** command, see the *IBM Tivoli Monitoring: Command Reference*.

To determine if a self-describing agent product installation has failed with a terminal error condition, run the `tacmd listappinstallrecs` command by using the `-e` option to show error records only. For any error records with a STATE value of ME, the installation is not retried.

Complete the following steps to retry the self-describing agent installation:

1. To avoid the same failure occurring again, you must first correct the condition that caused the installation to fail. The failure code is reported in the STATUS column of the `listappinstallrecs` output. In addition, the Tivoli Enterprise Monitoring Server message facilities (Audit, MSG2, and RAS1 messages) provide more information about the cause of the failure. Take corrective action to fix the condition or contact IBM Software support for assistance.
2. For each monitoring server, delete the failed installation records in the application properties table by running the `tacmd deleteappinstallrecs` command. Use this command to remove the blocking self-describing agent product installation record. See the *IBM Tivoli Monitoring: Command Reference* for more information about using the `tacmd deleteappinstallrecs` command.
3. When each Tivoli Enterprise Monitoring Server failed product install record has been cleared, the monitoring server self-describing agent facility immediately notifies any running self-describing agent that can provide this level of product support, to retry the product installation. For example, if the previous install attempt for product *pc* and version 06230000 failed with a STATE of ME, and you run the `deleteappinstallrecs` command, any running *pc* agent for version 06230000 immediately retries the installation.
4. Run the `tacmd listappinstallrecs -t <pc>` again for product *pc* to determine the current installation state.

Enabling or disabling self-describing agent capability

The self-describing agent environment variable is set to Y to enable the self-describing agent capability at all components by default, with the exception of the hub monitoring server. The environment variable KMS_SDA is disabled (KMS_SDA=N) by default only on the hub monitoring server. All components that connect to the hub monitoring server adjust their self-describing agent function to disabled if the hub monitoring server has the self-describing agent function disabled, or if the hub monitoring server is older and does not support self-describing agents. This allows control of all self-describing agent enablement from a single point - the hub monitoring server. For more information, see “Enabling self-describing agent capability at the hub monitoring server” on page 216. A best practice is to only use the KMS_SDA variable at the hub monitoring server to control self-describing agent enablement, but it can be controlled at each component:

- Tivoli Enterprise Monitoring Server uses the KMS_SDA environment variable.
- Tivoli Enterprise Portal Server uses the TEPS_SDA environment variable.
- Self-describing agents use the TEMA_SDA environment variable.

For more information on self-describing agent variables, see “Environment variables” on page 826.

Chapter 11. Monitoring your operating system via a System Monitor Agent

A special type of Tivoli Management Services agent, the System Monitor Agent, allows you to send OS monitoring data directly to any SNMP or EIF event receiver such as Tivoli Netcool/OMNIbus, Tivoli Enterprise Console, or Tivoli NetView without first passing the data to a Tivoli Enterprise Monitoring Server. These lighter-weight agents (they require a much smaller footprint than full-function Tivoli Monitoring OS agents) are configured locally on the agent node, which makes them ideal for customers who want fully autonomous monitoring of their desktop operating systems in environments where disk space or image transmission bandwidth is in short supply. This configuration enables them to be deployed autonomously (that is, they do not require the support of a Tivoli Enterprise Monitoring Server): they send SNMP or EIF event information directly to an Event Collector such as IBM Tivoli Netcool/OMNIbus.

These agents can run in agent-only environments that lack the standard Tivoli Monitoring servers (the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server). Local configuration files can be placed on the system directly, or the System Monitoring Agent can be configured to check in on a schedule with a Central Configuration Server to collect updated configuration files. (See *Central Configuration Servers* in the *IBM Tivoli Monitoring: Administrator's Guide*.)

You can also install or deploy System Monitor Agents using a Golden Master Image. To install System Monitor Agents using a Golden Master Image, use the following procedures.

To create the Golden Master Image:

1. Install a System Monitor Agent on a sample system.
2. Configure the agent as required:
   - You can provide the private configuration files, or provide the Central Configuration Connection information and test the image.
   - Once it is performing as desired, stop the agent.
3. Compress the files and directories within the IBM Tivoli Monitoring installation directory to create a Golden Master Image.

To install the Golden Master Image:

1. Transfer the compressed file to the target system.
2. Extract the image into the same installation directory.

   Note: The agent installation directory must be the same on the original image and the target system.
3. Once the files are in place run the initialization script and start the agent:
   - On Windows systems: `<instdir>\InstallITM\UpdateAutoRun.cmd`
   - On Linux systems: `<instdir>/bin/UpdateAutoRun.sh`

These agents, which run on Windows and on Linux/UNIX, effectively replace the OMNIbus System Service Monitors for monitoring of desktop operating systems. IBM Netcool System Service Monitor users may find these agents appropriate as an upgrade path.

Note: Only agents created with version 6.2.2 (or subsequent) of the Agent Builder tool that do not include a data provider that requires Java can be installed with a System Monitor Agent. No other agents should be installed with a System Monitor Agent, nor should they be installed on nodes where there is already a Tivoli Monitoring component in operation.

To support these new System Monitor Agents, there is a new silent installation process for both Windows and Linux/UNIX sites, along with operating system-specific configuration steps. There are also new uninstallation processes for these Windows and Linux/UNIX OS agents. All these processes are detailed in this chapter.

"Background information about agent autonomy" on page 65 introduces the concept of agents that operate autonomously (that is, without a connection to a Tivoli Enterprise Monitoring Server). For additional information, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Installing the System Monitor Agent on Windows systems

The System Monitor Agent packages provide IBM Tivoli Monitoring OS agents whose reduced footprint makes them ideal for customers who need fully autonomous OS agent monitoring in environments where disk space or image transmission bandwidth is in short supply. These OS agent packages offer a significantly reduced bundle size and installed footprint.

The System Monitor Agents support direct Netcool/OMNIbus integration without a connection to a Tivoli Enterprise Monitoring Server. The System Monitor Agent instead sends SNMP alerts directly to an SNMP Event Collector such as Netcool/OMNIbus.

Installation times are significantly reduced due to the small size of the System Monitor Agents. These agents use simplified flat-file, machine-independent configuration. Configuration files no longer contain references to the host name or system where the file is located; with this change, you can share and transport common configuration files across your network.

Notes:

1. A System Monitor Agent must not be installed into a system where existing IBM Tivoli Monitoring components (including other monitoring agents) are already installed, with this exception: Agents built with Agent Builder V6.2.2 or subsequent may be installed alongside a System Monitor Agent, provided they run in the same mode as the Windows System Monitor Agent itself: if the Windows agent runs in 32-bit mode, only 32-bit Agent Builder agents are supported; if the Windows agent runs in 64-bit mode, only 64-bit Agent Builder agents are supported. 32-bit Agent Builder agents can be regenerated using Agent Builder to create 64-bit Windows binaries that you can install with the 64-bit Windows System Monitor Agent.

   The following 64-bit Windows environments support 64-bit System Monitor Agents:
   - Windows Server 2003 Standard Edition R2 on x86-64 CPUs in 64-bit mode
   - Windows Server 2003 Enterprise Edition R2 on x86-64 CPUs in 64-bit mode
   - Windows Server 2003 Datacenter Edition R2 on Intel x86-64 CPUs in 64-bit mode
   - Windows Vista Enterprise Edition on Intel x86-64 CPUs in 64-bit mode
   - Windows Server 2008 Standard Edition on Intel x86-64 CPUs in 64-bit mode
   - Windows Server 2008 Enterprise Edition on Intel x86-64 CPUs in 64-bit mode
   - Windows Server 2008 Datacenter Edition on Intel x86-64 CPUs in 64-bit mode
2. The NT Micro Agent in the `system_monitor_agent` directory of the IBM Tivoli Monitoring installation media only supports 8.3 format path values, with no special characters and no spaces, as an installation path. The bat files that install the NT Micro Agent do not support long paths or paths that are enclosed in double quotes. Use a short path without special characters or spaces to install the NT Micro Agent.

You must invoke a silent installation procedure to install the System Monitor Agent that monitors the Windows operating system:

```
silentInstall.cmd -p response_file
```

where:

response_file
is the name of the updated version of the `nt_silent_install.txt` response file you created that names the components to be installed and where.

On completion of the install script, the OS agent for Windows is installed, configured with a default configuration, and started.

Contents of the silent response file

The silent response file supports the following records:

comments
Any line beginning with a semicolon (;) is treated as a comment.

Keyword=value
These lines define variables in the silent response file. Do not include spaces before the keyword or immediately before or after the equal sign (=). Values must not contain the following characters:

$
=
|

Note: By default the `silentInstall.cmd` installation script checks the processor architecture on which it is running; if it discovers it is running on a 64-bit x86_64 CPU, `silentInstall.cmd` installs the System Monitor Agent in 64-bit mode. Otherwise the agent is installed in 32-bit mode.

To force installation of the 32-bit System Monitor Agent on an x86_64 CPU, set variable **OS_ARCH** to 32 in the `nt_silent_response.txt` file. (**OS_ARCH** can also be set to 64, to force installation of a 64-bit agent. If **OS_ARCH** is blank or unspecified, the installer determines which agent to install based on the host processor.)

The silent response file supports the following variables:

License Agreement
(Required) This variable accepts the license agreement for the System Monitor Agent. In the sample response file, simply uncomment the line containing the **License Agreement** variable, or set the variable to **I agree to use the software only in accordance with the installed license.**

Install Directory
(Required) This variable specifies the directory where the System Monitor Agent will be installed. Characters supported for the **Install Directory** are alphanumeric characters, underscore (_), and hyphen (-).

KDC_FAMILIES_OVERRIDE
(Optional) This variable overrides the value assigned to the **KDC_FAMILIES** variable the agent uses at run time. In the sample response file, uncomment the line containing the **KDC_FAMILIES_OVERRIDE** variable and specify the wanted protocols to be enabled. If left commented out, the installer uses a value of:

```
IP.PIPE PORT:1918 IP use:n IP.SPIPE use:n SNA use:n
```

This example sets the default protocol to IP.SPIPE, using listening port 3660, and with all other protocols disabled:

```
KDC_FAMILIES_OVERRIDE=IP.SPIPE  PORT:3660  IP.PIPE use:n  IP use:n SNA use:n
```

Contents of the silent response file (cont'd)

For more information about the supported protocols and modifiers, see “Tivoli Monitoring protocol usage and protocol modifiers” on page 379.

Note: Failure to specify at least one active protocol results in the System Monitor Agent not starting. To rectify this you can manually edit the KDC_FAMILIES registry entry value.

`EncryptionKey`

(Optional) This variable sets the encryption key used for storing passwords and other protected content. The default value if not specified is **IBMTivoliMonitoringEncryptionKey**.

Notes:

1. Do not use any of the following characters in your key:

&	ampersand
|	pipe
'	single quote
=	equal sign
$	dollar sign

   In addition, do not specify double-byte (DBCS) characters.
2. Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

`InstallLog`

(Optional) This variable overrides the default location of the installation log created by the **silentInstall.cmd** command. The default value if not specified is `%CANDLE_HOME%\InstallITM\ibm_tivoli_micro_install.log`. In case of an installation error where the installation directory was not created, the installation log will be created in the %TEMP% directory as `ibm_tivoli_micro_install.log`. Characters supported for the **InstallLog** are alphanumeric characters, underscore (_), and hyphen (-).

A sample `nt_silent_install.txt` response file can be found in directory `\agents\system_monitor_agent\WINDOWS\` on the Agents DVD.

Configuring the System Monitor Agents on Windows

Environmental configuration for the autonomous OS agents is unchanged on Windows: Configuration parameters are contained in the agent's ENV file in the *`InstDir`*`\tmaitm6` directory, where *InstDir* is your installation directory. By directly editing this file, you can modify existing variables within that file or append additional environment variables.

User-defined local configuration files:

These configuration files are stored in a separate local configuration directory, *`InstDir`*`\localconfig\`*`pc`*, where *pc* is the agent's two-character product code. This segregation simplifies agent configuration through the use of flat files. It also segregates your user configurations, which preserves them during agent upgrade. They also allow redistribution of a configured agent to other systems while ensure their configurations remain identical.

To modify the default location of the operational configuration files, perform the following steps:

1. Edit the agent's ENV file (for example, KNTENV for the Windows OS agent).
2. Add or modify the IRA_LOCALCONFIG_DIR parameter to specify a different local configuration directory location.

After you have modified the ENV file and saved it, recycle the agent to implement your updated configuration.

Note: On Windows, there are new CLI commands to start and stop an agent:

```
%CANDLE_HOME%\InstallITM\itmcmd.cmd agent start nt
```

```
%CANDLE_HOME%\InstallITM\itmcmd.cmd agent stop nt
```

Private situation configuration files:

If a correctly named private situation configuration file is present in the System Monitor Agent's local configuration directory when the agent is started, the agent will run the private situations that the file defines. The file must be named `pc_situations.xml`, where *pc* is the agent's two-character product code.

You can use the IRA_PRIVATE_SITUATION_CONFIG parameter to specify a different file name or location. This variable is resolved relative to the IRA_LOCALCONFIG_DIR, or you can specify a fully qualified file name.

For more information about the private situation configuration files, see the *IBM Tivoli Monitoring: Administrator's Guide*.

SNMP trap configuration files:

If a correctly named trap configuration file is present in the System Monitor Agent's local configuration directory when the agent is started, the agent will emit the SNMP alerts that the file defines. The file must be named `pc_trapcnfg.xml`, where *pc* is the agent's two-character product code.

You can use the IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG parameter to specify a different file name or location. This variable is resolved relative to the IRA_LOCALCONFIG_DIR, or you can specify a fully qualified file name.

For more information about SNMP trap configuration files, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Uninstalling the Windows System Monitor Agent

There is a separate procedure for uninstalling the System Monitor Agent that monitors the Windows operating system and deleting the directory where it is stored. Issue the following command:

```
%CANDLE_HOME%\InstallITM\uninstall.cmd [-f]
```

You are prompted to confirm your request to uninstall the agent. Respond **Y** to continue the uninstallation.

The `-f` option forces the uninstallation command to bypass the confirmation prompt.

If the command is invoked while the current working directory is `%CANDLE_HOME%\InstallITM`, the directory is not deleted; you must delete it manually.

Notes:

1. Uninstalling the Windows operating system agent also uninstalls all Agent Builder agents installed in the same environment.
2. Directories are not removed on Microsoft Windows systems if the command that attempts to remove it is running from the directory being removed or if a file is locked within that directory. Thus, if you run `uninstall.cmd` from `%CANDLE_HOME%\BIN`, the directory is not removed; you must remove it yourself.

   Before running `uninstall.cmd`, it is recommended that you first close all processes (such as command prompts and text editors) currently accessing subdirectories of `%CANDLE_HOME%`. Then run the `uninstall.cmd` command from outside of `%CANDLE_HOME%`. Specify a fully qualified path.

If `uninstall.cmd` cannot remove a subdirectory, it displays the following message:

```
directory may have to be removed manually after this script completes.
```

This same message also is written to the uninstallation log file, `%TEMP%\IBM_Tivoli_Monitoring_System_Monitor_Agent_uninstall.log`.

Installing the System Monitor Agent on Linux or UNIX systems

The System Monitor Agent packages provide IBM Tivoli Monitoring operating-system agents whose reduced footprint makes them ideal for customers who need fully autonomous OS agent monitoring in environments where disk space or image transmission bandwidth is in short supply. These System Monitor Agent packages offer a significantly reduced bundle size and installed footprint.

The System Monitor Agents support direct Netcool/OMNIbus integration without a connection to a Tivoli Enterprise Monitoring Server. The System Monitor Agent instead sends SNMP alerts directly to an SNMP Event Collector such as Netcool/OMNIbus.

Installation times are significantly reduced due to the small size of the agents. System monitor agents use simplified flat-file, machine-independent configuration. Configuration files no longer contain references to the host name or system where the file is located; with this change, you can share and transport common configuration files across your network.

Note: A System Monitor Agent must not be installed into a system where existing IBM Tivoli Monitoring components (including other monitoring agents) are already installed, with this exception: Agent Builder agents built with Agent Builder V6.2.2 or subsequent may be installed alongside a System Monitor Agent.

There is a silent installation procedure that you must invoke to install these autonomous agents that monitor the Linux and UNIX operating systems:

```
silentInstall.sh -h installation_dir -p response_file
```

where:

`installation_dir`
is the directory on the target machine where the System Monitor Agent is to be installed.

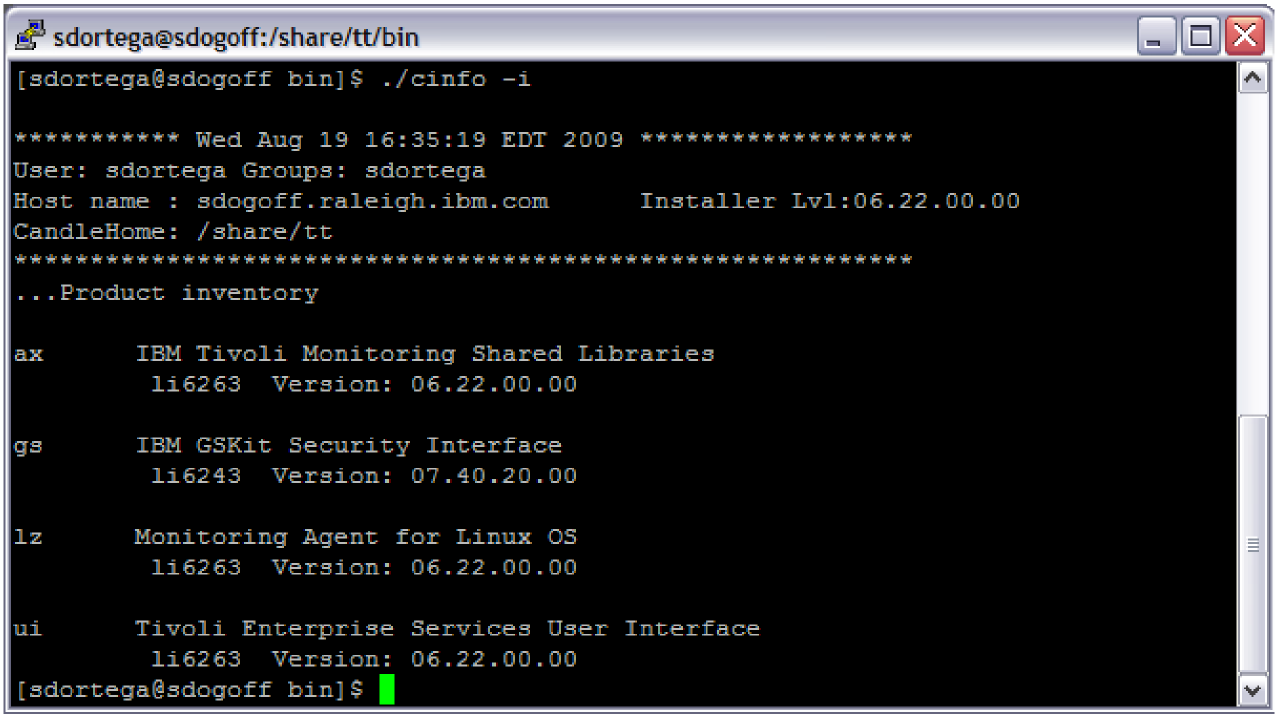
`response_file`
is the name of the response file you created that names the components to be installed and where.

Note: You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.

To verify the installation, enter this command:

```
InstDir/bin/cinfo -i
```

where *InstDir* is the directory where you installed the System Monitor Agent. The list of installed agent components is displayed, as shown in Figure 95 on page 355.



```
[sdortega@sdogoff bin]$ ./cinfo -i

*********** Wed Aug 19 16:35:19 EDT 2009 ******************
User: sdortega Groups: sdortega
Host name : sdogoff.raleigh.ibm.com      Installer Lvl:06.22.00.00
CandleHome: /share/tt
***********************************************************
...Product inventory

ax      IBM Tivoli Monitoring Shared Libraries
         li6263  Version: 06.22.00.00

gs      IBM GSKit Security Interface
         li6243  Version: 07.40.20.00

lz      Monitoring Agent for Linux OS
         li6263  Version: 06.22.00.00

ui      Tivoli Enterprise Services User Interface
         li6263  Version: 06.22.00.00
[sdortega@sdogoff bin]$
```

Figure 95. Output of the cinfo command

If no agents are listed, check the installation logs for more information.

Contents of the silent response file

The silent response file supports the following records:

`comments`
Any line beginning with a hash mark (#) is treated as a comment.

`Keyword=value`
These lines define variables in the silent response file. Do not include spaces before the keyword or immediately before or after the equal sign (=). Values must not contain the following characters:

$
=
|

The silent response file supports the following variables:

`License_Agreement`
(Required) This variable accepts the license agreement for the System Monitor Agent. In the sample response file, simply uncomment the line containing the **License Agreement** variable, or set the variable to **I agree to use the software only in accordance with the installed license.**

`INSTALL_PRODUCT`
(Required) Identifies the product you want installed; specify **lz** for Linux for **ux** for UNIX.

`INSTALL_FOR_PLATFORM`
(Optional) Identifies the platform for which the product should be installed; see Table 165 on page 816. If left commented out, the installer will use the platform of the machine on which the installation is performed.

`INSTALL_ENCRYPTION_KEY`
(Optional) This variable sets the encryption key used for storing passwords and other protected content. The default value if not specified is **IBMTivoliMonitoringEncryptionKey**.

Notes:

1. Do not use any of the following characters in your key:

&	ampersand
|	pipe
'	single quote
=	equal sign
$	dollar sign

   In addition, do not specify double-byte (DBCS) characters.
2. Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

`INSTALL_IGNORE_RUNNING_PROCESSES`
If set to **n**, this keyword aborts the installation of running IBM Tivoli Monitoring processes. If this parameter is set to **y**, the installation stops the running processes, installs the requested product, and restarts the processes that were running.

`KDC_FAMILIES_OVERRIDE`
(Optional) This variable overrides the value assigned to the **`KDC_FAMILIES`** variable that the agent uses at run time. In the sample response file, uncomment the line containing the **`KDC_FAMILIES_OVERRIDE`** variable and specify the wanted protocols to be enabled. If left commented out, the installer uses a value of:

```
IP.PIPE PORT:1918 IP use:n IP.SPIPE use:n SNA use:n
```

This example sets the default protocol to IP.SPIPE, using listening port 3660, and with all other protocols disabled:

```
KDC_FAMILIES_OVERRIDE='IP.SPIPE PORT:3660 IP.PIPE use:n IP use:n SNA use:n'
```

For more information about the supported protocols and modifiers, see “Tivoli Monitoring protocol usage and protocol modifiers” on page 379.

Note: Failure to specify at least one active protocol results in the System Monitor Agent not starting. To rectify this you can manually edit the KDC_FAMILIES environment parameter in the configuration file of the agent.

Sample `lz_silent_install.txt` and `ux_silent_install.txt` response files can be found in directory `/agents/system_monitor_agent/unix/` on the Agents DVD.

Configuring the System Monitor Agents on Linux or UNIX

The autonomous OS agents do not use the *pc*.ini files (where *pc* is the product code) for configuration. Instead these System Monitor Agents follow a new configuration model based on flat files. These files are designed to segregate and preserve your user configuration, which simplifies agent upgrade. They also allow redistribution of a configured agent to other systems while ensuring the original configuration is exactly duplicated.

The System Monitor Agents' environment files are stored in the *`InstDir`*`/config` directory, where *InstDir* is your installation directory. When the agent is installed, two files are added to this directory:

`.global.environment`
This file contains IBM Tivoli Monitoring-defined global environment settings that are available to all monitoring agents that use the same installation directory. *User modifications should not be made to this file*, as the changes may not be preserved in future upgrades. Instead, make your global user configuration changes in the `global.environment` file described below.

`.`*`pc`*`.environment`
Here *pc* is the two-character IBM Tivoli Monitoring product code (for example, `.ux.environment` is the product-provided UNIX System Monitor Agent environment configuration file, and

.lz.environment is the Linux System Monitor Agent environment configuration file). These files contain product-defined, agent-specific environment settings that are available only to the *pc* agents running in the same installation directory. Again, *user modifications should not be made to this file*, as the changes may not be preserved in future upgrades. Instead, make your user agent configuration changes in the *pc*.environment file described below.

Values defined in the .*pc*.environment files override those defined in the global environment files.

In the *InstDir*/config directory (where *InstDir* is your installation directory), you may create custom environment files that override the environment settings defined in the .global.environment and .*pc*.environment files. These user customization are preserved during future product upgrades. These files are:

global.environment

This file is not created when the System Monitor Agent is installed but must be created by you:

1. Copy the IBM Tivoli Monitoring-supplied .global.environment file to global.environment.
2. Delete all entries that do not need to be customized.
3. Add any custom global configuration variables that should be available to all agents running in the same installation directory.

User modifications should be made to this file as these changes will never be overwritten by future upgrades. Values specified in the global.environment configuration file override those in both the .global.environment file and the .*pc*.environment.

***pc*.environment**

This file is not created when the System Monitor Agent is installed but must be created by you:

1. Copy the IBM Tivoli Monitoring-supplied .*pc*.environment file to *pc*.environment.
2. Delete all entries that do not need to be customized.
3. Add any agent-specific configuration variables that should be available to all *pc* agents running in the same installation directory.

User modifications should be made to this file as these changes will never be overwritten by future upgrades. Values specified in the user pc.environment file override all others.

User-defined local configuration files:

These configuration files are stored in a separate local configuration directory, *InstDir*/localconfig/*pc*, and include this parameter:

IRA_LOCALCONFIG_DIR

Specifies a different local configuration directory. You may want to set the IRA_LOCALCONFIG_DIR parameter in the *pc*.environment file so each agent will have its own unique IRA_LOCALCONFIG_DIR setting. Alternatively, setting the IRA_LOCALCONFIG_DIR in the global.environment file allows all agents to share the same IRA_LOCALCONFIG_DIR setting.

Private situation configuration files:

If a correctly named private situation configuration file is present in the System Monitor Agent's local configuration directory when the agent is started, the agent will run the private situations that the file defines. The file must be named *pc*_situations.xml, where *pc* is the agent's two-character product code.

The **IRA_PRIVATE_SITUATION_CONFIG** environment variable may be used to specify a different file name or location. This variable is resolved relative to the **IRA_LOCALCONFIG_DIR** directory, or you may specify a fully qualified filename.

For more information about the private situation configuration files, see the *IBM Tivoli Monitoring: Administrator's Guide*.

SNMP trap configuration files:

If a correctly named trap configuration file is present in the System Monitor Agent's local configuration directory when the agent is started, the agent will emit the SNMP alerts that the file defines. The file must be named `pc_trapcnfg.xml`, where *pc* is the agent's two-character product code.

The **IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG** environment variable may be used to specify a different file name or location. This variable is resolved relative to the **IRA_LOCALCONFIG_DIR** directory, or you may specify a fully qualified filename.

For more information about SNMP trap configuration files, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Uninstalling the Linux or UNIX System Monitor Agent

There is a separate, manual procedure for uninstalling the System Monitor Agent that monitors the Linux or UNIX operating system:

1. Stop the agent by invoking either of the following commands:
   - For Linux, specify:
     ```
     InstDir/bin/itmcmd agent stop lz
     ```
   - For UNIX, specify:
     ```
     InstDir/bin/itmcmd agent stop ux
     ```

   where *InstDir* is the directory where you installed the System Monitor Agent.
2. Stop any other agents running from the same *InstDir*.
3. Issue one of the following commands :
   ```
   InstDir/bin/uninstall.sh
   ```

   or:
   ```
   InstDir/bin/uninstall.sh REMOVE EVERYTHING
   ```

   Invoking the `uninstall.sh` script with the `REMOVE EVERYTHING` parameter removes all agent files and deletes the installation subdirectory tree.

On UNIX and Linux, you can uninstall multiple, individual agents by listing each one in the uninstall.sh command.

Note: Uninstalling the UNIX or Linux OS System Monitor Agent also removes common files that are necessary for Agent Builder agents to function. If you are uninstalling the UNIX or Linux OS system monitor agent, you must also uninstall any Agent Builder agents in the same environment.

Defining common configuration parameters: accessing centralized configuration information

System Monitor Agents are started at the end of their silent installation. Unless private configuration files exist for them, these agents run but do not analyze situations or send events. Centralized configuration allows an agent to retrieve these files from the centralized configuration server and to begin using them immediately. You define these parameters in the `pc_silent_install.txt` silent response file that is used when installing a System Monitor Agent.

The System Monitor Agent installation process uses entries in the silent response file to create entries in the agent's environment file, based on statements of this form: **SETENV_***parameter***=***value* statements create *parameter***=***value* statements in the agent's environment file, whereas **SETENCR_***parameter***=***value* create *parameter***={AES256:keyfile:a}***encryptedvalue* statements.

Example: These silent response file entries connect to a centralized configuration server running on the node named linuxhost:

```
SETENV_IRA_CONFIG_SERVER_URL= http://linuxhost:1920///linuxhost_lz/linuxhost_lz/
SETENV_IRA_CONFIG_SERVER_USERID=itmuser
SETENCR_IRA_CONFIG_SERVER_PASSWORD=password
SETENV_IRA_CONFIG_SERVER_FILE_PATH=initloadlist/@PRODUCT@
SETENV_IRA_CONFIG_SERVER_FILE_NAME=cnfglist.xml
```

Agents recognize the following keywords and substitute for them runtime values retrieved from the client:

@PRODUCT@
Agent's lowercase, two-letter product code. Example: For a Windows OS agent, **@PRODUCT@_trapcnfg.xml** resolves to **nt_trapcnfg.xml**.

@ITMHOME@
IBM Tivoli Monitoring installation path. Example: If this is a Linux system and the default installation path is used, **@ITMHOME@** resolves to **/opt/IBM/ITM/**.

@MSN@
Agent's managed system name (not the subnode name). Example: If the agent's managed system name is `primary:icvw3d62:nt`, **@MSN@** resolves to **primary-icvw3d62-nt**.

@TASKNAME@
Agent's process name. Examples: **klzagent**; **kntcma**.

@VERSION@
Agent's product version string. Example: If the agent's version is Tivoli Monitoring 6.2.2 fix pack 2, **@VERSION@** resolves to **06-22-02**.

@HOSTNAME@
Computer host name. Example: **myhost**.

@IPADDRESS@
Computer network interface IP address. Example: If the agent's IP address is 9.42.38.333, **@IPADDRESS@** resolves to **9-42-38-333**.

@OSTYPE@
Operating system type. Examples: **linux**; **win2003**.

@OSVERSION@
Operating system version. Examples: Red Hat Enterprise Linux Version 5 (64 bit) resolves to **2-6-18-128-el5**; Windows 2003 (32 bit) with Service Pack 2 resolves to **5-2-sp2**

@SYSTEMID@
Computer system identifier. Example: System ID `icvr4a04.mylab.mycity.ibm.com` resolves to **icvr4a04-mylab-mycity-ibm-com**.

For detailed information on using the centralized configuration facility, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Notes:

1. All special characters in the parameter values for all keywords other than **@ITMHOME@** are converted to dashes (-). For example, if the IP address is 9.42.38.233, keyword **@IPADDRESS@** resolves to **9-42-38-233**.

   The value for **@ITMHOME@**, however, remains unchanged.
2. The value of **SETENCR_IRA_CONFIG_SERVER_PASSWORD** may be either plain text or encrypted when saved in the *pc*`_silent_install.txt` silent response file. Plain-text values are encrypted when created in the agent environment file. Encrypted values are created as specified. The `itmpwdsnmp` utility is used interactively on another system to encrypt the password string if desired; see the *IBM Tivoli Monitoring: Administrator's Guide*.

Part 4. Postinstallation configuration and customization

The chapters in this section cover configuration procedures that you can perform at any time after you complete the installation and basic configuration.

Chapter 12, “Configuring IBM Tivoli Monitoring components,” on page 363, introduces Manage Tivoli Enterprise Monitoring Services, a utility you can use to start and stop components and to configure or reconfigure components. It includes instructions for starting and stopping Manage Tivoli Enterprise Monitoring Services and for using the utility to reconfigure a Tivoli Enterprise Monitoring Server, to configure or change connections between agents and monitoring servers, to configure user authentication, and to configure failover support. This chapter also contains instructions for specifying network interfaces for the portal server to use when the monitoring server has multiple TCP/IP interfaces, controlling port number assignments, and configuring the heartbeat interval for agents and monitoring servers.

Chapter 13, “Additional Linux and UNIX configuration steps,” on page 383 contains instructions for disabling fsync() calls, configuring permissions for a monitoring server on non-NIS Solaris computers, and increasing virtual memory on AIX for large environments.

Chapter 15, “Additional Tivoli Enterprise Portal configurations,” on page 397 contains instructions for securing communication between clients and the portal server, creating a connection between a portal client and an external Web server, using network address translation (NAT) with a firewall, defining an interface for multiple network interface cards. It also provides several firewall scenarios.

Chapter 16, “Configuring IBM Tivoli Monitoring Web Services (the SOAP Server),” on page 415 provides instructions on controlling access to the SOAP server installed on hub monitoring servers and for configuring connections between hubs.

Chapter 17, “Performance tuning,” on page 421 discusses considerations for optimizing components within your Tivoli Monitoring environment.

Chapter 12. Configuring IBM Tivoli Monitoring components

Although the majority of configuration is done during the product installation, you can use the Manage Tivoli Enterprise Monitoring Services tool to configure components at any time. You can also use the Manage Tivoli Enterprise Monitoring Services tool to start and stop components.

Note: You can also perform many of these configuration and start and stop procedures from the command-line. Where this is possible, the command is included. See the *IBM Tivoli Monitoring: Command Reference* for a complete description, including parameters, of the commands that you can use in the installation and configuration of IBM Tivoli Monitoring.

You can perform the tasks in Table 63 with the Manage Tivoli Enterprise Monitoring Services tool:

Table 63. Configuration tasks available through Manage Tivoli Enterprise Monitoring Services

Task	Where to find information
Start Manage Tivoli Enterprise Monitoring Services	"Starting Manage Tivoli Enterprise Monitoring Services"
Change the configuration of the monitoring server	"Changing the configuration of the Tivoli Enterprise Monitoring Server" on page 364
Configure agents and other monitoring components	"Configuring or changing the monitoring server connection for agents" on page 366
Start and stop components	"Starting and stopping components" on page 368
Monitor the status of remote monitoring servers and agents by configuring heartbeat monitoring.	"Configuring the heartbeat interval" on page 372

Starting Manage Tivoli Enterprise Monitoring Services

Depending on the operating system you are using, the procedure for starting Manage Tivoli Enterprise Monitoring Services is different.

Starting Manage Tivoli Enterprise Monitoring Services on Windows computers

Use the following steps to start Manage Tivoli Enterprise Monitoring Services on a computer running Windows:

1. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

Starting Manage Tivoli Enterprise Monitoring Services on Linux or UNIX computers

Use the following steps to start Manage Tivoli Enterprise Monitoring Services on a computer running Linux or UNIX:

1. Change to the bin directory:

   ```
   cd install_dir/bin
   ```

2. Run the following command using the parameters described in Table 64:

   ```
   ./itmcmd manage [-h install_dir] [-s]
   ```

Table 64. Parameters for the itmcmd manage command

-h	(optional) An option used to specify the installation directory.
install_dir	The installation directory for IBM Tivoli Monitoring.

Table 64. Parameters for the itmcmd manage command (continued)

-s	(optional) Option to specify safe mode operation. Safe mode invokes the JRE with the **-nojit** option (no just-in-time compiler). If you encounter a Java failure error, try running the command as before, but also specifying the **-s** option. Entering the above commands with **-?** displays the syntax for using the **-s** option.

The Manage Tivoli Enterprise Monitoring Services utility is displayed.

Note: Note that the **Platform** column for agents lists the platform that the binary code was built on, not the platform that you are running on.

Changing the configuration of the Tivoli Enterprise Monitoring Server

You can change the basic configuration of the monitoring server through Manage Tivoli Enterprise Monitoring Services. Use the following steps:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click the monitoring server.
2. Click **Reconfigure** (on Windows) or **Configure** (on UNIX).
3. Identify the communications protocol for the monitoring server. You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify three methods for communication - this enables you to set up backup communication methods. If the method you've identified as Protocol 1 fails, Protocol 2 is used.
4. Click **OK**.
5. Complete the fields settings listed in Table 65 for the communications protocol for the monitoring server and click **OK**.

Table 65. Communications protocol settings

Field	Description
IP.UDP Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server. Note that the Tivoli Enterprise Monitoring Server supports both IPV4 and IPV6 addressing formats.
Port # or Port Pools	The listening port for the hub monitoring server.
IP.PIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server. Note that the Tivoli Enterprise Monitoring Server supports both IPV4 and IPV6 addressing formats.
Port Number	The listening port for the monitoring server. The default number is 1918.
IP.SPIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server. Note that the Tivoli Enterprise Monitoring Server supports both IPV4 and IPV6 addressing formats.
Port number	The listening port for the hub monitoring server. The default value is 3660.
SNA Settings	
Network Name	The SNA network identifier for your location.

Table 65. Communications protocol settings (continued)

Field	Description
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
TP Name	The transaction program name for the monitoring server.

6. If the monitoring server was running when you began the configuration process, after the reconfiguration is complete, you are asked if you want it restarted.

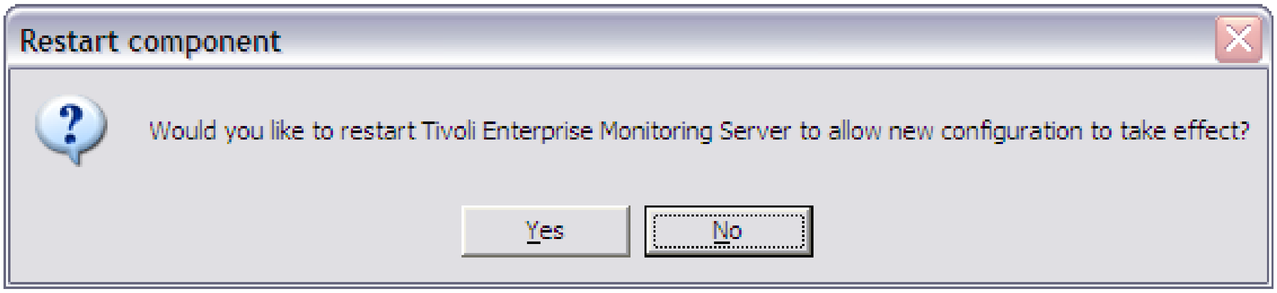

Restart component
Would you like to restart Tivoli Enterprise Monitoring Server to allow new configuration to take effect?
Yes
No

Figure 96. Restart Component window: Tivoli Enterprise Monitoring Server

Reply **Yes** or **No**.

On Linux and UNIX, you can also use the **itmcmd config -S** command to change the configuration of a monitoring server. When the CLI completes the reconfiguration, if the monitoring server was running when you began this process, you are asked if you want it restarted:

```
Would you like to restart the component to allow new configuration to take effect [1=Yes, 2=No]
(Default is: 2):
```

Reply **1** or **2**, as appropriate.

If you choose the restart, the monitoring server is stopped and then started again. These actions are necessary to force the server to read your changed configuration (which is always read at server startup). On UNIX platforms the component should be restarted with the same user that it previously ran on. If the monitoring server was not running when reconfigured, no action is performed, and the server remains stopped.

Notes:

1. Use caution when starting, stopping, or restarting the Tivoli Enterprise Monitoring Server, as it is a key component.
2. You cannot configure a hub monitoring server and a remote monitoring server on the same system because you cannot have two processes listen to the same IP port on the same system.
   - You can configure any number of remote monitoring servers on the same system as long as each reports to a different hub and uses a different port number.
   - You can configure any number of hub monitoring servers on the same system as long as each uses a different port number.
   - If a hub monitoring server and a remote monitoring server are configured on the same system, the remote monitoring server must report to a hub on another system using a port other than the one used by the hub running on the same system.

- You cannot have two monitoring servers talking to each other over IP from the same system unless one of them is a high-availability hub monitoring server, because a high-availability hub is isolated to a private IP address. See the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*.

Configuring or changing the monitoring server connection for agents

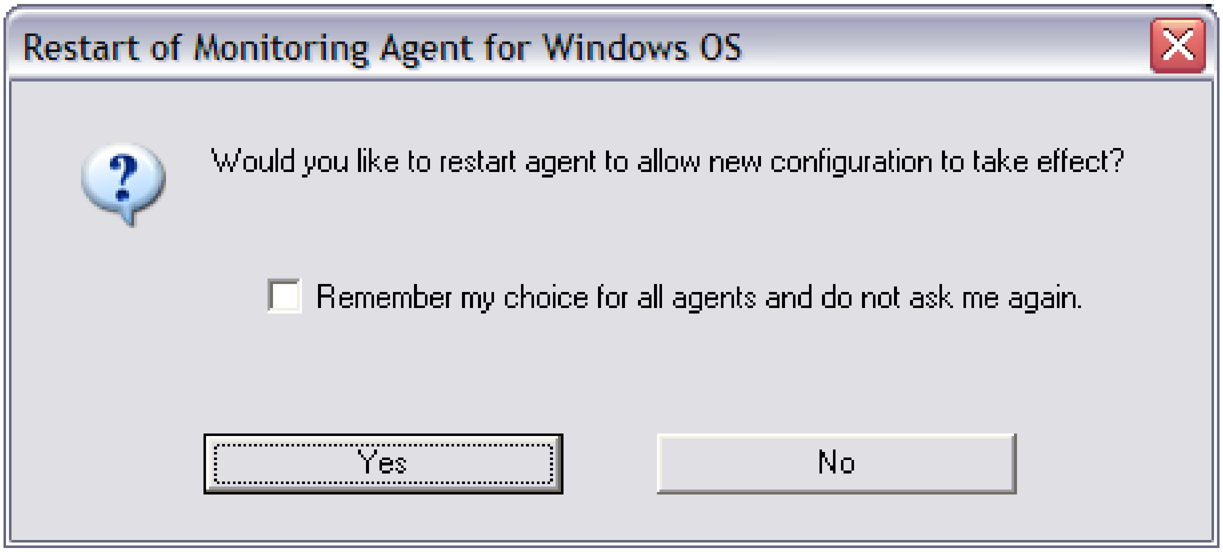
To configure or change the monitoring server connection to the agents, use the following procedure.

1. In the Manage Tivoli Enterprise Monitoring Services window, select the agent whose connection you want to configure. You can select multiple agents by holding down the Shift key or Control key and selecting agents.
2. Click **Actions** → **Reconfigure**.
3. Identify the communications protocol for communication with the monitoring server. You have four choices: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify three methods for communication - this enables you to set up backup communication methods. If the method you've identified as Protocol 1 fails, Protocol 2 is used.
4. Click **OK**.
5. Complete the following fields and click **OK**:

Table 66. Communications protocol settings

Field	Description
IP.UDP Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server. Note that the Tivoli Enterprise Monitoring Server supports both IPV4 and IPV6 addressing formats.
Port # or Port Pools	The listening port for the hub monitoring server.
IP.PIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server. Note that the Tivoli Enterprise Monitoring Server supports both IPV4 and IPV6 addressing formats.
Port Number	The listening port for the monitoring server. The default number is 1918.
IP.SPIPE Settings	
Hostname or IP Address	The host name or IP address for the hub monitoring server. Note that the Tivoli Enterprise Monitoring Server supports both IPV4 and IPV6 addressing formats.
Port number	The listening port for the hub monitoring server. The default value is 3660.
SNA Settings	
Network Name	The SNA network identifier for your location.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is "CANCTDCS."
TP Name	The transaction program name for the monitoring server.

6. If the agent was running when you began the configuration process, after the reconfiguration is complete, you are asked if you want the agent restarted.



Restart of Monitoring Agent for Windows OS

Would you like to restart agent to allow new configuration to take effect?

□ Remember my choice for all agents and do not ask me again.

Figure 97. Restart of Monitoring Agent window

Reply **Yes** or **No**.

On Linux and UNIX, you can also use the **itmcmd config -A** command to change the configuration of a monitoring agent. When the CLI completes the reconfiguration, if the agent was running when you began this process, you are asked if you want the agent restarted:

```
Would you like to restart the component to allow new configuration to take effect [1=Yes, 2=No]
(Default is: 1):
```

Reply **1** or **2**, as appropriate.

If you choose the restart, the agent is stopped and then started again. These actions are necessary to force the agent to read your changed configuration (which is always read at agent startup). On UNIX platforms the component should be restarted with the same user that it previously ran on. If the agent was not running when reconfigured, no action is performed, and the agent remains stopped.

Note: If you upgrade an agent and the upgrade includes new columns for an existing attribute group, you must stop and restart history collection to get the new attributes to be picked up by the Tivoli Enterprise Monitoring Server when history is being collected at the Tivoli Enterprise Monitoring Server. This applies to self-describing agents as well.

Configuring the handling of pure events by the monitoring server

When the Tivoli Enterprise Monitoring Server has EIF Event Integration enabled, and data arrives from agents at a rapid pace, multiple rows of data either get put in a single event going to the event receiver or they get discarded. You can use the following procedure to configure the monitoring server to support one event per row of pure data. This ensures that when multiple rows of data arrive from the agent in quick succession, they are not concatenated into a single event; each row of data will generate a corresponding event.

1. Create a file KPXATRGP on the Tivoli Enterprise Monitoring Server in the following directory:
   - $CANDLEHOME\CMS (C:\IBM\ITM\CMS) for Windows systems.
   - $CANDLEHOME/tables/ for UNIX systems.
2. Add the entries for pure event tables in the format <application-name>.<table-name> followed by a new line, for example:

   ```
   KNT.NTEVTLOG
   TES00.TES123450
   ```

3. Always restart the Tivoli Enterprise Monitoring Server after adding a new entry to this file.

This enables the feature for pure event tables only. Each row of data from the agent will result in an event. You should verify that all events from the agents show up in the EIF cache file or Event Server. The feature cannot be enabled for sample events. If there is a constant high volume of data from the agents, enabling this feature for all pure event tables will slow down the Tivoli Enterprise Monitoring Server performance. If you want to enable this feature you must enable it on all Tivoli Enterprise Monitoring Servers (hub and remote) in the environment.

Starting and stopping components

You can start and stop the IBM Tivoli Monitoring components from Manage Tivoli Enterprise Monitoring Services. Use the following steps:

1. Right-click the component (such as a specific agent or the Tivoli Enterprise Portal Server) that you want to start or stop.
2. Click **Start**, **Stop**, or **Recycle** (Windows only) from the menu.

Note: When a hub Tivoli Enterprise Monitoring Server is recycled, all current events are recycled.

You can also use the following commands to start and stop components, including agents built by the Agent Builder and System Monitor Agents:

itmcmd server
Starts and stops a UNIX monitoring server.

itmcmd agent
Starts and stops a UNIX monitoring agent.

tacmd startAgent
Starts both Windows, Linux, and UNIX monitoring agents.

tacmd stopAgent
Stops both Windows, Linux, and UNIX monitoring agents.

See the *IBM Tivoli Monitoring: Command Reference* for the syntax of these commands.

Specifying network interfaces

If there are multiple TCP/IP interfaces on the computer on which a monitoring server is running, you need to identify which interfaces monitoring agents or the Tivoli Enterprise Portal Server should use when connecting to the monitoring server. Setting network interfaces affects all the components installed on the local computer.

To specify the network interfaces to be used by the portal to connect to a hub monitoring server, or by a monitoring agent to connect to a hub or remote, complete these steps:

Note: For instructions on configuring a network interface list on z/OS, see the *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

1. In the Manage Tivoli Enterprise Monitoring Services window, select **Actions → Advanced → Set Network Interface**.
2. On the Set Desired Network Interface window specify the network interface or interface you want to use.

   Specify each network adapter by the host name or IP address to be used for input and output. Use a blank space to separate the entries. If your site supports DNS, you can specify IP addresses or short host names. If your site does not support DNS, you must specify fully qualified host names.
3. Click **OK** to close save the settings and close the window.

Controlling port number assignments

IBM Tivoli Monitoring assigns port numbers to each component in an installation to be used for communication with other components. Default well-known ports are reserved for major components such as monitoring servers and the portal server. For all other components, an algorithm calculates the listening port to reserve.

You might want to change or add to the default assignments under some conditions, for example, when the default port assigned to a monitoring agent has already been reserved by another application, or the portal server requires a second port number for communication through a firewall with Network Address Translation (NAT).

Configuring port number assignments for the monitoring server

The default IP.UDP and IP.PIPE listening port setting for the Tivoli Enterprise Monitoring Server is 1918. For IP.SPIPE, it is 3660. For SNA, it is 135. While you can specify a different port during or after installation, it is best to use the default setting. To reconfigure the port after installation, see “Configuring or changing the monitoring server connection for agents” on page 366.

Configuring port number assignments for the portal server

Communications between Tivoli Enterprise Portal clients and the Tivoli Enterprise Portal Server are controlled by a portal server interface definition. The default interface definition assigns port 15001 to the Tivoli Enterprise Portal Server and ports 1920 (for HTTP requests) and 3661 (for HTTPS) to the integrated Web server that is installed with the portal server. You can define additional interfaces to allow access through a firewall with NAT or through a secondary Network Interface Card (NIC). See “Firewall network address translation (NAT) or multiple network interface cards” on page 408.

The following sections describe how to change port number assignments on the portal server for connections to Tivoli Enterprise Portal clients (browser clients and desktop clients).

Changing the port number for browser client connections to the portal server

A portal server on Windows, Linux, or UNIX uses port 1920 for HTTP connections and 3661 for HTTPS connections from portal browser clients.

Do not change the default port settings, especially on multifunction UNIX and Linux systems, since many components might be located on the same system and some of these components might depend on the default values being used for HTTP and HTTPS ports.

If you need to change the default settings, you can change them by using the KDE_TRANSPORT environment variable:

On Windows:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server**, point to **Advanced**, and select **Edit ENV File** to open the KFWENV file.
2. Add the following line to the file:

   ```
   KDE_TRANSPORT=HTTP:1920 HTTPS:3661
   ```

   Substitute the port numbers you want to use.
3. If a KDC_FAMILIES environment variable exists in the file, copy the settings from that variable to KDE_TRANSPORT (except the ones you want to override). KDE_TRANSPORT supersedes and overrides KDC_FAMILIES.
4. Save the file.
5. Recycle the portal server. (Right-click **Tivoli Enterprise Portal Server** and select **Recycle**.)

On Linux or AIX:

1. Change to the *install_dir*/config directory (where *install_dir* is the IBM Tivoli Monitoring installation directory).
2. Add the following line to the cq.ini file:

   ```
   KDE_TRANSPORT=HTTP:1920 HTTPS:3661
   ```

   Substitute the port numbers you want to use.
3. If a KDC_FAMILIES environment variable exists in the file, copy the settings from that variable to KDE_TRANSPORT (except the ones you want to override). KDE_TRANSPORT supersedes and overrides KDC_FAMILIES.
4. Recycle the portal server.

Special settings:

The KDE_TRANSPORT environment variable keyword redefines the ports to be use by the HTTP and HTTPS daemons. Certain special settings are also provided to meet special needs.

HTTPS:0
Eliminates error messages when the HTTPS daemon server is active and failing to obtain or bind to family ip.ssl (ip.ssl.https:3661).

HTTP:0, HTTPS:0
These settings disable port allocation and port bind errors for the HTTP (1920) and HTTPS (3661) default ports.

HTTP_SERVER:n
This KDE_TRANSPORT environment variable keyword disables HTTP and HTTPS daemon services. Do not specify this for a hub monitoring server or for the portal server.

HTTP_CONSOLE:n
This KDE_TRANSPORT environment variable keyword disables the CT/Service Console facility of the HTTP daemon service. HTTP_CONSOLE:N removes the process from the published Tivoli service index; this makes the process inaccessible from the CT/Service Console.

Changing the port number for desktop client connections to the portal server

Use the following procedure for each desktop client instance you want to change:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click the Tivoli Enterprise Portal Desktop instance that you want to change, and select **Reconfigure**.

   The Configure Application Instance window opens.
2. In the **Parms** list, scroll down to `cnp.http.url.port` and double-click.
3. On the Edit window, perform the following steps:
   a. Change the port number value to the port number you want.
   b. Select the **In Use** check box.

      Note: If you fail to select this check box, the port number value that you entered will revert to the original value.
   c. Click **OK**.
4. On the Configure Application Instance window, verify that the port number value (in the **Value** column) for `cnp.http.url.port` has changed.
5. Click **OK**.

Configuring port number assignments for monitoring agents

IBM Tivoli monitoring uses the following algorithm to allocate port numbers for monitoring agents to reach the monitoring server:

reserved port = *well-known port* + (N*4096)

where:

- *well-known port* is the port number assigned to the monitoring server, for example, 1918.
- *N* indicates the position of the monitoring agent in the startup sequence for agents.

For example, if there are two monitoring agents on a system, and the monitoring server uses port 1918, the first monitoring agent in the startup sequence is assigned port 6014 (1918 + 1*4096) and the second agent to start is assigned port 10110 (1918 + 2*4096).

For "piped" protocols such as IP.PIPE and IP.SPIPE (but not IP or SNA), you can control the way port numbers are assigned to a monitoring agent by using the SKIP and COUNT parameters on the KDE_TRANSPORT environment variable for agents running on Windows or the KDC_FAMILIES environment variable for agents running on Linux/UNIX. See the following example:

```
KDE_TRANSPORT=IP.PIPE PORT:1918 COUNT:1 SKIP:2 IP use:n SNA use:n IP.SPIPE use:n
```

See also the following information about using the IP.PIPE and IP.SPIPE protocols and parameters:

- The PORT parameter specifies the well-known port for the monitoring server.
- The COUNT:*N* parameter is the mechanism for reserving IP.PIPE ports for components that connect to the monitoring server. *N* is the number of IP.PIPE ports to reserve on a host in addition to the well-known port for the monitoring server. Use the COUNT parameter to reserve ports for components that must be accessible from outside a firewall. Accessibility from outside the firewall requires IP.PIPE ports and because these ports must be permitted at the firewall, the ports must be predictable.

  For example, if the well-known port is 1918, COUNT:3 starts the search at port 6014 (1918 + 1*4096). If the agent process cannot bind to port 6014, the algorithm tries port 10110 (1918 + 2*4096). If port 10110 is not available, the search goes to port 14206 (1918 + 3*4096).

  The agent is assigned to the first available port encountered in the search. The process fails to start if the search reaches the highest port without a successful binding (port 14206 in this example).
- The SKIP:*N* parameter specifies the number of ports to skip when starting the search for an available port using the port assignment algorithm. Use the SKIP parameter for components that do not need access across a firewall.

  For example, if the well-known port is 1918, SKIP:2 specifies to start the search at port 10110 (1918 + 2*4096), skipping ports 1918 and 6014 (1918 + 1*4096). The algorithm continues searching until it finds an available port.
- The USE parameter enables or disables a protocol. To disable a protocol, specify `use:n`. To enable a protocol, specify `use:y`. This parameter has no default.

Note: Tivoli Monitoring agents allocate ports 1920 and 3661 as HTTP and HTTPS listener ports.

Example

The example in Table 67 shows the coding to use on a system that contains the components shown:

Table 67. Using COUNT and SKIP variables to assign port numbers

Component	Coding
Tivoli Enterprise Monitoring Server	The monitoring server uses port 1918.
Warehouse Proxy Agent Requires firewall access	`KDE_TRANSPORT=IP.PIPE COUNT:1` This coding reserves port 6014 (1918 + 1*4096) for the Warehouse Proxy Agent.

Table 67. Using COUNT and SKIP variables to assign port numbers (continued)

Component	Coding
Windows OS agent Does not require firewall access	`KDE_TRANSPORT=IP.PIPE SKIP:2` With this coding, the port assignment algorithm skips ports 1918 and 6014 (reserved for the monitoring server and Warehouse Proxy Agent), and starts at port 10110 (1918 + 2*4096). If the Windows OS agent fails to open port 10110, the agent tries port 14206 and so on, until it finds an available port or exhausts all possibilities.

Adding the KDE_TRANSPORT environment variable

The KDE_TRANSPORT environment variable must be added to the appropriate file (*ENV file on Windows).

Note: The KDE_TRANSPORT variable supersedes and overrides the KDC_FAMILIES variable. If a KDC_FAMILIES variable exists in the file, merge the KDC_FAMILIES settings with the KDE_TRANSPORT settings. Copy the KDC_FAMILIES settings that you want to keep to the new KDE_TRANSPORT variable.

Use the following procedures to add the KDE_TRANSPORT environment variable to the appropriate file:

On Windows:

Add the KDE_TRANSPORT environment variable to the ENV file for the component.

For example, the ENV file for the Windows OS agent is named KNTENV, where NT is the product code for the Windows OS agent. For a list of product codes, see Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815.

Edit the ENV file:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click the component that you want to change, point to **Advanced**, and select **Edit ENV File**.
2. Add a line similar to the following example:

   ```
   KDE_TRANSPORT=IP.PIPE PORT:1918 COUNT:N SKIP:N
           IP use:n SNA use:n IP.SPIPE use:n
   ```

   where *N* is the number of ports to reserve (COUNT:*N*) or the number of ports to skip (SKIP:*N*). This example uses the IP.PIPE protocol. It also applies to IP.SPIPE.
3. Search the file for a KDC_FAMILIES environment variable. If a KDC_FAMILIES variable exists in the file, merge its settings with the new KDE_TRANSPORT variable. The KDE_TRANSPORT variable supersedes and overrides the KDC_FAMILIES variable.
4. Save the file.
5. Recycle the component. (Right-click the component and select **Recycle**.)

Configuring the heartbeat interval

IBM Tivoli Monitoring uses a heartbeat mechanism to monitor the status of remote monitoring servers and monitoring agents. The different monitoring components in the monitoring architecture form a hierarchy (shown in Figure 98 on page 373) across which the heartbeat information is propagated.

The hub monitoring server maintains status for all monitoring agents. Remote monitoring servers offload processing from the hub monitoring server by receiving and processing heartbeat requests from monitoring agents, and communicating only status changes to the hub monitoring server.

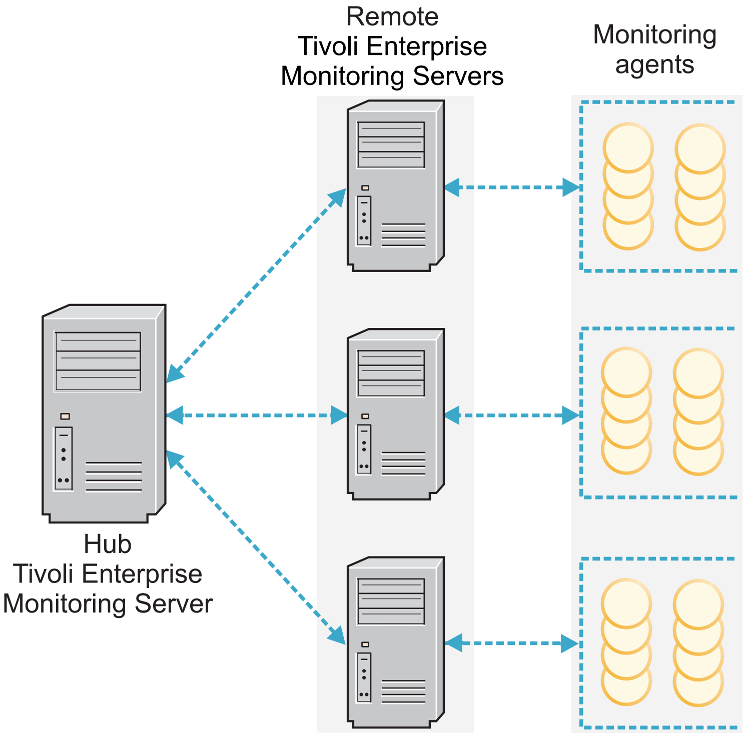


Figure 98. Hierarchy for the heartbeat interval

At the highest level, the hub monitoring server receives heartbeat requests from remote monitoring servers and from any monitoring agents that are configured to access the hub monitoring server directly (rather than through a remote monitoring server). The *default* heartbeat interval used by remote monitoring servers to communicate their status to the hub monitoring server is 3 minutes. The default heartbeat interval of 3 minutes for monitoring servers is suitable for most environments, and should not need to be changed. If you decide to modify this value, carefully monitor the system behavior before and after making the change.

At the next level, remote monitoring servers receive heartbeat requests from monitoring agents that are configured to access them. The *default* heartbeat interval used by monitoring agents to communicate their status to the monitoring server is 10 minutes.

You can specify the heartbeat interval for a node (either a remote monitoring server or a remote monitoring agent) by setting the **CTIRA_HEARTBEAT** environment variable. For example, specifying **CTIRA_HEARTBEAT=5** sets the heartbeat interval to 5 minutes. The *minimum* heartbeat interval that can be configured is 1 minute.

- For monitoring servers on Windows computers, you can set this variable by adding the entry to the KBBENV file. You can access this file from the Manage Tivoli Enterprise Monitoring Services utility by right-clicking **Windows OS Monitoring Agent** and clicking **Advanced -> Edit ENV File**. Note that you must stop and restart the monitoring server for the changes to the KBBENV file to take effect.
- For monitoring servers on Linux and UNIX computers, you can set the **CTIRA_HEARTBEAT** variable by adding the entry to the monitoring server configuration file. The name of the monitoring server configuration file is of the form *hostname*_ms_*temsname*.config. For example, a remote monitoring server named `REMOTE_PPERF06` running on host `pperf06` has a configuration filename of `pperf06_ms_REMOTE_PPERF06.config`. Note that you must stop and restart the monitoring server for the configuration changes to take effect.

- For remote monitoring servers, you can set this variable by adding an entry to the KBBENV file. You can access this file from Manage Tivoli Enterprise Monitoring Services by right-clicking **Tivoli Enterprise Monitoring Server** and clicking **Advanced → Edit ENV File**. You must stop and restart the monitoring server for changes to the KBBENV file to take effect.
- For Windows OS agents, you can set this variable by adding the entry to the KNTENV file. You can access this file from Manage Tivoli Enterprise Monitoring Services by right-clicking **Windows OS Monitoring Agent** and clicking **Advanced → Edit ENV File**. You must stop and restart the monitoring agent for the changes to the KNTENV file to take effect.
- For agents on Linux and UNIX computers, you can set the **CTIRA_HEARTBEAT** variable by adding an entry to the agent .ini file (for example, lz.ini, ux.ini, ua.ini). When the agent is stopped and restarted, the agent configuration file is recreated using settings in the .ini file.

When a monitoring agent becomes active and sends an initial heartbeat request to the monitoring server, it communicates the desired heartbeat interval for the agent in the request. The monitoring server stores the time the heartbeat request was received and sets the expected time for the next heartbeat request based on the agent heartbeat interval. If no heartbeat interval was set at the agent, the default value is used.

Changes to offline status typically require two missed heartbeat requests for the status to change. Offline status is indicated by the node being disabled in the portal client's Navigator View. If the heartbeat interval is set to 10 minutes, an offline status change would be expected to take between 10 and 20 minutes before it is reflected on the portal client's Navigator View.

Attention: Lower heartbeat intervals increase CPU utilization on the monitoring servers processing the heartbeat requests. CPU utilization is also affected by the number of agents being monitored. A low heartbeat interval and a high number of monitored agents could cause the CPU utilization on the monitoring server to increase to the point that performance related problems occur. If you reduce the heartbeat interval, you must monitor the resource usage on your servers. A heartbeat interval lower than 3 minutes is not supported.

Restarting the Tivoli Enterprise Portal Server after reconfiguration

The Tivoli Enterprise Portal Server must be stopped before it can be reconfigured.

- If the portal server is running and you ask to reconfigure it, a warning is first displayed informing you that the server must be stopped and asking if you want to continue.
- If the portal server is running and you allow it to be stopped for reconfiguration, you are asked when the configuration process ends if you want the server restarted.

Switching to a different Tivoli Enterprise Portal Server database

You can change to a different RDBMS after you have installed the Tivoli Enterprise Portal Server. To do this:

1. Bring up the Manage Tivoli Enterprise Monitoring Services main screen, and right-click the Tivoli Enterprise Portal Server entry. See Figure 99 on page 375.

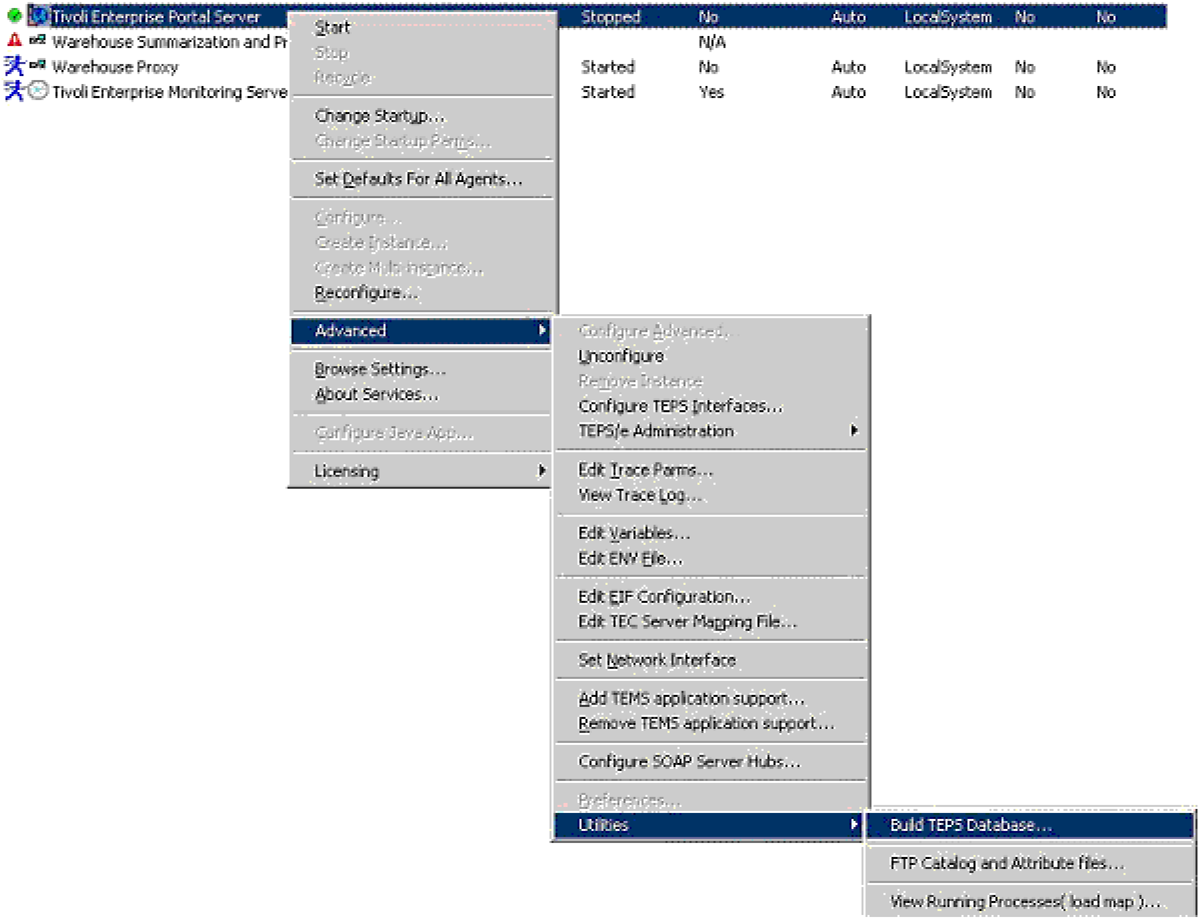


Figure 99. Manage Tivoli Enterprise Monitoring Services Advanced Utilities window

2. From the pop-up menu, select Advanced → Utilities → Build TEPS Database.
   - If the only RDBMSes installed on this computer are DB2 Database for Linux, UNIX, and Windows and the portal server's embedded Derby database, select the appropriate database manager, as shown in Figure 100.

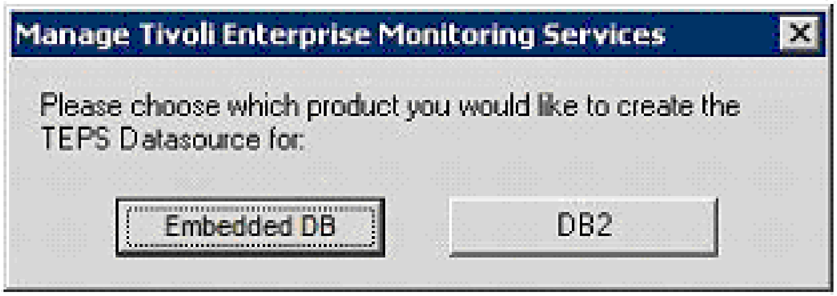
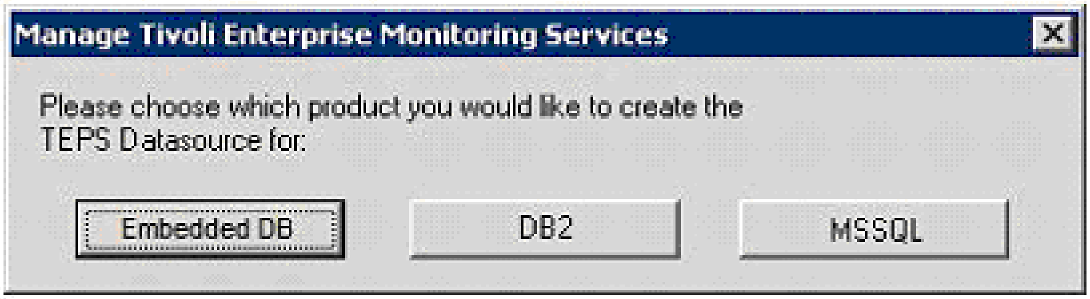

Manage Tivoli Enterprise Monitoring Services
Please choose which product you would like to create the TEPS Datasource for:
Embedded DB
DB2

Figure 100. The Manage Tivoli Enterprise Monitoring Services select the new portal server database window. The only available database managers are DB2 for Linux, UNIX, and Windows and Derby.

   - If this computer is running both DB2 Database for Linux, UNIX, and Windows, Microsoft SQL Server, and embedded Derby, select the appropriate database manager, as shown in Figure 101 on page 376.


Manage Tivoli Enterprise Monitoring Services

Please choose which product you would like to create the TEPS Datasource for:

Figure 101. The Manage Tivoli Enterprise Monitoring Services select the new portal server database window. The available database managers are DB2 for Linux, UNIX, and Windows, SQL Server, and Derby.

Note: Changing to a different database does not migrate the portal server data stored in it.

3. Continue with the database configuration, as explained in number 12 on page 188 under "Prerequisites for the single-computer installation" on page 187.

Silent configuration of the Performance Analyzer

The silent method of configuration is useful for advanced users who want to supply configuration information once through the silent configuration file for the Performance Analyzer.

Before you begin

With silent configuration, you can automate the configuration steps and avoid manual configuration of the Performance Analyzer.

Procedure

1. Locate the default silent configuration file for the Performance Analyzer.
   - On Windows systems, one sample `silent_server.txt` file is used by all agents for silent configuration. The sample file is provided on the product installation media.
   - On UNIX and Linux systems, each agent has a separate configuration file. For Performance Analyzer, the default configuration file is `pa_silent_config.txt`. For example: `<ITM_HOME>/samples/pa_silent_config.txt`.
2. Review the default configuration parameters in the silent configuration file. If required, update the values according to your requirements.
3. Save the file with a different file name.
4. Run the Performance Analyzer silent configuration command:
   - On Windows systems:
     - For silent configuration at Performance Analyzer installation, go to the IBM Tivoli Monitoring installation directory and run the command:

       ```
       start /wait setup /z"/sf<Complete path to silent_server.txt>" /s /f2"<complete
       path to log file>"
       ```
     - For silent configuration after Performance Analyzer installation, create the silent configuration file by copying Performance Analyzer-specific parameters and monitoring server connection-related parameters from the `silent_server.txt` file. Use the new silent configuration file in the command:

       ```
       <ITM_HOME>\InstallITM\kinconfg.exe -aK<pc> -n<Complete path_to_silent_file>
       ```

       Where <pc> = pa
   - On UNIX and Linux systems, run the following command (from the bin directory of Tivoli Monitoring if that is not added to PATH):

```
itmcmd config -A -h <ITM_HOME> -p <Complete path to silent_config_pa.txt> pa
```

Configuring historical data collection for the Performance Analyzer warehouse agent

You must configure the historical data collection for Tivoli Performance Analyzer to enable trend calculation visualization in the workspaces and reports. Complete the following steps to configure historical data collection for the Performance Analyzer warehouse agent:

1. Start Tivoli Enterprise Portal.
2. In the portal toolbar, select **Edit > History Configuration**. The **History Collection Configuration** window is displayed.
3. To obtain the calculation data, configure the historical data collection for each of your operating system agents, and for your domains by referring to the attribute groups that must be configured. Table 68 shows products and their associated attribute groups.

Table 68. Attribute groups that must be configured

Product		Attribute groups
Operating systems	Windows OS	• Network Interface (Superseded) • Logical Disk • Memory (Superseded) • Processor • System
	Linux	• Linux CPU Averages (Superseded) • Linux Disk (Superseded) • Linux Network (Superseded) • Linux System Statistics (Superseded) • Linux VM Stats (Superseded)
	UNIX OS	• Disk • Network • System
DB2		• Database (Superseded) • System Overview (Superseded) • Tablespace (Superseded)
Oracle		• Oracle Cache Totals • Oracle Database • Oracle Library Cache Usage • Oracle Server • Oracle Session Summary • Oracle Statistics Summary • Oracle Tablespaces

Table 68. Attribute groups that must be configured (continued)

Product		Attribute groups
ITCAM for RT	Client response time	• CRT Application Status • CRT Server Status • CRT SubTransaction Status • CRT Transaction Status
	Robotic response time	• RRT Application Status • RRT Robotic Playback Status • RRT SubTransaction Status • RRT Transaction Status
	Web response time	• WRT Application Status • WRT Client Status • WRT Server Status • WRT SSL Alert Current® Status • WRT SubTransaction Status • WRT TCP Status • WRT Transaction Status
System	VOIS Premium	• KVA LOGICAL PARTITION • KVA DISKS • KVA NETWORK ADAPTERS RATES
	AIX Premium	• KPX LOGICAL PARTITION
VMware		• KVM SERVER • KVM SERVER NETWORK • KVM VM CPU • KVM VM PARTITION • KVM VM NETWORK • KVM VM MEMORY

For each agent, do the following steps:

a. In the navigation pane, right-click the product whose agent you want to configure, and select **Create new collection settings**. The **Create New Collection Settings** window opens.

b. Select an attribute group for that agent, as listed in Table 68 on page 377, and click **OK**.

c. In the **Basic** tab, specify the following settings:

- In the **Collection Interval** field, set the polling interval, for example to **1 hour**.
- In the **Warehouse Interval** field, set the frequency at which data is written to the database.

d. In the **Distribution** tab, specify for which nodes the data should be collected and click **OK**.

e. Select the product again, and set its **Summarization** interval to at least **Hourly** and **Daily**. You can also select additional summarization intervals.

f. Set the **Pruning** interval to what is most appropriate for your system, and click **OK**.

g. Repeat the preceding steps for each agent support agent in your environment. Use collection, summarization, and pruning values that are appropriate to your system. You have now completed configuring the support agents for your environment.

4. Next, configure the historical data collection for Performance Analyzer Warehouse Agent to enable trend calculation visualization on the workspaces and reports:

   a. In the navigation pane, click Performance Analyzer Warehouse Agent.

b. Select the first attribute group for the agent.

c. In the Basic tab, specify the following settings:

   - In the **Collection Interval** field, set the polling interval to **1 hour** for System Health and Disk Health attribute groups and to **1 day** for other attribute groups.
   - In the **Warehouse Interval** field, set the interval to **1 day**.

d. In the **Distribution** tab, start collection for the **AFF_PERF_ANALYZER_WHSE_AGENT** group.

e. Repeat steps c and d for all attribute groups.

f. Select the Performance Analyzer Warehouse Agent application and set **Pruning** for detailed data for all attribute groups to **1 day**.

g. Finally, click **Close** when you have completed configuring the historical data collection for the agents.

Notes:

1. The out-of-the-box OS domain uses 32-bit attribute groups only – the metrics used in this domain are expressed in units that never exceed maximum value available for 32-bit numbers. There is no additional value in switching OS domain tasks, workspaces, and reports to use 64-bit attribute groups.
2. For step 4 on page 378: instead of using History Collection Configuration dialog in Tivoli Enterprise Portal to set up warehousing and pruning of Tivoli Performance Analyzer attribute groups, it is possible to achieve the same with the **tacmd** command. Sample scripts to automate this task can be found on the installation media under the **scripts/***<domain_name>* directory. For example, the OS domain scripts are:

```
scripts/os/histcoll_os.sh
or
scripts/os/histcoll_os.bat
```

Tivoli Monitoring protocol usage and protocol modifiers

IBM Tivoli Monitoring basic services communications are defined by the `KDE_TRANSPORT` environmental variable. In earlier releases `KDC_FAMILIES` was used but `KDE_TRANSPORT` is the successor. The two variables are processed identically, however many agent installers are only aware of the earlier variable. You should examine the agent installation and adopt whatever is used.

IBM Tivoli Monitoring uses other communication protocols such as the following:

- Tivoli Enterprise Portal Client to Tivoli Enterprise Portal Server uses CORBA IIOP after startup.
- Tivoli Enterprise Portal Server, Warehouse Proxy Agent, and Summarization and Pruning Agent can use ODBC or JDBC to communicate with the warehouse database.
- Most IBM Tivoli Monitoring to IBM Tivoli Monitoring communications use the TCP/IP protocol.

Modifiers to the protocols are of the form `attribute:value`. If the modifier occurs first, then it is global in effect. If the modifier occurs after a protocol and before the next protocol, then the effect is only on the protocol that precedes it. The protocol names and modifiers are not case sensitive although they are presented in uppercase in the following paragraphs.

KDE_TRANSPORT Structure

`KDE_TRANSPORT` Structure is a string that lists protocols and modifiers. All protocols are assumed as present. A protocol is activated only if an interface is available for use.

The USE modifier activates a specific protocol if Y is specified, otherwise it deactivates it. The scanning starts with an implicit global `USE:Y`, meaning that all protocols are assumed to be activated by default. For example:

```
IP.PIPE PORT:1918 USE:Y IP.SPIPE USE:N
```

This modifier means that IP.PIPE will be available but not IP.SPIPE. In addition, all the unnamed protocols listed later are activated.

KDE_TRANSPORT Transmission Control Protocol

Transmission Control Protocol (TCP) is a connection oriented protocol. Connection is made through a port number such as 0-65535. The connection continues until the application takes it down. Here are the protocol names:

```
IP.PIPE - tcp
IP.SPIPE - secure tcp
IP6.PIPE - ipv6 tcp
IP6.SPIPE - ipv6 secure tcp
```

The secure protocols are implemented with the Global Secure Toolkit (GSKIT) component.

The TCP protocol modifiers are listed in the sections below.

PORT

PORT defaults to 1918 for TCP and 3660 for secure TCP. These numbers are registered with the Internet Assigned Numbers Authority, for more information go to http://www.iana.org/assignments/port-numbers.

PORT defines the base port number. For a Tivoli Enterprise Monitoring Server, the base number is the listening port and the port that agents connect to. For agents, IBM Tivoli Monitoring processing attempts to open a listening port at number base+*N**4096, where *N* is a number from 1 to 15. If one is already in use Tivoli Monitoring attempts to open a listening port at the next higher iteration. If no Tivoli Enterprise Monitoring Server is present the base port is reserved, in the event a Tivoli Enterprise Monitoring Server will be started later.

An Agent uses the listening port for two main purposes:

- Tivoli Enterprise Monitoring Server requesting real-time data from the agent,
- An agent receiving notifications from the Tivoli Enterprise Monitoring Server, such as awareness of a Warehouse Proxy Agent re-registration at a new IP address or port number.

Example: `IP.PIPE PORT:1918 USE:Y`

Note: In this definition mode there are a maximum of 15 agents on a server. If you have a scenario with more than 15 agents on a server, see the **EPHEMERAL** modifier section below.

SKIP and COUNT

SKIP and COUNT modifiers are used to control the port search algorithm. Default search is for baseport+*N**4096 where *N* is a number from 1 to 15. The SKIP modifier forces it to start with *N* equal to the SKIP value in the above calculation. The COUNT modifier controls the number of attempts that are made.

The following modifier is commonly used for the Warehouse Proxy Agent:

```
IP.PIPE PORT:1918 SKIP:15 COUNT:1 USE:Y
```

The only port that will be checked with this modifier is `1918+15*4096` or `63358`. This means that the Warehouse Proxy Agent will have a fixed IP address even when a Tivoli Enterprise Monitoring Server and other agents are starting up. Having a fixed port number is required if firewall rules are in place.

EPHEMERAL

This modifier has three different values:

- A value of Y means that the connection to the Tivoli Enterprise Monitoring Server listening port is used for all communications. The agent does not require a separate listening port. This means fewer ports are used, which can be important when firewall rules are in place. It is also a way to avoid the 15 agent limitation. There is a disadvantage in the case of historical data collection. The historical data must be either stored at the Tivoli Enterprise Monitoring Server or, if the historical data is being stored at the agent, a Warehouse Proxy Agent must be running on the same server as the Tivoli Enterprise Monitoring Server that the agent is reporting to.

- A value of OUTBOUND means the same as the value Y above.
- A value of INBOUND can be used at a Tivoli Enterprise Monitoring Server. It means that every agent connecting to the Tivoli Enterprise Monitoring Server is configured to ephemeral mode.

POOL

IBM Tivoli Monitoring processes use opaque ports (also sometimes referred to as ephemeral ports) for communication with the server. These ports are not visible on the outside network. The range of possible opaque port numbers that are used can be controlled with the POOL option. See the following examples:

```
IP.PIPE POOL:50900-51923
IP.UDP POOL:01000-01023 POOL:01024-02048
```

Note: Each pool modifier can specify a maximum of 1024 pool ports, but you can have more than one such specification.

KDE_TRANSPORT User Datagram protocol

Typically this is not the best choice of protocol for IBM Tivoli Monitoring. Its advantage is a somewhat lower storage requirement. The disadvantages are as follows:

- Less reliability since applications are responsible for error recovery.
- Higher CPU resources are required.
- It cannot be used where firewall rules are in place.

The protocol names are as follows:

```
IP.UDP - User Datagram Protocol
IP - Synonym for IP.UDP
IP6.UDP - IP V6 version of IP.UDP
```

The POOL and PORT modifiers can be used for this scenario. The other modifiers are connection oriented.

KDE_TRANSPORT Hypertext Transfer Protocol

Each IBM Tivoli Monitoring process has an internal web server. The `KDE_TRANSPORT` Hypertext Transfer Protocols define what protocols are used to access the internal web server. The web server provides access to the IBM Tivoli Monitoring Service Console, the Tivoli Enterprise Portal client, the SOAP server tryout page, and the Agent Service Index pages (with Tivoli Monitoring V6.2.2 or higher).

The protocol names are as follows:

```
ip.tcp.http - http communications
ip.ssl.https - secure http communications
ip6.tcp.http - ipv6 http communications
ip6.ssl.https - ipv6 secure http communications
```

The protocol modifiers are as follows:

- `HTTP_SERVER`: Defaults to Y. If set to N then the internal web server is not started.
- `HTTP_CONSOLE`: Defaults to Y. If set to N then the IBM Tivoli Monitoring service console is not started.
- `HTTP`: Defaults to 1920. If set to 0, access to non-secure internal web server is disabled.
- `HTTPS`: Defaults to 3661. If set to 0, access to secure internal web server is disabled.

  Note: To meet the FIPS 140-2 requirement for secure communication, you should consider setting `HTTP:0` and only allowing internal web server access with `HTTPS`.
- `POOL`: The HTTP protocol also uses temporary ports and the usage is controlled by separate pool control settings. These are not protocols, but are required for the POOL setting:

```
ip.tcp - pool control for the ip.tcp.http protocol
ip.ssl - pool control for the ip.ssl.https protocol
ip6.tcp - pool control for the ip6.tcp.http protocol
ip6.ssl - pool control for the ip6.ssl.https protocol
```

Here is an example of usage: `IP.TCP.HTTP USE:Y  IP.TCP POOL:20000-20031`. You do not need to specify a protocol that is assumed enabled already, so you could simply add: `IP.TCP POOL:20000-20031`. To completely control pool usage, you must set the POOL value for all enabled protocols.

How to change KDE_TRANSPORT settings

If you decide to change KDE_TRANSPORT you must change the settings on all IBM Tivoli Monitoring tasks, when appropriate. For example, if you decide to turn off the IBM Tivoli Monitoring internal web server using the `HTTP_SERVER:N` modifier, you must make the same change for all IBM Tivoli Monitoring tasks.

Mass change to Windows and Linux/UNIX agents: For details on how to make a mass change to Windows and Linux/UNIX agents, see http://www-01.ibm.com/support/docview.wss?uid=swg21441836.

Linux/UNIX – Tivoli Enterprise Monitoring Server: This environment is similar to agents. The difference is the `config` file generation is only performed when you run the command:

```
itmcmd config -S -t <temsname>
```

Edit the `ms.ini` in the same way as the agent case above, and then reconfigure the Tivoli Enterprise Monitoring Server using the `itmcmd config` command.

i/5: The environment variable is manually changed in:

```
QAUTOTMP/KMSPARM(KBBENV)
```

z/OS: In z/OS these values are kept in the `RKANPARU(KDSENV)` member for Tivoli Enterprise Monitoring Server and in `RKANPARU(KppENV)` for agents. Configuration changes here are manual.

Interactions with other environment variables: `KDEB_INTERFACELIST` and `KDEB_INTERFACELIST_IPV6`

The dash “-“ option is used alone. These environment variables do not scan any of the related interfaces. In such a scenario a protocol might go unused even though it is specified in `KDE_TRANSPORT`. You can also eliminate all the interfaces by name to achieve the same result.

Chapter 13. Additional Linux and UNIX configuration steps

The following sections provide information about additional configuration that may be required for Linux and UNIX systems:

- "Disabling fsync() calls"
- "Configuring permissions for a monitoring server on a non-NIS Solaris computer"
- "Increasing virtual memory on AIX for large environments"
- "Linux requirements for the localhost host name" on page 385
- "Setting **ulimit** values for the Warehouse Proxy Agent" on page 385

Disabling fsync() calls

The KGLCB_FSYNC_ENABLED parameter was introduced in V6.2 for the Tivoli Enterprise Monitoring Server on UNIX and Linux operating systems. This variable can be used to specify whether the **fsync()** system call should be invoked after writes to the filesystem. This configuration variable may be set in the standard configuration file for the monitoring server.

For maximum reliability, by default **fsync()** is called. The **fsync()** system call flushes the filesystem's dirty pages to disk and protects against loss of data in the event of an operating system crash, hardware crash, or power failure. However, the call can have a significant negative effect on performance, because in many cases it defeats the caching mechanisms of the platform filesystem. On many UNIX platforms, the operating system itself syncs the entire filesystem on a regular basis. For example, by default the **syncd** daemon that runs on AIX syncs the filesystem every 60 seconds, which limits the benefit of **fsync()** calls by application programs to protecting against database corruption in the most recent 60-second window.

If the following line is added to the monitoring server configuration file, **fsync()** calls are omitted:

```
KGLCB_FSYNC_ENABLED='0'
```

Configuring permissions for a monitoring server on a non-NIS Solaris computer

If your monitoring server is installed on a non-NIS Solaris computer, you must set the permissions. You do not need to do this for monitoring servers running on AIX or Linux.

Set permissions for a monitoring server on a non-NIS Solaris system as follows:

1. Go to the /bin directory where file kdsvlunx is located (*install_dir*/*arch*/ms/bin, where *arch* is the operating system on which the monitoring agent was installed).
2. Move the file to the root user ID if you have the root password; otherwise obtain the password from an administrator:
   ```
   su root
   chown root kdsvlunx
   chmod u+s kdsvlunx
   ```
3. Return to your regular ID after you have moved the user ID to root.

Increasing virtual memory on AIX for large environments

The Tivoli Enterprise Portal Server process (KfwServices) is linked with the default memory model, which allows for data and stack of only 256 MB. In large environments, this can cause the portal server to crash at startup. In smaller environments, this may not cause a problem at startup, but may become one at some later point as more virtual storage is required. The problem can be predicted if the **topas** command shows KfwServices with a PgSp value of 180-250 MB. If a smaller environment is close to that value, it may occur when large queries are being handled. If the **topas** output shows that KfwServices is nearing these values, the memory model should be changed in smaller environments as well as large ones.

Note: This section applies only to a 32-bit portal server running on AIX. When installed for the first time on an AIX system with a 64-bit kernel, a 64-bit portal server is installed, and this procedure is not needed. When upgrading from previous versions, the 32-bit portal server will be rebuilt, so you must re-enable the large-memory model for the KfwServices process.

Changing the memory model requires that the KfwServices load header be modified. Also, changes must be made to your DB2 for Linux, UNIX, and Windows configuration. Complete the following steps to make these changes. These steps use the default directory. Use the directory locations appropriate to your system.

Important:: You must run the **ldedit** command below each time new Tivoli Enterprise Portal Server maintenance has been applied.

To make the required memory model and DB2 for Linux, UNIX, and Windows configuration changes:

1. Stop the Tivoli Enterprise Portal Server:
   ```
   cd /opt/IBM/ITM/bin ./itmcmd agent stop cq
   ```
2. Issue the following commands:
   ```
   cd /opt/IBM/ITM/aix533/cq/bin
   cp KfwServices KfwServices.orig
    /usr/ccs/bin/ldedit -bmaxdata:0x80000000 KfwServices
   ```
3. To verify that the maxdata value has been set issue the following command:
   ```
   dump -ov KfwServices
   ```

   This command displays the maxdata value in KfwServices. Maxdata should show as:
   ```
   maxSTACK maxDATA SNbss magic modtype 0x00000000 0x80000000 0x0003 0x010b 1L
   ```
4. Issue the following command:
   ```
   cd /opt/IBM/ITM/config
   ```
5. Using your preferred editor add the following line to file cq.ini:
   ```
   EXTSHM=ON
   ```
6. Using the DB2 for Linux, UNIX, and Windows installation user ID (the default value is db2inst1), make the DB2 for Linux, UNIX, and Windows configuration changes as follows:
   a. Stop the DB2 for Linux, UNIX, and Windows server if not already stopped:
      ```
      cd /db2inst1/sqllib/adm db2stop
      ```
   b. Issue the following configuration changes:
      ```
      export EXTSHM=ON
      db2set DB2ENVLIST=EXTSHM
      db2set -all
      ```
   c. Using your preferred editor add the following lines to /db2inst1/sqllib/db2profile:
      ```
      EXTSHM=ON
      export EXTSHM
      ```
   d. Restart DB2 for Linux, UNIX, and Windows:
      ```
      cd /db2inst1/sqllib/adm db2start
      ```
7. Restart the portal server:
   ```
   cd /opt/IBM/ITM/bin ./itmcmd agent start cq
   ```

For more information about using the large and very large address-space models to accommodate programs requiring data areas that are larger than those provided by the default address-space model, see http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.genprogc/doc/genprogc/lrg_prg_support.htm. For more information about using the EXTSHM environment variable to increate the number of share memory segments to which a single process can be attached, see http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic.

Linux requirements for the localhost host name

KDH servicepoint manipulation and the kdh (http/https) port-sharing code requires the availability of localhost: the **localhost** host name must exist and be resolvable to an address on the local system. Typically, **localhost** resolves to the loopback device with IP address `127.0.0.1`. If **localhost** does not exist or cannot be resolved to an IP address, http-based servers (the SOAP server, the IBM Tivoli Monitoring Service Console, the Tivoli Enterprise Portal Server) will fail to initialize.

When **localhost** resolves to an address other than `127.0.0.1`, a network interface must be locally available and configured to IP with that **localhost** network address.

Setting ulimit values for the Warehouse Proxy Agent

There is a limit on the number of open file descriptors that can be opened by the Warehouse Proxy process when it is running on UNIX and Linux systems. When a Tivoli Monitoring operating system monitoring agent connects to the Warehouse Proxy Agent, it uses a set of file descriptors for communication. When the number of operating system agent connections exceeds the file descriptor limit, the agent process consumes high amounts of CPU as it is unable to send data.

To correct this high CPU situation, the **ulimit** value must be set to a value higher than the maximum number of file descriptors that could be opened to the Warehouse Proxy Agent. The value that should be used is based upon the following conditions:

> The minimum number of file descriptors used by the Warehouse Proxy Agent process is `X + Y + 10`, where:
>
> **X** Is the number of agents warehousing to the Warehouse Proxy Agent
>
> **Y** Is the number of database connections between Warehouse Proxy Agent and the database server. (The default number of database connections is 10; you can change this number.)
>
> **10** Is the number of log and configuration files used by the Warehouse Proxy Agent.

The value used for the file descriptor **ulimit** should be high enough so that the limit is not met. A simpler formula is `X + 1000`, where X is the number of agents warehousing.

After you determine the value for the file descriptor **ulimit**, modify the **ulimit** as appropriate for the operating system. See the system documentation for the command (usually **ulimit**) and procedures to make this change permanently across system restarts, or contact your UNIX or Linux System Administrator.

Chapter 14. Configuring IBM Tivoli Monitoring components for IPv6 communication

This chapter provides instructions for configuring various Tivoli Monitoring components for IP version 6 (IPv6) communication. To enable IPv6 communication between a pair of endpoints, follow the instructions for configuring each endpoint. For example, if you want to enable IPv6 communication between a hub monitoring server on a Linux computer, and a remote monitoring server on Windows, follow the instructions for configuring the hub on the Linux computer and then the remote server on the Windows computer. For ease of reference, instructions are provided for each platform separately.

Note: In order to use IPv6 with the IBM Tivoli Monitoring components, you must ensure your operating system is set up and configured for IPv6.

For details on supported combinations of IPv6 with IPv4 across the various IBM Tivoli Monitoring components, see “Choose between IPv6 and IPv4” on page 127.

By default, Tivoli Monitoring uses the well known ports for communication between components. If the default ports that are used by Tivoli Monitoring components in the environment are changed, the change must be made consistently to the KDC_FAMILIES/KDE_TRANSPORT variables for all components, specifying the new port parameter for various protocols. For instance, if the port 11111 is to be used instead of the default 1918, and 7663 instead of the default 3660, the KDC_FAMILIES variable is as follows:

```
KDC_FAMILIES=ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ....
```

If IPv4 is also used, then the specification for the IPv4 protocol is also included. The syntax is platform-dependent.

Configuring AIX, Solaris, HP-UX, and Linux monitoring components

The instructions in this section apply to configuring monitoring components on the AIX, Solaris, HP-UX, and Linux platforms. $CANDLEHOME refers to the directory in which the Tivoli monitoring product is installed. In the following example, if a numeric IPv6 address is provided instead of a host name, the address must be enclosed in parentheses:

```
export CT_CMSLIST='ip6.pipe:(2002:930:9B04:305:9:48:133:98)'
```

Configuring the hub monitoring server

Hub and remote monitoring servers must be first configured using the platform-specific installation and configuration tools, which includes selecting a protocol and providing information about the standby hub.

If IPv6 is the preferred protocol for the hub, then the glb_site.txt file must be edited, providing the desired IPv6 family as the protocol. For example:

```
ip6.pipe:proton
```

The glb_site file is found in the tables directory for the hub monitoring server, for example, $CANDLEHOME/tables/$TEMSNAME.

On all platforms, if a port other than the default is used, then the KDC_FAMILIES variable must be added or updated in the configuration file for the hub in the $CANDLEHOME/config directory:

```
Export KDC_FAMILIES='ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ....'
```

On AIX only, the IPv6 protocol must be explicitly provided in the KDC_FAMILIES variable, before any IPv4 protocols. This variable is found in the configuration file for the hub in the $CANDLEHOME/config directory. For instance:

```
export KDC_FAMILIES='ip6.pipe use:y ip6.spipe use:y
ip6.tcp use:y ip6.ssl use:y ip.pipe use:y ip.spipe use:y
http use:y https use:y'
```

or simply:

```
export KDC_FAMILIES='+ipv6 +ipv4'
```

However, if a port other than the default is being used, the port must be explicitly declared, as described at the beginning of this paragraph.

On Solaris only, the hub can be configured either for IPv4 or IPv6 but not both. For IPv4, no action is required. For IPv6, the following must be added to the configuration file for the hub in the $CANDLEHOME/config directory:

```
export KDEB_INTERFACELIST='-'
```

On HP-UX on HP 9000 only, the hub can be configured either for IPv4 or IPv6 but not both. For IPv4, no action is required. For IPv6, the following must be added to the configuration file for the hub in the $CANDLEHOME/config directory:

```
export KDEB_INTERFACELIST='-'
```

If the hub has been configured to use a standby, then the MHM:MIRROR_SITE variable must be edited if the preferred protocol for communication with the standby hub is IPv6. For example:

```
export MHM:MIRROR_SITE='ip6.pipe:atlas'
```

This variable is found in the KBBENV file for the monitoring server in the tables directory, for example, $CANDLEHOME/tables/$TEMSNAME.

If the server has been configured to provide access to other SOAP server hubs, then the kshxhubs.xml file must be updated to enable IPv6 communication with those hubs. This file can be found in the HTML directory for the monitoring server under the tables directory, for example, $CANDLEHOME/tables/$TEMSNAME/HTML.

Example:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="hubdef.xsl"?>
<ENTERPRISE>
<HUB>
<CMS_Name>
ip6.pipe:saturn[1918]
</CMS_Name>
<Service_Name>newhub</Service_Name>
<Alias>newhub</Alias>
</HUB>
</ENTERPRISE>
```

If the server has been configured to use LDAP for user authentication, the server uses available protocols to connect to the LDAP server. The LDAP server host can be designated by name or IP address. For example:

```
KGL_LDAP_HOST_NAME='saturn'
KGL_LDAP_HOST_NAME='2002:930:9b04:305:9:48:157:7'
```

Configuring the remote monitoring server

Hub and remote monitoring servers must be first configured using the platform-specific installation and configuration tools, which includes selecting a protocol and providing information about the standby hub.

The remote monitoring server uses the entries in the glb_site.txt file to look up the hub monitoring server. If there is a standby hub, then there are at least two entries in this file. This file must be edited if the protocol for communicating with one or both the hubs must be set to IPv6. For example:

```
ip6.pipe:proton
ip.pipe:amoeba
```

On all platforms, if a port other than the default is used, then the KDC_FAMILIES variable must be added or updated in the configuration file for the hub the $CANDLEHOME/config directory:

```
export KDC_FAMILIES='ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ....'
```

On AIX only, the IPv6 protocol must be explicitly provided in the KDC_FAMILIES variable, before any IPv4 protocols. For instance:

```
export KDC_FAMILIES='ip6.pipe use:y ip6.spipe use:y
ip6.tcp use:y ip6.ssl use:y ip.pipe use:y ip.spipe use:y
http use:y https use:y'
```

or simply:

```
export KDC_FAMILIES='+ipv6 +ipv4'
```

However, if a port other than the default is being used the port must be explicitly declared, as previously described.

On Solaris only, the monitoring server can be configured either for IPv4 or IPv6 but not for both. For IPv4, no action is required. For IPv6, the following must be added to the configuration file for the monitoring server in the configuration directory:

```
export KDEB_INTERFACELIST='-'
```

On HP-UX on HP 9000 only, the monitoring server can be configured either for IPv4 or IPv6 but not for both. For IPv4, no action is required. For IPv6, the following must be added to the configuration file for the monitoring server in the configuration directory:

```
export KDEB_INTERFACELIST='-'
```

In this case, you must set the protocol to ip6.pipe in the glb_site file for this remote monitoring server.

Configuring the monitoring agents

Agents use the CT_CMSLIST variable to locate the monitoring server they are required to connect to. This variable may contain one or more entries, separated by a semi-colon. For example:

```
CT_CMSLIST='ip6.pipe:proton;ip6.pipe:atlas'
```

If IPv6 is the preferred protocol for the agent to communicate with the designated monitoring server, then this variable must be updated. Before doing this, ensure that the monitoring server is configured to communicate using IPv6.

The procedure includes the following steps:

1. Open the xx.ini file in the $CANDLEHOME/config directory for editing, where xx is the product code for the agent, such as lz.
2. Add the following line as the last line in the file (note that the line begins with a dot followed by a space):
   ```
   . $CANDLEHOME$/config/xx.ipv6config
   ```
3. Repeat Steps 1 and 2 for the xx.config file in the same directory.
4. Create the xx.ipv6config file in the $CANDLEHOME/config directory with the CT_CMSLIST specification, as in the following example:
   ```
   export CT_CMSLIST='ip6.pipe:proton;ip6.pipe:atlas'
   ```

On all platforms, if a port other than the default is used, then the KDC_FAMILIES variable must be added in the xx.ipv6config file:

```
export KDC_FAMILIES='ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ip.pipe port:11111 ....'
```

On AIX only, the IPv6 protocol must be explicitly provided in the KDC_FAMILIES variable, before any IPv4 protocols. For instance:

```
export KDC_FAMILIES='ip6.pipe use:y ip6.spipe use:y
ip6.tcp use:y ip6.ssl use:y ip.pipe use:y ip.spipe use:y
http use:y https use:y'
```

or simply:

```
export KDC_FAMILIES='+ipv6 +ipv4'
```

This definition must be provided in the xx.ipv6config file. However, if a port other than the default is being used, the port must be explicitly declared, as previously described.

On AIX, if the agent being configured is based on Tivoli Monitoring V6.1 but installed in a Tivoli Monitoring V6.2 environment, an additional step is required. For existing Tivoli Monitoring V6.1 agents, the preferred approach is to upgrade these agents to the Tivoli Monitoring V6.2 level. The upgrade can be done as follows:

- If there is a Tivoli Monitoring V6.1 Linux, HP-UX, Solaris, or AIX agent in the same environment, this agent can be upgraded to Tivoli Monitoring V6.2.
- If there is no Linux, HP-UX, Solaris, or AIX agent in the same environment, a Tivoli Monitoring V6.2 agent can be installed in this environment.

If the OS agent cannot be installed or upgraded in the agent environment for some reason, you should contact IBM support for assistance.

Check the LIBPATH variable in the xx.config file for the agent. If it refers to tmaitm6/aix513/lib, add the following line in the xx.ipv6config file:

```
export LIBPATH=$CANDLEHOME/tmaitm6/aix523/lib:$LIBPATH
```

If it refers to tmaitm6/aix516/lib, add the following line in the xx.ipv6config file:

```
export LIBPATH=$CANDLEHOME/tmaitm6/aix526/lib:$LIBPATH
```

On Linux, if the agent being configured is based on Tivoli Monitoring V6.1 but installed in a Tivoli Monitoring V6.2 environment, an additional step is required as described in the previous section for AIX.

Check the LD_LIBRARY_PATH variable in the xx.config file for the agent. If it refers to tmaitm6/ls3246/lib, add the following line in the xx.ipv6config file:

```
export LD_LIBRARY_PATH=$CANDLEHOME/tmaitm6/ls3266/lib
```

If it refers to tmaitm6/ls3243/lib, add the following line in the xx.ipv6config file:

```
export LD_LIBRARY_PATH=$CANDLEHOME/tmaitm6/ls3263/lib
```

If it refers to tmaitm6/li6243/lib, add the following line in the xx.ipv6config file:

```
export LD_LIBRARY_PATH=$CANDLEHOME/tmaitm6/li6263/lib
```

On Solaris only, the agent can be configured either for IPv4 or IPv6, but not for both. For IPv4, no action is required. For IPv6, the following line must be added:

```
export KDEB_INTERFACELIST='-'
```

This definition must also be provided in the xx.ipv6config file.

On HP-UX on HP 9000 (PA-RISC) only, the agent can be configured either for IPv4 or IPv6, but not for both. For IPv4, no action is required. For IPv6, the following line must be added:

```
export KDEB_INTERFACELIST='-'
```

This definition must also be provided in the xx.ipv6config file.

Note:

> Before starting an agent, be sure that the IPv6 address correctly resolves to the host name because numeric IPv6 addresses are not permitted in managed system names. This rule applies to all distributed agents and to other application agents where the agents provide the managed system name, and the name includes the host name.
>
> The resolved name must be a short name, and not a long fully qualified name, which can cause the managed system name to be truncated, resulting in unpredictable behavior. This rule is not an IPv6-specific constraint.

Warehouse Proxy Agent

The Warehouse proxy agent is configured as described in “Configuring the monitoring agents” on page 389. In addition, the glb_khd.txt file must be updated to include the IPv6 protocol. The glb_khd.txt file can be found in the hd/bin directory under the platform-specific directory in the installation home directory on UNIX, for example: $CANDLEHOME/li6263/hd/bin on Linux.

Tivoli Enterprise Portal Server

The instructions for “Configuring the monitoring agents” on page 389 also apply to the Tivoli Enterprise Portal Server (or portal server). The product code for the portal server is cq.

Adding Application Support to a monitoring server

If the monitoring server to which application support is to be added is on an IPv6-only system, or configured to use IPv6 only, then the Manage Tivoli Enterprise Management Services console cannot be used to add the application support, and a command-line interface must be used. Use the following procedure to add or remove application support to a management server:

1. Locate the executable kdstsns in the installed environment. It is usually found under $CANDLEHOME/$BINARCH/ui/bin or $CANDLEHOME/$BINARCH/ms/bin, where BINARCH is the platform architecture designated by IBM Tivoli Monitoring, for example, li6263 on 32 bit Linux, aix523 on 32 bit AIX, hp116 on 64 bit HP-UX, and so on.
2. Locate the SQLLIB directory where the SQL input files for application support reside. This directory is usually under $CANDLEHOME/tables/cicatrsq.
3. Change to the bin directory under $CANDLHOME directory, for example:

   ```
   cd /opt/IBM/ITM/bin
   ```

   Create a file called glb_ipv6tems.txt, and add a line to it to denote the host on which the monitoring server to be seeded resides, for example:

   ```
   ip6.pipe:atz1004.tivlab.austin.ibm.com
   ```
4. Create a shell command file called seedipv6tems.sh with the following lines (using as an example an installation on 32 bit Linux under /opt/IBM/ITM):

   ```
   #!/bin/ksh
   export CANDLEHOME=/opt/IBM/ITM
   export BINARCH=li6263
   export SQLLIB=$CANDLEHOME/tables/cicatrsq/SQLLIB
   export KDC_GLBSITES=glb_ipv6tems.txt
   $CANDLEHOME/$BINARCH/ui/bin/kdstsns $1 $2
   ```

   Save the file, and make it executable: `chmod +x seedtems.sh`

Note: You can run the following command to find out the **BINARCH** parameter name:

```
$CANDLEHOME/bin/cinfo -s ms | grep BINARCH
```

and look for a line that starts with **ms**, then the **TEMS** name, and then the **BINARCH** parameter name. For example:

```
$CANDLEHOME/bin/cinfo -s ms | grep BINARCH
ms     <TEMS-name>  BINARCH = lx8266
```

5. Use the following command from the command-line to add support for an application:

   ```
   seedipv6tems.sh <app-filename> <TEMS-name>
   ```

   where:

   app-filename
   is the name of the sql file to be added for the desired application.

   TEMS-name
   is the name of the monitoring server to which support is being added. For example:

   ```
   seedipv6tems kt4.sql TEMSFRT:CMS
   ```

   If multiple applications are to be added, then another script file can be created to run all the commands at once. For example, the file seedipv6all.sh.bat with the following can be used:

   ```
   seedipv6tems.sh kt4.sql TEMSFRT:CMS
   seedipv6tems.sh knt.sql TEMSFRT:CMS
   ```

6. Use the same procedure to remove application support by using the kxx_del.sql file for application kxx.

Command-line interface (CLI)

The CLI tacmd utility is used without change in all scenarios. On AIX only, the following line must be added to the tacmd script file in the $CANDLEHOME/bin directory:

```
export KDC_FAMILIES='+ip6 +ip4'
```

Configuring Windows monitoring components

The instructions in this section apply to configuring monitoring components on a Windows platform.

Configuring the hub monitoring server

Hub or remote servers must be first configured using the platform-specific installation and configuration tools, which includes selecting the protocol and providing information about the standby hub.

If IPv6 is the preferred protocol for the hub, then the glb_site.txt file must be edited, providing the desired IPv6 family as the protocol. For example:

```
ip6.pipe:proton
```

The glb_site.txt file is found in the cms directory under the installation home directory, which is typically \IBM\ITM.

If the hub has been configured to use a standby server, then the MHM:MIRROR_SITE variable must be edited if the preferred protocol for communication with the standby hub is IPv6. For example:

```
MHM:MIRROR_SITE=ip6.pipe:atlas
```

To set this variable, access the Manage Tivoli Services console, right-click the management server, select **Advanced** → **Edit Variables**, click **Add**, and then add the MHM:MIRROR_SITE variable with the desired value.

If a port other than the default is used, then the KDC_FAMILIES variable must be updated. To set this variable, access the Manage Tivoli Services console, right-click the management server, select **Advanced** → **Edit Variables**, click **Add**, and then add the KDC_FAMILIES variable with the desired value.

```
KDC_FAMILIES=ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ....
```

If the server has been configured to provide access to other SOAP server hubs, then the kshxhubs.xml file must be updated to enable IPv6 communication with those hubs. This file can be found in the cms\HTML directory under the installation directory, which is typically \IBM\ITM. For example:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="hubdef.xsl"?>
<ENTERPRISE>
<HUB>
<CMS_Name>
ip6.pipe:saturn[1918]
</CMS_Name>
<Service_Name>newhub</Service_Name>
<Alias>newhub</Alias>
</HUB>
</ENTERPRISE>
```

Configuring the remote monitoring server

The remote monitoring server uses the entries in the glb_site.txt file to look up the hub. If there is a standby hub, then there are at least two entries in this file. This file must be edited if the protocol for communicating with one or both the hubs must be set to IPv6. This file is found in the cms directory under the installation home directory, which is typically \IBM\ITM. For example:

```
ip6.pipe:proton
ip.pipe:amoeba
```

If a port other than the default is used, then the KDC_FAMILIES variable must be updated. To set this variable, access the Manage Tivoli Services console, right-click the management server, select **Advanced** → **Edit Variables**, click **Add**, and then add the KDC_FAMILIES variable with the desired value:

```
KDC_FAMILIES=ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ....
```

Configuring monitoring agents

Agents use the CT_CMSLIST variable to locate the monitoring server they are required to connect to. This variable may contain one or more entries, separated by a semi-colon. For example:

```
CT_CMSLIST='ip6.pipe:proton;ip6.pipe:atlas'
```

If IPv6 is the preferred protocol for the agent to communicate with the designated monitoring server, then this variable must be updated. Before doing this, ensure that the monitoring server is configured to communicate using IPv6.

To set this variable, access the Manage Tivoli Services console, right-click the agent, select **Advanced** → **Edit Variables**, click **Add**, and then add the CT_CMSLIST variable with the desired value.

If a port other than the default is used, then the KDC_FAMILIES variable must be updated. To set this variable, access the Manage Tivoli Services console, right-click the agent, select **Advanced** → **Edit Variables**, click **Add**, and then add the KDC_FAMILIES variable with the desired value:

```
KDC_FAMILIES=ip6.pipe port:11111 ip6 port:11111 ip6.spipe port:7663 ....
```

Note:

> Before starting an agent, you must ensure that the IPv6 address resolves to the correct host name because numeric IPv6 addresses are not permitted in managed system names. This rule applies to all distributed OS agents and to other application agents where the agents provide the managed system name, and the name includes the host name.

The resolved name must be a short name, and not a long fully qualified name, which can cause the managed system name to truncate, resulting in unpredictable behavior. This rule is not an IPv6-specific constraint.

Warehouse Proxy Agent

The Warehouse proxy agent is configured as described in “Configuring monitoring agents” on page 393. In addition, the file glb_khd.txt must be updated to include the IPv6 protocol. This file can be found in the TMAITM6 directory under the home directory on Windows (typically \IBM\ITM).

Tivoli Enterprise Portal Server

Configuration instructions for agents also apply to the portal server. The product code for the portal server is cq.

Adding Application Support to a monitoring server

If the monitoring server to which application support is to be added is on an IPv6-only system, or configured to use IPv6 only, then the Manage Tivoli Enterprise Management Services console cannot be used to add the application support, and a command-line interface must be used. Use the following procedure to add or remove application support to a management server:

1. Change to the CNPS directory in the installation home, for example:
   ```
   cd \IBM\ITM\cnps
   ```
2. Create a file called glb_ipv6tems.txt, and add a line to it to denote the host on which the monitoring server to be seeded resides, for example:
   ```
   ip6.pipe:atz1004.tivlab.austin.ibm.com
   ```
   Save the file.
3. Create another file called seedipv6tems.bat, with the following lines:
   ```
   set KDC_GLBSITES=glb_ipv6tems.txt
   kdstsns %1 %2
   ```
   If a port other than the default is being used for communication between components, also include the following as the first or the second line, replacing 11111 with the actual port number used in your environment:
   ```
   KDC_FAMILIES=ip6.pipe port:11111 ip6 port:11111
   ```
   Save the file.
4. Use the following command from the command-line to add support for an application:
   ```
   seedipv6tems.sh <app-filename> <TEMS-name>
   ```
   where:

   app-filename
   is the name of the sql file to be added for the desired application.

   TEMS-name
   is the name of the monitoring server to which support is being added. For example:
   ```
   seedipv6tems kt4.sql TEMSFRT:CMS
   ```
   If multiple applications are to be added, then another script file can be created to run all the commands at once. For example, the file seedipv6all.sh.bat with the following can be used:
   ```
   seedipv6tems.sh kt4.sql TEMSFRT:CMS
   seedipv6tems.sh knt.sql TEMSFRT:CMS
   ```
5. Use the same procedure to remove application support by using the kxx_del.sql file for application kxx.

Sending files to z/OS systems using FTP

Catalog and attribute files cannot be sent to z/OS systems over IPv6 using the file transfer capability provided by the Manage Tivoli Enterprise Services console provided on the Windows platform. These files should be transferred using other FTP utilities, such as the one available from the Windows command console.

Tivoli Enterprise Portal browser client

There are no configuration requirements for the browser client. The Internet Explorer browser on Windows 2003 and Windows XP does not allow the use of numeric IPv6 addresses in URLs, so symbolic names must be used to See IPv6 hosts.

Tivoli Enterprise Portal desktop client

Numeric IPv6 can be used to identify the portal server in the desktop client configuration. If a numeric IPv6 address is used, it must be enclosed in square brackets. For example:

```
[2002:930:9b04:305:9:48:133:98]
```

Command-line interface

The command-line utility tacmd can be used without any changes.

Configuring z/OS monitoring components

On the z/OS platform, the ICAT utility can be used for configuring components for using IPv6, as ICAT configuration panels offer IPv6 as one of the protocols. Manual editing of configuration files is required only if numeric IPv6 addresses must be provided for hosts, or if the IPv4 and IPv6 addresses are bound to two different host names.

The various configuration variables and members that must be updated to provide numeric IPv6 addresses are described in the following sections.

The KDEB_INTERFACELIST_IPv6 parameter in KppENV members in the RKANPARU data set provides a list of interfaces to be used for IPv6 communication. Numeric IPv6 addresses must be enclosed in parentheses. Multiple addresses are separated by spaces. For example:

```
KDEB_INTERFACELIST_IPv6=\

(2002:930:9b04:305:9:48:133:98)
(2002:930:9b04:305:9:48:133:100)
```

The CT_CMSLIST parameter in the KppENV members in the RKANPARU data set provides a list of monitoring servers agents can connect to, in the order provided. For example:

```
CT_CMSLIST=\
IP.PIPE:SP13;\
IP6.PIPE:( 2002:930:9b04:305:9:48:133:98)
```

The KppSSITE member in the RKANPARU data set provides a list of hosts. Numeric IPv6 addresses designating hosts, if provided, must be enclosed in parentheses. For example:

```
IP.PIPE:SP13
IP6.PIPE:( 2002:930:9b04:305:9:48:133:98)
```

The KSHXHUBS member in the RKANPARU data set is an XML document that specifies additional SOAP servers that can be accessed through this hub. If numeric IPv6 addresses are provided, they must be enclosed in parentheses. For example:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="hubdef.xsl"?>
<ENTERPRISE>
<HUB>
```

```
<CMS_Name>
IP6.PIPE:(2002:930:9b04:305:9:48:133:98)[1918]
</CMS_Name>
<Service_Name>newhub</Service_Name>
<Alias>newhub</Alias>
</HUB>
</ENTERPRISE>
```

Firewall considerations

For information on configuring firewall gateways and operation of other Tivoli Monitoring components in firewall environments, see “Determine if you require a firewall gateway” on page 35.

Chapter 15. Additional Tivoli Enterprise Portal configurations

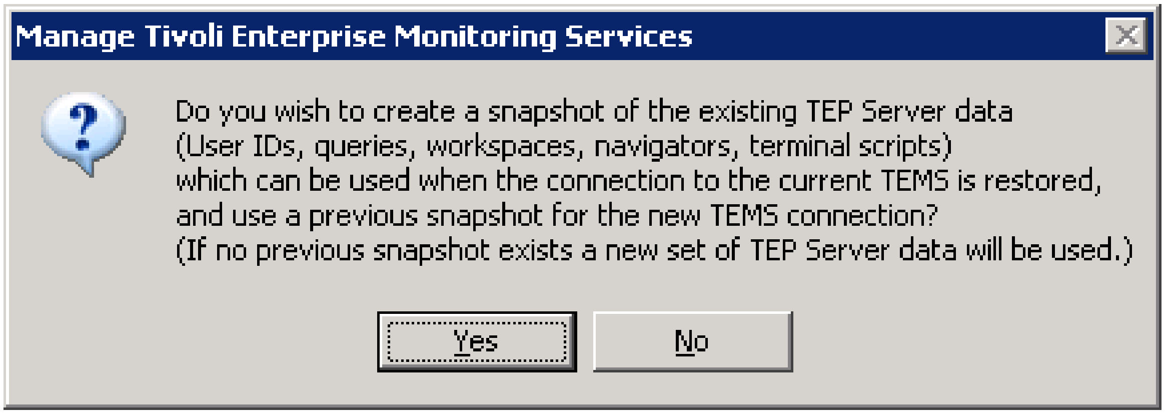
This chapter discusses how to perform the following advanced configuration of Tivoli Enterprise Portal components:

- "Connecting the Tivoli Enterprise Portal Server on Windows to a different monitoring server"
- "Using SSL between the portal server and the client" on page 398
- "Configuring an external Web server to work with Tivoli Enterprise Portal" on page 400
- "Configuring a portal client connection to an external Web server" on page 402
- "Configuring historical data collection for the Performance Analyzer warehouse agent" on page 377
- "Reverting from the IBM HTTP Server to the internal web server" on page 404
- "Configuring HTTP communication between the portal client and server" on page 405
- "Firewall network address translation (NAT) or multiple network interface cards" on page 408
- "Firewall scenarios for Tivoli Enterprise Portal" on page 409

It also includes illustrations of firewall scenarios that can help in defining the Tivoli Enterprise Portal Server interface.

Connecting the Tivoli Enterprise Portal Server on Windows to a different monitoring server

When reconfiguring the portal server on Windows for a different Tivoli Enterprise Monitoring Server, a pop-up window is displayed asking if a snapshot of the current Tivoli Enterprise Portal Server data should first be taken (see Figure 102).



Manage Tivoli Enterprise Monitoring Services

Do you wish to create a snapshot of the existing TEP Server data
(User IDs, queries, workspaces, navigators, terminal scripts)
which can be used when the connection to the current TEMS is restored,
and use a previous snapshot for the new TEMS connection?
(If no previous snapshot exists a new set of TEP Server data will be used.)

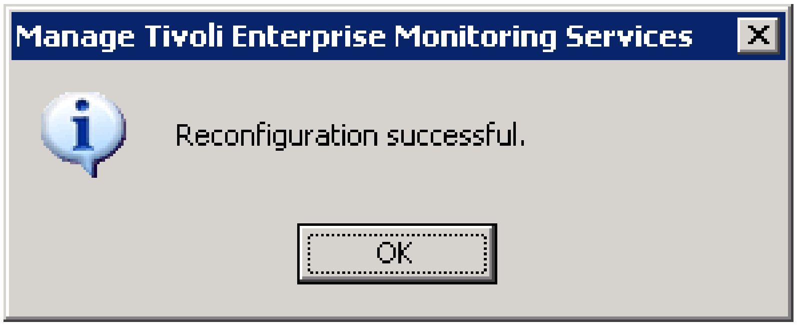
Figure 102. Tivoli Enterprise Portal Server snapshot request screen

The default response is **Yes**.

Note: When reconfiguring the portal server to communicate with a Hot Standby monitoring server, the correct response is **No**, as the same portal server data should be used for both the primary monitoring server and the secondary monitoring server. See the *IBM Tivoli Monitoring: High-Availability Guide for Distributed Systems*.

When **Yes** is selected, a snapshot of the portal server data is taken via the **migrate-export** process. The data is saved in a file named `saveexport.sql` and is placed in `%CANDLE_HOME%\CNPS\CMS\`*`hostname`*`:`*`port`*, where *hostname*:*port* is the current monitoring server's hostname and connection port number.

When this process completes, the verification screen shown in Figure 103 is displayed.


Manage Tivoli Enterprise Monitoring Services

Reconfiguration successful.

Figure 103. Tivoli Enterprise Portal Server snapshot verification screen

If no existing snapshot can be found for the monitoring server that is being switched to, a new set of portal server data is created, which means all existing customizations will not be included. To restore these for use with the new monitoring server (if needed), invoke the **migrate-import** process using as input the `saveexport.sql` file created during the previous snapshot request.

When reconfiguring the Tivoli Enterprise Portal Server to switch back to the previous Tivoli Enterprise Monitoring Server, answering **Yes** causes the previous snapshot to be automatically loaded, thus restoring your prior customizations.

Using SSL between the portal server and the client

You can choose to encrypt all communication between the portal server and portal client. IBM Tivoli Monitoring uses two protocols to provide this level of security between portal server and client server:

- Secure Hypertext Transport Protocol (HTTPS) to retrieve files and Interoperable Object Reference (IOR). The integrated browser in the client provides HTTPS support on the client side; for the server, consider using a third party web server that supports HTTPS, such as the IBM HTTP Server. See "Configuring an external Web server to work with Tivoli Enterprise Portal" on page 400 for more information.
- Internet Inter-ORB Protocol (IIOP) to secure the communications between the portal server and client. This uses the Secure Sockets Layer (SSL) API provided by VisiBroker. This secure communication uses public key cryptography.

When you install IBM Tivoli Monitoring, the Global Security Toolkit (GSKit) and iKeyman utilities are installed by default on all components. These utilities are used to create and manage the encryption of data between components through the use of digital certificates.

Digital certificates are the vehicle that SSL uses for public-key cryptography. Public-key cryptography uses two different cryptographic keys: a private key and a public key. Public-key cryptography is also known as *asymmetric cryptography*, because you can encrypt information with one key and decrypt it with the complement key from a given public/private key pair.

Public/private key pairs are simply long strings of data that act as keys to a user's encryption scheme. The user keeps the private key in a secure place (for example, encrypted on a computer's hard drive) and provides the public key to anyone with whom the user wants to communicate. The private key is used to digitally sign all secure communications sent from the user; the public key is used by the recipient to verify the sender's signature.

Public/private key pairs are validated by a trusted third party, called a Certificate Authority (CA). An example of a CA is Verisign. If you are setting up your own key pairs, you submit them to the CA for validation.

If you intend to use SSL for communication between the Tivoli Enterprise Portal Server and its clients, use the GSKit provided with IBM Tivoli Monitoring to manage certificates and keys. See the *IBM Tivoli Monitoring: Administrator's Guide* for instructions for setting up this encryption.

For additional information about using public/private key pairs, see the iKeyman documentation available at http://www-128.ibm.com/developerworks/java/jdk/security/50/.

Enabling and disabling SSL for the Tivoli Enterprise Portal Server

IBM Tivoli Monitoring is shipped with SSL disabled as the default.

If you want to use Secure Sockets Layer communication between the portal server and the portal client, use the following steps to enable it:

Note: This procedure disables the primary interface. For instructions on disabling additional interfaces, see Chapter 15, "Additional Tivoli Enterprise Portal configurations," on page 397.

On a Windows system:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server**.
2. Click **Advanced** → **Configure TEPS Interfaces**.
3. Highlight **cnps** and click **Edit** in the TEPS Interface Definitions window.
4. Select **Enable SSL for TEP Clients**.
5. Click **OK** to save your changes and close the window.
6. Recycle the service by stopping and starting it.

On a Linux system:

1. Change to the *install_dir*/bin directory
2. Run the following command:
   ```
   ./itmcmd manage
   ```
3. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server**.
4. Click **Configure**
5. In the first tab, select **Enable SSL for TEP Clients** to enable SSL in the Tivoli Enterprise Portal Server window.
6. Click **OK** to save your changes and close the window.
7. Recycle the service by stopping and starting it.

Disabling SSL

If you do not want to use Secure Sockets Layer communication between IBM Tivoli Monitoring components and the Tivoli Enterprise Portal Server, use the following steps to disable it:

Note: Each interface independently enables or disables SSL. If you are using multiple interfaces, you must disable all of them.

1. In Manage Tivoli Enterprise Monitoring Services, right-click **Tivoli Enterprise Portal Server**.
2. Click **Advanced** → **Edit ENV file**.
3. Find the following line:
   ```
   kfw_interface_cnps_ssl=Y
   ```
4. Change the `Y` to `N`.
5. Save the file and exit.
6. Click **Yes** when you are asked if you want to recycle the service.

Configuring an external Web server to work with Tivoli Enterprise Portal

If you want to use an external Web server to view the Tivoli Enterprise Portal, you need to configure that Web server. The following sections provide configuration information for Microsoft Internet Information Server versions 5.0 and 6.0, IBM HTTP Server, and Apache HTTP Server.

- “Configuring Internet Information Server V5.0”
- “Configuring Internet Information Server V6.0”
- “Configuring IBM HTTP Server and Apache HTTP Server” on page 401

Configuring Internet Information Server V5.0

Use the following steps to configure Internet Information Server V5.0 to work as a Tivoli Enterprise Portal browser client.

1. Start the Internet Services Manager.
2. Right click the WWW service (the system host name).
3. Select **Master Properties** and click **Edit**.
4. Click the **Directory Security** tab.
5. Click **Edit**.
6. Ensure that **Anonymous access** and **Integrated Windows authentication** are selected and click **OK**.
7. Click the **Documents** tab.
8. Click **Add**.
9. Type `index.html` in the **Default Document Name** field.
10. Click **OK** and then click **OK** to close the Master Properties notebook.

Use the following steps to set up the Web site:

1. From the Internet Services Manager, right-click the WWW service.
2. Click **New → Web Site**.
3. Click **Next**.
4. Type a description for the Web site (for example, "ITM").

   Note: See the documentation for Internet Information Server before changing any of the default values in the next steps.
5. If you have multiple IP addresses, select the one that is appropriate for this site.
6. If you want to use a separate port of this site, type the port number in the **TCP Port** field.
7. Type the path to the Tivoli Enterprise Portal browser client. The default path is C:\IBM\ITM\CNB.
8. Ensure that **Read** and **Run Scripts** are selected. Do *not* select **Execute**.
9. Click **Next** and then click **Finish**.

If the site does not start automatically, right-click it and click **Start**.

Configuring Internet Information Server V6.0

Use the following steps to configure Internet Information Server V6.0 on Windows 2003 to work as a Tivoli Enterprise Portal browser client:

1. Start IIS Manager.
2. Right-click **Web Sites** and click **New → Web Site**.
3. Click **Next**.
4. Type the name of a Web site (for example "Tivoli") and click **Next**.

5. Type the IP address for the Tivoli Enterprise Portal Server computer (this should be the same computer where IIS 6.0 is running) and click **Next**.
6. Type the path to the IBM Tivoli Monitoring home directory that is the root of the Web Content subdirectories. The default path is C:\IBM\ITM\CNB. Click **Next**.
7. Select **Read**, **Run scripts**, and **Execute**. Click **Next**.
8. Click **Finish**.
9. Right-click the new Web site and click **Properties**.
10. Click the **Documents** tab.
11. In the **Add Content Page** field, type `index.html`. This is the main page for the Tivoli Enterprise Portal.
12. Click the **Move Up** button to move **index.html** to the top of the list.
13. Click the **HTTP Headers** tab.
14. Click **MIME Types**.
15. Click **New** next to **MIME Types**.
16. Type `*.asp` in the **Extension** field.
17. Type `application/x-asp` in the **MIME Type** field.
18. Click **OK**.
19. Repeat Steps 15 to 18 for each of the following:

Extension	MIME Type
.class	application/java-class
.ior	application/octet-stream
.jar	application/java-archive
.jks	application/octet-stream
.jnlp	application/x-java-jnlp-file
.js	application/x-javascript
.llser	application/octet-stream
.pl	application/x-perl
.ser	application/java-serialized-object
.txt	text/plain
.zip	application/zip

20. Click **OK**.
21. Click **Apply**.
22. Click **OK**.

Configuring IBM HTTP Server and Apache HTTP Server

Use the following steps to configure IBM HTTP Server or Apache HTTP Server to work on a computer with a Linux, UNIX, or Windows operating system:

1. Install the server with the default settings. See the product documentation (www-306.ibm.com/software/webservers/httpservers/library) for additional information.
2. Open the **httpd.conf** file in a text editor.

   On the IBM HTTP Server, this file is located in the `conf` directory of the server installation directory. For the Apache server, the file is typically located in the `/etc/httpd/conf/` directory, but may be found in some other location specific to the platform.
3. Find the line that begins with `DocumentRoot`.

- On Linux and UNIX computers, change the value between the double quotation marks ("") to *itm_installdir*/*arch*/cw, where *itm_installdir* is the directory where IBM Tivoli Monitoring is installed and *arch* is the operating system type (li6263 for SLES9 for Intel systems, li3263 SLES9 for zSeries systems, or aix533for AIX systems). For example:

  ```
  /opt/IBM/ITM/ls3263/cw
  ```

- On Windows computers, change the value between the double quotation marks (“) to *itm_installdir*/CNB where *itm_installdir* is the directory where IBM Tivoli Monitoring is installed. Use forward slashes for the path. For example:

  ```
  DocumentRoot "C:/IBM/ITM/CNB"
  ```

4. Find the line that begins with `<Directory docRoot>`. Change the path to the value used for `DocumentRoot`. (From our previous examples, this would be *itm_installdir*/ls3263/cw on Linux and UNIX and *itm_installdir*/CNB on Windows.)
5. Save and close the file.
6. Open the **mime.types** file in a text editor and make the following changes. For the IBM HTTP Server, this file is located in the `conf` directory of the server installation directory. For an Apache server, this file is typically located in `/etc/mime.types`, but the location may differ by platform. You may need to search for the file.
   a. If the following lines are not in the file, add them:

   ```
   application/java-archive jar
   image/icon ico
   ```

   b. If you will be using Java Web Start and the following lines are not in the file, add them:

   ```
   application/x-java-jnlp-file jnlp
   image/x-icon ico
   ```

   c. Modify the line that begins with `application/octet-stream` to include `ior ser` at the end. For example:

   ```
   application/octet-stream bin dms lha lzh exe class so dll cab ior ser
   ```

7. Save and close the file.
8. Stop the IBM HTTP Server or Apache HTTP Server services, then start it again to enable the configuration changes.

Configuring a portal client connection to an external Web server

Use the following sections to configure your portal client to work with an external Web server:

- “Browser client”
- “Desktop client”
- “Web Start client” on page 403

Browser client

During installation, the IBM Tivoli integral Web server is installed as a component of the Tivoli Enterprise Portal Server. You can also use an external Web server on your Tivoli Enterprise Portal Server computer, as shown in “Firewall scenarios for Tivoli Enterprise Portal” on page 409.

Currently, IBM supports an external Web server for browser client access only when the external server is installed on the same computer as the Tivoli Enterprise Portal Server.

Desktop client

Although the desktop client does not need a Web server to start Tivoli Enterprise Portal, it does use it for common files stored on the Tivoli Enterprise Portal Server, such as the graphic view icons and style sheets. If your Tivoli Enterprise Portal Server setup disables the integral Web server and uses only an external Web server, you need to specify the Interoperable Object Reference (IOR) for every desktop client.

Updating the IOR for Windows

Use the following steps to specify the IOR for the desktop client:

1. On the computer where Tivoli Enterprise Portal desktop client is installed, open Manage Tivoli Enterprise Monitoring Services
2. Right-click **Tivoli Enterprise Portal – Desktop** and click **Reconfigure**.
3. Double-click **cnp.http.url.DataBus** in the parameters list.

   The Edit Tivoli Enterprise Portal Parm window is displayed.
4. In the **Value** field, type the Web server address where the `cnps.ior` can be found.

   For example, if the Web server name is `xyz.myserver.com` and the document root for the Web server is `\ibm\itm\cnb`, the value is `http://xyz.myserver.com/cnps.ior`.
5. Select **In Use** and click **OK**.
6. Click **OK** to close the window.

Updating the IOR for Linux

Use the following steps to specify the IOR for the desktop client:

1. On the computer where the Tivoli Enterprise Portal desktop client is installed, go to the *install_dir*/`bin` directory and edit the **cnp.sh** shell script.

   If you are configuring an instance of the desktop client, the name of the file is **cnp_*instancename*.sh**.
2. Add the following parameter to the last line of the file specifying the Web server address where the cnps.ior can be found for the value. For example, if the Web server name is `xyz.myserver.com` and the document root for the Web server is `/candle/cnb`, the value is `http://xyz.myserver.com/cnps.ior`.

   ```
   -Dcnp.http.url.DataBus=http://xyz.myserver.com/cnps.ior
   ```

   Note: The last line of **cnp.sh** is very long and has various options on it, including several other -D options to define other properties. It is very important to add the option in the correct place.

If the last line of your **bin/cnp.sh** originally looked like this:

```
${JAVA_HOME}/bin/java -showversion -noverify -classpath ${CLASSPATH}
-Dkjr.trace.mode=LOCAL -Dkjr.trace.file=/opt/IBM/ITM/logs/kcjras1.log
-Dkjr.trace.params=ERROR -DORBtcpNoDelay=true -Dcnp.http.url.host=
-Dvbroker.agent.enableLocator=false
-Dhttp.proxyHost=
-Dhttp.proxyPort=candle.fw.pres.CMWApplet 2>& 1 >> ${LOGFILENAME}.log
```

To specify the IOR, change the line to look like the following:

```
${JAVA_HOME}/bin/java -showversion -noverify -classpath ${CLASSPATH}
-Dcnp.http.url.DataBus=http://xyz.myserver.com/cnps.ior
-Dkjr.trace.mode=LOCAL -Dkjr.trace.file=/opt/IBM/ITM/logs/kcjras1.log
-Dkjr.trace.params=ERROR -DORBtcpNoDelay=true -Dcnp.http.url.host=
-Dvbroker.agent.enableLocator=false
-Dhttp.proxyHost=
-Dhttp.proxyPort=candle.fw.pres.CMWApplet 2>& 1 >> ${LOGFILENAME}.log
```

Web Start client

If you will be using a desktop client deployed using Web Start, you must edit several configuration templates to enable connection to an external web server.

Take the following steps to configure the Web Start client:

1. On the host of the Tivoli Enterprise Portal Server, change to the following directory:
   - Windows: *itminstall_dir*`\config` (for example, `c:\ibm\itm\config`)
   - UNIX and Linux: *itminstall_dir*`/config` (for example, `/opt/IBM/ITM/config`)
2. Open **tep.jnlpt** in a text editor, and make the following changes:
   - Replace the line:

```
codebase="http://$HOST$:$PORT$///cnp/kdh/lib"
```

with:

```
codebase= "http://$HOST$/"
```

.

- Add the following statement to the set of **<property>** elements underneath the **<resources>** section:

```
<property name="cnp.http.url.DataBus" value="http://$HOST$/cnps.ior"/>
```

3. Open **component.jnlpt** in a text editor, and make the following change:
   - Replace the line:

```
codebase="http://$HOST$:$PORT$///cnp/kdh/lib/"
```

   with the following line:

```
codebase="http://$HOST$/"
```

4. For these changes to take effect, reconfigure the Tivoli Enterprise Portal Server.

Use the following URL to launch the Java Web Start client:

http://*teps_hostname*/tep.jnlp

Reverting from the IBM HTTP Server to the internal web server

Starting with IBM Tivoli Monitoring V6.2.3, the IBM HTTP Server is the default web server used for communication between the portal client and server. This provides increased scalability, performance, and reliability for portal clients. The switch to IBM HTTP Server occurs automatically with no action required on your part. The IBM HTTP Server does require the following ports to be open in any firewall between the client and the server:

- Port: 15200 – HTTP
- Port: 15201 – HTTPS

If you do not want to open these ports you can use the following sections to revert from the IBM HTTP Server back to the internal web server:

- “Portal server”
- “Browser client”
- “Desktop client” on page 405
- “Web Start client” on page 405

Portal server

Take the following steps to configure the Portal client:

1. The variable KFW_IOR_IHS_REDIRECT=N must be added to the environment file:
   - On Windows systems, edit the `install_dir/CNPS/KFWENV` file.
   - On Linux or AIX systems, edit the `install_dir/config/cq.ini` file.
2. Add the variable KFW_IOR_IHS_REDIRECT=N to the environment file.

Browser client

Take the following steps to configure the Browser client:

1. The command `document.writeln( '<PARAM NAME= "cnp.http.useIHS" VALUE="false">' );` must be added to the `document.writeln` commands:
   - On Windows systems, edit the `install_dir/CNB/applet.html` file.
   - On Linux or AIX systems, edit the `install_dir/platform/cw/applet.html` file.

2. Scroll down to the bottom of the file, where a section of `document.writeln` commands create a set of <PARAM> tags. Add the `document.writeln( '<PARAM NAME= "cnp.http.useIHS" VALUE="false">' );` command.

Desktop client

Take the following steps to configure the Desktop client:

1. The JVM parameter `-Dcnp.http.useIHS=false` must be added to the parameter list:
   - On Windows systems, edit the `install_dir/CNP/cnp.bat` file and locate the line beginning with `set _CMD=`.
   - On Linux or AIX systems, edit the `install_dir/bin/cnp.sh` file and locate the line beginning with `${TEP_JAVA_HOME}/bin/java`.
2. Add the JVM parameter `-Dcnp.http.useIHS=false` to the parameter list.

Web Start client

Take the following steps to configure the Web Start client:

1. Change the JNLP template files:
   - On Windows systems, edit the **`install_dir/Config/tep.jnlpt`** file and locate the line beginning with `codebase=`.
   - On Linux or AIX systems, edit the **`install_dir/config/tep.jnlpt`** file and locate the line beginning with `codebase=`.
2. Change the value of the codebase variable to `codebase=`**`"http://$HOST$:1920///cnp/kdh/lib/"`**. Do not substitute the portal server hostname for the `$HOST$` variable.
3. Repeat steps 1 and 2 for the **`component.jnlpt`** file in the specified directory.
4. You must reconfigure the Tivoli Enterprise Portal Server to update all of the `.jnlp` files.

Configuring HTTP communication between the portal client and server

HTTP communication can be used between the portal client and server without the need for the CORBA communication protocol. HTTP communication requires the use of the IBM HTTP Server that is installed automatically with the portal server. Two new ports are available for client/server communication:

- Port: 15200 – HTTP
- Port: 15201 – HTTPS

Use the following section to change the port numbers used by IHS:

- “Changing the ports used by IHS” on page 407

Use the following sections to switch clients to use HTTP communications with the portal server:

- “Browser client”
- “Desktop client” on page 406
- “Web Start client” on page 406

Browser client

Take the following steps to configure the browser client:

On Windows systems:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal browser client**.
2. Select **Reconfigure**.
3. In the properties list, locate and double-click the property **tep.connection.protocol**.
4. In the Edit Tivoli Enterprise Portal Browser Parameter window, change the value to **`http`**.

5. Click **OK**.
6. Click **OK** to close the Configure Tivoli Enterprise Portal Browser window.

On Linux/UNIX systems:

1. Edit the `$CANDLEHOME/platform/cw/applet.html` file.
2. Scroll down to the section of `document.writeln` commands at the bottom of the file. Add the following line:
   ```
   document.writeln( '<PARAM NAME= "tep.connection.protocol" VALUE="http">' );
   ```
3. To enable secure communication using https, substitute https where http is specified for the new document.writeln parameter.

For the browser client, the valid options are **iiop** and **http**. These values instruct the client to use either the CORBA IIOP protocol or the HTTP protocol. The use of the secure HTTPS protocol is determined by the protocol specified in the URL used to connect to the portal server. For example, if you connect to the portal server using a URL that starts with **`http://`**, all communication with the server is unencrypted. The following table describes the browser client behavior:

Table 69. Browser client behavior

`tep.connection.protocol` **value**	**URL begins with**	**Result**
http	"http://"	All communication uses the HTTP protocol.
https	"https://"	All communication uses the HTTPS protocol.

Desktop client

Take the following steps to configure the desktop client:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal desktop client**.
2. Select **Reconfigure**.
3. In the properties list, locate and double-click the property **tep.connection.protocol**.
4. In the Edit Tivoli Enterprise Portal Browser Param window, change the value to **`http`** or **`https`**.
5. Click **OK**.
6. Click **OK** to close the Configure Tivoli Enterprise Portal Browser window.

Web Start client

Take the following steps to configure the Web Start client:

1. On the host of the Tivoli Enterprise Portal Server, change to the following directory:
   - On Windows systems: `*itminstall_dir*\config`. For example, `C:\ibm\itm\config`.
   - On Linux/UNIX systems: `*itminstall_dir*/config`. For example, `/opt/IBM/ITM/config`.
2. Open the `tep.jnlpt` file in a text editor and make the following changes:
   a. Under the `<resources>` section, add:
   ```
   <property name="tep.connection.protocol" value="http"/>
   ```
   b. To use HTTPS protocol, replace the **value** with the string **`https`**.
3. On the host of the Tivoli Enterprise Portal Server, change to the following directory:
   - On Windows systems: `%CANDLE_HOME%\CNB\`.
   - On Linux/UNIX systems: `$CANDLEHOME/platform/cw/`.
4. Open the `tep.jnlp` file in a text editor and make the following changes:
   a. Under the `<resources>` section, add the following parameter to the list of JVM parameters:

```
<property name="tep.connection.protocol" value="http"/>
```

   b. To use HTTPS protocol, replace the **value** with the string **`https`**.

5. You must reconfigure the Tivoli Enterprise Portal Server for these changes to take effect.

Changing the ports used by IHS

Complete the following steps to change the ports used by IHS to listen for HTTP/S communication between the client and server:

1. On the host of the Tivoli Enterprise Portal Server, change to the following directory:
   - On Windows systems: *itminstall_dir*\IHS\CONF
   - On Linux/UNIX systems: *itminstall_dir*/*arch*/iu/ihs/conf
2. Open the `httpd.conf` file in a text editor and change the following listen directive:

   ```
   Listen 0.0.0.0:15200
   ```

   to:

   ```
   Listen 0.0.0.0:<http port #>
   ```

   where **`<http port #>`** is the new port to use for HTTP communication.
3. To change the HTTPS port, change the following listen directive:

   ```
   Listen 0.0.0.0:15201
   ```

   to:

   ```
   Listen 0.0.0.0:<https port #>
   ```

   where **`<https port #>`** is the new port to use for HTTPS communication.
4. Change to the following directory:
   - On Windows systems: *itminstall_dir*\CNPSJ\profiles\ITMProfile\config\cells\ITMCell
   - On Linux/UNIX systems: *itminstall_dir*/*arch*/iw/profles/ITMProfile/config/cells/ITMCell
5. Open the `virtualhosts.xml` file in a text editor and change the following alias attribute:

   ```
   <aliases xmi:id="HostAlias_1" hostname="*" port="15200"/>
   ```

   to:

   ```
   <aliases xmi:id="HostAlias_1" hostname="*" port="<http port #>"/>
   ```
6. To change the HTTPS port, change the following alias definition:

   ```
   <aliases xmi:id="HostAlias_3" hostname="*" port="15201"/>
   ```

   to:

   ```
   <aliases xmi:id="HostAlias_3" hostname="*" port="<https port #>"/>
   ```
7. Restart the portal server to implement these changes.

For the Desktop client and Web Start client, you must explicitly define the new port number. The browser client determines the port based on the URL used to launch the client.

For the desktop client, take the following steps to change the port:

1. Open the client launcher in a text editor:
   - On Windows systems: edit *itminstall_dir*\CNP\cnp.bat
   - On UNIX and AIX systems: edit *itminstall_dir*/cnp.sh
2. Locate the line beginning with **`"set _CMD="`**.
3. Add the following parameter to the command line:

   ```
   -Dtep.connection.protocol.url.port=<port #>
   ```

where **<port #>** is the new port to use for communication with the server.

For the Web Start client, take the following steps to change the port:

1. On the host of the Tivoli Enterprise Portal Server, change to the following directory:
   - On Windows systems: `*itminstall_dir*\config`. For example, `C:\ibm\itm\config`
   - On Linux/UNIX systems: `*itminstall_dir*/config` For example, `/opt/IBM/ITM/config`
2. Open the `tep.jnlpt` file in a text editor and make the following changes:
   - Under the `<resources>` section, add:

     ```
     <property name="tep.connection.protocol.url.port" value="port #"/>
     ```
3. You must reconfigure the Tivoli Enterprise Portal Server for these changes to take effect.

Firewall network address translation (NAT) or multiple network interface cards

The URL for starting Tivoli Enterprise Portal in browser mode includes the Tivoli Enterprise Portal Server host name or IP address. The address for starting Tivoli Enterprise Portal is set for the desktop client during installation or through Manage Tivoli Enterprise Monitoring Services. If any of the following is true in your configuration, you need to define a Tivoli Enterprise Portal Server interface:

- A firewall with Network Address Translation (NAT) is used between the client and the Tivoli Enterprise Portal Server.
- The host of the Tivoli Enterprise Portal Server has more than one Network Interface Card (NIC).

On Windows you define an interface using Manage Tivoli Enterprise Monitoring Services. On Linux and AIX, you edit the `cq.ini` file manually.

Defining a Tivoli Enterprise Portal Server interface on Windows

Use the following steps to define a Tivoli Enterprise Portal Server interface on Windows:

1. On the computer where the Tivoli Enterprise Portal Server is installed, click **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**.
2. Right-click **Tivoli Enterprise Portal Server**.
3. Click **Advanced → Configure TEPS Interfaces**.

   Initially, the list has one definition named "cnps," using port 15001 for the Tivoli Enterprise Portal Server and the IBM Tivoli integrated Web server at http://*mysystem*:1920///cnp/client (where the variable *mysystem* is the host name). Port 80, for an external Web server, is assumed if the URL does not specify 1920 for the integrated Web server.

   Note: If IIS is being used as the external Web server for Tivoli Enterprise Portal over a firewall, you may experience performance problems like slow download times. Socket pooling may also cause problems under certain conditions. If you encounter problems, try using a port other than the default port 80 on the IIS server.
4. Click **Add**.
5. Define the interface. Complete the following fields:

   Interface Name
   Type a one-word title for the interface.

   Host If you are defining an interface for a specific NIC or different IP address on this computer, type the TCP/IP host address. Otherwise, leave this field blank.

   Proxy Host
   If you are using address translation (NAT), type the TCP/IP address used outside the firewall. This is the NATed address.

Port Type a new port number for the Tivoli Enterprise Portal Server. The default 15001 is for the server host address, so a second host IP address or a NATed address requires a different port number.

Proxy Port
If the port outside the firewall will be translated to something different than what is specified for **Port**, set that value here.

Enable SSL for TEP clients
Enable secure communications between clients and the portal server.

6. Click **OK** to add the new Tivoli Enterprise Portal Server interface definition to the list.

Defining a Tivoli Enterprise Portal Server interface on Linux or UNIX

To define an additional Tivoli Enterprise Portal interface on Linux or UNIX, edit the *install_dir/config/cq.ini* file as follows:

1. Locate the KFW_INTERFACES= variable and add the one-word name of the new interface, separating it from the preceding name by a space. For example:

```
KFW_INTERFACES=cnps myinterface
```

2. Following the entries for the default cnps interface, add the following variables as needed, specifying the appropriate values:

KFW_INTERFACE_*interface_name***_HOST=**
If you are defining an interface for a specific NIC or different IP address on this computer, specify the TCP/IP host address.

KFW_INTERFACE_*interface_name***_PORT=**
Type a port number for the Tivoli Enterprise Portal Server. The default 15001 is for the server host address, so a second host IP address or a NATed address requires a different port number.

KFW_INTERFACE_*interface_name***_PROXY_HOST=**
If you are using address translation (NAT), type the TCP/IP address used outside the firewall. This is the NATed address.

KFW_INTERFACE_*interface_name***_PROXY_PORT=**
If the port outside the firewall will be translated to something different than what is specified for **Port**, set that value here.

KFW_INTERFACE_*interface_name***_SSL=Y**
If you want clients to use Secure Sockets Layers (SSL) to communicate with the Tivoli Enterprise Portal Server, add the following variable.

Firewall scenarios for Tivoli Enterprise Portal

The following diagrams illustrate several firewall scenarios using various combinations of the IBM Tivoli integral Web server, an external Web server (such as Apache or IBM HTTP Server), NAT, and a second NIC on the Tivoli Enterprise Portal Server computer. These scenarios can help you to define the Tivoli Enterprise Portal Server interface.

Note: You can download the IBM HTTP Server for free at http://www-306.ibm.com/software/webservers/httpservers/.

Figure 104 on page 410 shows scenario with the following configuration:

- Has an intranet firewall
- Has no NAT
- Uses the integral Web server

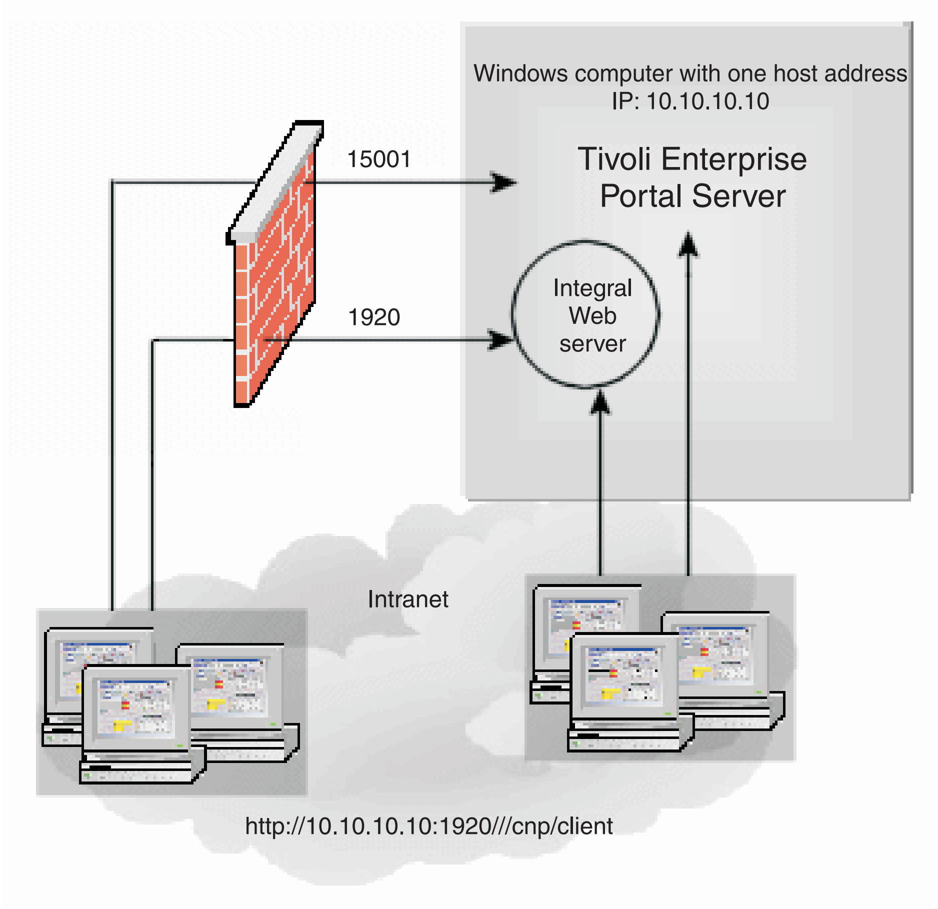


Figure 104. Intranet with integral Web server

The default Tivoli Enterprise Portal Server interface "cnps" is used. No additional interface definitions are needed. Browser mode users, whether going through the firewall or not, start Tivoli Enterprise Portal at:

```
http://10.10.10.10:1920///cnp/client
```

or substitute the host name for the IP address.

For configurations using the integrated Web server and these port numbers, use the default cnps interface definition.

In this scenario, the monitoring server and agents can be installed on the Tivoli Enterprise Portal Server computer.

Figure 105 on page 411 shows a scenario that has the following configuration:

- Has an intranet firewall
- Has no NAT
- Uses an external Web server (such as IBM HTTP Server, Apache or IIS)

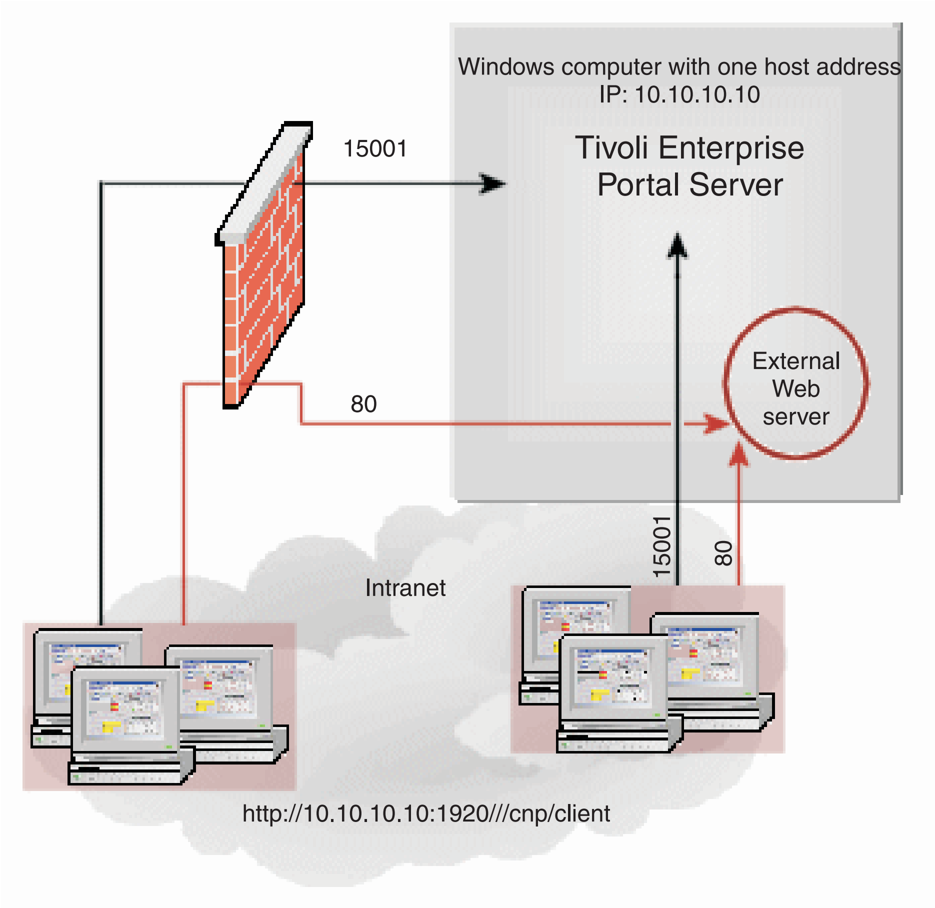


Figure 105. Intranet with external Web server

Browser mode users, whether going through the firewall or not, start Tivoli Enterprise Portal Server with

```
http://10.10.10.10 or http://10.10.10.10/mydirectory
```

(where mydirectory is the alias), or substitute the host name for the IP address.

For intranet configurations using an external Web server, with no NAT, you do not need to add a new interface definition. Web server port 80 is used automatically when none is specified in the URL.

In this scenario, the monitoring server and agents can be installed on the Tivoli Enterprise Portal Server computer.

Figure 106 on page 412 shows the following two-part configuration:

- Intranet firewall without NAT and using the integral Web server
- Internet firewall with NAT and using an external Web server

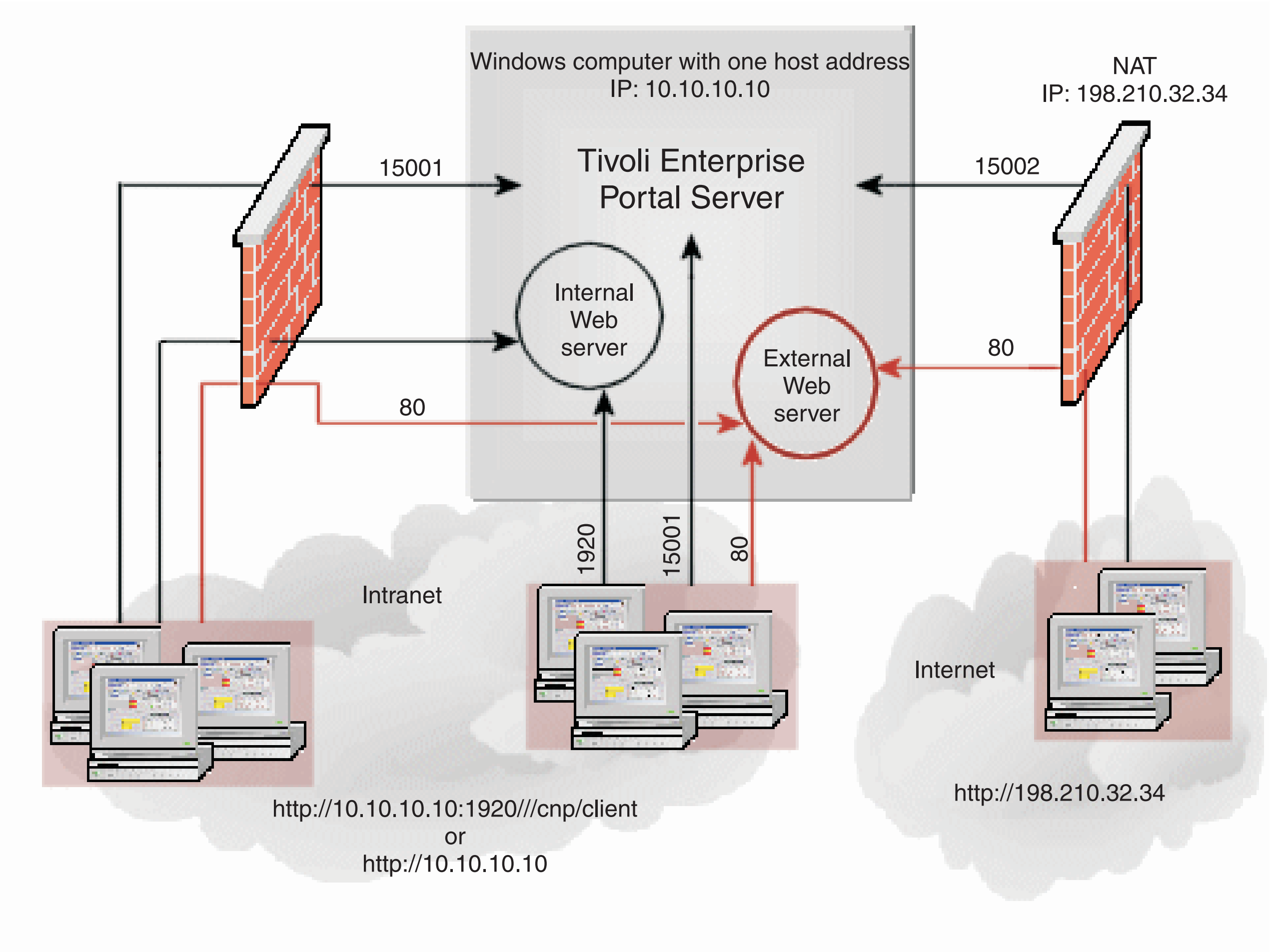


Figure 106. Intranet with integral Web server; Internet with external Web server

Intranet users can enter the URL for either the integral Web server or the external Web server:

```
 http//10.10.10.10:1920///cnp/client or http://10.10.10.10
```

Internet users enter the URL for the NATed address:

```
http://198.210.32.34/?ior=internet.ior
```

(or substitute the host name for the IP address).

The Internet configuration requires a new Tivoli Enterprise Portal Server interface named "internet", with proxy host address 198.210.32.34 and port number 15002. The intranet firewall uses the "cnps" definition.

In this scenario, the monitoring server and agents cannot be installed on the Tivoli Enterprise Portal Server computer.

Figure 107 on page 413 shows the following three-part configuration:

- Intranet firewall with NAT through the firewall to the external Web server

  ```
  http://192.168.1.100/?ior=intranet.ior
  ```

- Without NAT inside the DMZ to the integral Web server

  ```
   http://10.10.10.10:1920///cnp/client
  ```

- Internet firewall with NAT through the firewall to the external Web server

  ```
  http://198.210.32.34/?ior=internet.ior
  ```

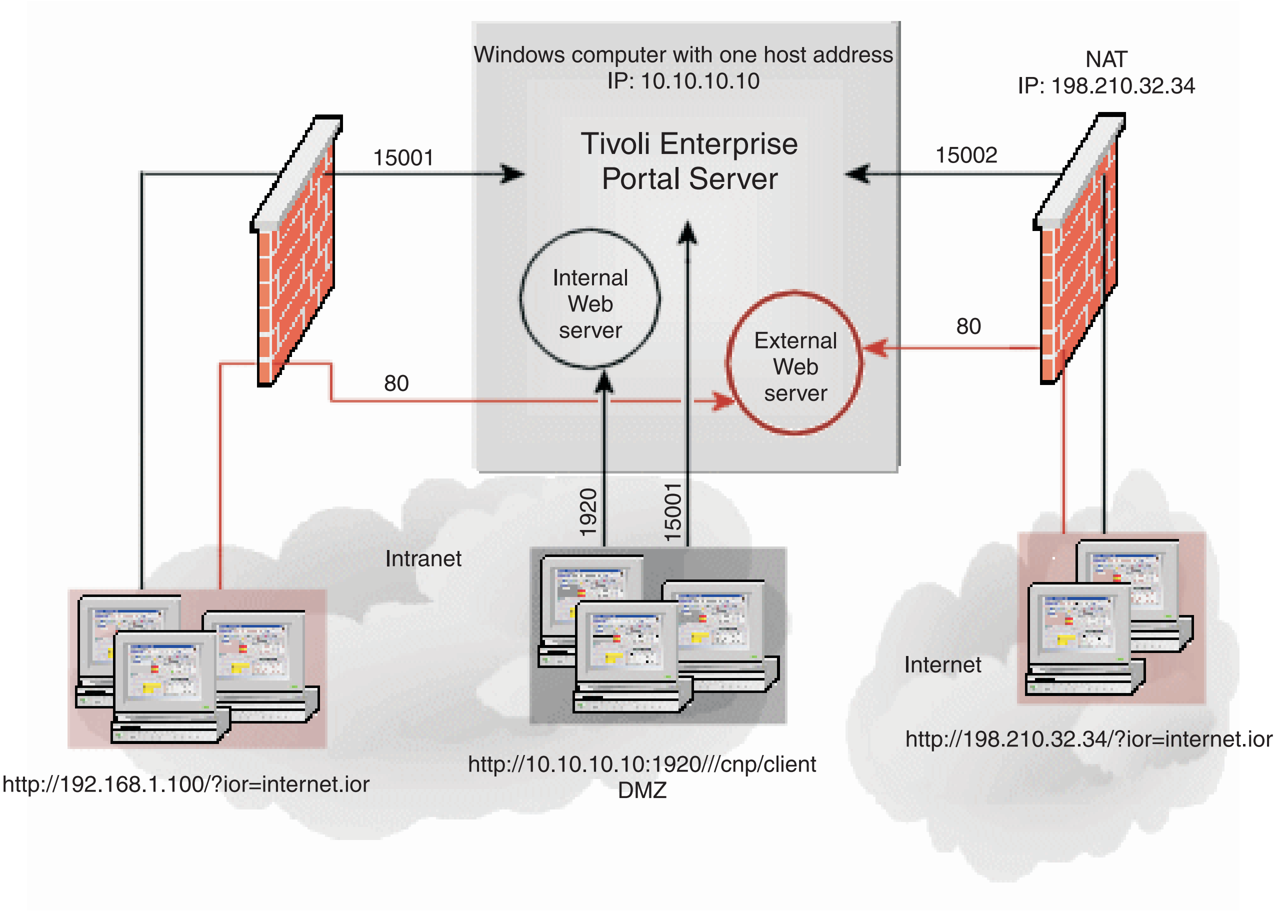

Windows computer with one host address
IP: 10.10.10.10
Tivoli Enterprise Portal Server
NAT
IP: 198.210.32.34
15001
15002
Internal Web server
External Web server
80
80
1920
15001
Intranet
Internet
http://192.168.1.100/?ior=internet.ior
http://10.10.10.10:1920///cnp/client
DMZ
http://198.210.32.34/?ior=internet.ior

Figure 107. Intranet and Internet with integral and external Web servers

The intranet firewall configuration requires a new Tivoli Enterprise Portal Server interface named "intranet", which uses proxy host 192.168.1.100 and port 15003.

The Internet DMZ configuration requires a new Tivoli Enterprise Portal Server interface definition.

The Internet configuration uses the same Tivoli Enterprise Portal Server "internet" interface definition as the previous scenario: proxy host 198.210.32.34 and port 15002.

In this scenario, the monitoring server and agents cannot be installed on the Tivoli Enterprise Portal Server computer.

Figure 108 on page 414 shows the following two-part configuration:

- Intranet firewall with NAT through the firewall to the external Web server using http://192.168.1.100, and without NAT inside the DMZ to the integral Web server uses http://10.10.10.10:1920///cnp/client
- Internet firewall with NAT through the firewall to the external Web server using http://198.210.32.34.

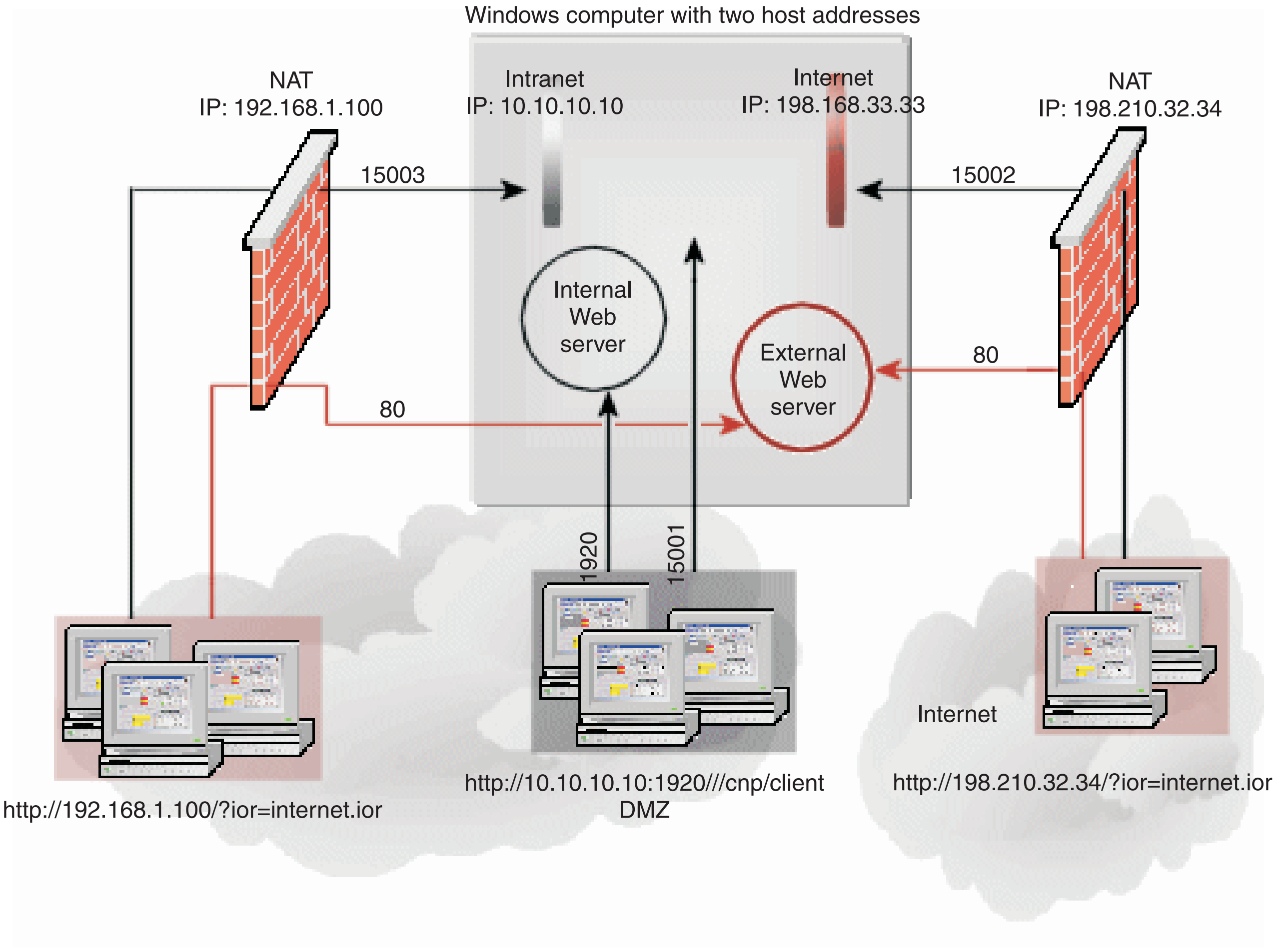


Figure 108. Two host addresses, intranet and Internet, with integral and external Web servers

The intranet firewall configuration uses the same Tivoli Enterprise Portal Server interface definition (named "intranet") as in the scenario shown in Figure 107 on page 413: http://10.10.10.10; proxy host 192.168.1.100; and port 15003.

The intranet DMZ configuration uses the default Tivoli Enterprise Portal Server interface definition: host 192.168.33.33; proxy host 198.210.32.34; port 15002; and proxy port 444.

In this scenario, the monitoring server and agents cannot be installed on the Tivoli Enterprise Portal Server computer.

Chapter 16. Configuring IBM Tivoli Monitoring Web Services (the SOAP Server)

IBM Tivoli Monitoring Web Services provide an industry-standard open interface into products that use the Tivoli Management Services framework. This open interface provides easy access to performance and availability data, allowing you to use this information for advanced automation and integration capabilities. Web Services implements a client/server architecture. The client sends SOAP requests to the SOAP server. The server receives and processes the SOAP requests from the client. Predefined SOAP methods let you perform many functions within the monitored environment. Using Web Services requires a basic understanding of SOAP, XML and XML Namespaces, and the Web Services Description Language (WSDL).

By default, the SOAP server is enabled on all hub monitoring servers. This chapter describes how to configure security on a SOAP server, and how to configure communication between SOAP servers.

Note: You cannot make SOAP requests from IBM Tivoli Monitoring to earlier SOAP servers (such as those in OMEGAMON Platform V350/360).

For complete information about using Web Services and customizing the SOAP interface for your site, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Table 70 outlines the steps required to configure the SOAP server.

Table 70. Overview of SOAP Server configuration steps

Task	Where to find information
Define the hubs with which your SOAP Server communicates.	“Defining hubs”
Create users and grant them access.	“Adding users” on page 418
Verify that you have successfully configured SOAP.	“Verifying the configuration” on page 420

Defining hubs

In this step you activate a SOAP server and define hubs with which it communicates. You can use Manage Tivoli Enterprise Monitoring Services to configure the SOAP server. On Linux and UNIX you can also use the **itmcmd config** command. After you configure the SOAP Server, you must restart the Tivoli Enterprise Monitoring Server.

- “Windows: Defining hubs”
- “UNIX and Linux: Defining hubs (GUI procedure)” on page 416
- “UNIX and Linux: Defining hubs (CLI procedure)” on page 417

Windows: Defining hubs

Use the following steps to define SOAP hubs on Windows:

1. Start Manage Tivoli Enterprise Monitoring Services (select **Start** → **(All) Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**).
2. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Monitoring Server**.
3. Click **Advanced** → **Configure SOAP Server Hubs**.

   The SOAP Server Hubs Configuration window is displayed. If the name of the local hub does not appear in the tree, define the local hub, including assigning user access, before defining the hubs with which it communicates.
4. Click **Add Hub**. The Hub Specification window is displayed.

5. Select the communications protocol to be used with the hub from the **Protocol** menu.
6. Specify an alias name in the **Alias** field.

   The alias for the local hub monitoring server must always be "SOAP". For hubs with which the local SOAP Server communicates, you may choose any alias (for example, `HUB2`). Alias names can be a minimum of 3 characters and a maximum of 8 characters.
7. Do one of the following:
   - If you are using TCP/IP or TCP/IP Pipe communications, complete the fields in Table 71:

Table 71. TCP/IP Fields in Hub Specification dialog

Field	Description
Hostname or IP Address	The host name or TCP/IP address of the host computer.
Port	The TCP/IP listening port for the host computer.

   - If you are using SNA communications, complete the fields in Table 72:

Table 72. SNA Fields in Hub Specification dialog

Field	Description
Network Name	Your site SNA network identifier.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU6.2 LOGMODE	The name of the LU6.2 logmode. Default: CANCTDCS.
TP Name	The Transaction Program name for the monitoring server.

8. Click **OK**. The server tree is displayed, with the newly defined hub.

You can define any hubs with which the local SOAP Server will communicate by repeating steps 4 - 7, or you can specify which user IDs can access the SOAP Server you just defined and what access privileges they have. See “Adding users” on page 418.

UNIX and Linux: Defining hubs (GUI procedure)

Use the following steps to define SOAP hubs on UNIX or Linux using Manage Tivoli Enterprise Monitoring Services:

1. Change to the *itm_install_dir*/bin directory and start Manage Tivoli Enterprise Monitoring Services by entering the following command:

   ```
   ./itmcmd manage
   ```

   The Manage Tivoli Enterprise Monitoring Services window is displayed.
2. Right-click **Tivoli Enterprise Monitoring Server** and select **Configure** from the popup menu.

   The Configure TEMS window is displayed.
3. Click **Save**.

   The SOAP Server Hubs Configuration window“UNIX and Linux: Defining hubs (GUI procedure)” is displayed.
4. Confirm that the host name or IP address, port number, and protocol displayed for the hub monitoring server are correct. If not, correct them.

   If the name of the local hub does not appear in the tree, define the local hub before defining the hubs with which it communicates. The alias for the local hub must always be "SOAP".
5. To add another hub:
   a. Type the name or IP address and the port number of the host in the appropriate fields. The port should be match the hub's protocol port number. By default, this is 1918.

   b. Specify an alias name in the **Alias** field.

      Alias names can be a minimum of 3 characters and a maximum of 8 characters (for example, HUB2).

   c. Select the communications protocol to be used with the hub from the **Transport** menu.

6. Click **Add Host**.

   The server tree is displayed, with the newly defined hub.

You can now define any hubs with which the local SOAP Server will communicate by repeating step 5, or you can specify which user IDs can communicate with the local SOAP Server and what access privileges they have.

UNIX and Linux: Defining hubs (CLI procedure)

Complete the following steps to configure the SOAP server:

1. On the host of the hub monitoring server on which you want to configure Web Services, change to the *install_dir*/bin directory and enter the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

2. Accept the default values, which should reflect the choices made during the last configuration, until you see the following prompt:

   ```
   *************************

   Editor for SOAP hubs list

   *************************

   Hubs
   ## CMS_Name
   1 ip.pipe:TEMS_NAME[port_#]

   1)Add, 2)Remove ##, 3)Modify Hub ##, 4)UserAccess ##, 4)Cancel, 5)Save/exit:
   ```

3. To add a hub with which the local hub can communicate:

   a. Type 1 and press Enter.

   b. Respond to the following prompts as shown in :

      ```
      Network Protocol [ip, sna, ip.pipe, or ip.spipe] (Default is: ip):
      CMS Name (Default is: local_host):
      Port Number (Default is: 1918):
      Alias (Default is: SOAP):
      ```

Table 73. SOAP hub configuration values

Network Protocol	The communications protocol configured for the hub monitoring server
CMS Name	The host name of the hub monitoring server. The host name must be fully
Port Number	The protocol port for the hub monitoring server.
Alias	An alias for the hub. Alias names can be a minimum of 3 characters and a maximum of 8 characters. The alias for the local hub must always be SOAP, but you must create a new alias for additional hubs (for example, HUB2).

   After you enter the alias, the list of hubs is displayed with the new hub added. For example,

   ```
   Hubs
   ##      CMS_Name
   1       ip.pipe:chihuahua[1918]
   2       ip:maple[1918]

   1)Add, 2)Remove ##, 3)Modify Hub ##, 4)UserAccess ##, 5)Cancel, 6)Save/exit:
   ```

   You can continue to add hubs, or you can proceed to define user access for the hubs you have already defined.

Adding users

Access to SOAP server can be secured in one of two ways: by enabling **Security: Validate User** and creating user accounts on the host of the hub monitoring server, or by adding specific users to the server definition.

If **Security: Validate User** is *not* enabled and no users are added to the server definition, the SOAP server honors all requests regardless of the sender. If **Security: Validate User** *is* enabled on the hub monitoring server, the SOAP server honors requests only from users defined to the operating system or security authorization facility of the monitoring server host.

However, if any users are added to the SOAP server definition, only those users who have also been defined to the operating system or the security authorization facility of the monitoring server host have access to the server, regardless of whether or not **Security: Validate User** is enabled.

In this step you define users to a SOAP Server and specify the access privileges for each user: Query or Update. The access privileges control what methods the user is allowed to use. Update access includes Query access. Table 74 lists the methods associated with each access. For information on using these methods, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Table 74. SOAP methods associated with access privileges

Method	Update	Query
CT_Acknowledge	x	
CT_Activate	x	
CT_Alert	x	
CT_Deactivate	x	
CT_Email	x	x
CT_Execute	x	x
CT_Export	x	x
CT_Get	x	x
CT_Redirect	x	x
CT_Reset	x	
CT_Resurface	x	
CT_Threshold	x	
WTO	x	x

After you configure the SOAP Server, you must restart the Tivoli Enterprise Monitoring Server.

Windows: Adding users

Use the following steps to define users to a SOAP server:

1. In the SOAP Server Hubs Configuration window, select the server (click anywhere within the server tree displayed).
2. Under Add User Data, type the user name.

   If **Security: Validate User** is enabled, user IDs must be identical to those specified for monitoring server logon validation. Access is restricted to only that monitoring server to which a user has access.
3. Click the type of user access: **Query** or **Update**.
4. Click **Add User**. The server tree is updated, showing the user and type of access.
5. To delete a user: Select the user name from the tree and click **Delete Item**.
6. To delete a hub: Click anywhere within the hub's tree and click **Clear Tree**.

UNIX or Linux: Adding users (GUI)

Complete the following steps to define which users can make requests to a hub and assign access privileges to those users, using a graphical user interface:

1. If you are not already in the SOAP Server Hubs Configuration window, perform steps 1 - 3 in “UNIX and Linux: Defining hubs (GUI procedure)” on page 416.
2. In the hub tree, select the **Access** node for the hub to which you want to define users.

   The values for that hub are displayed in the section above the tree.
3. Type the user ID you want to add in the **User** field.

   If the **Security: Validate User** option is enabled on the hub, the user ID must be a valid user ID on the hub system.
4. Use the radio buttons in the **Access** field to select the type of access you want to grant to the user.

   See Table 74 on page 418 for a list of the method each type of access includes.
5. Click **Add User**.

   The user ID appears under the appropriate subnode of the Access node of the selected hub.
6. To add additional users, repeat steps 2 through 5.
7. Click **OK** to close the window and save the changes.

UNIX or Linux: Adding users (CLI)

Complete the following steps to define users to a hub and assign access privileges using the command-line interface:

1. If you are not already at the following prompt, perform steps 1 and 2 in “UNIX and Linux: Defining hubs (CLI procedure)” on page 417:

   ```
   *************************

   Editor for SOAP hubs list

   *************************

   Hubs
   ## CMS_Name
   1 ip.pipe:TEMS_NAME[port_#]

   1)Add, 2)Remove ##, 3)Modify Hub ##, 4)UserAccess ##, 4)Cancel, 5)Save/exit:
   ```
2. To define users and assign access privileges, enter 4 followed by a space, and then the number (in the list) of the hub you want to configure. For example:

   ```
   1)Add, 2)Remove ##, 3)Modify Hub ##, 4)UserAccess ##, 5)Cancel, 6)Save/exit:4 1
   ```

   The following prompt is displayed:

   ```
   => Query Access:

   => Update Access:

   1)QueryAdd <user>, 2)UpdateAdd <user>, 3)Remove ##, 4)Exit :
   ```

   Any users already defined are listed under the corresponding access.
3. To define a user with Query access, enter `1` followed by a space and the user ID. To define a user with Update access, enter `2` followed by a space and the user ID. See Table 74 on page 418 for a list of the methods associated with each type of access.

   Note: If the **Security: Validate User** option is enabled, the user ID must be a valid user on the hub system.

   The prompt appears again with the new user added to the appropriate access list. You can continue to add users by repeating step 3 or complete the configuration by typing `4` and pressing Enter.

Verifying the configuration

In this step you verify that SOAP has been correctly configured by starting the SOAP client and making a request using the command-line utility kshsoap.

Use the following steps:

1. Verify that the monitoring server that you used to enable SOAP is running. If not, start it.
2. Open a command window.
3. Change to the *install_dir*\cms directory (on Windows) or the *install_dir*/TBD (on UNIX and Linux operating systems).
4. In the current directory, create an ASCII text file named SOAPREQ.txt that contains the following SOAP request:

   ```
   <CT_Get><object>ManagedSystem</object></CT_Get>
   ```

   Or if security has been enabled:

   ```
   <CT_Get><userid>logonid</userid><password>password</password> \
   <object>ManagedSystem</object></CT_Get>
   ```

5. Create another ASCII text file named URLS.txt that contains URLs that will receive the SOAP request. For example: http://hostname:1920///cms/soap
6. Run the following command:

   ```
   kshsoap SOAPREQ.txt URLS.txt
   ```

   SOAPREQ.txt is the name of the file that contains the SOAP request and URLS.txt is the name of the file that contains the URLs.

The kshsoap utility processes the SOAPREQ.txt file and displays the output of the SOAP request in the command window. The SOAP request is sent to each URL listed in URLS.txt, and the SOAP response from each URL displays in the DOS window.

Chapter 17. Performance tuning

This chapter contains information about optimizing the performance of several components within an IBM Tivoli Monitoring environment. The following topics are considered:

- "Configuring the heartbeat interval" on page 372
- "Tivoli Enterprise Monitoring Server"
- "Tivoli Enterprise Monitoring agents" on page 424
- "Agentless Monitoring" on page 425
- "Tivoli Enterprise Portal Server" on page 426
- "Tivoli Enterprise Portal client" on page 430
- "Tivoli Data Warehouse" on page 434
  - "Historical data collection" on page 434
  - "Warehouse Proxy Agent" on page 435
  - "Summarization and Pruning Agent" on page 439
  - "Database tuning" on page 442
- "Optimizing queries" on page 454
- "Optimizing situations" on page 457
- "Planning for platform-specific scenarios" on page 458

Review these topics and considerations for relevancy to your environment.

This chapter includes some sections from the redbook *IBM Tivoli Monitoring: Implementation and Performance Optimization for Large Scale Environments*.

Appendix B in the *IBM Tivoli Monitoring: Troubleshooting Guide* provides an extensive list of environment variables that can be customized for different components in the Tivoli Monitoring environment. The following sections highlight specific environment variables to consider in performance tuning of the Tivoli Monitoring environment.

Note: When changing environment variables in configuration files, note that whenever maintenance or reconfiguration takes place in your environment, these changes might be lost and need to be reapplied.

Tivoli Enterprise Monitoring Server

This section provides information about parameters you might consider editing to improve either hub or remote monitoring server performance. The parameters are set in the following files according to operating system:

Windows

ITM_HOME/cms/KBBENV

For example: `C:\IBM\ITM\cms\KBBENV`

Linux and UNIX

ITM_HOME/config/*tems_hostname*_ms_*tems_name*.config

For example: `/opt/IBM/ITM/config/edinburg_ms_labtems.config`

z/OS

&shilev.&rtename.RKANPARU(KDSENV)

For example: `ITM.SYP1.RKANPARU(KDSENV)`

Note: The `&shilev` and `&rtename` are variables that correspond to high-level qualifiers of the RKANPARU(KDSENV) partitioned data set. These variables can take 1 - 8 characters.

On each occasion maintenance or reconfiguration takes place in your environment these files might be recreated and changes lost and need to be reapplied.

The following lists the settings that might affect the monitoring server performance:

KDCFP_RXLIMIT

This parameter establishes the maximum number of 1 KB packets which might be transmitted to this endpoint in a single RPC request or response. The default value is 4096 KB (4 MB); the minimum is 1024 KB (1 MB); there is no maximum. If the remote endpoint (session partner) exceed this limit (that is, send more), the RPC request fails with status KDE1_STC_RXLIMITEXCEEDED. The intent of RXLIMIT is to prevent memory overrun by placing an upper-limit on a single RPC request or response. If sufficient capacity exists in a large-scale deployment, consider setting this value to 8192.

To increase the buffer size to 8 MB, include the following environment setting: KDCFP_RXLIMIT=8192

CMS_DUPER

This parameter enables or disables situation synchronization of common filter objects monitored by agents or endpoints. Enabling this setting in monitoring server environments with predominantly z/OS address space applications for example, OMEGAMON XE for CICS or Sysplex, improves performance and response time by limiting data collection samplings on behalf of running situations. Enable it by setting the value to YES. Disable by setting the value to NO. This parameter is enabled by default.

EVENT_FLUSH_TIMER

This parameter is set (in minutes) to set an interval at which time pending I/Os are committed to situation status history as a background writer and garbage collection task. This feature improves performance of incoming event throughput into the hub monitoring server per arriving situation statuses.

EIB_FLUSH_TIMER

This parameter is used to specify in minutes how long the monitoring server waits before resetting distribution and database event requests to an initial state. This frees held resources by the request if no event information has been able to get processed in the specified time. The default setting is 2 minutes. If event requests do not respond within 2 minutes you might consider allocating a higher minutes setting to allow requests more time to process. This is particularly valid in larger, more complex, environments.

DELAY_STOP

This parameter is used to specify in seconds how long to delay monitoring server shutdown for UNIX and Linux monitoring servers, as invoked by the **itmcmd server stop** *tems_name* command. The default value is 60 seconds. The delay is used to allow network connections to close before an immediate restart of the monitoring server with the **itmcmd server start** *tems_name* command. If you do not immediately restart the monitoring server after shutting it down, this parameter can be set to a lower value to cause the monitoring server shutdown to complete more quickly.

KGLCB_FSYNC_ENABLED

This parameter is not available on Windows systems, and is not available on IBM Tivoli Monitoring V6.1 systems.

For Linux and UNIX platforms, this variable can be used to specify whether the fsync() system call should be invoked after writes to the file system. You can set this configuration variable in the standard configuration file for the monitoring server. By default, for maximum reliability, fsync() is called. If, and only if, the following line is added to the monitoring server configuration file, fsync() calls are omitted:

```
KGLCB_FSYNC_ENABLED='0'
```

The default behavior is to call fsync() after writes, which is equivalent to the setting:

```
KGLCB_FSYNC_ENABLED='1'
```

The fsync() system call flushes the dirty pages from the file system to disk and protects against loss of data in the event of an operating system crash, hardware crash, or power failure. However, it can have a significant negative effect on performance because in many cases it defeats the caching mechanisms of the platform file system. On many UNIX platforms, the operating system itself syncs the entire file system on a regular basis. For example, by default the synced daemon that runs on AIX syncs the file system every 60 seconds. This limits the benefit of fsync() calls by application programs to protecting against database corruption in the most recent 60 second window.

Tivoli Enterprise Monitoring Server tuning recommendations for large-scale environments

This section provides tuning recommendations you might consider to improve either hub or remote monitoring server performance in large scale environments.

Avoid distributing unused situations

To reduce Tivoli Enterprise Monitoring Server memory usage and minimize the amount of CPU processing when many agents connect, consider reducing the total number of situations distributed by avoiding distribution of situations that are not being used. Some situations, including predefined situations, have the default distribution set as a managed system list. These situations are distributed to all managed Tivoli Enterprise Monitoring Servers in the managed system list, even if the situation is not being used. Limiting the distribution to only the managed Tivoli Enterprise Monitoring Server where the situation is used minimizes the total number of situations distributed from the remote Tivoli Enterprise Monitoring Server, and minimizes the CPU processing when many agents connect. The distribution specification for a situation can be changed using the Situation Editor or the `tacmd editsit` command.

Check ulimit settings for open file descriptors (UNIX/Linux only)

The Tivoli Enterprise Monitoring Server can use many file descriptors, especially in a large environment. On UNIX and Linux Tivoli Enterprise Monitoring Servers, the maximum number of file descriptors available to a process is controlled by user limit parameters. To display the current user limits, use the `ulimit –a` command.

The *nofiles* parameter is the number of file descriptors available to a process. When IP:PIPE or IP:SPIPE are used for agent connectivity, persistent TCP connections are maintained to each agent, and each connection requires a file descriptor. If a file descriptor is not available when needed, unexpected behavior can occur, including program failures. For the Tivoli Enterprise Monitoring Server process (kdsmain), the *nofiles* parameter should be set larger than the maximum number of agent connections that will be maintained. Consider increasing the value to 8000 file descriptors or more.

There are other user limit parameters that control how much data, stack, and memory are available to a process. For large environments, consider increasing these memory-related user limit parameters for the monitoring server (kdsmain) process using the `ulimit` settings.

The method for changing the user limit parameters is operating system specific. Consult the operating system manuals for information about how to configure the user limit parameters.

Disable extra fsync() calls by specifying KLGCB_FSYNC_ENABLED=0 (UNIX/Linux only)

The KGLCB_FSYNC_ENABLED parameter specifies whether the fsync() system call should be invoked after writes to the file system. The default value is 1, causing fsync() to be called after every write for maximum reliability. When KGLCB_FSYNC_ENABLED=0 is set, fsync() calls are omitted after every write. The fsync() system call flushes the file system dirty pages to disk and protects against loss of data in the event of an operating system crash, hardware crash, or power failure. However, it can have a significant negative effect on performance because in many cases it defeats the caching mechanisms of the platform file system. On many UNIX platforms, the operating system itself synchronizes the entire file system on a regular basis. For example, by default the syncd daemon that runs on AIX synchronizes the file system every 60 seconds which limits the benefit of fsync() calls by application programs to protecting against database corruption in the most recent 60 second window.

Using a value of **0** improves the performance of the Tivoli Enterprise Monitoring Server when updates are being made to the EIB tables, such as during status updates and event processing. The KGLCB_FSYNC_ENABLED=0 setting should be made to the Tivoli Enterprise Monitoring Server environment file `ms.ini`. If you do not want to reconfigure the Tivoli Enterprise Monitoring Server, you must also make the change to the configuration file `<hostname>_ms_<Tivoli Enterprise Monitoring Servername>.config`. This configuration file is regenerated from the values in `ms.ini` when the Tivoli Enterprise Monitoring Server is reconfigured (using the itmcmd config command).

Minimize Tivoli Enterprise Monitoring Server shutdown time by specifying DELAY_STOP=1 (UNIX/Linux only)

The DELAY_STOP parameter specifies in seconds how long to delay monitoring server shutdown, as invoked by the `itmcmd server stop <Tivoli Enterprise Monitoring Server_name>` command. The default value is 60 seconds. If you do not immediately restart the Tivoli Enterprise Monitoring Server after shutting it down, this parameter can be set to a lower value to cause the Tivoli Enterprise Monitoring Server shutdown to complete more quickly.

To minimize Tivoli Enterprise Monitoring Server shutdown time, specify DELAY_STOP=1. The setting should be made to the Tivoli Enterprise Monitoring Server environment file `ms.ini`. If you do not want to reconfigure the Tivoli Enterprise Monitoring Server, you must also make the change to the configuration file `<hostname>_ms_<Tivoli Enterprise Monitoring Servername>.config`. This configuration file is regenerated from the values in `ms.ini` when the Tivoli Enterprise Monitoring Server is reconfigured (using the itmcmd config command).

For historical data collection, store short-term history data at the agent rather than the Tivoli Enterprise Monitoring Server when possible

For best performance, when configuring historical data collection, store the short-term history data at the agent rather than the Tivoli Enterprise Monitoring Server. Remember that for some agent types, data is required to be stored at the Tivoli Enterprise Monitoring Server. Storing short-term history data at the agent reduces usage on the Tivoli Enterprise Monitoring Server to collect and store data for all of the connected agents. When an attribute group has short term-history stored at the Tivoli Enterprise Monitoring Server, data from all the agents is stored in a single short-term history file. When a request is made for short-term history for an agent, the Tivoli Enterprise Monitoring Server must read all of the agent data stored in the history file to satisfy the request. This process is much less efficient. Storing short-term history data at the Tivoli Enterprise Monitoring Server increases the Tivoli Enterprise Monitoring Server processing usage, and lowers the number of agents that can be supported by the Tivoli Enterprise Monitoring Server.

Tivoli Enterprise Monitoring agents

This section describes agent environment variables to consider when tuning your Tivoli Monitoring environment.

CTIRA_RECONNECT_WAIT

Time interval, in seconds, for the agent to wait between attempts to register with a Tivoli Enterprise Monitoring Server. The default value is 600 seconds. Consider setting this value to a value equivalent to the CTIRA_HEARTBEAT setting, which is specified in minutes. For example, if the CTIRA_HEARTBEAT value has been set at 3 minutes, consider setting CTIRA_RECONNECT_WAIT to 180 seconds.

CTIRA_MAX_RECONNECT_TRIES

Number of consecutive times without success the agent attempts to connect to a Tivoli Enterprise Monitoring Server before giving up and exiting. The default value is 720. Using the default value along with the default CTIRA_RECONNECT_WAIT setting, the agent attempts to connect to the Tivoli Enterprise Monitoring Server for 432000 seconds (5 days) before giving up and exiting. If you lower the value for CTIRA_RECONNECT_WAIT, consider increasing this value to maintain an equivalent retry period. For example, if you lower the CTIRA_RECONNECT_WAIT value to 180 seconds, consider increasing this value to 2400.

Agentless Monitoring

This section provides tuning recommendations you might consider to improve Agentless Monitoring performance in large-scale environments.

Adjust the thread pool size, refresh interval, and cache time-to-live parameters for your environment

For a complete list of Agentless Monitoring parameters, see the *IBM Tivoli Monitoring: Agent Builder User's Guide*. The following are some of the main parameters:

CDP_DP_THREAD_POOL_SIZE
Controls the number of threads in the thread pool.

CDP_DP_REFRESH_INTERVAL
Controls how often the thread pool collects the data for each attribute group. The refresh interval should also be correlated with CDP_DP_CACHE_TTL. Both are specified in seconds, and default to 60. If the refresh interval was stretched out to, for example, 300 while the cache ttl stayed at 60, then there would be a 4 minute period during which the cache was considered expired, and any queries that came in during that window would cause an on-demand collection to happen. The best practice is to keep these two values the same.

To determine if the value for CDP_DP_THREAD_POOL_SIZE is correct, check the Thread Pool Status attribute group and look at the Average Queue Length and Average Job Wait fields, and the Intervals Skipped field of the Performance Object status. If these numbers are getting large, and the processor load on the system is in the acceptable range, then the size of the thread pool can be increased. One thread per subnode might be a good starting point, although if there are fewer than 60 nodes, there is probably no reason to decrease below the default 60 threads. If the load on the system is too high, or if the Average Queue Length, Average Job Wait , and Intervals Skipped do not decrease, collecting the data less often can decrease the load. The default refresh interval is 60 seconds, so increasing that number, and the CDP_DP_CACHE_TTL as well, can decrease the overall load.

To monitor many monitoring servers, spread the target monitoring servers across multiple process instances

Although it is possible to monitor 100 target monitoring servers from a single process instance, if historical data collection is used the attribute group data from all of the target monitoring servers is written to a single attribute group file. This exhibits the same behavior as keeping agent historical data on the monitoring server rather than the monitoring agent. When a Tivoli Enterprise Portal client requests short-term historical data from a single target system, the entire attribute group file must be ready to satisfy the request.

To monitor a large number of target monitoring servers using agentless monitoring, it is better to use multiple process instances on the same system, and spread out the target monitoring servers across the process instances. This keeps the number of target monitoring servers with data in a single file to a smaller number, and improves response time for Tivoli Enterprise Portal requests for target system short-term historical data.

Avoid collecting historical data for the KR2_WMI_Event_Log attribute group

The KR2_WMI_Event_Log attribute group generates a high number of rows and collecting historical data for this attribute group can affect performance.

Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server (portal server) acts as a conduit for Tivoli Enterprise Portal clients requesting data for analysis from monitoring agents and other components within the enterprise. The portal server connects directly to the hub monitoring server, which it queries for enterprise information and receiving updates as they occur. As it is responsible for handling large amounts of data, it can be a potential bottleneck within the IBM Tivoli Monitoring environment. This section outlines considerations to optimize the portal server performance.

The portal server database stores information related to the presentation of monitored data at the portal client, including definitions related to users, workspaces and views. After IBM Tivoli Monitoring V6.1 Fix Pack 3, the portal server also stores information related to events and event attachments in the portal server database. Before IBM Tivoli Monitoring V6.1 Fix Pack 3, the portal server database required little or no tuning. In environments with a moderate to high rate of events, the portal server database might require some tuning to optimize performance. In particular, the KFWTSIT table, which is used to store events, can grow large.

If you are using DB2 for the portal server database, consider the following tuning recommendations:

The default buffer pool size is 250 4K pages. Increase the size of this buffer pool to minimize disk I/O activity.

The following commands illustrate how to increase the size of the buffer pool to 2000 4K pages (from a DB2 command prompt):

```
CONNECT TO TEPS;
ALTER BUFFERPOOL IBMDEFAULTBP IMMEDIATE SIZE 2000;
CONNECT RESET;
```

Because the KFWTSIT table is the most active table, use the runstats and reorg facilities on a regular (for example, daily) basis.

The following commands illustrate how to run the RUNSTATS facility on the KFWTSIT table (from a DB2 command prompt):

```
CONNECT TO TEPS;
RUNSTATS ON TABLE TEPS.KFWTSIT AND INDEXES ALL ALLOW WRITE ACCESS ;
CONNECT RESET;
```

The following commands illustrate how to run the REORG facility on the KFWTSIT table (from a DB2 command prompt):

```
CONNECT TO TEPS;
REORG TABLE TEPS.KFWTSIT INPLACE ALLOW WRITE ACCESS  START ;
CONNECT RESET;
```

Note: These tuning changes can also be made using the DB2 Control Center.

Configure an external Web server for large environments

If there are more than ten portal clients connected to the portal server concurrently, consider configuring an external Web server to work with the portal clients. An external Web server will offload processing from the portal server process (KfwServices), and offer enhanced scalability.

Instructions for configuring an external Web server to work with the Tivoli Enterprise Portal can be found in Chapter 15, “Additional Tivoli Enterprise Portal configurations,” on page 397.

Note: You can download the IBM HTTP Server for free at http://www-01.ibm.com/software/webservers/httpservers/

Portal server parameter tuning

This section provides some information about parameters you might consider editing to improve portal server performance. The parameters are set in the following files according to operating system:

Windows

ITM_HOME/CNPS/kfwenv

For example: C:\IBM\ITM\CNPS\kfwenv

Linux and UNIX

ITM_HOME/config/cq.ini

For example: /opt/IBM/ITM/config

On each occasion maintenance or reconfiguration takes place in your environment these files might be recreated and changes lost and must be reapplied.

Note: For parameter changes made in the portal server configuration file to take effect, the portal server must be stopped and restarted.

Parameters affecting event processing

KFW_CMW_EVENT_SLEEP

In a complex environment, you might have a number of events occurring simultaneously. This variable specifies a time in seconds the portal server waits between processing similar events. Consider setting this variable to a value of less than 10 seconds if you are experiencing slow performance, such as portal client refresh, as a result.

KFW_EVENT_ASSIST

The Event Assistant is an internal component within the portal server that allows the user to:

- Associate attachments to events, such as logs
- Assign ownership to events and transfer ownership between users
- View events specific to the current user for the convenience of working with events which they own
- View closed events along with any associated information provided by the user

The Event Assistant creates multiple tables within the portal server database, and processing overhead by the Event Assistant increases the portal server CPU, disk and memory usage.

The Event Assistant is enabled by default. To reduce the processing demands for the portal server, the Event Assistant can be disabled by setting KFW_EVENT_ASSIST=N in the portal server configuration file (KFWENV on Windows, cq.ini on Unix or Linux). Disabling the Event Assistant causes the following behavior:

- Event records are not written to the portal server database.
- The ability for any user to attach files to an event is disabled. This is equivalent to all users not having the Attach permission.
- The My Acknowledged Events view are not updated with events that are being acknowledged.
- The Event Notes view are not populated. However, to view notes and previously existing attachments, a user can still view notes through the Event Notes and Acknowledgement dialogs.
- The *Similar Events by ...* view is not populated.

KFW_EVENT_RETENTION

The number of days to keep a closed event. For example, to prune an event 2 days after it is close, specify 2. By default, no event pruning occurs. If the Event Assistant is disabled (KFW_EVENT_ASSIST=N), this parameter is ignored.

KFW_PRUNE_START

The time of day to start pruning data, specified in 24-hour notation (hh:mm). For example, to begin pruning data at 11:00 PM, specify 23:00. By default, no event pruning occurs. If the Event Assistant is disabled (KFW_EVENT_ASSIST=N), this parameter is ignored.

KFW_PRUNE_END
The time of day to stop pruning data, specified in 24-hour notation (hh:mm). For example, to stop pruning data at midnight, specify 24:00. By default, no event pruning occurs. If the Event Assistant is disabled (KFW_EVENT_ASSIST=N), this parameter is ignored.

You can control the size of file attachments for events either at the individual client level or at the monitoring environment level, by changing environment variables in the portal server environment file. Consider changing these if you want to restrict the number of large attachments to be held at the portal server:

KFW_ATTACHMENT_MAX
Use this parameter to control the maximum size of all the files attached to an acknowledgment. Consider editing this parameter if the size of event attachments are too large and are causing network issues, or alternatively you have excess capacity that may be used and attachments have been discarded. Enter the maximum size in bytes, such as 1000000 for 1 MB. The default value is 10000000 (10 MB). If the Event Assistant is disabled, this parameter is ignored.

KFW_ATTACHMENT_SEGMENT_MAX
This parameter enables you to set a size limitation for individual files attached to an acknowledgment. Enter the maximum size in bytes, such as 250000 for 250 KB. The default value is 1000000 (1 MB). If the Event Assistant is disabled, this parameter is ignored.

The portal server maintains an internal representation of the topology of the monitoring environment, which is called the topology tree. When changes to the topology occur, the portal server updates the topology tree. Examples of topology changes that cause updates to the topology tree include:

- A new agent or monitoring server connects to a monitoring server for the first time.
- Either the hostname or the IP address for managed system changes.
- A managed system is removed or cleared from the monitoring server using either the portal client or the **tacmd cleanms** command.

The process of updating the topology tree can be CPU-intensive for large environments, and the portal server can cause high CPU utilization for a brief period, depending on the size of the environment and the speed of the processor running the portal server. To minimize these processing requirements, the portal server batches topology updates to be processed as a group. The following portal server environment variables control how topology updates are batched together and affect the frequency of update processing for the topology tree.

Parameters affecting topology update processing

KFW_CMW_UPDATE_TOPOLOGY_CUTOFF=*xx*
This variable sets how long the portal server waits after receiving a topology update for new updates to arrive before updating the topology tree. If a new update arrives, the portal server waits for another cutoff interval for more updates to arrive. If no new updates appear, the update processing is started for topology tree. The default cutoff value is 20 seconds. Changing the value to 60 seconds increases the possibility that topology updates are batched together and can reduce the frequency of topology tree updates.

KFW_CMW_UPDATE_TOPOLOGY_MAX_WAIT=*xxx*
This variable sets the maximum time the portal server waits when topology updates are arriving before updating the topology tree. If multiple topology updates arrive and are batched together across several cutoff intervals, this variable sets the maximum time the first topology update waits before starting update processing for the topology tree. The default value is 120 seconds. Changing the value to 300 seconds allows for more batching of topology update requests.

Note: Settings made at the client level take precedence over the settings at the monitoring environment level defined here. If you have specified a setting at the client level, then any event variables defined within the portal server configuration are ignored. If you have not specified a setting at the client level, the portal server variables are used. If you have not specified any settings, the default values are used.

Figure 109 shows an example of how the topology update buffering algorithm would work using the following values:

- KFW_CMW_UPDATE_TOPOLOGY_CUTOFF=120
- KFW_CMW_UPDATE_TOPOLOGY_MAX_WAIT=300

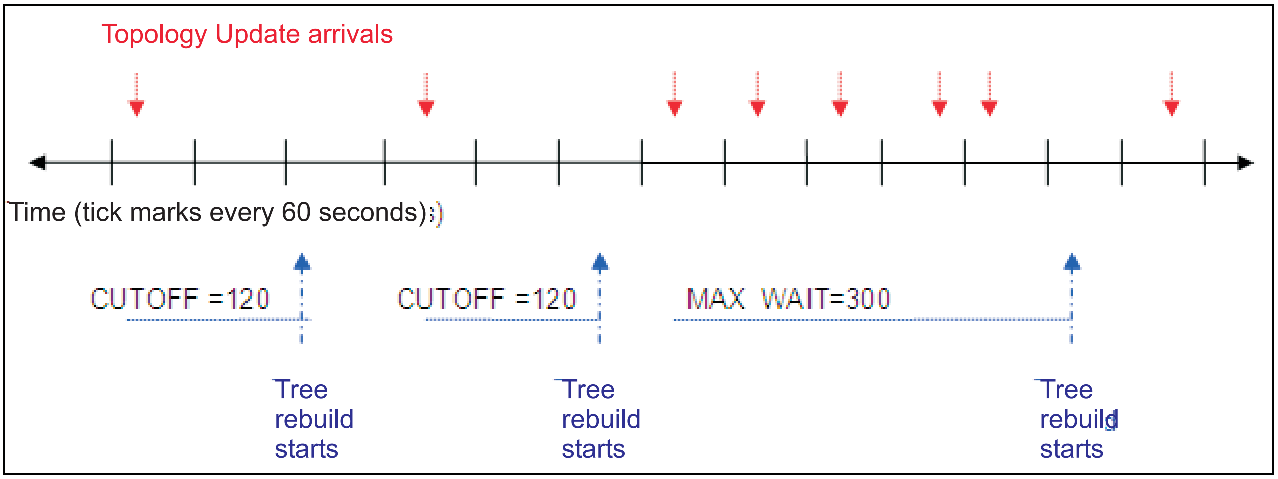


Figure 109. Topology update buffering algorithm example

The red arrows above the time line show when topology updates arrive. The blue arrows below the time line indicate when the tree rebuild would start. For the first two topology updates, there is a period of 120 seconds following their arrival when there were no additional arrivals. For the first two updates, the tree rebuild would occur after the CUTOFF interval. For the third topology update arrival, there are multiple arrivals after it, and no period of 120 seconds when no updates arrived. For these updates, the tree rebuild would start after the MAX_WAIT period of 300 seconds expired.

For machines with multiple network adapters, the recommended practice is to specify the IP address for the agent to use in the agent configuration file:

```
KDEB_INTERFACELIST=preferred_ipaddress
```

If you cannot update the agent definitions in the time required, as a short-term measure there is a Tivoli Enterprise Portal Server environment variable that you can set to prevent topology updates due to IP address changes. Use of this variable is not generally advised since it prevents the updated information from appearing in the Tivoli Enterprise Portal console.

```
KFW_CMW_DETECT_AGENT_ADDR_CHANGE=N
```

If agents are not correctly configured and generate excessive Navigator tree rebuilding, you can set this variable to have any discovery of changes or additions of IP address ignored.

For environments with more than 10 events per minute, consider disabling the Event Assistant

The Event Assistant is an internal component within the Tivoli Enterprise Portal Server that allows to perform the following actions:

- Associate attachments to events, such as logs.
- Assign ownership to events and transfer ownership between users.
- View events specific to the current user.

- View closed events along with any associated information provided by the user.

The Event Assistant creates multiple tables within the Tivoli Enterprise Portal Server database, and processing overhead by the Event Assistant increases the Tivoli Enterprise Portal Server CPU, disk, and memory usage. The Event Assistant is enabled by default. To reduce the processing demands for the portal server, you can disable the Event Assistant by setting the following parameter in the Tivoli Enterprise Portal Server configuration file ((KFWENV on Windows, cq.ini on Unix/Linux):

```
KFW_EVENT_ASSIST=N
```

Disabling the Event Assistant causes the following behavior:

- Event records are not written to the Tivoli Enterprise Portal Server database.
- The ability for any user to attach files to an event is disabled. This is equivalent to all users not having the Attach permission.
- The My Acknowledged Events view is not updated with events that are being acknowledged.
- The Event Notes view is not populated. However, to view notes and previously existing attachments you can still view notes through the Event Notes and Acknowledgement dialogs.
- The *Similar Events by...* view will is not populated.

For environments with more than 10 concurrent Tivoli Enterprise Portal clients, consider using an external web server

If there are more than 10 concurrent Tivoli Enterprise Portal clients connected to the Tivoli Enterprise Portal Server, consider configuring an external web server, for example IBM HTTP Server or Apache, to work with the portal clients. An external web server will offload processing from the Tivoli Enterprise Portal Server KfwServices process, and offer enhanced scalability.

For more information, see “Configuring an external Web server to work with Tivoli Enterprise Portal” on page 400.

Perform periodic table maintenance for Tivoli Enterprise Portal Server DB2 database tables

If you are using DB2 as your Tivoli Enterprise Portal Server database, you should perform periodic table maintenance on the Tivoli Enterprise Portal Server tables, including RUNSTATS. You can do this by turning on automatic table maintenance (AUTO_TBL_MAINT) and automatic runstats (AUTO_RUNSTATS).

Tivoli Enterprise Portal client

The Tivoli Enterprise Portal client issues requests to the Tivoli Enterprise Portal Server and renders the data that is returned. Depending on the choice of installation, the portal client can be started as a desktop application or as an applet embedded in a Web browser. This section outlines considerations to optimize the portal client performance:

- “Tuning the portal client JVM”
- “Portal client parameter tuning” on page 432
- “Avoid displaying a large number of events in the Situation Event Console” on page 433

Tuning the portal client JVM

The memory usage of the portal client JVM increases as the size of the monitoring environment increases. If the maximum Java heap size setting is too low, the JVM spends a significant amount of time performing garbage collection. This can result in poor portal client response time and high CPU utilization.

The default value for the maximum Java heap size differs by client type, as shown in Table 75 on page 431.

Table 75. Default Java heap size parameters by portal client type

Client type	Initial Java heap size (-Xms)	Maximum Java heap size (-Xmx)	Where specified
Browser (Internet Explorer)	4 MB *increase to 128 MB*	64 MB *increase to 256 MB*	IBM Control Panel for Java
Desktop	128 MB	256 MB	• cnp.bat (Windows) • cnp.sh (Linux) • in install_dir/CNP
Java Web Start Desktop client	128 MB	256 MB	tep.jnlp file on portal server

For the desktop client and Java Web Start desktop client, the default maximum Java heap size is 256 MB, which is a good setting for most environments.

For the browser client, the Java plug-in has a default maximum Java heap size of 64 MB on Windows, which is too low. The Tivoli Enterprise Portal browser client uses the IBM Java plug-in, which is automatically installed on your computer with the Tivoli Enterprise Portal. If the Java heap size parameters are not set properly, browser client performance will be slow and your workstation might receive HEAPDUMP and JAVACore files, an out-of-memory condition, when you are logged on.

To receive good performance from the browser client, you must increase the Java heap size parameters from their default values. Before you start the browser client, take the following steps:

1. In the Windows Control Panel, open the Java Control Panel.
2. If you have multiple Java versions, verify that you have the correct control panel open by confirming the Java Runtime is Version 6 and that the JRE is in the IBM path. To verify, click the **Java(TM)** tab and check the **Location** column for the JRE.
3. Set the Java Runtime Parameters:
   a. Click the **Java** tab.
   b. Click the Java Applet Runtime Settings **View** button.
   c. Double-click the **Java Runtime Parameters** field and enter `-Xms128m -Xmx256m -Xverify:none`.
   d. Click **OK**.

   The `-Xms128m` specifies the starting size of the Java heap (128 MB) and `-Xmx256m` specifies the maximum size.
4. Confirm that the Temporary Files settings are set to Unlimited:
   a. Click the **General** tab.
   b. Click **Settings**.
   c. Select **Unlimited** for the **Amount of disk space** to use.
   d. Click **OK**.
5. Clear the browser cache:
   a. In the **General** tab, click **Delete Files**.
   b. In the window that opens, select **Downloaded Applets** and click **OK**.

With either the desktop or browser client, if you observe symptoms of heap memory exhaustion using a maximum Java heap size setting of 256 MB, increase the maximum setting in increments of 64 MB until the symptoms disappear.

Make sure the client workstation has enough memory to handle the maximum heap size. To determine if the client workstation has sufficient memory, observe the available physical memory (as shown on the **Windows Task Manager Performance** tab) when the workstation is not running the Tivoli Enterprise

Portal client, but is running any other applications that need to run concurrently with the portal client. Verify that the client workstation has enough available physical memory to hold the entire maximum Java heap size for the Tivoli Enterprise Portal plus another 150 MB. The additional 150 MB provides an allowance for non-Java heap storage for the Tivoli Enterprise Portal and extra available memory for use by the operating system.

Additional Java heap tuning parameters for IBM Java runtime environments that you can be use to influence garbage collection and memory management are the minimum free percentage (-Xminf) and maximum free percentage (-Xmaxf) parameters. IBM Java documentation provides the following descriptions:

`-Xminf<number>`
A floating point number, 0 through 1, that specifies the minimum free heap size percentage. The heap grows if the free space is below the specified amount. The default is .3 (that is 30%).

`-Xmaxf<number>`
A floating point number between 0 and 1, which specifies the maximum percentage of free space in the heap. The default is 0.6, or 60%. When this value is set to 0, heap contraction is a constant activity. With a value of 1, the heap never contracts.

You can lower the amount of free space maintained in the Java heap at the expense of higher CPU utilization and longer response time by setting the minimum free and maximum free percentages to lower values.

- Default values: -Xminf0.30 -Xmaxf0.60
- Consider the following values: -Xminf0.15 -Xmaxf0.30

The IBM Java documentation warns that setting these values too low can cause poor Java performance. For more information on Java heap tuning parameters, see the *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 Diagnostics Guide*, which is available from http://www.ibm.com/developerworks/java/jdk/diagnosis.

Portal client parameter tuning

This section provides some information about parameters you might consider editing to improve portal client performance and usability. The parameters are set in the following files according to operating system:

Windows
`ITM_HOME``\CNP\cnp.bat` (or `cnp_`*`instance_name`*`.bat`)

For example: `C:\IBM\ITM\CNP\cnp.bat`

Linux *`ITM_HOME`*`/`*`OS_Specific_Directory`*`/cj/bin/cnp.sh` (or `cnp_`*`instance_name`*`.sh`)

For example: `/opt/IBM/ITM/li6243/cj/bin/cnp.sh`

Be aware that on each occasion maintenance or reconfiguration takes place in your environment these files might be recreated and changes lost and need to be reapplied.

cnp.databus.pageSize
Number of rows to fetch in single logical page for any workspace table. By default 100 rows are returned. Although there is no limit to what you can set here, the larger the page size, the more memory required at the client and portal server. To increase manageability you might want to edit this value to return more rows.

cnp.attachment.total.maxsize
Use this parameter to control the maximum size of all the files attached to an acknowledgment. Consider editing this parameter if the size of event attachments are too large and are causing

network issues, or alternatively you have excess capacity that might be used and attachments have been discarded. Enter the maximum size in bytes, such as 1000000 for 1 MB. The default value is 10 MB.

cnp.attachment.segment.maxsize
This parameter enables you to set a size limitation for individual files attached to an acknowledgment. Enter the maximum size in bytes, such as 250000 for 250 KB. The default value is 1 MB.

The following two parameters control the behavior when expanding items in the Navigator view which have a large number of items in the expanded list. The amount of time required to expand Navigator branches depends on how many items are in the expanded list. These parameters allow you to control how many items are expanded at a time.

cnp.navigator.branch.pagesize
The number of items to fetch on a Navigator branch expansion request. The default value is 25.

cnp.navigator.branch.threshold
The warning threshold for Navigator branch expansion requests. The default value is 100.

The following four parameters control the behavior of the Situation Event Console view when it is initially displayed on a workspace, and when it adds events to the display. The default values allow the portal client user to make selections while events are processed in the background.

cnp.siteventcon.initial_batchsize
Maximum number of events that the situation event console will process in the first event batch cycle. The default value is 100.

cnp.siteventcon.initial_dispatchrate
Number of milliseconds that will elapse after the first event batch cycle is processed by the situation event console. The default value is 5000 milliseconds (5 seconds).

cnp.siteventcon.batchsize
Maximum number of events that the situation event console will process in all subsequent event batch cycles. The default value is 100.

cnp.siteventcon.dispatchrate
Number of milliseconds that will elapse after each subsequent event batch cycle is processed by the situation event console. The default value is 1000 milliseconds (1 second).

Note: Settings made here at the client level take precedence over those at the monitoring environment level. If you have specified a setting at the client level, then any event variables defined within the portal server configuration are ignored. If you have not specified a setting at the client level, the portal server variables are used. If you have not specified any settings, the default values are used.

Avoid displaying a large number of events in the Situation Event Console

Tivoli Enterprise Portal performance starts to slow down as the number of events in the Situation Event Console view increases, especially if there are more than 1000 events. Consider the following approaches to reduce the number of events displayed by the Tivoli Enterprise Portal:

- Review the most frequently occurring events to determine if they represent a take action item for the user.
- Consider removing the Situation Event Console view from the default workspaces if you do not use it.
- If events are forwarded and managed using Tivoli Enterprise Console or Omnibus, and if event information is not needed in the Tivoli Enterprise Portal Navigator tree and situation event consoles, you can dissociate situations which eliminates Tivoli Enterprise Portal and Tivoli Enterprise Portal Server processing related to the event.

Tivoli Data Warehouse

The Tivoli Data Warehouse is the most intensely used component of the Tivoli Monitoring V6.2.3 infrastructure. The warehouse must support a large volume of data transactions during every warehousing period, daily summarization and pruning of the warehoused data and multiple queries that often return result sets that are thousands of rows long. The processing power required to support the warehouse database and the Summarization and Pruning Agent can be significant.

This section provides good guidelines for setting your Tivoli Data Warehouse. If you are interested in additional information, see the *Tivoli Management Services Warehouse and Reporting* Redbook.

This section outlines processes to optimize the performance of the warehousing process, including the Warehouse Proxy and the Summarization and Pruning Agents:

- "Historical data collection"
- "Warehouse Proxy Agent" on page 435
- "Summarization and Pruning Agent" on page 439
- "Database tuning" on page 442

Historical data collection

This section contains recommendations related to historical data collection and controlling the volume of data that is generated and loaded into the Tivoli Data Warehouse.

- Do not start collecting historical data for an attribute group without first estimating the data volume that will be generated and the disk space requirements on both the agent and the warehouse database. The Warehouse load projections spreadsheet helps you make these calculations. You can find this spreadsheet by searching for "warehouse load projections" or the navigation code "1TW10TM1Y" in Tivoli Integrated Service Management Library.
- Collect historical data only for attribute groups where there is a planned use for the information. Do not collect data that will not be used.
- The number of rows per day generated by a monitoring agent collecting historical data for an attribute group can be calculated with the following formula:

  (60 / *collection interval*) * 24 * (*# instances at each interval*)

  Where:

  60 Represents the 60 minutes in an hour.

  collection interval The data collection interval, in minutes. This value can be 1, 5, 15, 30, 60, or 1440 (1 day).

  24 Represents 24 hours in one day.

  # instances at each interval The number of instances recorded at each interval.

  The two variables in this formula are the collection interval and the number of instances recorded at each interval.

  – The collection interval is a configuration parameter specified in the Historical Data Collection configuration dialog.

  – The number of instances recorded at each interval depends on the nature of the attribute group and the managed system being monitored. Some attribute groups, such as NT_Memory, generate a single row of data per collection interval. Most attribute groups, however, generate multiple rows of data, one row for each monitoring instance (for example, one per CPU, one per disk, and so on). Certain attribute groups can be *instance-intense*, generating dozens or hundreds of rows of data per collection interval. Examples of instance-intense attribute groups would be those reporting information at the process level, thread level, disk level (for file servers), or network connection level.

  For attribute groups that return multiple rows, the number of instances recorded at each interval is configuration-dependent, and can be different from one monitoring environment to another. The

Warehouse load projections spreadsheet requires the user to specify the number of instances recorded at each interval. There are several approaches that you can use to come up with this number.

  - Using the portal client, build a table view for the monitoring agent and define a query to obtain data from the required attribute group. The number of rows shown in the table view is the number of instances that would be recorded at each interval. For details on how to define table views in the portal client, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.
  - Issue a SOAP call to collect data for this attribute group from the monitoring agent. The number of data rows returned by the SOAP call is the number of instances that would be recorded at each interval. For details on how to issue SOAP calls, see Appendix A “Tivoli Enterprise Monitoring Web services” in the *IBM Tivoli Monitoring: Administrator's Guide*.
  - If you have a test environment, you can create a monitoring server to use in historical data collection testing. Enable historical data collection for the required attribute group under this remote monitoring server, and configure a representative agent to connect to this monitoring server. When the agent uploads data to the Warehouse Proxy Agent, you can query the WAREHOUSELOG table to see how many rows were written by the agent for the attribute group.

  To minimize the data volume (rows per day) generated by a monitoring agent for an attribute group, consider the following two recommendations:

  - Use the longest data collection interval (1, 5, 15, 30, 60 or 1440 minutes) that will provide the required level of information.
  - Avoid or minimize data collection for instance-intense attribute groups, which can generate many rows per data collection interval.

- When configuring historical data collection for an attribute group, you specify which monitoring servers collects the data. Since historical data collection is configured at the monitoring server level, restrict the number of agents collecting data for an attribute group, if possible, by enabling historical data collection on a subset of the monitoring servers.
- To minimize the performance impact on the monitoring server, configure historical data to keep short-term history files at the agent, if possible, rather than at the monitoring server.
- Enable warehouse collection only for attribute groups where there is a planned use for the information. For attribute groups with historical data collection enabled but not configured for warehouse collection, you must schedule regular tasks to prune the short-term history files (the supplied programs are described in the *IBM Tivoli Monitoring: Administrator's Guide*).
- To spread the warehouse collection load across the day, configure warehouse collection to occur hourly rather than daily.

Warehouse Proxy Agent

The Warehouse Proxy Agent provides a means of translating and transferring historical data collected by other monitoring agents to the Tivoli Data Warehouse. The amount of historical data generated can be huge, particularly in environments with thousands of monitoring agents or when instance intense attribute groups are enabled.

This section describes Warehouse Proxy Agent configuration parameters that affect performance and throughput. For most environments, the default settings for these parameters provide good performance. If the Warehouse Proxy Agent starts to encounter errors when exporting data, these parameters may need adjustment. If the Warehouse Proxy Agent is not able to export data into the warehouse, or if it cannot keep up with the volume of export requests, export errors will be encountered at the storage location. The storage location can be the monitoring agent or the monitoring server, whichever is specified in the historical data collection configuration dialog.

Configuration parameters for the Warehouse Proxy Agent are set in the following files according to operating system:

Windows

```
ITM_HOME\TMAITM6\khdenv
```

For example: `C:\IBM\ITM\TMAITM6\khdenv`

Linux and UNIX

ITM_HOME/config/hd.ini

For example: `/opt/IBM/ITM/config/hd.ini`

Be aware that whenever maintenance or reconfiguration takes place in your environment, these files might be recreated and configuration changes can be lost and require reapplication.

Warehouse Proxy internals

To understand the performance-related configuration parameters, it is first helpful to understand how the Warehouse Proxy Agent collects and transfers data to the warehouse. The Warehouse Proxy Agent comprises three internal components; the intelligent remote agent (IRA) communication framework, the work queue, and the exporter threads. These three components work together to collect, translate, and transmit historical data to the warehouse. Historical data flows from one component to the next, undergoing specific processing at each step before being passed on.

For more information about Warehouse Proxy internals, work queues, exporter threads, database connection pool size, and batch inserts, see "Warehouse Proxy internals" in Chapter 5 of the *IBM Tivoli Monitoring: Implementation and Performance Optimization for Large Scale Environments* redbook.

Tuning the Warehouse Proxy Agent on AIX and Linux systems

For a Warehouse Proxy Agent running on AIX or Linux, make sure that the user limit value for number of open file descriptors (the nofiles parameter) is set higher than the number of agents that are uploading data through the Warehouse Proxy Agent. See “File descriptor (maxfiles) limit on UNIX and Linux systems” on page 135 for more details.

Warehouse Proxy Agents running on AIX and Linux systems use a Java virtual machine (JVM), which uses the JDBC (Java Database Connectivity) interface to communicate with the warehouse database. If the maximum Java heap size of the JVM is set to a low value, performance can be degraded by frequent garbage collections.

The maximum Java heap size is controlled by the -Xmx option. By default, this option is not specified in the Warehouse Proxy Agent configuration file. If this option is not specified, the default value used by Java applies as follows:

AIX Half the available memory with a minimum of 16 MB and a maximum of 512 MB.

Linux Half the available memory with a minimum of 16 MB and a maximum of 512 MB.

Note: The above values are from the *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 Diagnostics Guide*.

To set the size of maximum Java heap size for the Warehouse Proxy Agent, edit the `hd.ini` configuration file and modify the KHD_JAVA_ARGS variable as shown below:

```
KHD_JAVA_ARGS=-Xmx256m
```

A maximum Java heap size of 256 megabytes is more than adequate for most environments.

Setting KHD_JNI_VERBOSE=Y in the configuration file will enable logging of the garbage collector’s actions. If the Java log contains an excessive number of garbage collection entries during a single warehousing interval, consider increasing the size of the Java heap.

Using multiple Warehouse Proxy Agents

Tivoli Monitoring supports multiple warehouse proxies within a single hub monitoring server environment. The provision for multiple warehouse proxies provides for greater scalability and performance in historical data collection, and more importantly, improves reliability by providing a failover mechanism. If a Warehouse Proxy is unavailable, data can be inserted into the warehouse by a different Warehouse Proxy

Agent (if the agents are configured properly for failover). “Installing and configuring multiple Warehouse Proxy Agents” on page 608 contains detailed information about configuring multiple Warehouse Proxy Agents.

- If you are collecting and warehousing historical data in a monitoring environment with more than 1500 monitoring agents, consider using multiple Warehouse Proxy Agents to handle the volume of data that will be uploaded and inserted into the warehouse.
- To reduce the number of servers running Tivoli Monitoring components, you can install the Warehouse Proxy Agent on the same servers running the remote monitoring servers. Servers running both a monitoring server and a Warehouse Proxy Agent should have two processors.
- By default, each Warehouse Proxy Agent opens 10 exporter threads and 10 database connections for inserting data into the warehouse database (controlled by the **KHD_EXPORT_THREADS** and **KHD_CNX_POOL_SIZE** parameters, respectively). If you are using more than five Warehouse Proxy Agents, consider reducing the number of exporter threads and database connections for each Warehouse Proxy Agent, to limit the number of database connections inserting data into the warehouse database.

Check ulimit settings for open file descriptors (UNIX/Linux only)

The Warehouse Proxy Agent is similar to the Tivoli Enterprise Monitoring Server in that it can use a large number of file descriptors, especially in a large environment. The same recommendation for ulimit settings for the Tivoli Enterprise Monitoring Server applies to the Warehouse Proxy Agent as well. For more information, see “Check ulimit settings for open file descriptors (UNIX/Linux only)” on page 423.

Disable logging to the WAREHOUSELOG table and use trace to monitor Warehouse Proxy Agent export activity

During normal processing, the WAREHOUSELOG table is appended each time an agent successfully exports a batch of attribute group rows to the warehouse database. The entries in the WAREHOUSELOG table record the agent name, attribute group name, number of rows exported, the Warehouse Proxy Agent involved, and the time of the export. For large environments, the WAREHOUSELOG table can grow very large, and managing the table can add unnecessary overhead for the Summarization and Pruning Agent. A more efficient approach for monitoring Warehouse Proxy Agent export activity is to disable logging to the WAREHOUSELOG table, and instead turn on a trace setting that records similar information to the Warehouse Proxy Agent trace log.

Note: With Tivoli Monitoring V6.2.3 statements from the Warehouse Proxy Agent into the WAREHOUSELOG table are disabled by default. The self-monitoring workspaces provided by the Warehouse Proxy Agent provide sufficient information to determine if the agent is operating correctly.

To disable logging to the WAREHOUSELOG table manually, add the following to the Warehouse Proxy Agent configuration file:

```
KHD_WHLOG_ENABLE=N
```

To turn on tracing for Warehouse Proxy Agent export activity, set the trace options on the KBB_RAS1 entry in the Warehouse Proxy Agent configuration file:

```
KBB_RAS1=ERROR (UNIT:khdxdbex OUTPUT)
```

This trace setting writes an entry to the trace log every time an agent exports a batch of attribute group rows to the warehouse database. The trace entry contains similar information to the information that is written to the WAREHOUSELOG, that is agent name, attribute group name, and number of rows.

You can tail the Warehouse Proxy Agent trace log to monitor Warehouse Proxy Agent export activity in real-time.

Example:

```
# tail -20f prfr1s33_hd_4c582197-02.log
(4C5DC26D.00EA-14:khdxdbex.cpp,3022,"endProcessSample") Inserted 4 rows of data into
```

```
"Linux_VM_Stats" (LNXVM appl KLZ) for "177:prfr2s34:LZ" , status 0
(4C5DC26D.00EB-BD:khdxdbex.cpp,3022,"endProcessSample") Inserted 4 rows of data into
"Linux_VM_Stats" (LNXVM appl KLZ) for "228:prfr2s34:LZ" , status 0
(4C5DC26D.00EC-16:khdxdbex.cpp,3022,"endProcessSample") Inserted 16 rows of data into
"Linux_Network" (LNXNET appl KLZ) for "125:prfr2s34:LZ" , status 0
```

The trace setting described in this section writes export information in the trace log, which does wrap and will not grow indefinitely.

If using DB2 for z/OS for Tivoli Data Warehouse database, increase the idle thread timeout parameter

When using DB2 for z/OS 9.1 for the Tivoli Data Warehouse database, the Warehouse Proxy Agent can encounter repeated disconnections from the database. The default idle thread timeout value (DSN6FAC IDTHTOIN in DSNZPxxx) is 120 seconds. The Warehouse Proxy Agent uses a pool of database connections to process export requests from monitoring agents. The warehousing interval used by agents can be set to values ranging from 15 minutes up to 24 hours. The database connections are idle between export requests, and if the idle thread timeout value is less than the warehousing interval, the database connections might timeout. This results in numerous error messages written to the Warehouse Proxy Agent log. The Warehouse Proxy Agent *Statistics* workspace also shows many disconnections in the *Failure / Disconnections* view.

To avoid repeated disconnections, consider increasing the DB2 idle thread timeout value to a value higher than the warehousing interval. Specifying a value of 0 disables time-out processing. If time-out processing is disabled, idle server threads remain in the system and continue to hold their resources, if any.

Configuration parameter changes used in large-scale environments

Configuration parameters for the Warehouse Proxy Agent are set in the following files according to operating system:

Windows

ITM_HOME\TMAITM6\khdenv

For example: `C:\IBM\ITM\TMAITM6\khdenv`

Linux and UNIX

ITM_HOME/config/hd.ini

For example: `/opt/IBM/ITM/config/hd.ini`

This section contains configuration settings to the Warehouse Proxy Agent configuration files, that you might consider to improve performance in large-scale environments.

KHD_WAREHOUSE_TEMS_LIST=REMOTE_<hostname>

This value is needed when using multiple Warehouse Proxy Agents in a configuration. This value is set so that the Warehouse Proxy Agent services agents that are connected to the local remote monitoring server.

KHD_EXPORT_THREADS=5

With multiple Warehouse Proxy Agents in the configuration, you can lower this value from the default of 10 to reduce the number of concurrent connections to the warehouse database.

KHD_CNX_POOL_SIZE=5

With multiple Warehouse Proxy Agents in the configuration, you can lower this value from the default of 10 to reduce the number of concurrent connections to the warehouse database.

KHD_WHLOG_ENABLE=N

This setting eliminates logging of every agent export to the WAREHOUSELOG table, which reduces load on the warehouse database.

KBB_RAS1=ERROR (UNIT:khdxdbex OUTPUT)

You can add the trace setting *(UNIT:khdxdbex OUTPUT)* to the KBB_RAS1 entry. This setting causes export messages to write to the Warehouse Proxy Agent trace logs, which provides useful information for performance analysis with negligible overhead. This setting is useful if you are specifying KHD_WHLOG_ENABLE=N.

CTIRA_HEARTBEAT=3
Sets the agent heartbeat interval to 3 minutes.

CTIRA_RECONNECT_WAIT=180
Sets the agent reconnect wait time to 180 seconds, or 3 minutes. If the Warehouse Proxy Agent gets disconnected from the hub, this is the amount of wait time until it attempts to reconnect.

DB2CODEPAGE=1208 (DB2 only)
This setting avoids an initialization error message of *DB2 client code page not found*, which appears in the Warehouse Proxy workspace. The value that you use for the DB2CODEPAGE variable must be the same as what you specify on your CREATE DATABASE statement. For more information about setting the DB2CODEPAGE variable, see the IBM DB2 Database information center at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp.

NLS_LANG=AMERICAN_AMERICA.AL32UTF8 (Oracle only)
This setting is used in place of DB2CODEPAGE if using Oracle for the warehouse database.

KHD_JAVA_ARGS=-Xmx256m (UNIX/Linux only)
Sets the maximum Java heap size for the khdxprtj Java virtual machine. This setting is useful in lowering the process size (SZ) and virtual size (VSZ) for the khdxprtj process on the Tivoli Enterprise Monitoring Server.

Summarization and Pruning Agent

The Summarization and Pruning Agent is a multi-threaded, Java based application. It interacts with the warehouse using a JDBC driver appropriate for the database of the warehouse. The number of worker threads available and the heap size of the JVM affect the performance of the Summarization and Pruning Agent and the length of time of its processing runs. The installation location of the Summarization and Pruning Agent is another important aspect of Summarization and Pruning Agent performance tuning.

On Windows, the Summarization and Pruning Agent environment parameters are set in the following configuration files according to operating system:

Windows
`ITM_HOME\TMAITM6\KSYENV`

For example: `C:\IBM\ITM\TMAITM6\KSYENV`

Linux and UNIX
`ITM_HOME/config/sy.ini`

For example: `/opt/IBM/ITM/config/sy.ini`

Whenever maintenance or reconfiguration takes place in your environment, these files might be recreated, and configuration changes can be lost and require reapplication.

Number of worker threads

The Summarization and Pruning Agent creates a pool of worker threads during initialization for performing the summarization and pruning tasks. Each worker thread operates independently of all others, concentrating on one attribute group and its associated warehouse tables. After a worker thread finishes an attribute group, it locates the next attribute group scheduled for processing. If there are no more attribute groups to process, the thread ends, leaving the remaining threads to finish their work.

This threading model provides concurrency while minimizing lock activity, since threads are working on different tables. If there are a few attribute group tables that are considerably larger than the other attribute group tables, the run time of the Summarization and Pruning Agent will be gated by the processing times for the largest attribute group tables.

The number of worker threads can be set in the **Additional Parameters** tab of the configuration window, or by setting the variable KSY_MAX_WORKER_THREADS in the configuration file (`KSYENV` on Windows, `sy.ini` on UNIX or Linux).

- In Tivoli Monitoring V6.2.3, the default value is 2.
- The suggested number of worker threads is 2 or 4 times the number of processors on the host server. If the Summarization and Pruning Agent is running on a separate server from the warehouse database server, set the value based on the number of processors on the warehouse database server.
- Configuring more threads than attribute groups will not decrease the processing time, because each thread works on one attribute group at a time.

Setting the maximum Java heap size

The Summarization and Pruning Agent runs in a Java virtual machine (JVM), which uses the JDBC (Java Database Connectivity) interface to communicate with the warehouse database. If the maximum Java heap size of the JVM is set to a low value, performance can be degraded by frequent garbage collections.

The maximum Java heap size is controlled by the **-Xmx** option. By default, this option is not specified in the Summarization and Pruning Agent configuration file. If this option is not specified, the default value used by Java applies as follows:

- **AIX** Half the available memory with a minimum of 16 MB and a maximum of 512 MB.
- **Linux** Half the available memory with a minimum of 16 MB and a maximum of 512 MB.
- **Windows** Half the real memory with a minimum of 16 MB and a maximum of 2 GB.

Note: The above values are from the *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 Diagnostics Guide*.

To set the size of maximum Java heap size for the Summarization and Pruning Agent, edit the configuration file (`KSYENV` on Windows, `sy.ini` on UNIX or Linux) and modify the KSZ_JAVA_ARGS variable as shown below:

```
KSZ_JAVA_ARGS=-Xmx256m
```

- A maximum Java heap size of 256 megabytes (shown in the example above) is adequate for most environments.
- In addition to the "**mx**" Java parameter, you can also specify the **-verbose:gc** Java runtime parameter, which causes diagnostic messages related to garbage collection to be written to the log. If there are an excessive number of garbage collection entries consider increasing the size of the Java heap.
- For more information about Java heap tuning parameters, see the *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 Diagnostics Guide*, which is available from http://www.ibm.com/developerworks/java/jdk/diagnosis.

Enabling more detailed trace in log files

The Summarization and Pruning Agent Java internal trace files contain diagnostic messages showing the number of rows read and pruned for each attribute group and agent. The names of these files are of the form `hostname_sy_java_timestamp-n.log`. For large environments, these trace files might be overwritten during a single Summarization and Pruning Agent processing cycle, and some of the diagnostic information may be lost.

By default, the Java-based internal trace wraps at 5 files, and each file contains 300000 lines. To change the default values, you can specify Java runtime parameters in the Summarization and Pruning Agent configuration file (`KSYENV` on Windows, `sy.ini` on UNIX or Linux):

```
KSZ_JAVA_ARGS=-Dibm.tdw.maxNumberDetailTraceFiles=A
-Dibm.tdw.maxLinesForDetailTraceFile=B
```

where:

A Specifies the maximum number of Java-based internal trace files that can exist at any one time for a single launch.

B Specifies the maximum number of lines per Java-based internal trace file.

Consider increasing these log parameters so that you have a few days worth of data in the logs for diagnostic purposes.

Consider disabling shifts and vacations

The Summarization and Pruning Agent configuration settings can be specified in the configuration window, which is described in the various Summarization and Pruning Agent installation and configuration steps in Part 5, "Setting up data warehousing," on page 463. The **Work Days** tab in this configuration window allows you to specify shift information and vacation settings.

When you enable and configure shifts, IBM Tivoli Monitoring produces three separate summarization reports:

- Summarization for peak shift hours
- Summarization for off-peak shift hours
- Summarization for all hours (peak and off-peak)

Similarly, when you enable and configure vacations, IBM Tivoli Monitoring produces three separate summarization reports:

- Summarization for vacation days
- Summarization for non-vacation days
- Summarization for all days (vacation and non-vacation)

Enabling shifts and vacations causes increased processing and database space usage by the Summarization and Pruning Agent. If you do not require the separate shift and vacation summarization reports, verify that shift information and vacation settings are not enabled in the Summarization and Pruning Agent configuration window. Alternatively, to disable shifts and vacations, you can specify the following parameter settings in the configuration file (`KSYENV` on Windows, `sy.ini` on UNIX or Linux):

KSY_SHIFTS_ENABLED=N

KSY_VACATIONS_ENABLED=N

For all but the simplest environments, database administrator skills are needed to manage the warehouse database

Database tuning is a complex task, and for most databases it requires the skills of a database administrator. For the Tivoli Data Warehouse, a database located on a single disk using default setting database parameters is only suitable for small test environments. For all other environments, careful planning, monitoring, and tuning are required to achieve satisfactory performance. For information about tuning recommendations for the warehouse database, see "Tivoli Data Warehouse" on page 434.

Before turning on historical collection, estimate the data volume and disk space requirements using the Warehouse Load Projections spreadsheet

Do not start collecting historical data for an attribute group without first estimating the data volume that is generated and the disk space requirements on both the agent and the warehouse database. The Warehouse load projections spreadsheet helps you make these calculations. You can find this spreadsheet by searching for *warehouse load projections* or the navigation code *1TW10TM1Y* in the IBM Tivoli Integrated Service Management Library Web site.

Use the longest collection interval that will provide the required level of information

Values for some attribute groups do not change quickly, such as OS configuration, IP address, disk space usage, and so on. A longer collection interval (1 hour, 1 day) will provide the information with much less storage and processing overhead.

Avoid or minimize history collection for instance-intense attribute groups

Some attribute groups, for example, Process, Thread, and so on, generate many rows per collection interval. To minimize collection for these groups, consider using the granular collection and filtering capabilities in IBM Tivoli Monitoring.

Minimize the number of summarization intervals used

Keep the number of summarization intervals used, such as hourly, daily, weekly, to a minimum since each additional intervals increase the Summarization and Pruning Agent workload significantly. The summarization intervals are processed independently, so to use daily summarization it is not necessary to also use hourly summarization.

Consider disabling shifts and vacations

In the Summarization and Pruning Agent configuration window, the Work Days tab allows you to specify shift information and vacation settings. When you enable and configure shifts, the Summarization and Pruning Agent produces three separate summarization reports:

- Summarization for peak shift hours
- Summarization for off-peak shift hours
- Summarization for all hours (peak and off-peak)

Similarly, when you enable and configure vacations, the Summarization and Pruning Agent produces three separate summarization reports:

- Summarization for vacation days
- Summarization for non-vacation days
- Summarization for all days (vacation and non-vacation)

Enabling shifts and vacations causes increased processing and database space usage. If you do not require the separate shift and vacation summarization reports, verify that shift information and vacation settings are not enabled in the Summarization and Pruning Agent configuration window. Alternatively, to disable shifts and vacations, you can specify the following parameter settings in the configuration file (KSYENV on Windows, sy.ini on UNIX/Linux):

```
KSY_SHIFTS_ENABLED=N
KSY_VACATIONS_ENABLED=N
```

Set the max worker threads and max rows per transaction appropriately for your environment

In the large scale environment, you might want to change the following parameters in the Summarization and Pruning Agent configuration file (sy.ini for UNIX/Linux or ksyenv for Windows):

KSY_MAX_WORKER_THREADS

This parameter specifies the number of worker threads for processing attribute group tables. The default value is 2. For a large database server with multiple processors, increasing this value might improve concurrency and throughput. A good starting value is two times the number of processors on the database server.

KSY_MAX_ROWS_PER_TRANSACTION

This parameter specifies the maximum number of rows per database transaction, and can be thought of as a batching factor. When the number of worker threads is greater than 1, the batching factor for a given worker thread is equal to the maximum rows per transaction divided by the number or worker threads. When you set the number of worker threads, you should also consider increasing the value, to maintain a batching factor of 200 or more.

Database tuning

Database tuning is a complex task, and for important databases, it requires the skills of a database administrator. For the Tivoli Data Warehouse, a database located on a single disk using default database parameters is suitable only for small test environments. For all other environments, careful planning, monitoring, and tuning are required to achieve satisfactory performance.

There are a number of sources of database configuration and tuning information that should be helpful in the planning, monitoring, and tuning of the Tivoli Data Warehouse:

1. "Understanding the disk requirements for your database" on page 473 describes factors to consider in planning the disk subsystem to support the Tivoli Data Warehouse.
2. The paper "Relational Database Design and Performance Tuning for DB2 Database Servers" is available from the Tivoli Integrated Service Management Library by searching for "database tuning" or navigation code "1TW10EC02" at Tivoli Integrated Service Management Library. This paper is a concise reference describing some of the major factors that affect DB2 performance. This document is a good starting point to use before referencing more detailed information in manuals and Redbooks® devoted to DB2 performance. While DB2 specific, many of the concepts are applicable to Relational Databases in general, such as Oracle and Microsoft SQL Server.
3. The "Tivoli Data Warehouse tuning" chapter of the *Redbook Tivoli Management Services Warehouse and Reporting (SG24-7443)* builds upon the previously referenced Tivoli Integrated Service Management Library paper, supplementing it with information about Oracle and Microsoft SQL Server. This chapter is almost 100 pages in length.
4. The "Optimizing the performance" chapter of the *Redbook IBM Tivoli Monitoring Implementation and Performance Optimization for Large Scale Environments (SG24-7443)* contains a section on database tuning considerations for the Tivoli Data Warehouse. This section makes suggestions about specific tuning parameters for DB2, Oracle and MS SQL. At approximately 12 pages, this section is much shorter than the tuning chapter in the previously referenced Redbook (item number 3).

The remainder of this section is a summarized version of material from the Tivoli Integrated Service Management Library paper in number 2 above, which has been supplemented by identifying specific parameters relevant to Tivoli Data Warehouse and providing some suggested ranges of values.

Relational database design and performance tuning for DB2 Database servers

This section explains some of the major factors that affect DB2 performance on distributed platforms such as Windows, UNIX, and Linux. Many of the concepts are applicable to the topic of relational databases such as Oracle and Microsoft SQL Server. This information does not replace detailed information about DB2 performance that is available in various manuals and Redbooks. You can find the Redbooks at http://www.redbooks.ibm.com. You can find the DB2 library of documents at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp.

Terminology

The following terms are useful for understanding performance issues:

Throughput

The amount of data transferred from one place to another or processed in a specified amount of time. Data transfer rates for disk drives and networks are measured in terms of throughput. Typically, throughputs are measured in kilobytes per second, Mbps, and Gbps.

Optimizer

When an SQL statement is run, the SQL compiler must determine the access plan to the database tables. The optimizer creates this access plan, using information about the distribution of data in specific columns of tables and indexes if these columns are used to select rows or join tables. The optimizer uses this information to estimate the costs of alternative access plans for each query. Statistical information about the size of the database tables and available indexes heavily influences the optimizer estimates.

Clustered Index

An index whose sequence of key values closely corresponds to the sequence of rows that are stored in a table. Statistics that the optimizer uses measure the degree of this correspondence.

Cardinality

The number of rows in the table or for indexed columns the number of distinct values of that column in a table.

Prefetch

An operation in which data is read before its use when its use is anticipated. DB2 supports the following mechanisms:

Sequential prefetch

A mechanism that reads consecutive pages into the buffer pool before the application requires the pages.

List prefetch or list sequential prefetch

Prefetches a set of non-consecutive data pages efficiently.

Performance factors

The following performance factors, which are thoroughly detailed in subsequent sections, affect overall performance of any application:

- Database Design
- Application Design
- Hardware Design and Operating System Usage

This section identifies areas where you can influence performance of the Tivoli Data Warehouse database.

Database design details

Key factors for database design include table spaces, buffer pools, and logging. This section includes information about the following topics:

- Files that are created to support and manage your database
- Amount of required space for storing your data
- Determining how you use the table spaces that are required to store your data
- Setting the various database parameters to fit your environment

Table spaces: A *table space* is a physical storage object that provides a level of indirection between a database and the tables stored within the database. It is made up of a collection of containers into which database objects are stored.

A *container* is an allocation of space to a table space. Depending on the table space type, the container can be a directory, device, or file. The data, index, long field, and LOB portions of a table can be stored in the same table space, or can be individually broken out into separate table spaces.

When you are working with database systems, the main objective is your ability to store and retrieve data as quickly and efficiently as possible. One important consideration when designing your database or analyzing a performance problem on an existing database is the physical layout of the database itself.

DB2 provides support for two types of table spaces:

System Managed Space (SMS)

Stores data in operating system files. This type is an excellent choice for general-purpose use, providing good performance with little administration cost.

Database Managed Space (DMS)

Includes database manager control of the storage space. A list of devices or files is selected to belong to a table space when it is defined. The DB2 database manager manages the space on those devices or files. Some additional administration cost is incurred with DMS table spaces because the size of the pre-allocated files is monitored and adjusted. Altering an existing container or adding a new container to it can easily increase the DMS table space size.

Performance and table space types: DMS table spaces usually perform better than SMS table spaces because they are pre-allocated and do not use up time extending files when new rows are added. DMS table spaces can be either raw devices or file system files. DMS table spaces in raw device containers provide the best performance because double buffering does not occur. *Double buffering*, which occurs when data is buffered first at the database manager level and subsequently at the file system level, might be an additional cost for file containers or SMS table spaces.

If you use SMS table spaces, consider using the **db2empfa** command on your database. The **db2empfa** command, which runs the Enable Multipage File Allocation tool, runs multipage file allocation for a database. With multipage file allocation enabled for SMS table spaces, disk space is allocated one extent rather than one page at a time, improving INSERT throughput. In version 8 of DB2, this parameter is turned on by default.

If you are using a RAID device, create the table space with a single container for the entire array. When a database is created in DB2, a default table space called USERSPACE1 is created, and by default, Tivoli Monitoring uses this table space when creating its tables in the DB2 database. You can create a new default table space in DB2 by creating a table space with the name IBMDEFAULTGROUP. If a table space with that name, and a sufficient page size, exists when a table is created without the IN tablespace clause, it is used. You can create the table space in a different location, for example, a RAID array. You could also create it as a DMS table space if you prefer, as in the following example:

```
CREATE REGULAR TABLESPACE IBMDEFAULTGROUP IN DATABASE
PARTITION GROUP IBMDEFAULTGROUP PAGESIZE 4096 MANAGED
BY SYSTEM
  USING ('E:\DB2\NODE0000\SQL00001\IBMDEFAULTGROUP')
  EXTENTSIZE 32
  PREFETCHSIZE AUTOMATIC
  BUFFERPOOL IBMDEFAULTBP
  OVERHEAD 12.670000
  TRANSFERRATE 0.180000
  DROPPED TABLE RECOVERY ON;
```

File system caching on a Windows system: For Windows systems caching, the operating system might cache pages in the file system cache for DMS file containers and all SMS containers. For DMS device container table spaces, the operating system does not cache pages in the file system cache. On Windows, the DB2 registry variable DB2NTNOCACHE specifies whether DB2 opens database files with the NOCACHE option. If DB2NTNOCACHE is set to `ON`, file system caching is eliminated. If DB2NTNOCACHE is set to `OFF`, the operating system caches DB2 files. This standard applies to all data except for files that contain LONG FIELDS or LOBs. Eliminating system caching allows more available memory for the database, and the buffer pool or SORTHEAP can be increased.

Buffer pools: A *buffer pool* is an area of memory into which database pages are read, modified, and held during processing. On any system, accessing memory is faster than disk I/O. DB2 uses database buffer pools to attempt to minimize disk I/O. Although the amount of memory to dedicate to the buffer pool varies, in general it is true that more memory is preferable. Start with 50 - 75% of your system's main memory devoted to buffer pools if the machine is a dedicated database server. Because it is a memory resource, buffer pool usage must be considered along with all other applications and processes that are running on a server. If your table spaces have multiple page sizes, create only one buffer pool for each page size.

Logging: Maintaining the integrity of your data is important. All databases maintain log files that record database changes. DB2 *logging* involves a set of primary and secondary log files that contain log records that show all changes to a database. The database log is used to roll back changes for units of work that are not committed and to recover a database to a consistent state.

DB2 provides the following strategies for logging:

- Circular logging
- Log retention logging

Circular logging: *Circular logging* is the default log mode in which log records fill the log files and later overwrite the initial log records in the initial log file. The overwritten log records are not recoverable. This type of logging is not suited for a production application.

Log retain logging: With log retain logging, each log file is archived when it fills with log records. New log files are made available for log records. Retaining log files enables roll-forward recovery, which reapplies changes to the database based on completed units of work (transactions) that are recorded in the log. You can specify that roll-forward recovery is done to the end of the logs, or to a specific point in time before the end of the logs. Because DB2 never directly deletes archived log files, the application is responsible for maintaining them (for example, archiving, purging, and so on).

Log performance: Ignoring the logging performance of your database can be a costly mistake, especially in terms of time. Optimize the placement of the log files for write and read performance, because the database manager must read the log files during database recovery. To improve logging performance, use the following suggestions:

- Use the fastest disks available for your log files. Use a separate array or channel if possible.
- Use Log Retain logging.
- Mirror your log files.

  Increase the size of the database configuration Log Buffer parameter (LOGBUFSZ). This parameter specifies the amount of the database heap to use as a buffer for log records before writing these records to disk. The log records are written to disk when one of the following points occurs:
  - A transaction commits, or a group of transactions commit, according to the definition in the MINCOMMIT configuration parameter.
  - The log buffer is full.
  - Another internal database manager event occurs.
- Buffering the log records supports more efficient logging file I/O because the log records are written to disk less frequently and more log records are written each time.
- Tune the Log File Size (LOGFILSIZ) database configuration parameter so that you are not creating excessive log files.

Database maintenance: Regular maintenance, which involves running the REORG, RUNSTATS, and REBIND facilities in that order on the database tables, is a critical factor in the performance of a database environment. A regularly scheduled maintenance plan is essential for maintaining peak performance of your system. Implement at least a minimum weekly maintenance schedule.

REORG: After many INSERT, DELETE, and UPDATE changes to table data, often involving variable length columns activity, the logically sequential data might be on non-sequential physical data pages. The database manager must perform additional read operations to access data. You can use the **REORG** command to reorganize DB2 tables, eliminating fragmentation and reclaiming space. Regularly scheduled REORGs improve I/O and significantly reduce elapsed time. Implement a regularly scheduled maintenance plan.

DB2 provides the following types of REORG operation, classic REORG and In-Place REORG. If you have an established database maintenance window, use the classic REORG. If you operate a 24 by 7 operation, use the In-Place REORG.

- Classic REORG
  - The fastest method of REORG
  - Indexes rebuilt during the reorganization
  - Ensures perfectly ordered data
  - Access limited to read-only during the UNLOAD phase, with no access during other phases
  - Not restartable
- In-Place REORG

- – Slower than the Classic REORG and takes more time to complete
- – Does not ensure perfectly ordered data or indexes
- – Requires more log space
- – Can be paused and restart
- – Can allow applications to access the database during reorganization

RUNSTATS: The DB2 optimizer uses information and statistics in the DB2 catalog to determine optimal access to the database based on the provided query. Statistical information is collected for specific tables and indexes in the local database when you run the RUNSTATS utility. When significant numbers of table rows are added or removed, or if data in columns for which you collect statistics is updated, execute **RUNSTATS** again to update the statistics.

Use the RUNSTATS utility to collect statistics in the following situations:

- When data was loaded into a table and the appropriate indexes were created
- When you create a new index on a table. Execute **RUNSTATS** for the new index only if the table was not modified since you last ran **RUNSTATS** on it.
- When a table has been reorganized with the REORG utility
- When the table and its indexes have been extensively updated by data modifications, deletions, and insertions. “Extensive” in this case might mean that 10 to 20 percent of the table and index data was affected.
- Before binding or rebinding, application programs whose performance is critical
- When you want to compare current and previous statistics. If you update statistics at regular intervals, you can discover performance problems early.
- When the prefetch quantity is changed
- When you have used the REDISTRIBUTE DATABASE PARTITION GROUP utility

The **RUNSTATS** command has several formats that primarily determine the depth and breadth or statistics that are collected. If you collect more statistics, the command takes more time to run. The following options are included:

- Collecting either SAMPLED or DETAILED index statistics
- Collecting statistics on all columns or only columns used in JOIN operations
- Collecting distribution statistics on all, key, or no columns. Distribution statistics are useful when you have an uneven distribution of data on key columns.

Take care when running **RUNSTATS**, because the collected information impacts the selection of access paths by the optimizer. Implement **RUNSTATS** as part of a regularly scheduled maintenance plan if some of the conditions occur. To ensure that the index statistics are synchronized with the table, execute **RUNSTATS** to collect table and index statistics at the same time.

Consider some of the following factors when deciding what type of statistics to collect:

- Collect statistics only for the columns that join tables or in the WHERE, GROUP BY, and similar clauses of queries. If these columns are indexed, you can specify the columns with the ONLY ON KEY COLUMNS clause for the **RUNSTATS** command.
- Customize the values for num_freqvalues and num_quantiles for specific tables and specific columns in tables.
- Collect detailed index statistics with the SAMPLE DETAILED clause to reduce the amount of background calculation performed for detailed index statistics. The SAMPLE DETAILED clause reduces the time required to collect statistics and produces adequate precision in most cases.
- When you create an index for a populated table, add the COLLECT STATISTICS clause to create statistics as the index is created.

REBIND: After running **RUNSTATS** on your database tables, you must rebind your applications to take advantage of those new statistics. Rebinding ensures that DB2 is using the best access plan for your SQL statements. Perform a **REBIND** after running **RUNSTATS** as part of you normal database maintenance procedures. The type of SQL that you are running determines how the rebind occurs.

DB2 provides support for the following types of SQL:

- Dynamic SQL
- Static SQL

Dynamic SQL: *Dynamic SQL* statements are prepared and executed at run time. In dynamic SQL, the SQL statement is contained as a character string in a host variable and is not precompiled. Dynamic SQL statements and packages can be stored in one of the DB2 caches. A rebind occurs under the following conditions when you are using dynamic SQL:

- If the statement is not in the cache, the SQL optimizer "binds" the statement and generates a new access plan.
- If the statement is in the cache, no rebind occurs. To clear the contents of the SQL cache, use the FLUSH PACKAGE CACHE SQL statement.

Static SQL: *Static SQL* statements are embedded within a program, and are prepared during the program preparation process before the program is executed. After preparation, a static SQL statement does not change, although values of host variables that the statement specifies can change. These static statements are stored in a DB2 object called a *package*.

A rebind occurs under the following conditions when you are using static SQL:

- Explicitly, if an explicit REBIND package occurs
- Implicitly, if the package is marked "invalid", which can happen if an index that the package was using was dropped.

Application design details

The Tivoli Monitoring application was designed and tested with performance and scalability in mind. Consider reuse of the database connection using the Tivoli Monitoring connection pooling feature. *Connection pooling* is a process in which DB2 drops the inbound connection with an application that requests disconnection, but keeps the outbound connection to the host in a pool. When a new application requests a connection, DB2 uses one from the existing pool. Using the existing connection reduces the total amount of connection time and the high processor connection cost on the host.

For the data warehouse, connection pooling is implemented using the Tivoli Monitoring environment variable KHD_CNX_POOL_SIZE. The default value is 10. You can use the volume of work that the database is processing to decide whether to increase or decrease this value. This parameter is applicable to other database managers such as SQL Server and Oracle.

For information about other environment variables, see "Warehouse Proxy Agent" on page 41 and "Warehouse Summarization and Pruning Agent" on page 41.

Hardware design and operating system usage

For any database system, a number of common areas must be addressed and sized appropriately to support your application workload. This section covers common and platform-specific hardware and operating system components and explains main considerations for hardware design and operating system usage. It does not include detailed calculations for capacity planning purposes.

Key factors for hardware design and operating system usage include memory, CPU, I/O and network considerations.

Memory: Understanding how DB2 organizes memory helps you tune memory use for good performance. Many configuration parameters affect memory usage. Some parameters might affect memory on the server, some on the client, and some on the server and the client. Furthermore, memory is allocated and de-allocated at different times and from different areas of the system.

While the database server is running, you can increase or decrease the size of memory areas inside the database shared memory. Understand how memory is divided among the different heaps before you tune to balance overall memory usage on the entire system. See the *DB2 Administration Guide: Performance* for a detailed explanation of the DB2 memory model and all of the parameters that affect memory usage.

CPU: CPU utilization should be about 70 to 80% of the total CPU time. Lower utilization means that the CPU can cope better with peak workloads. Workloads between 85% to 90% result in queuing delays for CPU resources, which affect response times. CPU utilization above 90% usually causes unacceptable response times. While running batch jobs, backups, or loading large amounts of data, the CPU might be driven to high percentages such as to 80 to 100% to maximize throughput.

DB2 supports the following processor configurations:

Uni-Processor
A single system that contains only one single CPU

SMP (symmetric multiprocessor)
A single system that can contain multiple CPUs. Scalability is limited to the CPU sockets on the motherboard.

MPP (massively parallel processors)
A system with multiple nodes connected over a high speed link. Each node has its own CPUs. Adding new nodes achieves scalability.

Notes:

1. Inefficient data access methods cause high CPU utilization and are major problems for database system. Regular database maintenance is an important factor.
2. Paging and swapping require CPU time. Consider this factor while planning your memory requirements.

I/O: Improving I/O can include making accurate calculations for total disk space that an application requires, improving disk efficiency, and providing for parallel I/O operations.

Calculating disk space for an application: Use the following guidelines to calculate total disk space that an application requires:

- Calculate the raw data size. Add the column lengths of your database tables and multiply by the number of expected rows.
- After calculating the raw data size, use the following scaling up ratios to include space for indexing, working space, and so on:
  - OLTP ratio: 1:3
  - DSS ratio: 1:4
  - Data warehouse ratio: 1:5

Disk efficiency: You can improve disk efficiency with attention to the following concerns:

- Minimize I/O. Access to main memory is much faster than accessing the disk. Provide as much memory as possible to the database buffer pools and various memory heaps to avoid I/O.
- When I/O is needed, reading simultaneously from several disks is the fastest method. You can use several smaller disks rather than one large disk, or place the disk drives on separate controllers.

Selecting disk drives: Disks tend to grow larger every year, doubling in capacity every 18 months, and the cost per GB is lower each year. The cost difference of the two smallest drives diminishes until the smaller drive is not practical. The disk drives improve a little each year in seek time. The disk drives get smaller in physical size. While the disk drives continue to increase capacity with a smaller physical size, the speed improvements, seek, and so on, are small in comparison. A database that would have taken 36 * 1 GB drives a number of years ago can now be placed on one disk.

This growth trend highlights database I/O problems. For example, if each 1 GB disk drive can do 80 I/O operations a second, the system can process a combined 2880 I/O operations per second (36 multiplied by 80). But a single 36-GB drive with a seek time of 7 milliseconds can process only 140 I/O operations per second. Although increasing disk drive capacity is beneficial, fewer disks cannot deliver the same I/O throughput.

Parallel operations: Provide for parallel I/O operations. Use the smallest disk drives possible to increase the number of disks for I/O throughput. If you purchase larger drives, use only half the space, especially the middle area, for the database. The other half is useful for backups, archiving data, off hour test databases, and extra space for accommodating upgrades.

Network: After a system is implemented, the network should be monitored to ensure that its bandwidth is not being consumed more than 50%. The network can influence overall performance of your application, especially if a delay occurs in the following situations:

- Lengthy time between the point a client machine sends a request to the server and the server receives this request
- Lengthy time between the point the server machine sends data back to the client machine and the client machine receives the data

Tuning

This section explains relevant database and database manager configuration parameters and includes guidelines for setting their values.

Database manager configuration tuning: Each instance of the database manager has a set of database manager configuration parameters, also called *database manager parameters*, which affect the amount of system resources that are allocated to a single instance of the database manager. Some of these parameters are used for configuring the setup of the database manager and other information that is not related to performance. You can use either the DB2 Control Center or the following command to change the parameters:

```
UPDATE DATABASE MANAGER CONFIG USING keyword
```

Parameters: The following database manager configuration parameters have a high impact on performance:

Note: See the *DB2 Administration Guide: Performance* for detailed explanations about all the database manager configuration parameters.

AGENTPRI
Controls the priority that the operating system scheduler gives to all agents and to other database manager instance processes and threads. Use the default value unless you run a benchmark to determine the optimal value.

ASLHEAPSZ
Represents a communication buffer between the local application and its associated agent. The application support layer heap buffer is allocated as shared memory by each database manager agent that is started. If the request to the database manager or its associated reply do not fit into the buffer, they are split into two or more send-and-receive pairs. The size of this buffer should be set to handle the majority of requests using a single send-and-receive pair.

INTRA_PARALLEL
Specifies whether the database manager can use intra-partition parallelism on an SMP machine. Multiple processors can be used to scan and sort data for index creation. Usually, this parameter is set to `YES`, especially if you are running on a dedicated database server. The default value is `NO`.

MAX_QUERYDEGREE

Specifies the maximum degree of intra-partition parallelism that is used for any SQL statement that is executing on this instance of the database manager. An SQL statement does not use more than this number of parallel operations within a partition when the statement is executed. The intra_parallel configuration parameter must be set to a value greater than 1 to enable the database partition to use intra-partition parallelism. Setting the value to `ANY` enables use of all partitions. Usually, this parameter should be set to `ANY`, especially if you are running on a dedicated database server. The default value is `ANY`.

SHEAPTHRES

Determines the maximum amount of memory available for all the operations that use the sort heap, including sorts, hash joins, dynamic bitmaps that are used for index ANDing and Star Joins, and operations where the table is in memory. Set the sort heap threshold parameter to a reasonable multiple of the largest SORTHEAP parameter in your database manager instance. This parameter should be at least twice as large as the largest sort heap defined for any database within the instance, but you must also consider the number of concurrent sort processes that can run against your database.

Database configuration tuning: Each database has a set of the database configuration parameters, which are also known as *database parameters*. These parameters affect the amount of system resources that are allocated to that database. Furthermore, some database configuration parameters provide descriptive information only and cannot be changed, and others are flags that indicate the status of the database. You can use the DB2 Control Center or the UPDATE DATABASE CONFIG FOR *dbname* USING *keyword* command to change those parameters. For more information about the numerous database configuration parameters, see the *DB2 Administration Guide: Performance*.

The following data configuration parameters have a high impact on performance:

DBHEAP

Contains control block information for tables, indexes, table spaces, and buffer pools, and space for the log buffer (LOGBUFSZ) and temporary memory that the utilities use. Each database has only one database heap, and the database manager uses it on behalf of all applications connected to the database. The size of the heap is dependent on a large number of variables. The control block information is kept in the heap until all applications disconnect from the database. The DB2 default value is typically too low, particularly for the Tivoli Data Warehouse. Start with a value between 2000 and 8000.

DFT_DEGREE

Specifies the default value for the CURRENT DEGREE special register and the DEGREE bind option. The default value is 1, which means no intra-partition parallelism. A value of -1 means that the optimizer determines the degree of intra-partition parallelism based on the number of processors and the type of query. The degree of intra-partition parallelism for an SQL statement is specified at statement compilation time using the CURRENT DEGREE special register or the DEGREE bind option. The maximum runtime degree of intra-partition parallelism for an active application is specified using the **SET RUNTIME DEGREE** command. The Maximum Query Degree of Parallelism (max_querydegree) configuration parameter specifies the maximum query degree of intra-partition parallelism for all SQL queries. The actual runtime degree that is used is the lowest of one of the following:

- The max_querydegree configuration parameter
- Application runtime degree
- SQL statement compilation degree

For a multi-processor machine, set this to `−1` (ANY), to allow intra-partition parallelism for this database.

CHNGPGS_THRESH

Improves overall performance of the database applications. Asynchronous page cleaners write changed pages from the buffer pool or the buffer pools to disk before a database agent requires the space in the buffer pool. As a result, database agents should not have to wait for changed pages to be written out so that they might use the space in the buffer pool. Usually, you can start out with the default value.

LOCKLIST

Indicates the amount of storage that is allocated to the lock list. This parameter has a high impact on performance if frequent lock escalations occur. The DB2 default value is typically too low, particularly for the Tivoli Data Warehouse. Start with a value of between 500 and 800.

MAXLOCKS

Maximum percent of lock list before escalation. Use this parameter with the LOCKLIST parameter to control lock escalations. Increasing the LOCKLIST parameter augments the number of available locks.

LOGBUFSZ

Specifies the amount of the database heap (defined by the dbheap parameter) to use as a buffer for log records before writing these records to disk. Buffering the log records supports more efficient logging file I/O because the log records are written to disk less frequently, and more log records are written at each time. The DB2 default value is typically too low, particularly for the Tivoli Data Warehouse. Start with a value of between 256 and 768.

NUM_IOCLEANERS

Specifies the number of asynchronous page cleaners for a database. These page cleaners write changed pages from the buffer pool to disk before a database agent requires the space in the buffer pool. As a result, database agents do not wait for changed pages to be written out so that they can use the space in the buffer pool. This parameter improves overall performance of the database applications. The DB2 default value is typically too low. Set this parameter equal to the number of physical disk drive devices that you have. The default value is 1.

NUM_IOSERVERS

Specifies the number of I/O servers for a database. I/O servers perform prefetch I/O and asynchronous I/O by utilities such as backup and restore on behalf of the database agents. Specify a value that is one or two more than the number of physical devices on which the database is located. The DB2 default value is typically too low. Set this parameter equal to the number of physical disk drive devices that you have, and add two to that number. The default value is 3.

PCKCACHESZ

The package cache is used for caching sections for static and dynamic SQL statements on a database. Caching packages and statements eliminates the requirement to access system catalogs when reloading a package so that the database manager can reduce its internal overhead. If you are using dynamic SQL, caching removes the need for compilation.

SORTHEAP

Defines the maximum number of private memory pages to be used for private sorts, or the maximum number of shared memory pages to be used for shared sorts. Each sort has a separate sort heap that is allocated as needed by the database manager. This sort heap is the area where data is sorted. Increase the size of this parameter when frequent large sorts are required. The DB2 default value might be too low, particularly for the Tivoli Data Warehouse. Start with a value of 256 - 1024. When changing this parameter, you might want to change the SHEAPTHRES database manager parameter too.

LOGFILSIZ

Defines the size of each primary and secondary log file. The size of these log files limits the number of log records that can be written to them before they become full, which requires a new log file. The DB2 default value is too low, particularly for the Tivoli Data Warehouse. Start with a value of 4096 - 8192.

LOGPRIMARY

Specifies the number of primary log files to be pre-allocated. The primary log files establish a fixed amount of storage that is allocated to the recovery log files. The DB2 default value might be too low, particularly for the Tivoli Data Warehouse. Start with a value of 6 - 10.

Buffer pools: The buffer pool is the area of memory where database pages, table rows or indexes, are temporarily read and manipulated. All buffer pools are located in global memory, which is available to all applications using the database. The purpose of the buffer pool is to improve database performance. Data can be accessed much faster from memory than from disk. Therefore, as the database manager is able to read from or write to more data (rows and indexes) to memory, database performance improves.

The default buffer pool allocation is usually not sufficient for production applications, and must be monitored and tuned before placing your application in production. The DB2 default value is typically too low, particularly for the Tivoli Data Warehouse. Start with a value of 50 -75 percent of your system memory if the database server is dedicated. You can use the SQL statement ALTER BUFFERPOOL to change this value.

Registry variables: Each instance of the database manager has a set of registry and environment variables that affect various aspects of DB2 processing. You can change the value of DB2 registry variables using the **DB2SET** command. Although numerous other registry and environment variables exist, the DB2_PARALLEL_IO variable has a high impact on performance.

Note: See the *DB2 Administration Guide: Performance* for a detailed explanation of all the registry and environment variables.

While reading or writing data from and to table space containers, DB2 may use parallel I/O for each table space value that you specify. The prefetch size and extent size for the containers in the table space determine the degree of parallelism. For example, if the prefetch size is four times the extent size, there are four extent-sized prefetch requests. The number of containers in the table space does not affect the number of prefetchers.

To enable parallel I/O for all table spaces, you can specify the asterisk (*) wildcard character. To enable parallel I/O for a subset of all table spaces, enter the list of table spaces. For several containers, extent-size pieces of any full prefetch request are broken down into smaller requests that are executed in parallel based on the number of prefetchers. When this variable is not enabled, the number of containers in the table space determines the number of prefetcher requests that are created.

Monitoring tools

DB2 provides the following tools that can be used for monitoring or analyzing your database:

- Snapshot Monitor, which captures performance information at periodic points of time and is used to determine the current state of the database
- Event Monitor, which provides a summary of activity at the completion of events such as statement execution, transaction completion, or when an application disconnects
- Explain Facility, which provides information about how DB2 accesses the data to resolve the SQL statements
- db2batch tool, which provides performance information as a benchmarking tool

SNAPSHOT and EVENT monitors: DB2 maintains data as the database manager runs about its operation, its performance, and the applications that are using it. This data can provide important performance and troubleshooting information. For example, you can track the following developments:

- The number of applications connected to a database, their status, and which SQL statements each application is executing, if any
- Information that shows how well the database manager and database are configured, helping you make tuning decisions
- Information about the time that deadlocks occurred for a specified database, the applications that were involved, and the locks that were in contention

- The list of locks held by an application or a database. If the application cannot proceed because it is waiting for a lock, additional information is on the lock, including which application is holding it.

Collecting performance data introduces some overhead on the operation of the database. DB2 provides monitor switches to control which information is collected. You can use the following DB2 commands to turn on these switches:

```
UPDATE MONITOR SWITCHES USING BUFFERPOOL  ON ;
UPDATE MONITOR SWITCHES USING LOCK        ON ;
UPDATE MONITOR SWITCHES USING SORT        ON ;
UPDATE MONITOR SWITCHES USING STATEMENT   ON ;
UPDATE MONITOR SWITCHES USING TABLE       ON ;
UPDATE MONITOR SWITCHES USING UOW         ON ;
```

You can access the data that the database manager maintains either by taking a snapshot or by using an event monitor.

SNAPSHOTs: Use the **GET SNAPSHOT** command to collect status information and format the output for your use. The returned information represents a snapshot of the database manager operational status at the time the command was issued. Various formats of this command obtain different kinds of information, and the specific syntax can be obtained from the DB2 Command Reference. The following formats are useful:

GET SNAPSHOT FOR DATABASE
Provides general statistics for one or more active databases on the current database partition.

GET SNAPSHOT FOR APPLICATIONS
Provides information about one or more active applications that are connected to a database on the current database partition.

GET SNAPSHOT FOR DATABASE MANAGER
Provides statistics for the active database manager instance.

GET SNAPSHOT FOR LOCKS
Provides information about every lock held by one or more applications connected to a specified database.

GET SNAPSHOT FOR BUFFERPOOLS
Provides information about buffer pool activity for the specified database.

GET SNAPSHOT FOR DYNAMIC SQL
Returns a point-in-time picture of the contents of the SQL statement cache for the database.

You can create some simple scripts and schedule them to get periodic snapshots during your test cycles.

DB2BATCH: A benchmark tool called DB2BATCH is provided in the sqllib/bin subdirectory of your DB2 installation. This tool can read SQL statements from either a flat file or standard input, dynamically describe and prepare the statements, and return an answer set. You can specify the level of performance information that is supplied, including the elapsed time, CPU and buffer pool usage, locking, and other statistics collected from the database monitor. If you are timing a set of SQL statements, DB2BATCH also summarizes the performance results and provides both arithmetic and geometric means. For syntax and options, type `db2batch`.

Optimizing queries

This section contains information about tuning the queries that are processed to display the tables, charts, and graphs that make up workspace views within the Tivoli Enterprise Portal.

Processing queries

The query assigned to a chart or table view requests data from a particular attribute group. It executes when you open or refresh the workspace. Queries make up the processing load for on-demand data collection. You can reduce the frequency and amount of data sampling by:

- Customizing the query to filter out unwanted data. This reduces the number of selection criteria (rows) and attributes (columns) collected.
- Applying the same query to other views in the workspace. This reduces the number of data samples required: one query uses a single sample for multiple views.
- Disabling automatic refresh of workspace views or adjust the refresh rate to longer intervals. This causes Tivoli Enterprise Monitoring Agent data to be collected less frequently.
- Consider how you want to display the data returned by a query. A graphical view workspace might require less data compared to a table view, because it uses only the truly numeric data and leaves out the character data.

Do not confuse custom queries with view filters from the **Filters** tab of the Query Properties editor. View filters fine-tune the data after it has been retrieved by the query and do not reduce network traffic, data collection processing, or memory demands.

The following general recommendations and observations might be considered as well:

- Some attributes are more expensive to retrieve than others. One *expensive* column in a table makes any workspace view or situation that references that table more expensive. An example of an expensive attribute is one that must run long storage chains to determine its value, such as using a process table to look for looping tasks. Where possible, ensure that you only retrieve attributes that are required for the monitoring process.
- Column function (such as MIN, MAX, AVG, and so on) requires post processing of the query results set after data is returned to Tivoli Enterprise Monitoring Server.
- Use more efficient data manipulating functions, such as substring instead of string scan. If you know the position of the string to search, do not scan the whole string to check for the value.

Post-filters versus pre-filters

Performance is improved for each view if you pre-filter the required view information and send to the portal client only what is needed in that view. This reduces the amount of data sent through the network and you do not have to post-process it either. However, there is one exception to the rule.

Think of a workspace that contains many views. Each of those views has a query associated with it, which is issued when the workspace is accessed. This might result in many queries that must be processed in parallel.

A better way of doing this might be to create one query that returns all data that is required in the workspace. In this case, the query will only be issued once and the data can then be post-filtered for each view to only display information as it applies to each view.

One important consideration however is that queries are saved globally and are not user ID dependent. This means that only administrators will be able to modify queries in most installations. For the end user to be able to modify filters, the preferred method might therefore be the filters applied in the view properties **Filters** tab.

Defining custom queries

Custom queries reduce network traffic, processing at the agent and Tivoli Enterprise Monitoring Server, and memory usage at the Tivoli Enterprise Portal Server and Tivoli Enterprise Portal client. Custom queries accomplish this by limiting the number of rows and columns passed from the Tivoli Enterprise Monitoring Agent to the Tivoli Enterprise Monitoring Server.

Most of the predefined, predefined queries request all columns and all rows of an attribute group, of which only a few might be of interest to you. Removing the unwanted columns (attributes) reduces the amount of data transferred from monitoring agent to Tivoli Enterprise Monitoring client via the portal server, hub monitoring server, and remote monitoring server. Additionally, in the case of OMEGAMON Monitoring Agents that are located on z/OS, you also reduce the amount of EBCDIC/ASCII conversion required when data is passed between mainframe and distributed platforms.

It is recommended to tune any queries servicing workspaces that are frequently executed or return large quantities of data. Query execution always requires resources and intermittent large reports causes a spike in memory requirements and network resources.

Restricting the number of rows

Most predefined queries return all rows and columns. You can create a custom query to filter out the irrelevant or uninteresting metrics. Not only does this make it easier to read the report, but it saves Tivoli Enterprise Portal Server memory, client memory, and CPU because there are fewer rows to translate, sort, and transmit.

You might be using view filters inappropriately. View filters work only on the current page returned from the query. For example, if page size is 100 lines and the filter reduces it to five lines on page 1 and similarly on subsequent pages, the row set cannot be seen on one page. Do not increase the workspace page size to see everything on one page. Increased page size actually increases portal client memory requirements.

Instead, avoid this condition by creating a custom query that filters the data at query execution time.

Restricting the number of columns

Most predefined queries return all columns (or attributes). A particular attribute group might contain 50 columns, yet all you need is five. Creating a custom query to retrieve only the required five attributes reduces portal server and client CPU and memory.

Use the same query in a workspace

If you have multiple views in a workspace requiring data from different attribute groups, you will need a different query for each group. However, if the views have data from the same attribute group, then use a single query to accommodate both, even if the attributes (columns) that each view requires are different. Two unique queries will each drive data collection at the agent and increase resource overhead. For each workspace, have one query that can be shared by all views using that attribute group. Remember that the entire results set for each query is stored on the portal server, this technique avoids duplicate result sets being stored.

Collect agent data less frequently

A good practice is to avoid the use of workspace automatic refresh when possible. The Navigator view and event views (message log, event console, and graphic console) refreshes automatically. This provides you with instantaneous alerts, and you can navigate to their event workspaces with actual data. The graphic view, the graphical equivalent of the Navigator, which shows alerts but no data, affects client memory but not the portal server.

Reduce the distribution of queries

A query can be assigned to a particular managed system or to a group of systems by defining a managed system group. The default group usually includes all known instances for that application or operating system.

It might be that you want this query to be applied to only certain managed systems and so distribution to all managed systems is an unnecessary waste of resources. Modify MSLs to reduce the distribution of the query. Also remove the MSLs for queries that are of no interest to the user. Even if you are not viewing the results of the query, there might be a use of system resources that can be avoided by restricting the distribution of the unneeded queries.

Optimizing situations

Situations are conditions that you want to monitor on managed systems. Each situation contains a name, a predicate formula, special attributes for specific situation processing, information about whether the situation is automatically started or not, and the sampling interval. It can also contain a command to execute when the situation is true, and advice to give the client when an alert for the situation is surfaced, and so on.

A situation predicate is a formula that uses attributes, functions, and other situations as operands along with threshold values to describe a filtering criterion. The situation formula is made up of one or more logical expressions.

Each expression takes the form:

[Attribute name] / [logical operator] / [value]

For example: `PROCESS_ID == 0`

The situation predicates are similar to a WHERE clause in SQL. In IBM Tivoli Monitoring, predicates are processed sequentially, and perform better if you put the most restrictive condition as the first predicate. You can use the situation editor to create/edit situations by choosing attributes, logical operator, value, and sampling interval, and so on. Situations are assigned to run on behalf of managed systems or lists of managed systems.

When a situation is running on a monitoring agent, the agent collects the current values of the attributes specified in the formula, and tests the values against a threshold. When the condition is met, that is the threshold is exceeded or a value is matched, the agent passes the collected data back to the monitoring server to which it is connected, and an event is generated.

However, some types of situations cannot be tested at the agent level:

- Situations involving group functions, such as MIN, MAX, AVG, SUM, COUNT
- Embedded situations
- Correlated situations

For these types of situations, the agent returns all rows back to the monitoring server to which it is connected, and the server performs the testing. In large-scale environments, especially if the sampling interval for the situation is short, evaluation for such situations dramatically increases the workload and memory usage for the monitoring server.

In general, there is limited situation processing at the hub monitoring server if there are no agents directly connected to the hub. For remote monitoring servers, there is a direct correlation to the number of agents, situations, and data rows to the amount of memory required. Therefore, the number of situations and size of the data might be the limiting factor that determines how many agents an individual remote monitoring server can support.

Since the performance of the monitoring server is greatly affected by the volume and frequency of situation state changes, do not run a situation and collect data unless you are interested in the results.

The following recommendations provide guidance for how to write more efficient situations and how to reduce the situation processing requirements:

1. If possible, avoid use of group functions, embedded situations, or correlated situations. Processing demands for such situations are much higher, since all attribute group rows must be sent to the monitoring server for testing, increasing the processing demand and memory usage on the monitoring server.
2. Put the most stringent condition at the beginning of the formula because the conditions are evaluated sequentially, left to right.

Consider a situation that has the first condition test on real storage use, the result set may contain multiple rows; then the second condition tests whether a given process name is among the returned rows. It would be more efficient to first test on process name (the result will be one row), followed by the test on the storage usage, just on the single row.

Consider the following in creating the condition tests:

a. Numeric attributes are processed more quickly than text attributes.

b. String checking with substring (STR) is more efficient than the string scan (SCAN), especially for long strings. If you know the exact location of the text or characters to be evaluated, use a substring.

3. Use longer sampling intervals where possible, especially for situations that are distributed to many agents.
4. Minimize the number of situations on attribute groups that can return many rows, such as process or disk attribute groups.
5. Put mission critical systems in a separated managed system group, and distribute the heavy workload situations to them only if necessary.
6. Consider spreading business-critical situations among several different remote Tivoli Enterprise Monitoring Servers. For example, a database monitoring situation might be high load, but business-critical. Instead of having all 500 database agents report through the same remote monitoring server, consider configuring the database agents across five remote monitoring servers, which are supporting other less-demanding agents.

Planning for platform-specific scenarios

This section describes several platform-specific scenarios.

Disabling TCP-delayed acknowledgments on AIX systems

On AIX systems, the default behavior for TCP connections results in delayed acknowledgments (Ack packets). When tcp_nodelayack is set to 0 (the default setting), TCP delays sending Ack packets by up to 200ms, the Ack attaches to a response, and system overhead is minimized.

Setting the tcp_nodelayack parameter to 1 causes TCP to send immediate acknowledgment (Ack) packets to the sender.

Setting tcp_nodelayack to 1 will cause slightly more system overhead, but can result in much higher performance for network transfers if the sender is waiting on the acknowledgment from the receiver.

Measurements of communication between Tivoli Monitoring components have shown that setting tcp_nodelayack to 1 can significantly improve performance.

To make the parameter setting, issue the following:

```
# no -p -o tcp_nodelayack=1
Setting tcp_nodelayack to 1
Setting tcp_nodelayack to 1 in nextboot file
```

The `-p` flag makes the change persistent, so that it is still in effect at the next boot. This is a dynamic change that takes effect immediately.

Validating your installation

You can use the **kincinfo** command to validate your installation. On Windows systems, view information for your monitoring server, including inventory of installed IBM Tivoli products, configuration settings, installed CD versions, and a list of running IBM Tivoli processes.

CLI syntax

```
kincinfo
[-d]
[-i]
[-r]
[-l]
[-t]
```

Where:

- **-d** Displays a list of installed products, which can be parsed.
- **-i** Lists the inventory in English.
- **-r** Displays a list of running agents.
- **-l** Displays the log switch.
- **-t** Displays the product name, version, build information, and installation date for all of the products installed in the installation directory.

  You can also use this option to review the installed support of self-described agents, which is displayed in a table. The following example shows the table for the r2 agent:

  ```
  ************* Thursday, May 20, 2010 3:23:29 PM *************
  User : Administrator Group : NA
  Host Name : NC045161 Installer : Ver: 062300000
  CandleHome : C:\IBM\ITM
  Installitm : C:\IBM\ITM\InstallITM
  *************************************************************
  ...Product Inventory
  PC PRODUCT DESC PLAT VER
  BUILD INSTALL DATE
  R2 Agentless Monitoring for Windows Operating Sy WINNT 06.23.00.00
  201004121113 20100520 1459
  PC SELF-DESCRIBED APPLICATION SUPPORT PACKAGE PLAT APP VER
  R2 Agentless Monitoring for Wind - r2tms_support CMS 06.23.00.00
  R2 Agentless Monitoring for Wind - r2tps_support CNS 06.23.00.00
  R2 Agentless Monitoring for Wind - r2tpw_support XEB 06.23.00.00
  PC APPLICATION SUPPORT DESC PLAT APP VER
  BUILD INSTALL DATE
  R2 Agentless Monitoring for Windows Operating Sy WICMS 06.23.00.00
  201004121113 20100520 1512
  ```

CLI example

The following example shows all installed products:

```
kincinfo -i
```

The following is the output of this example:

```
kincinfo -i *********** Mon May 07 14:19:20 Eastern Daylight Time 2007 ********
User : Administrator Group : NA
Host Name : FVWIN18 Installer: Ver: 0NOVALUE00000
CandleHome: C:\IBM\ITM
****************************************************
...Product Inventory
A4 i5/OS Support
WINNT Version: 06.10.05.01 Build: 200702230014
A4 i5/OS Support
WINNT Version: 06.10.05.01 Build: 200702230014
A4 i5/OS Support
WINNT Version: 06.10.05.01 Build: 200702230014
A4 i5/OS Support
WINNT Version: 06.10.05.01 Build: 200702230014
AX Tivoli Enterprise Monitoring Agent Framework
WINNT Version: 03.50.03.00 Build: 200510061051
CJ Tivoli Enterprise Portal Desktop Client
WINNT Version: 06.10.05.01 Build: 200705012123
```

```
CQ Tivoli Enterprise Portal Server
WINNT Version: 06.10.05.01 Build: 200705012135
CW Tivoli Enterprise Portal Browser Client
WINNT Version: 06.10.05.01 Build: 200705012123
GL Tivoli Enterprise Monitoring Agent Framework
WINNT Version: 06.10.05.01 Build: 200705012139
HD Warehouse Proxy
WINNT Version: 06.10.05.01 Build: 200705012139
IT TEC GUI Integration
WINNT Version: 06.10.05.01 Build: 200611010030
IT TEC GUI Integration
WINNT Version: 06.10.05.01 Build: 200611010031
IT TEC GUI Integration
WINNT Version: 06.10.05.01 Build: 200611010030
KF IBM Eclipse Help Server
WINNT Version: 06.10.05.01 Build: 200704171107
LZ Linux OS Support
WINNT Version: 06.10.05.01 Build: 200704301644
LZ Linux OS Support
WINNT Version: 06.10.05.01 Build: 200704301644
LZ Linux OS Support
WINNT Version: 06.10.05.01 Build: 200704301644
LZ Linux OS Support
WINNT Version: 06.10.05.01 Build: 200704301644
MS Tivoli Enterprise Monitoring Server
WINNT Version: 06.10.05.01 Build: 200705012139
NS NLS Support
WINNT Version: 03.50.03.00 Build: 200510061051
NT Monitoring Agent for Windows OS
WINNT Version: 06.10.05.01 Build: 200701160818
NT Windows OS Support
WINNT Version: 06.10.05.01 Build: 200701160818
NT Windows OS Support
WINNT Version: 06.10.05.01 Build: 200701160818
NT Windows OS Support
WINNT Version: 06.10.05.01 Build: 200701160818
NT Windows OS Support
WINNT Version: 06.10.05.01 Build: 200701160818
SY Summarization and Pruning agent
WINNT Version: 06.10.05.01 Build: 200705012135
SY Summarization and Pruning agent
WINNT Version: 06.10.05.01 Build: 200705012135
TM IBM Tivoli Monitoring 5.x Endpoint Support
WINNT Version: 06.10.05.01 Build: 200604051327
TM IBM Tivoli Monitoring 5.x Endpoint Support
WINNT Version: 06.10.05.01 Build: 200604051327
UI Tivoli Enterprise Services User Interface
WINNT Version: 06.10.05.01 Build: 200705012139
UL UNIX Logs Support
WINNT Version: 06.10.05.01 Build: 200701120847
UL UNIX Logs Support
WINNT Version: 06.10.05.01 Build: 200701120847
UL UNIX Logs Support
WINNT Version: 06.10.05.01 Build: 200701120847
UL UNIX Logs Support
WINNT Version: 06.10.05.01 Build: 200701120847
UM Universal Agent
WINNT Version: 06.10.05.01 Build: 200705012131
UM Universal Agent Support
WINNT Version: 06.10.05.01 Build: 200705012131
UM Universal Agent Support
WINNT Version: 06.10.05.01 Build: 200705012131
UM Universal Agent Support
WINNT Version: 06.10.05.01 Build: 200705012131
UM Universal Agent Support
WINNT Version: 06.10.05.01 Build: 200705012131
UX UNIX OS Support
```

```
WINNT Version: 06.10.05.01 Build: 200701161329
UX UNIX OS Support
WINNT Version: 06.10.05.01 Build: 200701161329
UX UNIX OS Support
WINNT Version: 06.10.05.01 Build: 200701161329
UX UNIX OS Support
WINNT Version: 06.10.05.01 Build: 200701161329
C:\ >
```

Part 5. Setting up data warehousing

The chapters in this section provide instructions for installing and configuring the components that collect and manage historical data in the Tivoli Data Warehouse.

Chapter 18, “Tivoli Data Warehouse solutions,” on page 465 introduces and summarizes your options for database platforms, operating systems, and communications between warehousing components.

Chapter 20, “Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows,” on page 489 provides information and instructions for implementing a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows for the warehouse database.

Chapter 21, “Tivoli Data Warehouse solution using DB2 on z/OS,” on page 519 provides information and instructions for implementing a Tivoli Data Warehouse solution using DB2 on z/OS for the warehouse database.

Chapter 22, “Tivoli Data Warehouse solution using Microsoft SQL Server,” on page 547 provides information and instructions for implementing a Tivoli Data Warehouse solution using Microsoft SQL Server for the warehouse database.

Chapter 23, “Tivoli Data Warehouse solution using Oracle,” on page 571 provides information and instructions for implementing a Tivoli Data Warehouse solution using Oracle for the warehouse database.

Chapter 24, “Tivoli Data Warehouse solutions: common procedures,” on page 595 contains information and procedures common to the Tivoli Data Warehouse solutions that use any of the supported database platform. You should read the chapter that pertains to the database platform you are using for the warehouse before you read this chapter.

Chapter 18. Tivoli Data Warehouse solutions

The term *Tivoli Data Warehouse solution* refers to the set of IBM Tivoli Monitoring components, successfully installed and configured, that interact to collect and manage historical data. These *warehousing components* include the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server, the Tivoli Data Warehouse database, the Warehouse Proxy Agent, and the Summarization and Pruning Agent.

While all Tivoli Data Warehouse solutions include these four components, the details of any particular solution vary according to the size of the environment, which database management systems are used, and which operating systems are involved. This chapter introduces and summarizes your options for database platforms, operating systems, and communications between warehousing components.

- "Planning assumptions" on page 474
- "Firewall considerations for the Tivoli Data Warehouse" on page 476
- "Summary of supported operating systems" on page 478

Each of the next four chapters describes how to implement solutions for a variety of configurations using one of the supported database platforms: IBM DB2 for Linux, UNIX, and Windows, IBM DB2 on z/OS, Microsoft SQL Server, and Oracle.

New in Version 6.2.3

See the following changes to the Tivoli Data Warehouse component for Tivoli Monitoring V6.2.3:

New 64-bit Windows Warehouse Proxy Agent

The 32-bit Warehouse Proxy Agent (the only agent supported on Windows platforms in earlier releases) requires a 32-bit ODBC driver, which is difficult to configure on a 64-bit Windows system. If the ODBC driver is not correctly configured, the agent cannot connect to the data warehouse, and the resulting error messages can be very confusing.

To eliminate this need for a 32-bit ODBC driver, a 64-bit Warehouse Proxy Agent is now provided for users running 64-bit Windows environments. The 64-bit agent can run with the 64-bit ODBC driver, which is available by default on a 64-bit Windows system.

At installation time, you must choose which Warehouse Proxy Agent to install on your Windows systems because the 32-bit agent cannot coexist with the 64-bit agent. If you currently run the Tivoli Data Warehouse with a 32-bit Warehouse Proxy Agent on Windows and you want to upgrade to the new 64-bit agent, you must first uninstall the 32-bit agent, and then install the 64-bit agent.

If choosing the 32-bit agent, use the `ODBCAD32.EXE` file found in the Windows `SysWow64` directory. The file on the Windows **Start** menu is not the correct one.

Database compression for Tivoli Data Warehouse tables and indexes

You can now enable database compression for Tivoli Data Warehouse tables and indexes to reduce the amount of disk space used by the database, and improve database performance. This functionality is for newly created tables only and might require additional Relational Data Base Management System licensing. For more information, see the *IBM Tivoli Warehouse Proxy Agent User's Guide* and the *IBM Tivoli Summarization and Pruning Agent User's Guide*.

New compression before upload feature

New in Tivoli Monitoring V6.2.3 is the compression before upload feature that allows you to compress the data uploaded to the Warehouse Proxy Agent.

New default configuration for inserts into Tivoli Data Warehouse tables

Statements from the Warehouse Proxy Agent into the WAREHOUSELOG table are now disabled by default, and statements from the Summarization and Pruning Agent into the WAREHOUSEAGGREGLOG table are also disabled by default. The self-monitoring workspaces provided by the Warehouse Proxy Agent and Summarization and Pruning Agent provide sufficient information to determine if both agents are operating correctly. Tivoli Data Warehouse log tables can grow very large and require regular pruning. This new default configuration decreases the Summarization and Pruning Agent processing time, and decreases database resource utilization and contention. To restore the old behavior, you must edit the Warehouse Proxy Agent configuration file and change the `KHD_WHLOG_ENABLE` variable for the Warehouse Proxy Agent, and the `KSY_WHLOG_ENABLE` variable for the Summarization and Pruning Agent. For more information, see the *IBM Tivoli Warehouse Proxy Agent User's Guide* and the *IBM Tivoli Summarization and Pruning Agent User's Guide*.

The Schema Publication Tool uses the compression setting and generates appropriate DDL statements

The Schema Publication Tool can now generate DDL scripts that take into account the database compression option. For more information, see “Using the Schema Publication Tool for database compression” on page 487.

Configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent is completed by using a new graphical interface

A similar graphical interface is now used on all platforms. Windows systems use an ODBC connection, which means ODBC fields are included in the graphical interface. UNIX and Linux systems use a JDBC connection, which means JDBC fields are included in the graphical interface. The following actions that happened during installation now no longer occur and must be done via a checkbox on the Tivoli Enterprise Portal Server configuration panel:

- The Warehouse Database is not created for DB2.
- The ODBC Datasource is no longer created if it does not already exist on the Warehouse Proxy Agent system.
- The Windows ITMUser is not created on the Warehouse Proxy Agent system.

The new functionality allows for remote configuration of the agents from the Tivoli Enterprise Portal and the command-line interface. You can also remotely deploy, remotely manage, and silently install the agents. Configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent takes place during the setup of the IBM Tivoli Monitoring infrastructure. For more information, see “Configuring the Warehouse Proxy Agent and Summarization and Pruning Agent using the new graphical interface” on page 619.

During an upgrade installation to Tivoli Monitoring V6.2.3 on Windows systems, configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent preserves the Tivoli Data Warehouse connection settings for the Tivoli Enterprise Portal, the Warehouse Proxy Agent, and Summarization and Pruning Agent. This preservation of settings reduces the time and effort required to perform the upgrade. You should be aware of the following changes:

- The **Synchronize TEPS Warehouse Information** option on the Warehouse connection configuration window is no longer available. The Installer does not synchronize the Warehouse connection information between the Tivoli Enterprise Portal Server and the Warehouse Proxy Agent. If you change the connection settings for one of these components, you must also manually change the settings in the other. The Summarization and Pruning Agent and the Performance Analytics Agent also connect to the Warehouse database, and these components must also be manually reconfigured.
- The database type **Other** is no longer available when configuring the Tivoli Enterprise Portal Server Warehouse connection. You can choose from DB2, MSSQL and Oracle.

- On Windows systems, when configuring a Warehouse connection to the Tivoli Enterprise Portal Server with a DB2 database, you can now select a checkbox to have the Warehouse database and Data Source created by the Installer. For this option, your DB2 database must be on the same system.

  Note: Before the Warehouse Proxy Agent is configured, you must manually create the Warehouse database, database user, and ODBC DSN. Use the **Create database and Warehouse data source** checkbox for quick, proof-of-concept installations and not for production installations. The installer tool creates the database and ODBC DSN only if the checkbox is selected.

  The **Database Name**, **DB Admin User ID** and **DB Admin Password** fields are no longer required fields on the Warehouse connection window.

Using the silent response file to configure the Warehouse Proxy Agent and Summarization and Pruning Agent

Use the silent-mode configuration option to perform an unattended and automatic installation and configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent as well as the remote configuration of these two agents. Configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent takes place during the setup of the IBM Tivoli Monitoring infrastructure. On UNIX systems, installation and configuration are separate steps with separate response files. For more information, see "Installing and configuring the Warehouse Proxy Agent and Summarization and Pruning Agent in silent mode, using the silent response file" on page 620.

Using the CLI and the Tivoli Enterprise Portal to remotely configure the Warehouse Proxy Agent and the Summarization and Pruning Agent

By using the CL and the Tivoli Enterprise Portal, you can now remotely configure the Warehouse Proxy Agent and the Summarization and Pruning Agent. For more information, see "Configuring the Warehouse Proxy Agent using the Tivoli Enterprise Portal" on page 632 and "Configuring the Summarization and Pruning Agent using the Tivoli Enterprise Portal" on page 633.

Using the Tivoli Enterprise Portal to remotely manage the Warehouse Proxy Agent and the Summarization and Pruning Agent

You can now use the Tivoli Enterprise Portal to remotely manage the Warehouse Proxy Agent and the Summarization and Pruning Agent. For more information, see "Remotely starting and stopping the Warehouse Proxy Agent using the Tivoli Enterprise Portal" on page 633 and "Remotely starting and stopping the Summarization and Pruning Agent using the Tivoli Enterprise Portal" on page 634.

Remote deployment of the Warehouse Proxy Agent and the Summarization and Pruning Agent

You can now remotely deploy the Warehouse Proxy Agent and the Summarization and Pruning Agent. You can use this function whenever the Warehouse Proxy Agent or the Summarization and Pruning Agent must be installed remotely. For more information, see "Remotely deploying the Warehouse Proxy Agent" on page 635 and "Remotely deploying the Summarization and Pruning Agent" on page 636.

New in V6.2.2 fix pack 2

To support enterprise monitoring agents that are running autonomously, it is now possible to run the Warehouse Proxy and Summarization and Pruning agents autonomously.

If a Warehouse Proxy Agent is configured to run in autonomous mode, it does not register its location with the global location broker on the hub monitoring server to make it available to agents. Instead, agents obtain the location of the Warehouse Proxy Agent from their local configuration file or from a central configuration server (see the *IBM Tivoli Monitoring: Administrator's Guide* for more information on the central configuration server). Providing the location of the Warehouse Proxy Agent in the configuration information allows an agent to pass historical data to the Warehouse Proxy Agent for insertion into the Tivoli Data Warehouse database without a connection to a monitoring server and prevents the loss of historical information.

Beginning with V6.2.2 fix pack 2, historical data collection and summarization and pruning settings are stored in a table in the Tivoli Data Warehouse database, instead of in the Tivoli Enterprise Portal database. In addition, if the Summarization and Pruning Agent is configured to run in autonomous mode, it looks for required application support files in a specified location instead of obtaining the information from the portal server. This means that the Summarization and Pruning Agent can operate without connecting to a Tivoli Enterprise Portal Server.

Historical data collection and summarization and pruning settings previously saved to the Tivoli Enterprise Portal database are migrated to the Tivoli Data Warehouse database automatically the first time the Summarization and Pruning Agent is started after the WAREHOUSESUMPRUNE table is created in the Tivoli Data Warehouse. The table is automatically created when the first time the V6.2.2FP2 Summarization and Pruning Agent is started. Any settings specified subsequently are stored directly in the Tivoli Data Warehouse's WAREHOUSESUMPRUNE table. Once these parameters have been migrated, you can continue to configure historical data collection and summarization and pruning using the Tivoli Enterprise Portal, or you can configure them directly in the warehouse database table using the SQL insert command.

Notes:

1. The Summarization and Pruning Agent can operate without a connection to a Tivoli Enterprise Portal Server, but a portal server must be installed and application support for all agents that will be collecting historical data must be installed on the portal server for required application support files to be available.
2. Once the Tivoli Enterprise Portal Server has migrated the Summarization and Pruning Agent's configuration parameters to the WAREHOUSESUMPRUNE database table, the Tivoli Enterprise Portal must be able to connect to that table. Otherwise, users (including possibly your database administrator) will be unable to even view the Summarization and Pruning parameters.

For more information, see “Running the warehouse agents autonomously” on page 609.

New in V6.2.2 fix pack 1

The Tivoli Data Warehouse now supports IBM DB2 Database for Linux, UNIX, and Windows version 9.7.

New in V6.2.1

- Starting with IBM Tivoli Monitoring V6.2.1, the Warehouse Proxy Agent and the Summarization and Pruning Agent can use a DB2 version 9.1 (or subsequent) environment running on z/OS as the data repository. Instructions for implementing a Tivoli Data Warehouse using DB2 on z/OS are in Chapter 21, “Tivoli Data Warehouse solution using DB2 on z/OS,” on page 519.
- The Tivoli Data Warehouse now supports 64-bit agent data.
- The Tivoli Data Warehouse now supports IBM DB2 Database for Linux, UNIX, and Windows version 9.5.
- With the new schema publication tool, you can now generate the SQL statements needed to create the database objects (data warehouse tables, indexes, functions, views, and ID table inserts) required for initial setup of the Tivoli Data Warehouse; see Chapter 19, “Schema Publication Tool,” on page 483.

New in Version 6.2

The following features are new in IBM Tivoli Monitoring Version 6.2:

- Use of variable length character columns:

  Starting with V6.2, character columns in raw data and summary tables larger than 16 characters are now created as variable length columns (VARCHAR in DB2 for Linux, UNIX, and Windows, VARCHAR2 in Oracle, VARCHAR or NVARCHAR in SQL Server) rather than fixed length CHAR or NCHAR columns. This significantly reduces disk space requirements for tables and improves the performance and scalability of the warehouse.

- Most key columns in summary tables defined as NOT NULL:

  Many columns in summary tables can never have a NULL value. Defining these columns as NOT NULL reduces disk space requirements. This is particularly true for DB2 since these columns are indexed. Index disk space requirements are significantly reduced when using DB2 for Linux, UNIX, and Windows.
- Improved indexing on summary tables

  Indexes on summary tables now include all of the key columns needed for SNP processing. This significantly improves SNP performance since many tablespace scans have been eliminated.

These improvements are available only on tables or indexes created by the 6.2 version of the Warehouse Proxy and Summarization and Pruning Agents. Currently there is no migration utility to convert tables and indexes that were created in 6.1 versions of these agents, but those tables and indexes can be used by all of the V6.2 components.

Planning considerations for the Tivoli Data Warehouse

In a large or enterprise environment, you have the option of using multiple databases for the Tivoli Data Warehouse or having all of your hub monitoring servers use one single database. The benefit of using one database across your environment is that the information is stored in one location, meaning that you can more easily and accurately generate reports that reflect your entire environment (as opposed to needing to collate reports from several different databases). However, because the amount of data that is generated during history collection across a large or enterprise environment can be immense, you need to carefully plan the scale and performance of that database.

In addition to the planning information below, the IBM redbook *Tivoli Management Services Warehouse and Reporting*, SG24-7290, provides database performance tuning information. You can download the book at the following location: http://www.redbooks.ibm.com/abstracts/sg247290.html?Open.

Estimating the required size of your database

One of the factors to consider when planning the size of database that you need is the amount and type of information you will collect for agent history data collection. Each of the agent user's guides provide capacity planning information to help you calculate the amount of disk space required by data for each attribute group. Use this information to complete the following calculations to determine how large your data warehouse database needs to be:

- "Step 1: Determine the number of detailed records per day for each attribute group"
- "Step 2: Determine the hard disk drive footprint for each attribute group" on page 470
- "Step 3: Determine the amount of detailed data for each attribute group" on page 470
- "Step 4: Calculate the amount of aggregate data for each attribute group" on page 470

Use the worksheets in Appendix A, "Installation worksheets," on page 777 to record the values from these calculations. To print these, go to the IBM Tivoli Monitoring information center: http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/.

You might use the Warehouse Load Projection tool available in the IBM Tivoli Integrated Service Management Library. (Search on "Warehouse Load Projects at the following site: http://www.ibm.com/software/tivoli/opal.) The tool does all the calculations for you and includes data for nearly all of the IBM Tivoli Monitoring V6.x-based monitoring agents.

Step 1: Determine the number of detailed records per day for each attribute group

Determine the number of detailed records per day for each attribute group that you want to collect data for. Use the following equation:

```
(60 / collection interval) * (24) * (# instances at each interval)
```

where:

60 Represents the 60 minutes in an hour.

collection interval
The data collection interval, in minutes. This value can be 1, 5, 15, 30, 60, or 1440 (1 day).

24 Represents 24 hours in one day.

# instances at each interval
The number of instances recorded at each interval. See the agent user's guide for this value.

Step 2: Determine the hard disk drive footprint for each attribute group

Determine the hard disk drive footprint for each attribute group. The result generated by this formula gives an estimate of the amount of disk space used for this attribute group for 24 hours worth of data for a single agent.

Use the following equation:

```
(# detailed records) * (attribute group detailed record size) / 1024
```

where:

# detailed records
The number of detailed records for the attribute. This is the value you calculated in "Step 1: Determine the number of detailed records per day for each attribute group" on page 469.

attribute group detailed record size
The detailed record size for the attribute group. See the agent user's guide for this value.

1024 Represents 1 KB and causes the equation to generate a kilobyte number instead of a byte number.

Step 3: Determine the amount of detailed data for each attribute group

Determine the amount of detailed data in the warehouse database for each attribute group. Use the following equation:

```
(attribute group disk footprint) * (# of agents) * (# days of detailed data) / 1024
```

where:

attribute group disk footprint
The disk footprint for the attribute group. This is the value you calculated in "Step 2: Determine the hard disk drive footprint for each attribute group."

# of agents
The number of agents of the same agent type in your environment.

# days of detailed data
The number of days for which you want to keep detailed data in the warehouse database.

1024 Represents 1 KB and causes the equation to generate a megabyte number.

Step 4: Calculate the amount of aggregate data for each attribute group

Determine the amount of aggregate data in the warehouse for each attribute group.

First, calculate the number of aggregate records per agent. Use the following equation:

```
(#hourly + #daily + #weekly + #monthly + #quarterly + #yearly) * (# instances at each interval)
```

where:

#hourly
The number of hourly records for the attribute group. For example, if you have hourly records for 60 days, the number of hourly records is 1440 (60 multiplied by 24 hours per day).

#daily The number of daily records for the attribute group. For example, if you have daily records for 12 months, the number of daily records is 365.

#weekly
The number of weekly records for the attribute group. For example, if you have weekly records for a 2-year period, the number of weekly records is 104 (2 multiplied by 52 weeks per year).

#monthly
The number of monthly records for the attribute group. For example, if you have monthly records for a 2-year period, the number of monthly records is 24.

#quarterly
The number of quarterly records for the attribute group. For example, if you have quarterly records for a 2-year period, the number of quarterly records is 8 (2 years multiplied by 4 quarters in a year).

#yearly
The number of yearly records for the attribute group. For example, if you have yearly reports for a 10-year period, the number of yearly records is 10.

# instances at each interval
The number of instances recorded at each interval. See the agent user's guide for this value.

Next, use the following equation to calculate the amount of attribute data in the warehouse for the attribute group:

```
(# aggregate records per agent) * (attribute group aggregate record size) * (# agents) / 1048576
```

where:

# aggregate records per agent
The number of aggregate records per agent for the attribute group.

attribute group aggregate record size
The size of the attribute group aggregate group. See the agent user's guide for this value.

# agents
The number of agents of the same agent type in your environment.

1048576
Represents 1 MB and causes the equation to generate a megabyte number.

Step 5: Determine the estimated size of your database

First, determine the total space required for each attribute group. Add the amount of detailed data and the amount of aggregate data for the attribute group. Use the following equation:

```
(detailed data size) + (aggregate data size)
```

Second, determine the total space required for all attribute groups for the agent. Add the total space for each attribute group that you want to collect. Use the following equation:

```
aggGroup1 + aggGroup2 + aggGroup3 ...
```

Third, determine the total space required for all agents. Add the total space for each agent. Use the following equation:

```
agent1 + agent2 + agent3 ...
```

Finally, to estimate the total disk space requirement for the database, multiplying the total amount of data (detailed + aggregate for all attribute groups) by 1.5 (to increase the number by 50%). Compare this number to the **Database data** row in Table 77 on page 473 to determine the number of disks you need for your database.

Use the following worksheet to estimate the size of your data warehouse database.

Table 76. Tivoli Data Warehouse database size estimation worksheet

		Data from the User's Guide										
Attribute group	**Number of agents**	**Record size–detailed**	**Record size–aggregated**	**Interval instances**	**Collection interval**	**Detailed records per day***	**Attribute agent disk space (KB)***	**Days of detailed data kept**	**Warehouse space–detailed (MB)***	**Aggregate records**	**Warehouse space–aggregated (MB)***	**Total warehouse space for attribute group (MB)**
Total warehouse data size (sum of all attribute group total warehouse space)												
Total database size (total warehouse data size * 1.5)												

* Use the equations in “Estimating the required size of your database” on page 469 to obtain these values.

Understanding the disk requirements for your database

Consider the following factors when designing a disk subsystem to support your database processing:

- Disk crash: With sufficient funds and planning, you can build a system that can continue running without interruption or be recovered in a few hours, despite a system crash. You can provide disk protection by using some level of RAID on parts, or all, of the disk subsystem. Some of the more popular types of RAID include the following:
  - RAID 1, also known as *disk mirroring*, uses data mirroring to achieve a high level of redundancy. In a RAID 1 configuration, two copies of the data are kept on separate disks, each mirroring the other.
  - RAID 5 uses block-level striping with distributed parity. RAID 5 stripes both data and parity information across three or more drives.
- The database log needs disk protection to enable database recovery of recent transactions. The other disks can optionally be protected. If they are protected, you can eliminate downtime while data is recovered.
- Consider the OS and paging space disks.
- Also, consider including one or two additional disks to speed up recovery and reduce the risks while running after a disk failure.
- Because of the increasing capacity of disk drives, the configurations listed below result in excess disk capacity but increase the number of disks available for I/O throughput.

The following table provides some example sizes for a database:

Table 77. Database size examples

	Number of disks to use			
	Absolute minimum disks	**Small RDBMS**	**Small and safe RDBMS**	**Large RDBMS**
Operating System	1	1	1 + mirror	1
Paging and RDBMS code	Use above	1	1 + mirror	1
RDBMS data	1	1	1 + mirror	8
RDBMS indexes	1	1	1 + mirror	6
RDBMS temp	Use above	1	1 + mirror	6
RDBMS logs	1 + mirror	1 + mirror	1 + mirror	2
Database data	12 GB	24 GB	48 GB	108+ GB
Number of disks	5	7	12	24

The **Absolute minimum disks** column specifies the minimum number of disks for an RDBMS. In this column, the index and temporary space is allocated onto one disk. While not an ideal arrangement, this might work in practice because databases tend to use indexes for transactions or temporary space for index creation and sorting full table scan large queries, but not both at the same time. This is not a recommended minimum disk subsystem for a database, but it does have the lowest cost.

The **Small RDBMS** column represents a minimum disk subsystem, although there might be limits in I/O rates because of the data being placed on only one disk. Striping the data, indexes, and temporary space across these three disks might help reduce these I/O rate limits. This disk subsystem arrangement does not include disk protection for the database or other disks (apart from the mandatory log disk protection for transaction recovery).

The **Small and safe RDBMS** column adds full disk protection and can withstand any disk crash with zero database downtime.

The **Large RDBMS** column represents a typical size database for a database subsystem. Disk protection is not included in these sizings but can be added to increase the stability of the database.

Increasing the size of your database (DB2 for Linux, UNIX, and Windows only)

DB2 for Linux, UNIX, and Windows Workgroup Edition has a default table size limit of 64 GB with a page size of 4 KB. To increase the capacity of your DB2 for Linux, UNIX, and Windows database, you can create a new tablespace, IBMDEFAULTGROUP, and choose a larger page size (up to 32 KB). This increases the capacity of the database up to 512 GB per table.

The following example creates the IBMDEFAULTGROUP tablespace with a page size of 16 K. This increases the table size capacity to 256 GB.

```
CREATE REGULAR TABLESPACE IBMDEFAULTGROUP IN DATABASE PARTITION GROUP IBMCATGROUP
PAGESIZE 16384 MANAGED BY DATABASE
  USING (FILE 'E:\DB2\NODE0000\SQL00001\IBMDEFAULTGROUP.001'1500000)
  EXTENTSIZE 32
  PREFETCHSIZE AUTOMATIC
  BUFFERPOOL IBM16KBP
  OVERHEAD 12.670000
  TRANSFERRATE 0.180000
  DROPPED TABLE RECOVERY ON
```

If 512 GB of space per table is not enough for your environment, move to DB2 for Linux, UNIX, and Windows Enterprise Edition and using physical or logical partitioning.

The following steps outline the process for using database partitioning with DB2 for Linux, UNIX, and Windows Enterprise Edition:

1. Add a new database partition to the DB2 for Linux, UNIX, and Windows instance by running the **db2ncrt** command.
2. Use the ALTER TABLE statement to add a partitioning key to the tables that you want to partition. For example:

   ```
   ALTER TABLE "CANDLE "."NT_System"     ADD PARTITIONING KEY ("Server_Name")
        USING HASHING
   ```
3. Use the ALTER DATABASE PARTITION GROUP statement to assign the new partition to the database partition group. You can do this either from the command-line or from the DB2 Control Center.
4. Redistribute the data in the database partition group, using the Redistribute Data Wizard in the DB2 Control Center.

For additional information about database partitioning, see the following DB2 for Linux, UNIX, and Windows sources:

- *IBM DB2 Universal Database Administration Guide: Planning*
- *IBM DB2 Universal Database Administration Guide: Implementation*
- DB2 for Linux, UNIX, and Windows Information Center at http://publib.boulder.ibm.com/infocenter/db2help/index.jsp.

For additional database performance and tuning information, see the *IBM Tivoli Monitoring: Administrator's Guide*.

Planning assumptions

The information in this and the next four chapters is based on the following assumptions:

- You have completed the preliminary planning required to determine the size and topology of your environment and your warehousing needs. You may have already installed a monitoring server, a portal server, and some monitoring agents, but you have not yet created the Tivoli Data Warehouse.

- You are not installing IBM Tivoli Monitoring on one computer.
- You will create the Tivoli Data Warehouse database on a different computer from the Tivoli Enterprise Portal Server.
- If installing the warehouse database on Microsoft SQL Server, you will also install the Tivoli Enterprise Portal Server on a Windows-based computer. This restriction applies even if the warehouse database and portal server are installed on separate computers. For example, a portal server on Linux does not support a warehouse database using Microsoft SQL Server.
- If you are upgrading from IBM Tivoli Monitoring V6.1, you have performed the necessary agent warehouse database upgrade steps.

The following sections discuss these assumptions in more detail.

Preliminary planning is complete

For guidance on planning the size of your environment and determining the warehousing capacity you need, see “Sizing your Tivoli Monitoring hardware” on page 46 and “Planning considerations for the Tivoli Data Warehouse” on page 469. The deployment scenarios illustrate your options for where to locate the warehousing components, whether to have multiple Warehouse Proxy Agents reporting to the same hub monitoring server, and whether to deploy single or multiple hub installations.

Environment assumptions

The information in the current chapter and the next four chapters assume an environment with more than one computer. In environments where monitoring and warehousing needs are minimal, such as a test environment, you can install all IBM Tivoli Monitoring components on one computer: a monitoring server, portal server, data warehouse, Warehouse Proxy Agent, and Summarization and Pruning Agent. If you install all components on a Windows system with either DB2 for Linux, UNIX, and Windows or Microsoft SQL Server, many of the warehouse configuration tasks are automated.

If you want to deploy IBM Tivoli Monitoring on one Windows computer with either of these database platforms, follow the instructions in Chapter 8, “Installing IBM Tivoli Monitoring on one computer,” on page 187.

The data warehouse is remote from the portal server

There are two databases involved in IBM Tivoli Monitoring:

- The *Tivoli Enterprise Portal Server database* (or *portal server database*) stores user data and information required for graphical presentation on the user interface. The portal server database is created automatically during configuration of the portal server. It is always located on the same computer as the portal server.
- The *Tivoli Data Warehouse database* (also called the *warehouse database* or *data warehouse*) stores historical data for presentation in historical data views. While it is possible for the warehouse database to be located on the portal server (as in the single computer deployment), it is best to create the database on a remote computer to handle the warehousing needs of most production environments.

The procedures described in the next four chapters are based on the assumption that you will install the Tivoli Data Warehouse database remotely from the portal server.

No assumption is made about where you will install the Warehouse Proxy Agents and Summarization and Pruning Agent. Either of these agents may be installed on the same computer as the Tivoli Data Warehouse or on a different computer:

- Installing the Warehouse Proxy Agent and Summarization and Pruning Agent on the same computer as the data warehouse minimizes server deployments, simplifies configuration, and eliminates network transmission overhead.
- If you install these components on different computers, ensure that you have a high-speed network connection for best performance.

- In environments with more than one Warehouse Proxy Agent reporting to the same hub monitoring server, install each Warehouse Proxy on a separate computer.

Agent warehouse database upgrade

Some of the monitoring agents have made changes to the warehouse tables that require performing upgrade procedures from IBM Tivoli Monitoring V6.1 before running the V6.2 Warehouse Proxy and Summarization and Pruning Agents. See "Upgrading the warehouse" on page 169 for more information.

Firewall considerations for the Tivoli Data Warehouse

Firewalls can be a significant factor in most Tivoli Data Warehouse implementations. At a minimum:

- The Warehouse Proxy Agent must be able to communicate with:
  - The database
  - The hub monitoring server
  - The agents that are exporting data
- The Summarization and Pruning Agent must be able to communicate with:
  - The database
  - The monitoring server it is configured to communicate with (not necessarily the hub)
  - The Tivoli Enterprise Portal Server

  Note: The default Tivoli Enterprise Portal Server interface port of 15001 is also used after the initial connection of the Summarization and Pruning Agent to the portal server over port 1920. Any firewalls between the two need to allow communications on either 15001 or whichever port is defined for any newTivoli Enterprise Portal Server interface used per the instructions in "Defining a Tivoli Enterprise Portal Server interface on Windows" on page 408.
- The agents exporting data to the Warehouse Proxy Agent must be able to communicate with the Warehouse Proxy Agent.

Appendix C, "Firewalls," on page 799 contains information on the firewall options available. The IBM redbook *Tivoli Management Services Warehouse and Reporting*, available from www.redbooks.ibm.com, discusses firewall considerations specific to the Tivoli Data Warehouse in detail.

Compressing historical data transmission

The data from the agent or the monitoring server can be compressed before it is transmitted to the Warehouse Proxy Agent. The data is then decompressed at the Warehouse Proxy Agent before it is inserted in the warehouse database, allowing for efficient usage of network bandwith between the historical client and the Warehouse Proxy Server. The following reductions can occur:

- A reduction in timeout errors due to a low bandwith between clients and Warehouse Proxy Server. Timeout errors on the client side occur when it takes more than 15 minutes for an export of data to complete and be inserted in the warehouse database.
- A reduction in bandwith usage. Uploading historical data to the warehouse can tie up a significant amount of bandwidth, and also impact other critical applications sharing the network link.
- A reduction in the elapsed time for the upload over slow links.

Use this functionality when a slow network is used or when the network is shared with other critical applications. However, compressing the data does increase CPU usage, both at the agent where it is uploaded and at the Warehouse Proxy Agent where it is decompressed. CPU costs differently on a distributed source than on a z/OS source. In a mixed environment, it is possible to enable or disable the compression option (which is a server option), depending on the source operating system of the client. It is also possible to overwrite a server compression option.

Configuration parameters for the Warehouse Proxy Agent are set in the following files according to operating system:

Windows

ITM_HOME\TMAITM6\khdenv

For example: C:\IBM\ITM\TMAITM6\khdenv

Linux and UNIX

ITM_HOME/config/hd.ini

For example: /opt/IBM/ITM/config/hd.ini

The Warehouse Proxy Agent configuration file now contains two new variables that you can modify if necessary:

KHD_SERVER_DIST_COMPRESSION_ENABLE

If this variable is set to Y, then the Warehouse Proxy server allows distributed clients to send compressed data. This variable is set to Y by default.

KHD_SERVER_Z_COMPRESSION_ENABLE

If this variable is set to Y, then the Warehouse Proxy server allows z/OS clients to send compressed data. This variable is set to N by default.

The agent configuration file, if historical data stored at the Tivoli Enterprise Monitoring Agent, or the monitoring server configuration file, and if historical data is stored at the monitoring server, accepts a new warehouse variable:

KHD_CLIENT_COMPRESSION_ENABLE

If set to N, and even if the Warehouse Proxy Agent server has allowed the compression, the historical data is not compressed on this source. This variable does not exist by default.

The new Warehouse Proxy Agent variables are visible in a view in the configuration workspace of the Warehouse Proxy Agent. They also exist on the configuration GUI of the Warehouse Proxy Agent.

You can see if the compression is occurring by using a tracing level to see the data compressed at the client and uncompressed at the server: ERROR (UNIT: khdxmts DETAIL) in the warehouse proxy agent. To check uncompression in the server log file, see if the file contains the following text:

```
khdxmts.cpp,154,"KHD_SendBuff_from_xmit_rep") Uncompressing the data
```

Use ERROR (UNIT: khdxdacl DETAIL) to see the message about compression in the client. To check that the client has overwritten the server variable to not compress the data, see if the file contains the following text:

```
khdxmts.cpp,167,"KHD_SendBuff_from_xmit_rep" Client did not compress the data
```

If compression has failed, you will see the following text in the Warehouse Proxy Server log file, khdxrpcs.cpp:

```
"Client compression failed, data was sent uncompressed"
```

If compression has failed, you will see the following text in the client log file, khdxdacl.cpp:

```
"Warehouse client unable to compress the data, data sent uncompressed"
```

If uncompression has failed, you will see the following text in the Warehouse Proxy Server log file, khdxrpcs.cpp:

```
"Uncompress error"
```

If uncompression has failed, you will see the following text in the client log file, khdxdacl.cpp:

```
"Warehouse client unable to compress the data, data sent uncompressed"
```

Complete the following steps to setup compression:

1. Install or upgrade to the newest release.
2. Configure the Warehouse Proxy Agent, using the GUI. The GUI can pop up during install time, or can be launched by right clicking on the warehouse Proxy entry in the manage Tivoli Monitoring Services panel. The new server compression variables are part of the GUI and can be modified as needed.
3. Compression variables can also be reconfigured using the command-line interface (**itmcmd config hd**).
4. Compression variables can also be reconfigured in the configuration file itself. Configuration parameters for the Warehouse Proxy Agent are set in the following files according to operating system:

   Windows

   `ITM_HOME\TMAITM6\khdenv`

   For example: `C:\IBM\ITM\TMAITM6\khdenv`

   Linux and UNIX

   `ITM_HOME/config/hd.ini`

   For example: `/opt/IBM/ITM/config/hd.ini`

In mixed environments with a previous version of the client and newer version of the server, or new version of the client and a previous version of the server, the compression should not happen.

Next steps

After you have completed your preliminary planning, you are ready to implement your Tivoli Data Warehouse solution:

- Review the *Summary of supported operating systems* section in this chapter to understand your options for operating system platforms, database platforms, and communications between warehousing components, all summarized in a single composite graphic. This section also describes the relationships between warehousing components.
- The next four chapters are organized by database platform. Follow the instructions in the chapter that pertains to the database platform you are using for the Tivoli Data Warehouse database: IBM DB2 for Linux, UNIX, and Windows, IBM DB2 on z/OS, Microsoft SQL Server, or Oracle.

Summary of supported operating systems

Figure 110 on page 479 summarizes the supported operating system platforms for the various warehousing components, the supported database products, and the connections between components. For more specific information about supported operating systems and database products, including product names and versions, see “Hardware and software requirements” on page 138. In this diagram, UNIX refers to any UNIX platform supported by the RDBMS.

Note: The diagram is not meant to suggest that each warehousing component must be installed on a separate computer. Multiple components can exist on the same computer provided there is operating system support for all installed components. Connections between components must be configured whether or not they exist on the same computer. (Although the diagram makes no assumptions about where components are installed, the procedures described in the next four chapters assume a Tivoli Data Warehouse that is remote from the portal server.)

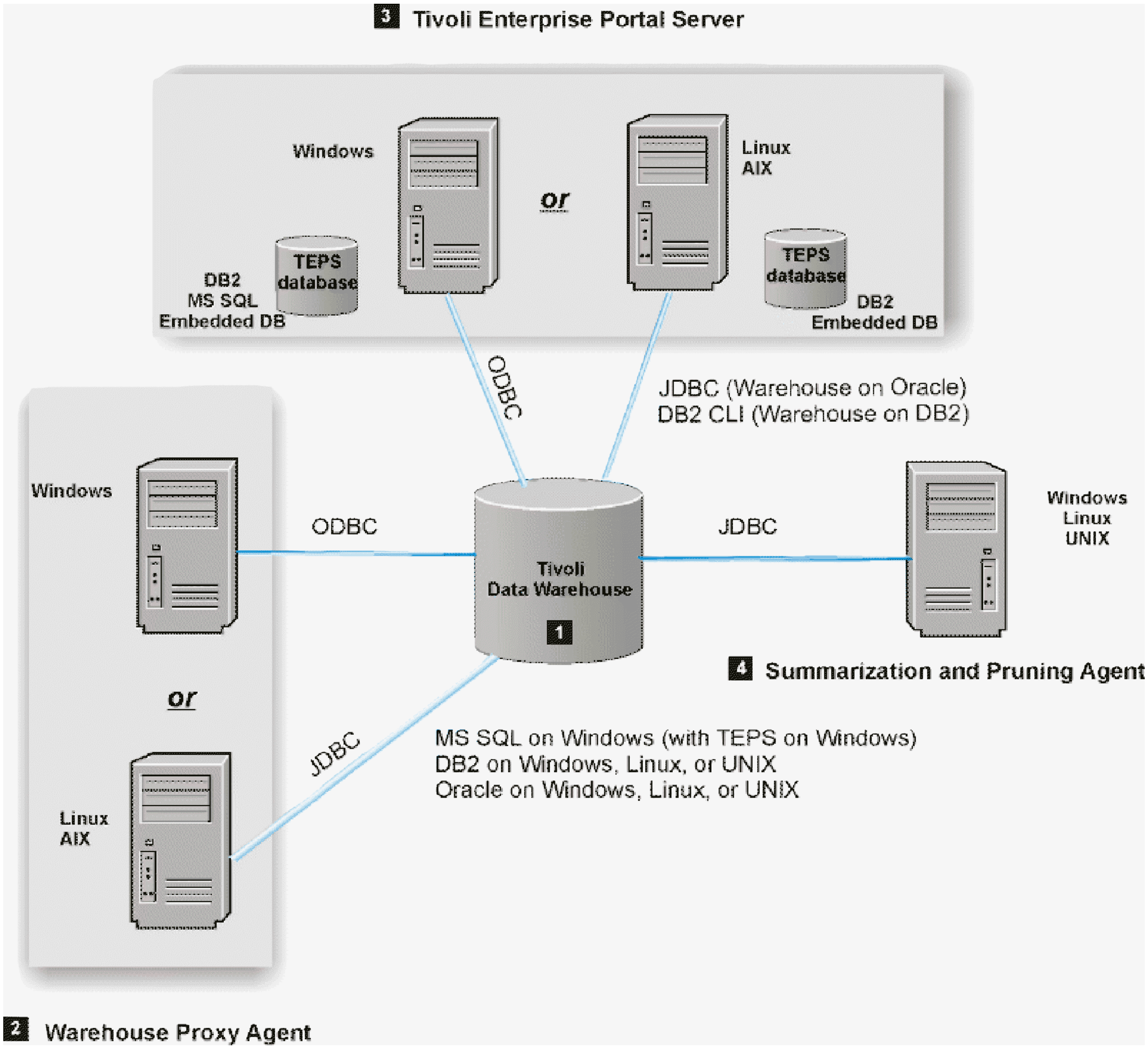

3 Tivoli Enterprise Portal Server
Windows
or
Linux
AIX
DB2
MS SQL
Embedded DB
TEPS database
TEPS database
DB2
Embedded DB
ODBC
JDBC (Warehouse on Oracle)
DB2 CLI (Warehouse on DB2)
Windows
ODBC
Tivoli Data Warehouse
1
JDBC
Windows
Linux
UNIX
4 Summarization and Pruning Agent
or
JDBC
MS SQL on Windows (with TEPS on Windows)
DB2 on Windows, Linux, or UNIX
Oracle on Windows, Linux, or UNIX
Linux
AIX
2 Warehouse Proxy Agent

Figure 110. Summary of support for the Tivoli Data Warehouse

In the following discussion, numbered product components correspond to the numbers on the diagram.

1 Tivoli Data Warehouse database

Data collected by monitoring agents is stored at intervals in short term history files. The data in the short term history files is referred to as *historical data*. The short term history files are located either at the agents or at the monitoring server (hub or remote) to which the agents report. (An administrator can determine where the data is stored. The monitoring agents and monitoring server are not shown in the diagram.)

The historical data is sent from its temporary storage location (at the monitoring agents or monitoring server) to the Warehouse Proxy Agent at a preset interval (either every hour or every 24 hours). The Warehouse Proxy Agent inserts the data it receives to the Tivoli Data Warehouse.

Data in the short term history files that is older than 24 hours is pruned when the monitoring agent or monitoring server receives an acknowledgment that the data has been successfully inserted into the Tivoli Data Warehouse. (The pruning of binary files is not the pruning performed by the Summarization and

Pruning Agent. The Summarization and Pruning Agent prunes data in the Tivoli Data Warehouse.) The result of these operations is that, at any given time, the short-term historical files contain data that is less than 24 hours old, and the Tivoli Data Warehouse contains long-term historical data that is older than 24 hours.

The Tivoli Data Warehouse database can be created using Microsoft SQL Server, IBM DB2 Database for Linux, UNIX, and Windows, Oracle, or IBM DB2 on Z/OS on the indicated operating system platforms. Note that the warehouse database is supported on Microsoft SQL Server only if the Tivoli Enterprise Portal Server is installed on Windows. This condition applies even if the warehouse database and portal server are installed on separate computers. For example, a portal server on Linux does not support a warehouse database on Microsoft SQL Server.

2 Warehouse Proxy Agent

A Warehouse Proxy Agent running on Windows uses an **ODBC** connection to send the collected data to the warehouse database. On a 32-bit Windows system, this requires a 32-bit ODBC driver, but on 64-bit Windows, it requires a 64-bit ODBC driver by default (although sites running a 32-bit Warehouse Proxy Agent under 64-bit Windows can configure a 32-bit ODBC driver for the agent to use; the necessary instructions are provided in this manual). The type of ODBC driver your site uses is determined not by the operating system but rather by the type of Warehouse Proxy Agent your site runs: the 32-bit agent requires a 32-bit ODBC driver, whereas the native 64-bit agent requires a 64-bit ODBC driver.

A Warehouse Proxy Agent running on Linux or AIX uses a **JDBC** connection.

The Warehouse Proxy Agent will do the following as needed:

- Create new tables and indexes.
- Alter existing tables.

  For example, add a new column to a table. This can occur if the table was created for an older version of an agent but data is being exported for the first time from a newer version of the agent that has more attributes.

Firewall considerations:

- The Warehouse Proxy Agent must be able to communicate with the hub monitoring server to register its network address. This address will then be replicated to all remote monitoring servers . If communication is impossible for firewall reasons, a variable indicating the Warehouse Proxy Agent network addresses can be set on the hub monitoring server and remote monitoring servers: KPX_WAREHOUSE_LOCATION.
- The Warehouse Proxy Agent must be able to communicate with the RDBMS.
- The agents exporting data must be able to communicate with the Warehouse Proxy, *and* the Warehouse Proxy Agent must be able to communicate with the agents (in other words, there must be two-way communication between agents and the Warehouse Proxy Agent).

Appropriate ports must be open to support these communication channels (see Appendix C, “Firewalls,” on page 799).

Depending on the database system you're using with your Tivoli Data Warehouse, make sure you complete one of these steps when installing the Warehouse Proxy Agent:

- If your site hosts its Tivoli Data Warehouse on DB2 Database for Linux, UNIX, and Windows, complete this step: “Configuring an ODBC data source for a DB2 data warehouse” on page 501.
- If your site hosts its Tivoli Data Warehouse on Microsoft SQL Server, complete this step: “Configuring an ODBC data source for a Microsoft SQL data warehouse” on page 555.
- If your site hosts its Tivoli Data Warehouse on Oracle, complete this step: “Configuring an ODBC data source for an Oracle data warehouse” on page 579.

Note: If your site has previously installed the 32-bit agent and you wish to upgrade to the 64-bit agent, you must first uninstall the 32-bit agent; see “Uninstalling the Warehouse Proxy” on page 859. This might also require removing the 32-bit ODBC driver you configured for the Warehouse Proxy Agent; see also “Removing the ODBC data source connection” on page 859.

When uninstalling a 32-bit agent so your site can upgrade to a 64-bit agent, remember that any customizations your site has made to the KHDENV configuration file must be redone. Also, a 64-bit ODBC driver must be installed for the 64-bit agent to use.

Usual process for activating the 64-bit Warehouse Proxy Agent:

1. Install the ODBC 64-bit data source, if not installed already.
2. Install the 64-bit agent.
3. Configure the agent to use a 64-bit data source.
4. Start the agent.

To determine which Warehouse Proxy Agent is running (32-bit or 64-bit), open the Windows Task Manager. The 32-bit agent has "*32" next to the process name.

3 Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server retrieves historical data for display in historical data views in the Tivoli Enterprise Portal. It retrieves short-term historical data from the binary files on the monitoring agents or monitoring server. It retrieves long-term historical data from the Tivoli Data Warehouse.

In Figure 110 on page 479, the Tivoli Enterprise Portal Server is shown with the *portal server database* (designated as *TEPS database* in the diagram). The portal server database stores user data and information required for graphical presentation on the user interface. Before you install and configure the portal server, you must install the database platform (RDBMS) to be used for the portal server database (DB2 for Linux, UNIX, and Windows or Microsoft SQL Server) on the same computer. The portal server database is created automatically during configuration of the portal server.

Although the portal server database is not considered a warehousing component, it is included in the diagrams in this and the following chapters because it can affect the installation and configuration tasks required for the Tivoli Data Warehouse database. For example, the database client already installed for the portal server database can connect to a remote warehouse database, provided both databases use the same database platform. There is no need to manually install another client.

If the portal server is installed on Windows, it uses an ODBC connection to request and retrieve historical data from the warehouse database. If the portal server is installed on Linux or AIX, it communicates with the warehouse database through a JDBC connection, if the warehouse is installed on Oracle, or through a proprietary DB2 CLI connection if the warehouse is installed on DB2 for Linux, UNIX, and Windows.

4 Summarization and Pruning Agent

The Summarization and Pruning Agent retrieves detailed data from the warehouse database, aggregates or prunes the data, and returns the processed data to the warehouse. Communication takes place through a JDBC connection, regardless of the operating system on which the Summarization and Pruning Agent is installed.

The Summarization and Pruning Agent will create tables, indexes, and views as needed. This could happen in either of the following situations:

- Summarization is enabled for the first time for an attribute group.
- Additional summarizations are enabled for an attribute group.

In addition, the Summarization and Pruning Agent, like the Warehouse Proxy Agent, may alter existing tables by adding new columns.

Firewall considerations:

- The Summarization and Pruning Agent must be able to communicate with the RDBMS.
- The Summarization and Pruning Agent must be able to communicate with the Tivoli Enterprise Portal Server.
- The Summarization and Pruning Agent must be able to communicate with the Tivoli Enterprise Monitoring Server it is configured to use.

Chapter 19. Schema Publication Tool

With the schema publication tool (an SQL editor), you can perform the following tasks:

- Generate the SQL statements needed to create the database objects required for initial setup of the Tivoli Data Warehouse.
- Create the necessary database objects whenever either historical collection is enabled for additional attribute groups or additional summarizations are enabled. This is referred to as running the schema publication tool in **updated** mode. Updated mode is also used if agents are upgraded to a new release. An upgrade might add new columns to existing tables and the schema publication tool will generate the appropriate alter table statements.

Generating SQL for data warehouse tables

You can use the schema publication tool to generate the SQL statements needed to create the database objects required for initial setup of the Tivoli Data Warehouse.

Before you begin

The schema publication tool is installed with the Summarization and Pruning agent. You should perform this task after product installation and after configuring the Summarization and Pruning agent, but before starting the Warehouse Proxy agent and the Summarization and Pruning agent for the first time.

About this task

By default, the database objects required for the data warehouse are created automatically by the installer, the Warehouse Proxy agent and the Summarization and Pruning agent. The schema publication tool enables you to create the data warehouse database objects manually rather than allowing them to be created automatically. There are several situations in which you might want to do this:

- You might want to use a separate database administration user ID for creating the tables, rather than granting permission to the Tivoli Data Warehouse user ID specified during installation.
- You might want to customize the SQL before creating the tables in order to accommodate performance considerations, security policies, or other issues unique to your environment.

The schema publication tool is a script that generates the SQL required to create the data warehouse tables, indexes, functions, views, and ID table inserts required for the selected products.You can then modify the generated SQL files before using them to create the tables and indexes.

Procedure

1. Create a new response file by making a copy of the sample response file:
   - On Windows systems: *itm_install_dir*\TMAITM6\tdwschema.rsp
   - On Linux and UNIX systems: *itm_install_dir*/*arch*/bin/tdwschema.rsp
2. Using an ASCII text editor, edit the response file to indicate the options you want to use. The keywords in the response file affect which SQL statements are generated, as well as other options:

   `KSY_PRODUCT_SELECT=`*`category`*

   The category of products for which you want to generate SQL files:

Value	Description
`installed`	All installed Tivoli Enterprise Portal Server products. **Note:** This can produce DDL for a large number of tables.

Value	Description
`configured`	All configured portal server products. Products that have historical collections defined. DDL is only generated for the attribute groups that have historical collections defined. DDL for summary tables is generated based on the summarization and pruning settings. For example, if an attribute group has hourly and monthly summarization enabled, DDL is generated for only the hourly and monthly summary tables.
`updated`	Configured portal server products with configuration changes that have not yet been deployed to the database. The **updated** value captures the changes that are necessary to bring the Tivoli Data Warehouse up to date to the current configuration. If you require the entire schema that is configured, you must use the **configured** value. Examples of updates: • Historical collections created for an attribute group that previously had no collections. • Enabling additional summarizations for one or more attribute groups. • Software update that changes existing attribute groups. Existing warehouse tables will need to have columns added. • Database compression is enabled at the database and the Summarization and Pruning agent is configured to use database compression. DDL is generated to compress existing tables and indexes.

This keyword is required.

`KSY_PRODUCT_FILTER=`*`product_filter`*

An optional filter to indicate that only certain specific products are included. (If you do not specify a filter, all products in the specified category are included by default.) Specify the three-letter product codes of the products you want to include, separated by commas. You can find these codes by using the **`tacmd histListProduct`** command (for more information, refer to the *IBM Tivoli Monitoring Command Reference*).

`KSY_SUMMARIZATION_SELECTION=`*`summarization_filter`*

An optional filter to indicate that only certain summarization options are to be included in the generated tables:

Value	Description
`H`	Hourly
`D`	Daily
`W`	Weekly
`M`	Monthly
`Q`	Quarterly
`Y`	Yearly

This keyword is valid only with `KSY_PRODUCT_SELECT=installed`.

`KSY_SQL_OUTPUT_FILE_PATH=`*`path`*

An optional path to the directory where the generated SQL files are to be written. If you do not include this keyword, the current working directory is used.

For more details and the complete syntax for each keyword, refer to the comments in the tdwschema.rsp sample response file.

3. Make sure the Tivoli Enterprise Portal Server is started.
4. Run the schema publication tool script using the appropriate syntax for your operating system.
   - Windows systems:

     ```
     tdwschema -rspfile response_file
     ```
   - Linux and UNIX systems:

     ```
     tdwschema.sh -rspfile response_file
     ```

   The SQL files for the products specified in the response file are generated and written to the directory indicated by the KSY_SQL_OUTPUT_FILE_PATH keyword (or to the current working directory, if no output directory is specified).
5. Make any necessary changes to the generated SQL files. For example, you might want to partition tables or assign tables to table spaces.

   Note: Do not change the names of any tables specified in the generated SQL files.
6. Use the appropriate tools to run the SQL queries to create the warehouse tables, indexes, views, inserts, and functions for your relational database. Execute the scripts in this order:
   a. tdw_schema_table.sql
   b. tdw_schema_index.sql
   c. tdw_schema_view.sql
   d. tdw_schema_insert.sql
   e. tdw_schema_function.sql

   The following examples are for the DB2 commands:

   ```
   db2 -tvf tdw_schema_table.sql
   db2 -tvf tdw_schema_index.sql
   db2 -tvf tdw_schema_view.sql
   db2 -tvf tdw_schema_insert.sql
   db2 -td# -f tdw_schema_function.sql
   ```

   Note: The different invocation for the `tdw_schema_function.sql`.

Using the schema publication tool in updated mode

Use the schema publication tool's updated mode to create the necessary database objects whenever either historical collection is enabled for additional attribute groups or additional summarizations are enabled.

About this task

Note: The schema publication tool is not intended for database migration. For information about migrating warehouse data from a previous version, refer to Chapter 4. If you run the schema publication tool in updated mode on an existing IBM Tivoli Monitoring version 6.1 Tivoli Data Warehouse database, the following will not be done:

- In an IBM Tivoli Monitoring V6.1 Tivoli Data Warehouse database, indexes on aggregate tables for a given attribute group do not include all of the key columns for the attribute group. This causes performance problems with the Summarization and Pruning agent. The updated mode does not generate SQL to recreate the indexes to optimize performance.
- In IBM Tivoli Monitoring V6.1, all character data was stored in fixed-length CHAR columns. In IBM Tivoli Monitoring V6.2, this was changed to VARCHAR, greatly reducing disk requirements and improving performance. The updated mode does not generate SQL to convert CHAR columns to VARCHAR.
- In IBM Tivoli Monitoring V6.1, all columns allowed NULL values. In IBM Tivoli Monitoring V6.2, some columns that can never be NULL were changed to include a NOT NULL constraint. In

large tables this can save significant disk space. For DB2, this greatly reduced the disk requirements for indexes. The updated mode does not generate SQL to set columns to NOT NULL.

- If using DB2 for Linux, UNIX, and Windows and if the IBM Tivoli Monitoring V6.1 database is not migrated properly, the schema tool may produce SQL that fails. The tool may generate ALTER TABLE statements that cause a table not to fit into the table's page size.

Procedure

1. Create a new response file by copying the sample response file:
   - On Windows systems, copy *itm_install_dir*\TMAITM6\tdwschema.rsp.
   - On Linux and UNIX systems, copy *itm_install_dir*/arch/bin/tdwschema.rsp2.
2. Using an ASCII text editor, edit the response file as follows: **`KSY_PRODUCT_SELECT=updated`**. (See the description above.)

   You may also specify an output path via the **`KSY_SQL_OUTPUT_FILE_PATH=`***path* parameter, as explained above.
3. Using either the historical configuration interface within the Tivoli Enterprise Portal or the historical configuration CLI, make the desired changes to your site's historical configuration. If enabling historical collection for new attribute groups, configure but do not start historical collection. If collection is started, the warehouse proxy agent may attempt to create the database objects before you have a chance to generate, edit, and execute the SQL.
4. Ensure the Tivoli Enterprise Portal Server is started.
5. Run the schema publication tool script:
   - On Windows systems:
     ```
     tdwschema -rspfile response_file
     ```
   - On Linux and UNIX systems:
     ```
     tdwschema.sh -rspfile response_file
     ```

   where *response_file* is the name of the response file you edited in step 1. The SQL files for the products specified in the response file are generated and written to the directory indicated by the **`KSY_SQL_OUTPUT_FILE_PATH`** keyword (or to the current working directory, if no output directory is specified).
6. Make any necessary changes to the generated SQL files. For example, you might want to partition tables or assign tables to table spaces.

   Note: Do not change any of the following in the generated SQL files:
   - Table or view names
   - Table column names
   - Table column sizes or data types
7. Use the appropriate tools to run the SQL queries to create the warehouse tables, indexes, views, inserts, and functions for your relational database. Run the scripts as the Tivoli Data Warehouse user in this order:
   a. tdw_schema_table.sql
   b. tdw_schema_index.sql
   c. tdw_schema_view.sql
   d. tdw_schema_insert.sql
   e. tdw_schema_function.sql
8. Using either the historical configuration interface within the portal or the historical configuration CLI, start historical collection for the newly configured attribute groups.

Using the Schema Publication Tool for database compression

The Schema Publication Tool allows you to generate DDL scripts that can be used against a database capable of compression. Those scripts will alter the tables and indexes, including the log control tables. Executing those scripts will help you to reduce the amount of disk space used and also improve database performance. Reducing storage subsystem costs can result in substantial cost savings. The Schema Publication Tool generated scripts can be modified to accommodate local security concerns, point to a particular tablespace, or otherwise customize the generated DDL.

Notes:

1. The Summarization and Pruning Agent must be configured and have an available database connection.
2. At this time, compression on DB2 z/OS is not supported.
3. Compression increases CPU use by the database application.
4. When using DB V9.1 a reorganization is required for each table and index created with compression enabled, whether it is existing or new data. The reorganization creates the required compression dictionary.
5. DB2 V9.5 and greater automatically creates the required compression dictionary. A reorganization is required to compress existing data, otherwise only new data is compressed.
6. The Schema Publication tool uses only the compression setting stored in the Summarization and Pruning configuration file for all tables and indexes. This setting includes tables normally generated by the Warehouse Proxy Agent such as raw tables and associated control tables.
7. If the compression setting is turned off after table and indexes are compressed, the Schema Publication Tool will not generate the DDL to change them back to uncompressed mode.

Ensure a compression enabled database application server is installed and properly licensed. Summarization and Pruning Agent must be configured and an active database connection available. When using Oracle, compression is on a block basis (while on DB2 it is on the entire table). A larger block size could result in significantly larger compression. Block size is defined at database creation and cannot be changed, and block size needs to be a multiple of the underlying operating system block size.

The amount of storage reduction depends on a number of factors including the following consideration: the structure of the data makes an enormous difference. More compression is generated if there are repeated values in the data since all of the database applications use dictionary compression algorithms. The total amount of storage saved is proportional to the size of the table. However, the compression ratio is based on the data itself.

The compression settings and any warnings can be found in the Summarization and Pruning Agent log files and in the Warehouse Proxy Agent and Summarization and Pruning Agent's Tivoli Enterprise Portal configuration workspaces. If the Schema Publication Tool finds the DBMS does not support compression and DB compression is enabled, then the DB compression will be set to NO, a log entry made, and also the script will issue a message to that effect. All DDL will then be created without compression.

Complete the following steps to enable compression:

1. In the Summarization and Pruning Agent environment file, set the KSY_DB_COMPRESSION environment variable to Y. The default is N.

   For Windows systems, the environment file name and location is `*install dir*\TMAINT6\KSYENV`. For UNIX systems, the environment file name and location is *install dir*`/config/sys.ini`.
2. Update the required settings in the Schema Publication Response file and run the `tdwschema.bat` (Windows) or `tdwschema.sh` (UNIX) command. The response file property for this setting is: KSY_PRODUCT_SELECT = "updated".

Note: In the following situations there will be separate values and potentially conflicting options for the compression option:

- The Warehouse Proxy Agent and the Summarization and Pruning Agent have separate compression options and can be out of sync.
- If there are multiple Warehouse Proxy Agents, the values could be out of sync.

In all cases, the Schema Publication Tool uses only the Summarization and Pruning Agent setting. No mechanism is provided to coordinate the settings.

Chapter 20. Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows

Use the information and instructions in this chapter to implement a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows for the warehouse database. The following table lists the goals for creating a DB2 for Linux, UNIX, and Windows solution.

Table 78. Goals for creating a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows

Goal	Where to find information
Review your options, specific to a DB2 for Linux, UNIX, and Windows solution, for operating system platforms and communications between warehousing components.	"Supported components" on page 490
Install prerequisite software before implementing your Tivoli Data Warehouse solution.	"Prerequisite installation" on page 491
Understand how to use the instructions for implementing your Tivoli Data Warehouse solution.	"Implementing a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows" on page 492
Complete the steps for implementing your Tivoli Data Warehouse solution using DB2 for the data warehouse.	"Step 1: Create the Tivoli Data Warehouse database" on page 493 "Step 2: Install and configure communications for the Warehouse Proxy Agent" on page 499 "Step 3: Configure communications between the Tivoli Enterprise Portal Server and the data warehouse" on page 509 "Step 4: Install and configure communications for the Summarization and Pruning Agent" on page 513 "Step 5: Install and configure communications for the Tivoli Performance Analyzer" on page 514

Supported components

Figure 111 presents the options for a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows for the warehouse database. The diagram summarizes the supported operating system platforms for the various warehousing components, the supported database products, and the connections between components. For more specific information about supported operating systems and database products, including product names and versions, see "Hardware and software requirements" on page 138.

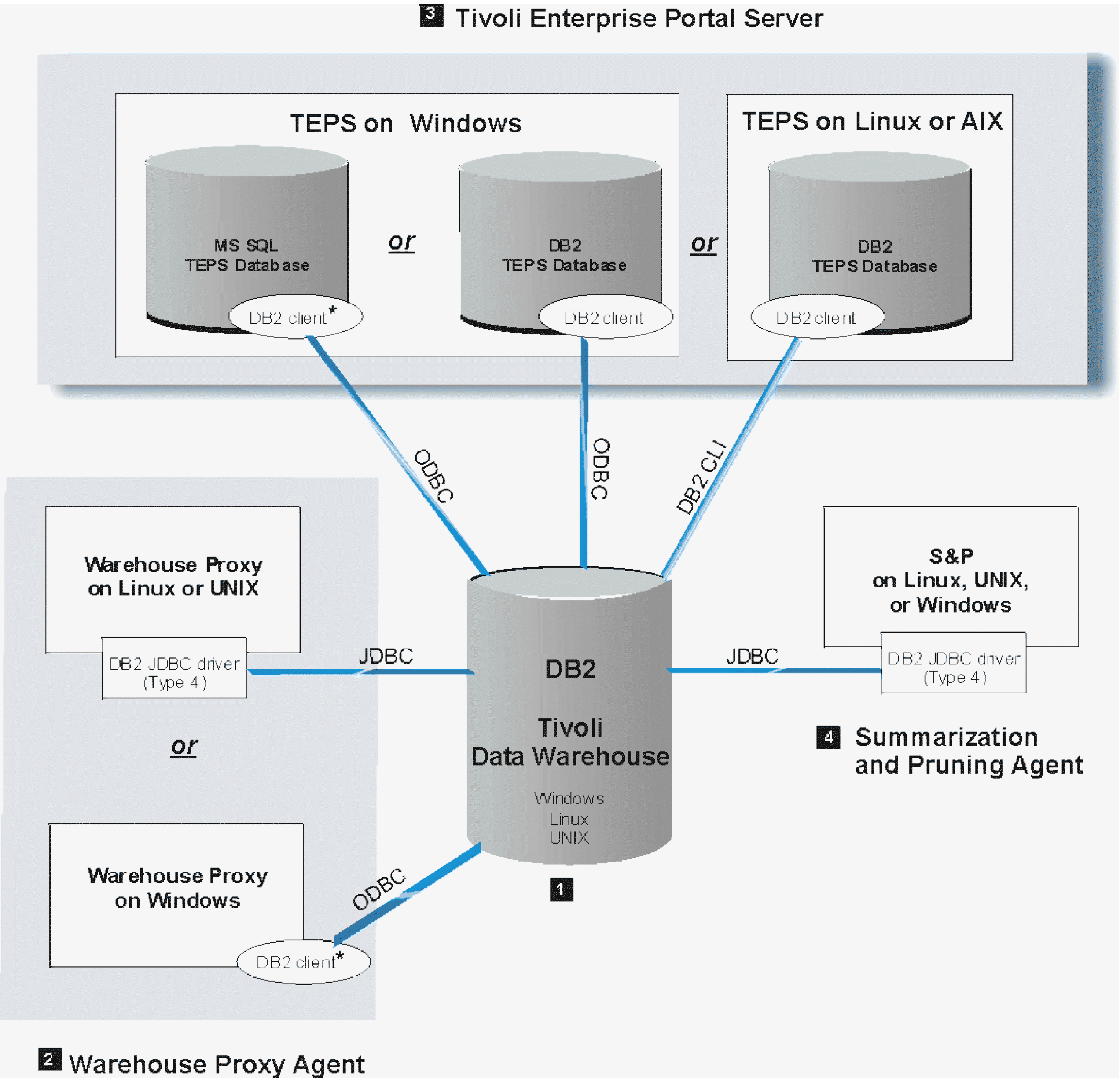


Figure 111. Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows

Note: An asterisk (*) next to a database client indicates that you must manually install the client if it does not already exist.

In the following discussion, numbered product components correspond to the numbers on the diagram.

1 Tivoli Data Warehouse on DB2 for Linux, UNIX, and Windows

A Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows can be installed on supported Windows, Linux, or any UNIX platform that is supported by DB2 for Linux, UNIX, and Windows. Ensure that the DB2 TCP/IP listeners are active in order to accept connections from a DB2 client or JDBC driver.

2 Warehouse Proxy Agent

A Warehouse Proxy Agent on Linux or AIX communicates with the warehouse database through a JDBC connection. Install a Type 4 driver (DB2 for Linux, UNIX, and Windows JDBC Universal Driver) on the computer where the Warehouse Proxy Agent is located.

A Warehouse Proxy Agent on Windows communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the DB2 for Linux, UNIX, and Windows client. If the Tivoli Data Warehouse is located on a remote computer, install a DB2 client on the local computer where the Warehouse Proxy Agent is located. Also, catalog the remote node and database on the local computer.

3 Tivoli Enterprise Portal Server

A Tivoli Enterprise Portal Server on Windows, Linux, or AIX can connect to a DB2 for Linux, UNIX, and Windows data warehouse through a DB2 for Linux, UNIX, and Windows client installed on the portal server. If the *portal server database* (designated as *TEPS database* in the diagram) uses DB2 for Linux, UNIX, and Windows, the DB2 client already exists. *Manually* install a DB2 for Linux, UNIX, and Windows client on the portal server only if the portal server database uses Microsoft SQL Server.

A portal server on Windows communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the DB2 for Linux, UNIX, and Windows client. Catalog the remote warehouse node and database on the portal server.

A portal server on Linux or AIX uses the DB2 for Linux, UNIX, and Windows CLI interface, a proprietary connection, to communicate with the warehouse database. If the warehouse database is installed on a different computer from the portal server, catalog the remote database and remote node on the portal server.

4 Summarization and Pruning Agent

The Summarization and Pruning Agent communicates with the warehouse database through a JDBC connection from any supported operating system. Install a DB2 for Linux, UNIX, and Windows Type 4 JDBC driver (*DB2 for Linux, UNIX, and Windows JDBC Universal Driver*) on the computer where the Summarization and Pruning Agent is located.

Prerequisite installation

Before you implement your Tivoli Data Warehouse solution, complete one or more hub installations, *excluding the warehousing components*. Include the following components in each hub installation:

- The hub Tivoli Enterprise Monitoring Server
- *(Optional)* One or more remote monitoring servers
- The Tivoli Enterprise Portal Server, including the prerequisite RDBMS for the portal server database (DB2 for Linux, UNIX, and Windows or Microsoft SQL Server)
- An IBM DB2 for Linux, UNIX, and Windows server on the computer where you will create the Tivoli Data Warehouse database. (The Tivoli Data Warehouse database can be shared in a multi-hub installation or dedicated to a single hub.)
- *(Optional)* A portal desktop client
- *(Optional)* Monitoring agents, and the application support for the monitoring agents

Note: The term *monitoring agent*, as used here, refers to agents that collect data directly from managed systems, not the Warehouse Proxy Agent or Summarization and Pruning Agent.

- *(Optional)* The Tivoli Performance Analyzer
- *(Optional)* Language packs for all languages other than English

See Table 79 for related information:

Table 79. Information topics related to installation of prerequisite software for a Tivoli Data Warehouse solution

Topic	Where to find information
Single and multiple hub installations	To understand the terminology related to single and multiple hub installations, see “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46.
Installation procedures for prerequisite components	The detailed instructions for installing the prerequisite components are described in Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207. See your database documentations for instructions on how to install a supported database server.
Supported RDBMS versions	For specific information about the supported database platforms for the portal server database and the Tivoli Data Warehouse, see “Hardware and software requirements” on page 138.

Implementing a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows

Use the instructions in the remainder of this chapter to implement a Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows for the data warehouse.

Assumptions

The implementation instructions are based on the following assumptions:

- You will create the Tivoli Data Warehouse database on a different computer from the Tivoli Enterprise Portal Server.
- You will create a single Tivoli Data Warehouse database, to be used either within a single hub installation or to be shared in a multi-hub installation. If you have multiple independent hub installations, repeat the implementation steps for each hub installation. (See “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46 for information about hub installations.)
- No assumption is made about where you will install the Warehouse Proxy Agent and Summarization and Pruning Agent. Either of these agents may be installed on the same computer as the Tivoli Data Warehouse or on a different computer.

Solution steps

To implement your Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows, complete the four major steps described in the remaining sections of this chapter, in the order listed:

1. Create the Tivoli Data Warehouse database.
2. Install and configure communications for the Warehouse Proxy Agent.
3. Configure communications between the Tivoli Enterprise Portal Server and the data warehouse.
4. Install and configure communications for the Summarization and Pruning Agent.

Each major step consists of a series of installation and configuration tasks, listed and described in a table. Use the step tables as a road map for implementing your solution. The step tables describe the tasks at a high level, account for variations among configuration options (such as which operating system is used for a component), and reference the appropriate sections for detailed implementation procedures. To implement your solution successfully:

- Perform the tasks in the order listed in the table.
- Do not skip a table to the procedures that follow it.

  Be aware that some of the implementation procedures referenced in a table are included in this chapter and some are documented elsewhere. In some cases, the task is described in the table, without referencing a separate procedure. Read and follow all instructions in the tables.

Step 1: Create the Tivoli Data Warehouse database

Complete the tasks described in the following table to create a Tivoli Data Warehouse database using DB2 for Linux, UNIX, and Windows and to make it accessible to clients.

Table 80. Tasks for creating the Tivoli Data Warehouse database

Task	Procedure
Create the Tivoli Data Warehouse database on one of the supported Windows, Linux, or UNIX operating systems. To comply with the assumptions described in the introduction to this chapter, create the warehouse database on a different computer from the Tivoli Enterprise Portal Server.	For guidance on planning the size and disk requirements for the warehouse database, see “Planning considerations for the Tivoli Data Warehouse” on page 469. For information about creating the warehouse database using DB2, see “Creating the warehouse database on DB2 for Linux, UNIX, and Windows” on page 494.
Create an operating system (OS) user account (user name and password) with administrator authority on the computer where the warehouse database is located. The warehousing components (portal server, Warehouse Proxy Agents, and Summarization and Pruning Agent) will use this OS user account to access the database. Create only one user account for all warehousing components to use. This user is referred to in this chapter as the *warehouse user*.	“Creating a warehouse user on Windows” on page 494 “Creating a warehouse user on Linux or UNIX” on page 495
(*Optional*) Restrict the authority of the warehouse user. Initially, the warehouse user is created with OS administrative authority. Use the referenced procedure if you want to limit the authority of the warehouse user to just those privileges required to access and use the data warehouse.	“Limiting the authority of the warehouse user” on page 495
(*Tivoli Data Warehouse on Linux or AIX only*) Activate the DB2 for Linux, UNIX, and Windows TCP/IP listeners on the DB2 server where the Tivoli Data Warehouse is installed. The TCP/IP listener processes on the DB2 for Linux, UNIX, and Windows server must be active in order to accept connections from a DB2 for Linux, UNIX, and Windows client or JDBC driver. The DB2 listeners are automatically activated on Windows systems. Perform the referenced procedure to activate the DB2 listeners if the warehouse database is located on a Linux or AIX system.	“Activating the DB2 listeners on a UNIX DB2 server” on page 497

Creating the warehouse database on DB2 for Linux, UNIX, and Windows

This section provides guidelines for creating the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows. For specific instructions on how to create a DB2 database, see the DB2 for Linux, UNIX, and Windows documentation or have a database administrator create the database for you.

When you create the warehouse database using DB2, follow these guidelines:

- Create the database with UTF-8 encoding.
- Create a name for the warehouse database, and an operating system (OS) user account (user name and password) that the warehousing components (portal server, Warehouse Proxy Agent, and Summarization and Pruning Agent) can use to access the data warehouse. In these instructions, this user account is referred to as the *warehouse user*.
- Consider using the default values shown in Table 81 for the warehouse name and warehouse user. The default values are used in the configuration procedures for connecting the warehousing components to the warehouse database. (For example, see the entries in Figure 113 on page 504.)

Table 81. Default values for Tivoli Data Warehouse parameters

Parameter	Default value
Tivoli Data Warehouse database name	WAREHOUS
User name	itmuser
User password	itmpswd1

- Give the warehouse user administrative authority to the database initially. After that, you can optionally limit the authority of the warehouse user to just the privileges required for interacting with the data warehouse. See the following sections for information about creating and limiting the authority of the warehouse user.
  - "Creating a warehouse user on Windows"
  - "Creating a warehouse user on Linux or UNIX" on page 495
  - "Limiting the authority of the warehouse user" on page 495
- For a Tivoli Data Warehouse on Linux or AIX, ensure that the DB2 for Linux, UNIX, and Windows server is configured for TCP/IP communications. See "Activating the DB2 listeners on a UNIX DB2 server" on page 497.

Creating a warehouse user on Windows

Complete the following steps on the computer where the warehouse database is installed to create a Windows OS user with Administrator authority:

1. Right-click the **My Computers** icon on the Windows desktop and click **Manage**.
2. In the navigation pane of the Computer Management window, expand **Local Users and Groups** by clicking on the plus sign (+).
3. Right-click the **Users** folder and click **New User**.
4. Type a user name and password in the **User Name** and **Password** fields. Confirm the password by typing it again in the **Confirm password** field.
5. Clear **User must change password at next logon**.
6. Click **Close**.
7. Click the **Groups** folder.
8. Double-click **Administrators** in the right pane of the window.
9. Click **Add** in the Administrator Properties window.
10. Locate the new user you created and select it.
11. Click **Add**.

12. Click **OK** and then **OK** again to close the Administrator Properties window.
13. Close the Computer Management window.

Creating a warehouse user on Linux or UNIX

Complete the following procedure on the computer where the warehouse database is installed to create a Linux or UNIX OS user with administrative authority to the warehouse:

- To create the user, follow the instructions in the documentation for the specific Linux or UNIX product and version that is installed on the computer where the warehouse database is located.
- To give this user administrative authority to the data warehouse, add the user to the DB2 for Linux, UNIX, and Windows SYSADM group. Run the following command to find the name of the SYSADM group:

  ```
  db2 get dbm cfg | grep SYSADM
  ```

  For example:

  ```
  db2 get dbm cfg | grep SYSADM
  SYSADM group name          (SYSADM_GROUP) = DB2GRP1
  ```

  In this example, the name of the DB2 for Linux, UNIX, and Windows SYSADM group is DB2GRP1. If you created an OS user named ITMUSER, add ITMUSER to DB2GRP1.

Limiting the authority of the warehouse user

If you do not want the warehouse user to have broad administrative authority, you can limit the authority of the warehouse user to just those privileges required for accessing and using the data warehouse. These more limited privileges include the authority to create and update tables, to insert or delete information from the tables, to create indexes for the tables, and to grant public authority to the tables.

Before you begin

The Tivoli Data Warehouse requires one bufferpool and three tablespaces to begin its operation. The bufferpool and tablespaces are created by the warehouse user before the Warehouse Proxy Agent starts, provided the warehouse user has administrative authority to the database. A warehouse user with limited authority cannot create the required bufferpool and tablespaces. Therefore, this procedure to limit the authority of the warehouse user includes steps to create the bufferpool and tablespaces in advance. Be sure to perform this procedure before the Warehouse Proxy Agent starts.

Use the script shown in Table 82 on page 496 to create the required bufferpool and tablespaces. Create the script on a computer from which you can connect to the Tivoli Data Warehouse database server. The name of the script is KHD_DB2_crt_BP_TBSP.sql.

Table 82. Script for creating required bufferpool and tablespaces for the Tivoli Data Warehouse

```
-- CREATE a Bufferpool of page size 8K
CREATE BUFFERPOOL ITMBUF8K IMMEDIATE  SIZE 250[1] PAGESIZE 8 K;

-- CREATE a Regular Tablespace using the 8K Bufferpool
CREATE REGULAR TABLESPACE ITMREG8K PAGESIZE 8 K
    MANAGED BY SYSTEM
    USING ('itmreg8k')[2] BUFFERPOOL ITMBUF8k;

-- CREATE a System tablespace using the 8K Bufferpool
CREATE SYSTEM TEMPORARY TABLESPACE ITMSYS8K PAGESIZE 8 K
    MANAGED BY SYSTEM
    USING ('itmsys8k')[2] BUFFERPOOL ITMBUF8k;

-- CREATE a User tablespace using the 8K Bufferpool

CREATE USER TEMPORARY  TABLESPACE ITMUSER8K PAGESIZE 8 K
    MANAGED BY SYSTEM
    USING ('itmuser8k')[2] BUFFERPOOL ITMBUF8k;
```

Notes:

1. SIZE is the number of 8K pages to allocate for the bufferpool. If there is sufficient memory, performance can be improved by increasing this number.
2. A fully qualified path can be specified here. As shown this will be created in a default directory.
3. IBM Tivoli Monitoring creates SMS tablespaces, which cannot be extended across multiple drives. This can be customized by the database administrator.

Procedure

To limit the authority of the warehouse user, complete the following steps:

1. Connect to the data warehouse with db2admin privileges:

   ```
   db2 connect to warehouse user db2admin using password
   ```

   where *warehouse* is the name of the Warehouse database, *db2admin* is the DB2 for Linux, UNIX, and Windows administrator ID, and *password* is the password of the *db2admin* user ID. The user ID must be a DB2 user with SYSADM authority
2. Change to the directory where the KHD_DB2_crt_BP_TBSP.sql script is located.
3. Run the script to create the required bufferpool and tablespaces:

   ```
   db2 -stvf KHD_DB2_crt_BP_TBSP.sql
   ```
4. Remove administrative privileges from the warehouse user (OS user) that you created when you created the warehouse database:
   - On Windows, remove the warehouse user from the Administrator group.
   - On Linux or UNIX, remove the warehouse user from the SYSADM group to which it was assigned (for example, DB2GRP1). (See "Creating a warehouse user on Linux or UNIX" on page 495.)
5. Grant these database authorities to the warehouse user:

   CONNECT
   CREATETAB

   Note: The user might already implicitly have CONNECT and CREATEAB authority through the PUBLIC role.

   USE OF TABLESPACE

   CONNECT authority grants the user access to the database. CREATETAB authority allows the user to create tables. The authority to drop tables, alter tables, create and drop indexes for the tables, insert, delete, or update data in the tables, are all implicitly granted. USE OF TABLESPACE grants the user authority to use particular tablespaces, in this case ITMREG8K.

To grant these authorities, you can use either the DB2 Control Center (Database Authorities window) or the command-line interface. If you use the command-line interface, run commands similar to the following. In this example, the name of the warehouse user is `itmuser`.

```
db2 "GRANT CONNECT ON DATABASE TO USER itmuser"
db2 "GRANT CREATETAB ON DATABASE TO USER itmuser"
db2 "GRANT USE OF TABLESPACE ITMREG8K TO itmuser"
```

Perform the additional steps if your security policy prohibits use of the CREATTAB authority:

1. Use the schema tool to generate the DDL that create the database objects.

   Note: You should create the historical collections that you want first and then configure the Summarization and Pruning Agent so that you can use the schema tool's configured mode.
2. Execute the generated scripts as described in Chapter 19, “Schema Publication Tool,” on page 483. Run the scripts as the Tivoli Data Warehouse user so that the Tivoli Data Warehouse user has sufficient privileges on the tables.
3. Revoke the CREATETAB privilege from the Tivoli Data Warehouse user and from the PUBLIC role.

Setting database and instance configuration values

The default values for many of the configuration values for the DB2 for Linux, UNIX, and Windows database and the DB2 for Linux, UNIX, and Windows instance may not be acceptable for a production environment. Following are a few of the values you should consider adjusting. Consult the DB2 for Linux, UNIX, and Windows documentation or your DB2 database administrator for other adjustments that can improve performance.

LOGPRIMARY
Number of primary transaction logs. In a production environment, this value should be larger than the default value.

LOGSECOND
Number of secondary log files. In a production environment, this value should be larger than the default value.

NEWLOGPATH
Location of transaction logs. To improve performance, the logs should be placed on a separate physical disk from the tables and indexes.

LOGFILSIZ
Size of transaction log files. In a production environment, this value should be larger than the default value.

LOCKTIMEOUT
Defaults to infinite wait, which can cause Warehouse Proxy and Summarization and Pruning processing to appear to be locked up. This should be set to a reasonable value such as 120 seconds.

Activating the DB2 listeners on a UNIX DB2 server

The TCP/IP listener processes on the DB2 for Linux, UNIX, and Windows server where the Tivoli Data Warehouse database is installed must be active in order to accept connections from a DB2 for Linux, UNIX, and Windows client or a JDBC Type 4 driver (*DB2 for Linux, UNIX, and Windows JDBC Universal Driver*). On a Windows system, the DB2 listeners are automatically activated. Run the following commands on a UNIX system where the Tivoli Data Warehouse database is installed to activate the DB2 for Linux, UNIX, and Windows listeners:

```
db2set -i instance_name DB2COMM=tcpip
db2 update dbm cfg using SVCENAME port_number
db2stop
db2start
```

where *instance_name* is the name of the instance in which you created the warehouse database and *port_number* is the listening port for the instance. (The port number is specified in the file /etc/services.) For example:

```
db2set -i db2inst1 DB2COMM=tcpip
db2 update dbm cfg using SVCENAME 60000
db2stop
db2start
```

Step 2: Install and configure communications for the Warehouse Proxy Agent

You can install one or more Warehouse Proxy Agents to collect and send historical data to the Tivoli Data Warehouse database. Complete the tasks described in the following table, in the order listed, to install and configure each Warehouse Proxy Agent.

Table 83. Tasks for installing and configuring communications for the Warehouse Proxy Agent

Task	Procedure
Install one or more Warehouse Proxy Agents. If you want to install a Summarization and Pruning Agent on the same computer as one of the Warehouse Proxy Agents, use the referenced procedures to install both agents at the same time. If you are installing more than one Warehouse Proxy Agent, each agent must be installed on a separate computer. The installation procedure for Windows includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (*Configuring the monitoring agent*) and an X11 GUI is required to configure the agent. Alternatively, you can run the following command to utilize an X terminal emulation program (such as Cygwin) that is running on another computer: `export DISPLAY=my_windows_pc_IP_addr:0.0` where *my_windows_pc_IP_addr* is the IP address of a computer that is running an X terminal emulation program. See the information at right. Be sure to perform all referenced installation and configuration procedures. **Note for sites setting up autonomous operation::** The installation procedure includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring Server. On Windows operating systems, if you want to run the Warehouse Proxy Agent without a connection to the hub, accept the defaults for the connection information, but specify a nonvalid name for the monitoring server. On UNIX and Linux operating systems, check **No TEMS** on the **TEMS Connection** tab of the configuration window.	To install a Warehouse Proxy Agent on Windows, complete the procedure “Windows: Installing a monitoring agent” on page 253. To install a Warehouse Proxy Agent on Linux or AIX, complete the procedure “Linux or UNIX: Installing a monitoring agent” on page 259, including the following subsections: • *Installing the monitoring agent* • *Configuring the monitoring agent* • *Changing the file permissions for agents* (if you used a non-root user to install the Warehouse Proxy) *Do not complete the procedure for starting the agent*.
(*Warehouse Proxy Agent on Windows only*) • Install a DB2 for Linux, UNIX, and Windows client on the computer where the Warehouse Proxy Agent is installed if *both* of the following statements are true: – The Warehouse Proxy is installed on Windows, and – The Warehouse Proxy needs to connect to a remote data warehouse. • Catalog the remote data warehouse on the Windows computer where you installed the DB2 for Linux, UNIX, and Windows client. You must perform this step before configuring an ODBC data source. (See the next row.) • Set the following system variable on the computer where the Warehouse Proxy Agent is installed. Restart the computer after setting the variable. DB2CODEPAGE=1208 Set the environment variable whether or not the warehouse database is local or remote.	See the DB2 for Linux, UNIX, and Windows documentation for instructions on how to install a DB2 for Linux, UNIX, and Windows client. To catalog a remote data warehouse, see “Cataloging a remote data warehouse” on page 500.

Table 83. Tasks for installing and configuring communications for the Warehouse Proxy Agent (continued)

Task	Procedure
(*Warehouse Proxy Agent on Windows only*) On the computer where the Warehouse Proxy Agent is installed, configure an ODBC data source for the data warehouse. Perform this procedure whether or not the Warehouse Proxy Agent and the warehouse database are installed on the same computer.	"Configuring an ODBC data source for a DB2 data warehouse" on page 501
(*Warehouse Proxy Agent on Linux or AIX only*) If the data warehouse is located on a remote computer, copy the *DB2 for Linux, UNIX, and Windows JDBC Universal Driver* (Type 4 driver) JAR files, included with the DB2 for Linux, UNIX, and Windows product installation, to the local computer where the Warehouse Proxy Agent is installed. You can copy the files to any directory on the local computer.	The Type 4 driver file names and locations are as follows: *db2installdir*/java/db2jcc.jar *db2installdir*/java/db2jcc_license_cu.jar where *db2installdir* is the directory where DB2 for Linux, UNIX, and Windows was installed. The default DB2 for Linux, UNIX, and Windows Version 9 installation directory is as follows: • On AIX: /usr/opt/db2_09_01 • On Linux: /opt/IBM/db2/V9.1
Configure the Warehouse Proxy Agent to connect to the data warehouse. Perform this procedure whether or not the Warehouse Proxy Agent and the warehouse database are installed on the same computer.	For a Warehouse Proxy Agent on Windows, see "Configuring a Warehouse Proxy Agent on Windows (ODBC connection)" on page 502. For a Warehouse Proxy Agent on Linux or AIX, see "Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)" on page 505.
If you are installing more than one Warehouse Proxy Agent within the same hub monitoring server installation, associate each Warehouse Proxy Agent with a subset of monitoring servers (hub or remote) within the installation. Each Warehouse Proxy Agent receives data from the monitoring agents that report to the monitoring servers on the list. Use the environment variable KHD_WAREHOUSE_TEMS_LIST to specify a list of monitoring servers to associate with a Warehouse Proxy Agent.	For instructions about installing and configuring multiple Warehouse Proxy Agents within a single hub monitoring server installation, see "Installing and configuring multiple Warehouse Proxy Agents" on page 608.
(*Optional*) Customize the configuration of the Warehouse Proxy Agent for tuning performance.	"Tuning the performance of the Warehouse Proxy" on page 617
Start the Warehouse Proxy Agent.	"Starting the Warehouse Proxy" on page 508

Cataloging a remote data warehouse

Perform this procedure on a computer where a DB2 for Linux, UNIX, and Windows client is installed to enable communication between the client and a *remote* DB2 for Linux, UNIX, and Windows server where the data warehouse is installed. For example, use this procedure to set up communication to a remote DB2 data warehouse server from:

- The DB2 for Linux, UNIX, and Windows client on the computer where the Tivoli Enterprise Portal Server is installed (on any platform)
- The DB2 client on a Windows computer where a Warehouse Proxy Agent installed

Do *not* perform this procedure on the computer where the data warehouse (DB2 server) is installed or on a computer where there is no DB2 client (for example, on a computer where a Type 4 DB2 for Linux, UNIX, and Windows JDBC driver is used to communicate with the remote DB2 server).

Complete the following steps on the computer where the DB2 for Linux, UNIX, and Windows client is installed (the *local* computer):

1. Catalog the remote TCP/IP node where the warehouse database is installed:

   ```
   db2 catalog tcpip node node_name remote host_name server port
   db2 terminate
   ```

   where the indicated variables identify the location and port of the remote DB2 for Linux, UNIX, and Windows server. For *host_name*, specify the host name or IP address. The default port for a DB2 server is 60000. For example:

   ```
   db2 catalog tcpip node amsc2424 remote 8.53.36.240 server 60000
   db2 terminate
   ```

2. Catalog the remote Tivoli Data Warehouse database:

   ```
   db2 catalog db db_name as db_alias at node node_name
   db2 terminate
   ```

   where

   db_name is the name of the remote warehouse database.

   db_alias is the nickname or alias used to identify the remote warehouse database on the local computer. The local alias for the warehouse database must match the name that you specify in the configuration procedure for the portal server, Warehouse Proxy Agent, or Summarization and Pruning Agent.

   node_name is the name of the node where the warehouse database is located.

   Example:

   ```
   db2 catalog db WAREHOUS as WAREHOUS at node amsc2424
   db2 terminate
   ```

3. Test the connection to the remote warehouse database:

   ```
   db2 connect to db_alias user user_name using user_password
   ```

   where:

   db_alias
   Is the nickname or alias used to identify the remote warehouse database on the local computer.

   user_name
   Is the user ID that the local DB2 client uses to access the warehouse database.

   user_password
   Is the password for that *user_name*.

   These values must match the values that you specify in the configuration procedures for the portal server, Warehouse Proxy Agent, or Summarization and Pruning Agent.

   Example:

   ```
   db2 connect to WAREHOUS user itmuser using itmpswd1
   ```

Configuring an ODBC data source for a DB2 data warehouse

A DB2 for Linux, UNIX, and Windows client on Windows requires an ODBC connection to the data warehouse. For the Warehouse Proxy Agent, you must configure the ODBC connection manually.

Before you begin

- If the warehouse database is remote from the Warehouse Proxy Agent, catalog the remote database before you configure the ODBC data source. See "Cataloging a remote data warehouse" on page 500. The ODBC connection does not work if the remote database has not been cataloged prior to performing this procedure.

- This procedure uses default values for the data source name, warehouse alias, and warehouse user ID. (Default values are used in configuration procedures for warehousing components.) Substitute different values if you do not want to use the default values.

Procedure

Complete the following procedure to set up an ODBC connection for a Warehouse Proxy Agent on Windows to a local or remote Tivoli Data Warehouse.

1. On the computer where the Warehouse Proxy Agent is installed, open the Control Panel.
2. Click **Administrative Tools → Data Sources (ODBC)**
3. Click **Add** in the **System DSN** tab in the ODBC Data Source Administrator window.
4. Select **IBM DB2 ODBC DRIVER** from the list.
5. Click **Finish**.
6. In the ODBC DB2 Driver - Add window, perform the following steps:
   a. Enter `ITM Warehouse` in **Data source name**.
   b. Enter `Warehous` in **Database Alias**.

      If the Tivoli Data Warehouse is located on a remote computer, ensure that the database alias matches the alias that you used when cataloging the remote data warehouse. See "Cataloging a remote data warehouse" on page 500.

      If local, ensure that the database alias matches the name used for the warehouse database.
   c. Click **OK**.
7. Test the ODBC database connection before continuing:
   a. In the ODBC Data Source Administrator window, select **ITM Warehouse**.
   b. Click **Configure**.
   c. In the CLI/ODBC Settings - ITM Warehouse window, you see the data source name, **ITM Warehouse**.
   d. Enter `ITMUser` for the **User ID**.
   e. Type a password for the user in the **Password** field. The default password is `itmpswd1`.
   f. Click **Connect**.
   g. A **Connection test successful** message is displayed.
   h. Click **OK**.
   i. Click **OK** to close the window.

Configuring a Warehouse Proxy Agent on Windows (ODBC connection)

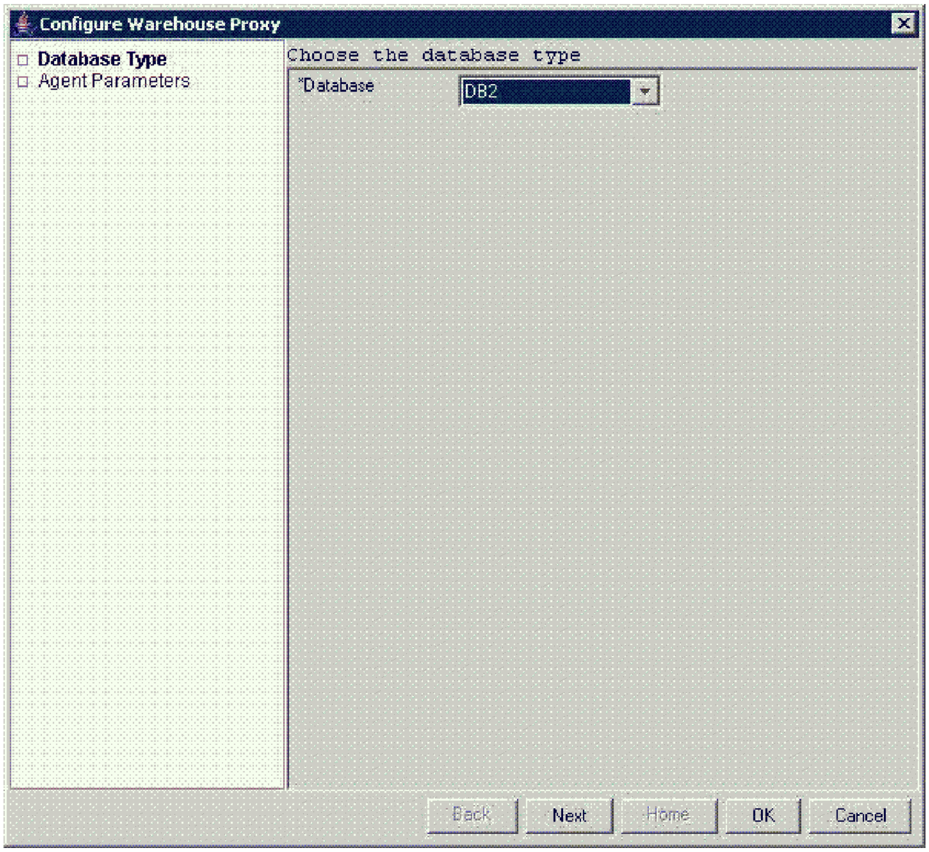
Use this procedure to configure a Warehouse Proxy Agent on Windows to connect to a DB2 for Linux, UNIX, and Windows data warehouse:

1. Log on to the Windows system where the Warehouse Proxy Agent is installed and begin the configuration:
   a. Click **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Warehouse Proxy** and click **Configure Using Defaults**.

      Click **Reconfigure** if the Warehouse Proxy is installed on the same computer as the portal server.
   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Warehouse Proxy: Agent Advanced Configuration) contain the settings for the connection between the Warehouse Proxy Agent and the hub monitoring server. These settings were specified when the Warehouse Proxy Agent was installed. Click **OK** on each window to accept the settings.

3. Select **DB2** from the list of selectable databases (see Figure 112), and click **Next**.

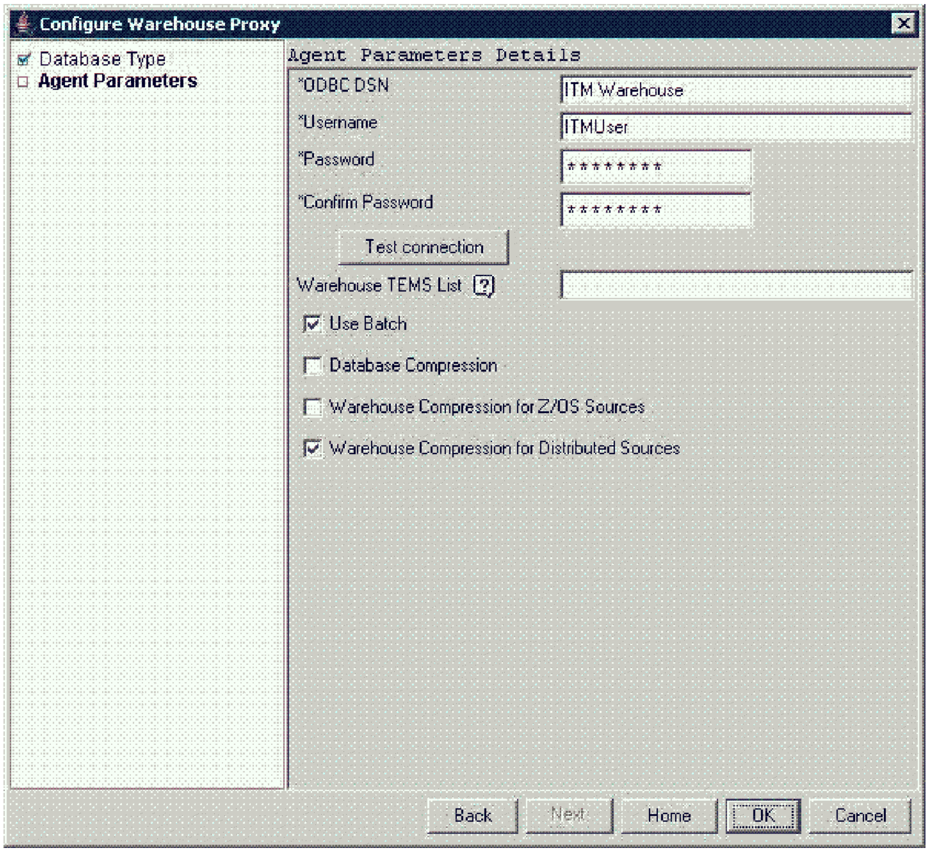


Choose the database type

*Database DB2

Figure 112. Warehouse Proxy Database Selection screen

The following configuration window is displayed.


Configure Warehouse Proxy

☑ Database Type
□ Agent Parameters

Agent Parameters Details

*ODBC DSN: ITM Warehouse
*Username: ITMUser
*Password: ********
*Confirm Password: ********

Warehouse TEMS List:

☑ Use Batch
☐ Database Compression
☐ Warehouse Compression for Z/OS Sources
☑ Warehouse Compression for Distributed Sources

Figure 113. Configure DB2 Data Source for Warehouse Proxy window

4. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in Table 84.

   Note: The values for the data source name, database name, and database user ID and password must match the values that you used when configuring an ODBC connection for the Warehouse Proxy Agent. (See “Configuring an ODBC data source for a DB2 data warehouse” on page 501.)

Table 84. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.
Username	ITMUser	The name of the user that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Confirm Password	itmpswd1	Confirm the password by entering it again.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password. **Note:** Test Connection is not available if configuring remotely from the Tivoli Enterprise Portal.

Table 84. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows (continued)

Field	Default value	Description
Warehouse TEMS List		Environment variable containing a space delimited list of TEMS names, which are given during the configuration of HTEMS or RTEMS. A TEMS name in this field indicates that all the agents connected to this TEMS will have their historical data sent to this Warehouse Proxy Agent. This variable is used when the ITM environment contains multiple Warehouse Proxy Agents and the workload has to be balanced using specific Warehouse Proxy Agents.
Use Batch		Batch inserts introduced ITM V6.2.2 fix pack 2, can greatly increase the data insertion rate of the Warehouse Proxy Agent. This is especially true if the proxy and the warehouse are located on different hosts. Batch inserts are supported for ODBC warehouse connections. Using batch inserts is recommended in all configurations, but they will place increased load on the data warehouse.
Database Compression		If the database compression mode is supported by the database, the Warehouse Proxy Agent will create all tables and indexes in the Tivoli Data Warehouse with compression enabled. This option reduces the storage costs of the Tivoli Data Warehouse.
Warehouse Compression for Z/OS Sources		Select this option for the Warehouse Proxy server to allow clients installed on Z/OS machines to send compressed data.
Warehouse Compression for Distributed Sources		Select this option for the Warehouse Proxy server to allow clients installed on distributed machines (Linux/UNIX, Windows) to send compressed data.

5. Click **OK**.

Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)

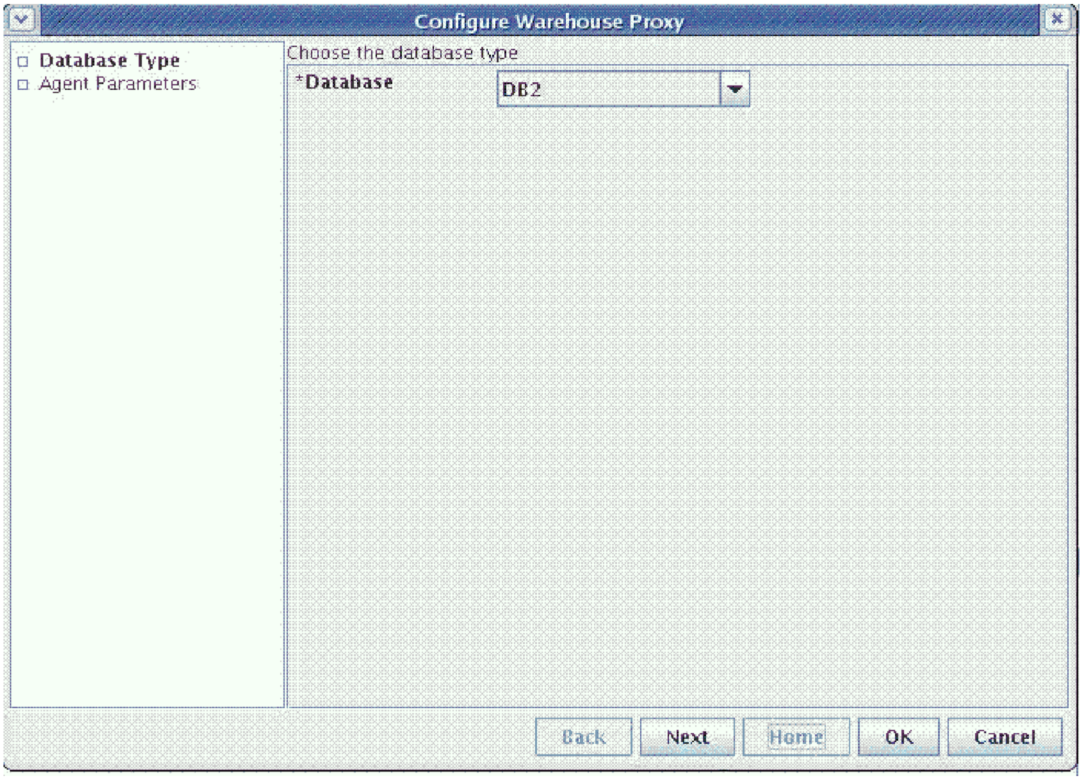
Use this procedure to configure a Warehouse Proxy Agent on Linux or UNIX to connect to a DB2 for Linux, UNIX, and Windows Tivoli Data Warehouse on any operating system:

1. Log on to the computer where the Warehouse Proxy Agent is installed and begin the configuration.
   a. Change to the *install_dir*/bin directory and run the following command:
   ```
   ./itmcmd manage [-h install_dir]
   ```
   where *install_dir* is the installation directory for IBM Tivoli Monitoring. The default installation directory is /opt/IBM/ITM.

   The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Warehouse Proxy** and click **Configure**.

   The Configure Warehouse Proxy window is displayed.

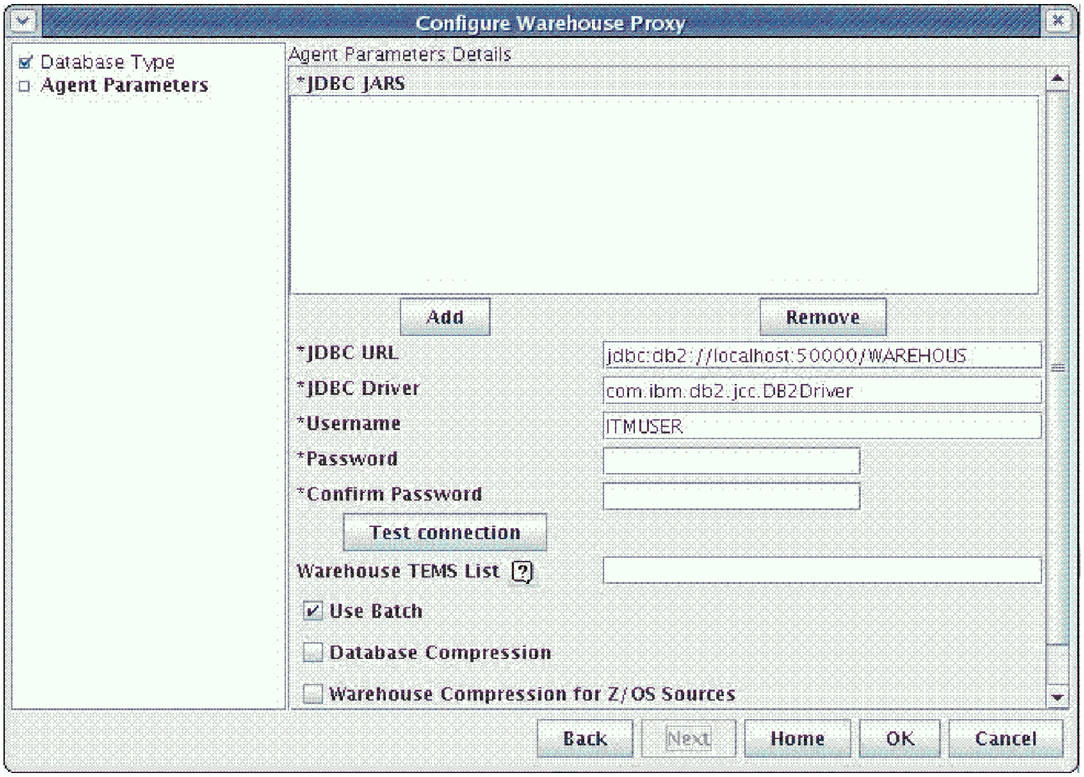


Choose the database type

*Database: DB2

Figure 114. Configure Warehouse Proxy window (Database Type)

2. Select **DB2** from the list of selectable databases, and click **Next**. The following configuration window is displayed.



*JDBC JARS

*JDBC URL: jdbc:db2://localhost:50000/WAREHOUS

*JDBC Driver: com.ibm.db2.jcc.DB2Driver

*Username: ITMUSER

*Password:

*Confirm Password:

Warehouse TEMS List ?:

☑ Use Batch

□ Database Compression

□ Warehouse Compression for Z/OS Sources

Figure 115. Configure Warehouse Proxy window (Agent Parameters)

3. Review the settings for the connection between the Warehouse Proxy Agent and the hub monitoring server. Correct the settings if necessary.

   The Warehouse Proxy Agent must use the same protocols used by the application agents and by the hub monitoring. If the proxy agent does not have the same protocol as the hub monitoring server, it cannot register with the hub. If the proxy does not have the same protocol as the application agents, then the application agents cannot communicate with the proxy when they to try create a route to it.
4. Add the names and directory locations of the JDBC JARS files to the **JDBC JARS** list box:
   a. Click **Add** to display the file browser window. Navigate to the location of the JARS files on this computer and select the following driver files:

      ```
      db2jcc.jar
      db2jcc_license_cu.jar
      ```
   b. Click **OK** to close the browser window and add the JDBC JARS files to the list.

   If you need to delete an entry from the list, select the entry and click **Remove**.
5. Change the default value displayed in the **JDBC URL** field if it is not correct. The default Tivoli Data Warehouse URL for IBM DB2 for Linux, UNIX, and Windows is as follows:

   ```
   jdbc:db2://localhost:50000/WAREHOUS
   ```
   - If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of `localhost`.
   - Change the port number if it is different.
   - If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name. (See “Creating the warehouse database on DB2 for Linux, UNIX, and Windows” on page 494.)

Note: When specifying the DB2 for Linux, UNIX, and Windows database name on Linux and UNIX, case is ignored. In other words, it makes no difference whether you provide the database name in lowercase or uppercase.

6. Verify the JDBC driver name, which is displayed in the **JDBC Driver** field. (Note that the **JDBC Driver** field displays the *driver name*, in contrast to the *JDBC JARS* that are listed in the **JDBC JARS** field.)

   The default DB2 JDBC Driver is:

   com.ibm.db2.jcc.DB2Driver
7. If necessary, change the entries in the **Username** and **Password** fields to match the user name and password that were created for the Tivoli Data Warehouse. (See "Creating the warehouse database on DB2 for Linux, UNIX, and Windows" on page 494.) The default user name is `itmuser` and the default password is `itmpswd1`.
8. Click **Test connection** to ensure you can communicate with the Tivoli Data Warehouse database.

   Note: Test Connection is not available if configuring remotely from the Tivoli Enterprise Portal.
9. A Tivoli Enterprise Monitoring Server name in the **Warehouse TEMS List** field indicates that all the agents connected to this Tivoli Enterprise Monitoring Server will have its historical data sent to this Warehouse Proxy Agent. This variable is used when the IBM Tivoli Monitoring environment contains multiple Warehouse Proxy Agents and the workload has to be balanced using specific Warehouse Proxy Agents.
10. Select the **Use Batch** check box if you want the Warehouse Proxy Agent to submit multiple execute statements to the Tivoli Data Warehouse database for processing as a batch.

    In some situations, such as crossing a network, sending multiple statements as a unit is more efficient than sending each statement separately. Batch processing is one of the features provided with the JDBC 2.0 API.
11. Select the **Database Compression** check box for the Warehouse Proxy Agent to create all tables and indexes in the Tivoli Data Warehouse with compression enabled, if the database compression mode is supported by the database. This option reduces the storage costs of the Tivoli Data Warehouse.
12. Select the **Warehouse Compression for Z/OS Sources** check box for the Warehouse Proxy server to allow clients installed on Z/OS machines to send compressed data.
13. Click **OK** to save your settings and close the window.

Starting the Warehouse Proxy

- To start the Warehouse Proxy Agent from the Manage Tivoli Enterprise Monitoring Services window, right-click **Warehouse Proxy** and select **Start**.
- (*Linux or AIX only*) To start the Warehouse Proxy Agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is /opt/IBM/ITM.

  ```
  ./itmcmd agent start hd
  ```

  where `hd` is the product code for the Warehouse Proxy Agent.

Step 3: Configure communications between the Tivoli Enterprise Portal Server and the data warehouse

Complete the tasks described in the following table, in the order listed, to configure communications between the portal server and the data warehouse.

Table 85. Tasks for configuring communications between the portal server and a DB2 for Linux, UNIX, and Windows data warehouse

Task	Procedure
(*Portal server on Windows only*) If the *portal server database* was created on Microsoft SQL Server, install a DB2 for Linux, UNIX, and Windows database client on the portal server. If the portal server database was created on DB2 for Linux, UNIX, and Windows, the DB2 client already exists on the portal server.	See the DB2 for Linux, UNIX, and Windows documentation for instructions on how to install a DB2 for Linux, UNIX, and Windows client.
On the computer where the portal server is installed, catalog the remote data warehouse. You must perform this step before configuring the portal server to connect to the data warehouse. (See the next row.) Cataloging the remote data warehouse enables communications between the DB2 for Linux, UNIX, and Windows client on the portal server and the remote DB2 for Linux, UNIX, and Windows server where the data warehouse is located. Complete this task regardless of which platforms are used by the portal server or the data warehouse.	"Cataloging a remote data warehouse" on page 500
Configure the portal server to connect to the data warehouse. The configuration procedure on Windows automatically configures an ODBC connection to the data warehouse.	For a portal server on Windows, see "Configuring a Windows portal server (ODBC connection)." For a portal server on Linux or AIX, see "Configuring a Linux or AIX portal server (DB2 for Linux, UNIX, and Windows CLI connection)" on page 511.
Restart the portal server.	"Starting the portal server" on page 512
Test the connection between the portal server and the Tivoli Data Warehouse by creating a customized query in the Tivoli Enterprise Portal.	"Testing the connection between the portal server and the Tivoli Data Warehouse" on page 614

Configuring a Windows portal server (ODBC connection)

The procedure described in this section uses the Manage Tivoli Enterprise Monitoring Services window to configure an ODBC connection between a Windows portal server and the data warehouse. You do not need to configure the ODBC data source through the Control Panel in Windows.

Before you begin

Catalog the remote warehouse database before you configure the Windows portal server to connect to the database. See "Cataloging a remote data warehouse" on page 500. The ODBC connection does not work if the remote database has not been cataloged prior to performing this procedure.

Procedure

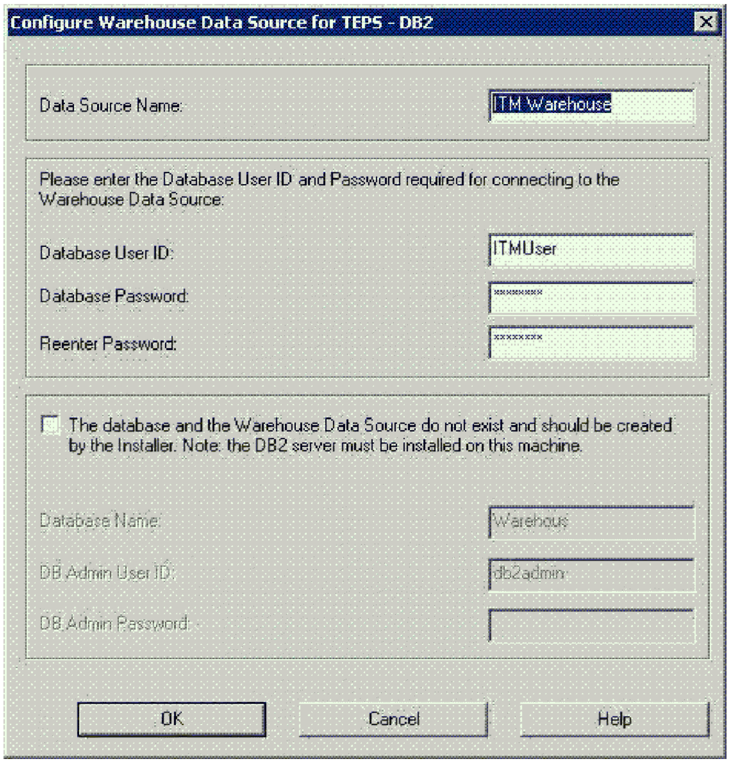
Complete the following procedure to configure a portal server on Windows to connect to a DB2 for Linux, UNIX, and Windows data warehouse:

1. Log on to the Windows system where the portal server is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Tivoli Enterprise Portal Server** and click **Reconfigure**.
2. The next two windows (entitled TEP Server Configuration) contain the settings for the connection between the portal server and the hub monitoring server. These settings were specified when the portal server was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to reconfigure the warehouse information for the Tivoli Enterprise Portal Server.
4. Select **DB2** from the list of databases and click **OK**.

   The following configuration window is displayed.



Configure Warehouse Data Source for TEPS - DB2

Data Source Name: ITM Warehouse

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

Database User ID: ITMUser

Database Password: ********

Reenter Password: ********

□ The database and the Warehouse Data Source do not exist and should be created by the Installer. Note: the DB2 server must be installed on this machine.

Database Name: Warehous

DB Admin User ID: db2admin

DB Admin Password:

Figure 116. Configure DB2 Data Source for Warehouse window

5. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in the following table.

Table 86. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows

Field	Default value	Description
Data Source Name	ITM Warehouse	The name of the data source.
Database User ID	ITMUser	The name of the Windows OS user that the portal server will use to access the Tivoli Data Warehouse database.
Database Password	itmpswd1	The password for the Windows user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Reenter Password	itmpswd1	Confirm the password by entering it again.

6. Click **OK**.

Configuring a Linux or AIX portal server (DB2 for Linux, UNIX, and Windows CLI connection)

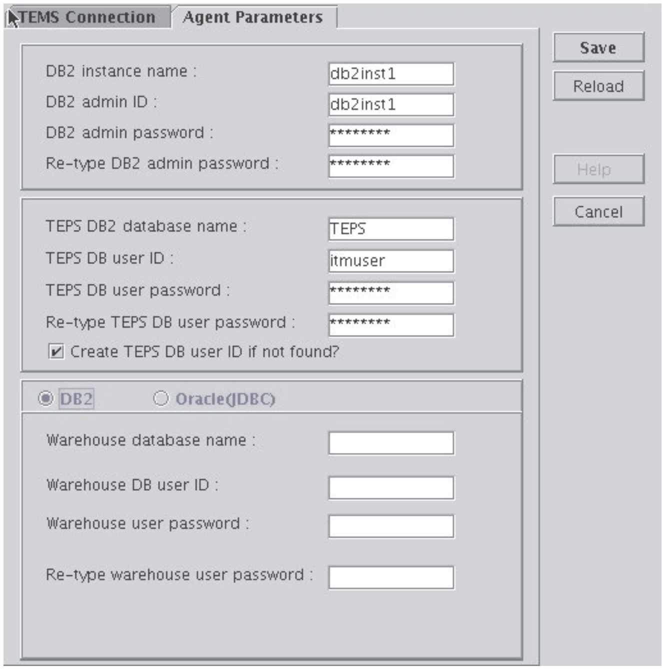
Use this procedure to configure a portal server on Linux or AIX to connect to a DB2 for Linux, UNIX, and Windows Tivoli Data Warehouse on any operating system.

1. Log on to the computer where the Tivoli Enterprise Portal Server is installed and begin the configuration.
   a. Change to the `install_dir/bin` directory and run the following command:
      ```
      ./itmcmd manage [-h install_dir]
      ```
      where *install_dir* is the installation directory for IBM Tivoli Monitoring. The default installation directory is /opt/IBM/ITM.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Tivoli Enterprise Portal Server** and click **Configure**.

      The Configure Tivoli Enterprise Portal Server window is displayed.
2. On the **TEMS Connection** tab, review the settings for the connection between the portal server and the hub monitoring server. These settings were specified when the portal server was installed.
3. Click the **Agent Parameters** tab.
4. Select the **DB2** radio button.

   The fields for configuring the connection to a DB2 for Linux, UNIX, and Windows data warehouse are displayed at the bottom of the window.



DB2 instance name : db2inst1
DB2 admin ID : db2inst1
DB2 admin password : ********
Re-type DB2 admin password : ********

TEPS DB2 database name : TEPS
TEPS DB user ID : itmuser
TEPS DB user password : ********
Re-type TEPS DB user password : ********
☑ Create TEPS DB user ID if not found?

● DB2 ○ Oracle(JDBC)
Warehouse database name :
Warehouse DB user ID :
Warehouse user password :
Re-type warehouse user password :

Figure 117. Configuring the connection to a DB2 for Linux, UNIX, and Windows data warehouse

5. Enter information in the fields described in the following table:

Table 87. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows

Field	Default value	Description
Warehouse database name	WAREHOUS	The name of the Tivoli Data Warehouse database.
Warehouse database user ID	itmuser	The login name of the database user that the portal server will use to access the Tivoli Data Warehouse database.

Table 87. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows (continued)

Field	Default value	Description
Warehouse user password	itmpswd1	The password for the database login user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Re-type warehouse user password	itmpswd1	Confirm the password by entering it again.

6. Click **Save** to save your settings and close the window.

Starting the portal server

Use the following steps to start the portal server:

- To start the portal server from the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server** and select **Start**.
- (*Linux or AIX only*) To start the portal server from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is /opt/IBM/ITM.

  ```
  ./itmcmd agent start cq
  ```

  where cq is the product code for the portal server.

Step 4: Install and configure communications for the Summarization and Pruning Agent

Complete the tasks described in the following table, in the order listed, to install and configure the Summarization and Pruning Agent.

Table 88. Tasks for installing and configuring communications for the Summarization and Pruning Agent

Task	Procedure
Install the Summarization and Pruning Agent if you have not already installed it. For best performance, install the Summarization and Pruning Agent on the same computer as the data warehouse. The installation procedure for Windows includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (*Configuring the monitoring agent*). See the information at right. Be sure to perform all referenced installation and configuration procedures. **Note**: The Summarization and Pruning Agent is not automatically started after installation. Do not complete any step or procedure for starting the agent at this point.	To install a Summarization and Pruning Agent on Windows, complete the procedure "Windows: Installing a monitoring agent" on page 253. To install a Summarization and Pruning Agent on Linux or UNIX, complete the procedure "Linux or UNIX: Installing a monitoring agent" on page 259, including the following subsections: • *Installing the monitoring agent* • *Configuring the monitoring agent* • *Changing the file permissions for agents* (if you used a non-root user to install the Warehouse Proxy) *Do not complete the procedure for starting the agent.*

If the data warehouse is located on a remote computer, copy the *DB2 for Linux, UNIX, and Windows JDBC Universal Driver* (Type 4 driver) JAR files, included with the DB2 for Linux, UNIX, and Windows product installation, to the local computer where the Summarization and Pruning Agent is installed. You can copy the files to any directory that the user that the Summarization and Pruning Agent process runs as has access to.	The Type 4 driver file names and locations are as follows:
	`db2installdir/java/db2jcc.jar`
	`db2installdir/java/db2jcc_license_cu.jar`
	where *db2installdir* is the directory where DB2 for Linux, UNIX, and Windows was installed. The default DB2 for Linux, UNIX, and Windows Version 9 installation directory is as follows:
	• On Windows: `C:\Program Files\IBM\SQLLIB`
	• On AIX: `/usr/opt/db2_09_01`
	• On Linux or Solaris: `/opt/IBM/db2/V9.1`

Configure the Summarization and Pruning Agent. When you configure the Summarization and Pruning Agent, you configure the connection to the Tivoli Data Warehouse and you specify settings that control the operation of the Summarization and Pruning Agent. Perform this procedure whether or not the Summarization and Pruning Agent and the warehouse database are installed on the same computer.	"Configuring the Summarization and Pruning Agent (JDBC connection)" on page 595
Configure the Summarization and Pruning Agent to connect to the Tivoli Enterprise Portal Server. Perform this procedure whether or not the Summarization and Pruning Agent and the warehouse database are installed on the same computer.	See Step 9 of "Configuring the Summarization and Pruning Agent (JDBC connection)" on page 595.

Table 88. Tasks for installing and configuring communications for the Summarization and Pruning Agent (continued)

Task	Procedure
Configure history collection. When you configure history collection, you specify settings for how often to collect, aggregate, and prune data for individual monitoring agents and attribute groups. Configure history collection from the Tivoli Enterprise Portal.	See the *IBM Tivoli Monitoring: Administrator's Guide* for instructions on how to configure history collection.
Start the Summarization and Pruning Agent.	"Starting the Summarization and Pruning Agent" on page 608

Step 5: Install and configure communications for the Tivoli Performance Analyzer

You can install Tivoli Performance Analyzer to a server which also has other Tivoli Monitoring components installed, or you can install it to a separate machine. The installation procedure is similar to that for monitoring agents. Complete the tasks described in the following table, in the order listed, to install and configure Tivoli Performance Analyzer.

Table 89. Tasks for installing and configuring communications for the Tivoli Performance Analyzer

Task	Procedure
The installation procedure for Windows includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (Configuring the monitoring agent) and an X11 GUI is required to configure the agent. Alternatively, you can run the following command to utilize an X terminal emulation program (such as Cygwin) that is running on another computer: `export DISPLAY=my_windows_pc_IP_addr:0.0` where `my_windows_pc_IP_addr` is the IP address of a computer that is running an X terminal emulation program. See the information opposite. Be sure to perform all referenced installation and configuration procedures. **Note:** for sites setting up autonomous operation, the installation procedure includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring Server.	To install Tivoli Performance Analyzer on Windows, complete the procedure "Windows: Installing a monitoring agent" on page 253. To install Tivoli Performance Analyzer agent on Linux or AIX, complete the procedure "Linux or UNIX: Installing a monitoring agent" on page 259, including the following subsections: • Installing the monitoring agent • Configuring the monitoring agent • Changing the file permissions for agents Do not complete the procedure for starting the agent.
On the computer where Tivoli Performance Analyzer is installed, configure an ODBC data source for the data warehouse.	"Configuring an ODBC data source for a DB2 data warehouse" on page 501

If the data warehouse is located on a remote computer, copy the JDBC Universal Driver (Type 4 driver) JAR files, included with the DB2 for Linux, UNIX, and Windows product installation, to the local computer where Tivoli Performance Analyzer is installed. You can copy the files to any directory on the local computer.	The Type 4 driver file names and locations are the following:
	`db2installdir/java/db2jcc.jar`
	`db2installdir/java/db2jcc_license_cu.jar`
	where `db2installdir` is the directory where DB2 for Linux, UNIX, and Windows was installed. The default DB2 for Linux, UNIX, and Windows Version 9 installation directory is as follows:
	• for AIX: `/usr/opt/db2_09_01`
	• for Linux: `/opt/IBM/db2/V9.1`

Table 89. Tasks for installing and configuring communications for the Tivoli Performance Analyzer (continued)

Task	Procedure
Configure Tivoli Performance Analyzer to connect to the data warehouse.	For Tivoli Performance Analyzer on Windows, see “Configuring Tivoli Performance Analyzer on Windows (ODBC connection)” on page 516. For a Tivoli Performance Analyzer on Linux or AIX, see “Configuring Tivoli Performance Analyzer on Linux or UNIX (JDBC connection)” on page 517.
Start Tivoli Performance Analyzer.	“Starting Tivoli Performance Analyzer” on page 518

Configuring an ODBC data source for a DB2 data warehouse

A DB2 for Linux, UNIX, and Windows client on Windows requires an ODBC connection to the data warehouse.

Before you begin

- This procedure uses default values for the data source name, warehouse alias, and warehouse user ID. (Default values are used in configuration procedures for warehousing components.) Substitute different values if you do not want to use the default values.

Procedure

Complete the following procedure to set up an ODBC connection for Tivoli Performance Analyzer on Windows to a local or remote Tivoli Data Warehouse.

1. On the computer where the Tivoli Performance Analyzer is installed, open the Control Panel.
2. Click **Administrative Tools → Data Sources (ODBC)**
3. Click **Add** in the **System DSN** tab in the ODBC Data Source Administrator window.
4. Select **IBM DB2 ODBC DRIVER** from the list.
5. Click **Finish**.
6. In the ODBC DB2 Driver - Add window, perform the following steps:
   a. Enter `ITM Warehouse` in **Data source name**.
   b. Enter `Warehous` in **Database Alias**.

      If the Tivoli Data Warehouse is located on a remote computer, ensure that the database alias matches the alias that you used when cataloging the remote data warehouse. See “Cataloging a remote data warehouse” on page 500.

      If local, ensure that the database alias matches the name used for the warehouse database.
   c. Click **OK**.
7. Test the ODBC database connection before continuing:
   a. In the ODBC Data Source Administrator window, select **ITM Warehouse**.
   b. Click **Configure**.
   c. In the CLI/ODBC Settings - ITM Warehouse window, you see the data source name, **ITM Warehouse**.
   d. Enter `ITMUser` for the **User ID**.
   e. Type a password for the user in the **Password** field. The default password is `itmpswd1`.
   f. Click **Connect**.
   g. A **Connection test successful** message is displayed.
   h. Click **OK**.
   i. Click **OK** to close the window.

Configuring Tivoli Performance Analyzer on Windows (ODBC connection)

Use this procedure to configure Tivoli Performance Analyzer on Windows to connect to a DB2 for Linux, UNIX, and Windows data warehouse:

1. Log on to the Windows system where Tivoli Performance Analyzer is installed and begin the configuration:
   a. Click Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services. The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click Performance Analyzer and click **Reconfigure**.
   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Performance Analyzer: Agent Advanced Configuration) contain the settings for the connection between Tivoli Performance Analyzer and the hub monitoring server. These settings were specified when Tivoli Performance Analyzer was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to configure the ODBC data source.
4. Select **ODBC** from the list of selectable agent database connection types.
5. Set your Database Type to DB2.
6. Specify the Data Source Name - Agent ODBC DSN (ITM Warehouse is the default).

   Note: Tivoli Performance Analyzer does not create this DSN - it must already exist. If you are installing the agent on the same machine where TEP Server is installed, you can use the existing data source created by Tivoli Monitoring. Otherwise, you must create a new System DSN manually, prior to re-configuring Tivoli Performance Analyzer.

   On 64-bit versions of Windows, data sources created by the default ODBC Data Source Administrator applet available from the Control Panel are not available for 32-bit applications. Therefore you must use the 32-bit version of the ODBC Data Source Administrator applet from `<WINDOWS>\SysWOW64\odbcad32.exe`.

   Values for the data source name, database name, and database user ID and password must match the values that you used when configuring an ODBC connection for Tivoli Performance Analyzer. For more information, see “Configuring an ODBC data source for a DB2 data warehouse” on page 515.
7. Type the Username and Password. The entries in these fields are used to connect to the Tivoli Data Warehouse and are the same credentials as those used by the Tivoli Enterprise Portal Server, the Warehouse Proxy Agent, the Summarization and Pruning Agent, and the Performance Analyzer to communicate with the Tivoli Data Warehouse.
8. Click **Next** to proceed to the Advanced Configuration window.
9. You can enable Advanced Configuration to specify Tivoli Data Warehouse schema and The Tivoli Data Warehouse database schema. If you do not select **Enable advanced configuration** these options are greyed out.
10. You can also choose whether you want the agent to Initialize PA tables and OS domain tasks.

    Note: Setting Initialize PA tables to YES will remove and recreate all previously created tables deleting all user tasks and reverting each OS task to its default.
11. Use the **Bypass connection tests** option to finish the configuration without running connection tests.
12. Click **OK** to finish the configuration process.

Table 90. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.

Table 90. Configuration information for the Tivoli Data Warehouse database on DB2 for Linux, UNIX, and Windows (continued)

Field	Default value	Description
Username	ITMUser	The name of the Windows OS user that the Tivoli Performance Analyzer will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password for the Windows OS user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password.

Configuring Tivoli Performance Analyzer on Linux or UNIX (JDBC connection)

Use this procedure to configure Tivoli Performance Analyzer on Linux or UNIX to connect to a DB2 for Linux, UNIX, and Windows data warehouse on any operating system:

1. To begin the configuration, log on to the computer where Tivoli Performance Analyzer is installed.
   a. Change to the `install_dir/bin` directory and run the following command:

      ```
      ./itmcmd manage [-h install_dir]
      ```

      where `install_dir` is the installation directory for IBM Tivoli Monitoring. The default installation directory is `/opt/IBM/ITM`. The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Performance Analyzer** and click **Configure**. The Configure Tivoli Performance Analyzer window is displayed.
2. Set the Database Type to DB2.
3. Type the username and the password. The entries in these fields are used to connect to the Tivoli Data Warehouse.
4. Review all the defaults in the Agent Configuration window and change as required.
   a. If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of localhost.
   b. Change the port number if necessary (the default port number for DB2 is 50000).
   c. If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name. (See “Creating the warehouse database on DB2 for Linux, UNIX, and Windows” on page 494.)

      Note: when specifying the DB2 for Linux, UNIX, and Windows database name on Linux and UNIX, case is ignored. In other words, it makes no difference whether you provide the database name in lowercase or uppercase.
5. Specify the JDBC Driver. The default driver name for DB2 is `com.ibm.db2.jcc.DB2Driver`.
6. Specify the JDBC Driver Path, which should be provided as a list of JAR files with the full path separated by ":".

   Note: The driver files names for DB2 for Linux, UNIX, and Windows are:

   ```
   db2jcc.jar
   db2jcc_license_cu.jar
   ```

   Fast path: You can use the Browse button to specify the path. In such a case a file list is added at the end of the JDBC Driver Path text field, separated from the existing content by a path separator.

Attention: For compatibility with version 6.1.1 it is also possible to provide a single directory name containing all the JAR files which should be used, although this solution is not recommended. Ensure that the directory does not contain the `db2jcc4.jar` file. Performance Analyzer only supports Java version 1.5 and requires the `db2jcc.jar` file. Including `db2jcc4.jar` in the classpath may prevent the agent from being successfully configured.

7. You can use the **Test connection** button to check whether the connection can be initiated.
8. Click **Next** to proceed to the Advanced Configuration window.
   a. You can enable Advanced Configuration to specify TDW Schema and Configuration schema. If you do not select Enable advanced configuration, all of these options are greyed out.
   b. You can also choose whether you want the agent to initialize PA tables.

      Note: Setting Initialize PA tables to YES will remove and recreate all previously created tables deleting all user tasks and reverting each OS task to its default.
   c. Use the **Bypass connection tests** option to finish the configuration without running connection tests.
9. Click **Save** to save your settings and close the window.

Starting Tivoli Performance Analyzer

To start Tivoli Performance Analyzer from the Manage Tivoli Enterprise Monitoring Services window, right-click Tivoli Performance Analyzer and select Start. To start the Tivoli Performance Analyzer agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is `/opt/IBM/ITM`.

```
./itmcmd agent start pa
```

where pa is the product code for Tivoli Performance Analyzer agent.

Chapter 21. Tivoli Data Warehouse solution using DB2 on z/OS

Use the information and instructions in this chapter to implement a Tivoli Data Warehouse solution using mainframe-based DB2 running on z/OS as your warehouse database. The following table lists the goals for creating a DB2 on z/OS solution.

Table 91. Goals for creating a Tivoli Data Warehouse solution using DB2 on z/OS

Goal	Where to find information
Review your options, specific to a DB2 on z/OS solution, for operating system platforms and communications between warehousing components.	"Supported components" on page 520
Install prerequisite software before implementing your Tivoli Data Warehouse solution.	"Prerequisite installation" on page 521
Understand how to use the instructions for implementing your Tivoli Data Warehouse solution.	"Implementing a Tivoli Data Warehouse solution using DB2 on z/OS" on page 522
Complete the steps for implementing your Tivoli Data Warehouse solution using DB2 on z/OS for the data warehouse.	"Step 1: Connect the Warehouse Proxy node to your DB2 on z/OS database" on page 524 "Step 2: Configure the Tivoli Data Warehouse agents" on page 537

Supported components

Figure 118 presents the IBM Tivoli Monitoring environment when implementing a Tivoli Data Warehouse solution using DB2 on z/OS as the warehouse database. The diagram summarizes the supported operating system platforms for the various warehousing components, the supported database products, and the connections between components. For more specific information about supported operating systems and database products, including product names and versions, see “Hardware and software requirements” on page 138.

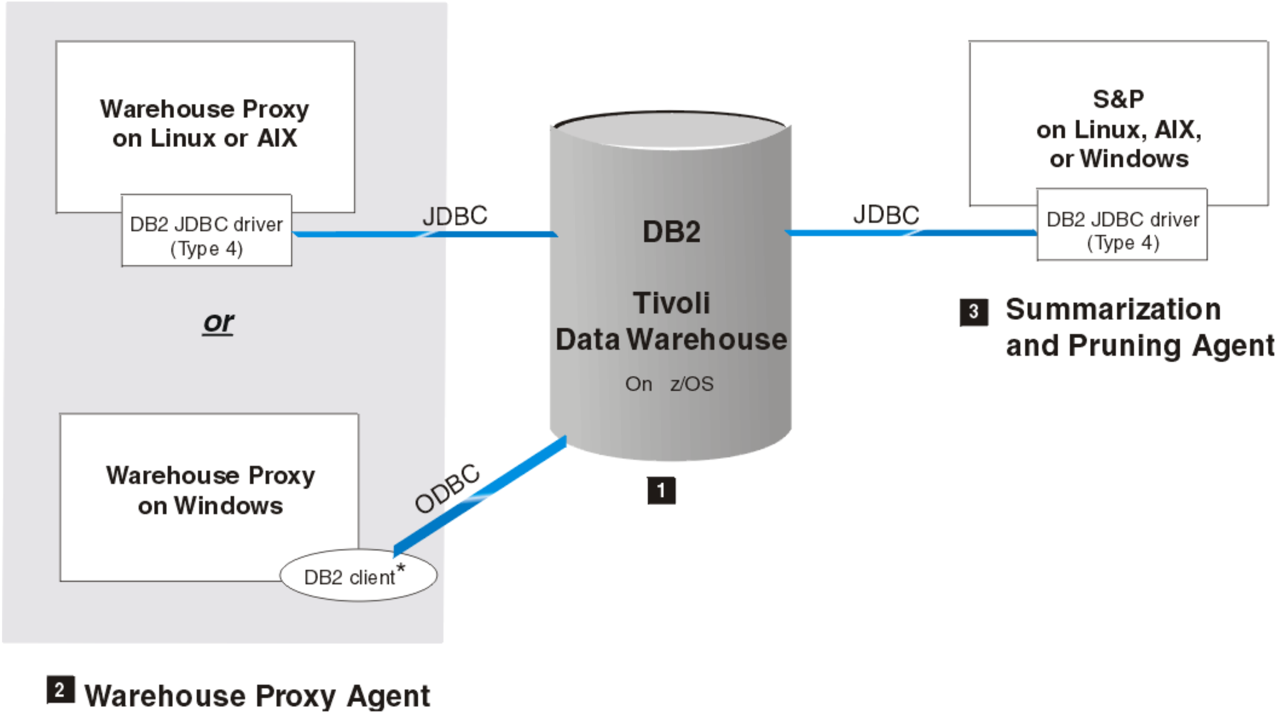


Figure 118. Tivoli Data Warehouse solution using DB2 on z/OS

Note: An asterisk (*) next to a database client indicates that you must manually install the client if it does not already exist.

In the following discussion, numbered product components correspond to the numbers on the diagram.

1 Tivoli Data Warehouse on DB2 on z/OS

A Tivoli Data Warehouse repository on DB2 on z/OS can be accessed by any ITM-supported platform that can run the Warehouse Proxy Agent—Windows, Linux, or AIX—as well as the DB2 Connect software.

2 Warehouse Proxy Agent

A Warehouse Proxy Agent on Linux or AIX communicates with the warehouse database through a JDBC connection. Install a Type 4 driver (DB2 on z/OS JDBC Universal Driver) on the computer where the Warehouse Proxy Agent is located.

A Warehouse Proxy Agent on Windows communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the DB2 on z/OS client. You must install a DB2 client on the Windows computer where the Warehouse Proxy Agent is located, and then catalog the remote node and database on the local computer.

Note: If you install a version 9 DB2 client, you also must install DB2 Connect Server Edition to connect to a DB2 on z/OS data server.

3 Summarization and Pruning Agent

The Summarization and Pruning Agent communicates with the warehouse database through a JDBC connection from any supported operating system. Install a DB2 on z/OS Type 4 JDBC driver (*DB2 on z/OS JDBC Universal Driver*) on the computer where the Summarization and Pruning Agent is located.

Prerequisite installation

Before you implement your Tivoli Data Warehouse solution, complete one or more hub installations, including the warehousing components (see the appropriate chapter within this section for the necessary installation instructions). Include the following components in each hub installation:

- The hub Tivoli Enterprise Monitoring Server
- *(Optional)* One or more remote monitoring servers
- The Tivoli Enterprise Portal Server, on either Windows, Linux, or UNIX
- An IBM DB2 on z/OS server on the computer where you will create the Tivoli Data Warehouse database. (The Tivoli Data Warehouse database can be shared in a multi-hub installation or dedicated to a single hub.)
- DB2 Connect Server Edition.
- *(Optional)* A portal desktop client
- *(Optional)* Monitoring agents and the application support for the monitoring agents
- The Warehouse Proxy Agent and the Summarization and Pruning Agent
- *(Optional)* The Tivoli Performance Analyzer
- *(Optional)* Language packs for all languages other than English

See Table 92 for related information:

Table 92. Information topics related to installation of prerequisite software for a Tivoli Data Warehouse solution

Topic	Where to find information
Single and multiple hub installations	To understand the terminology related to single and multiple hub installations, see “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46.
Installation procedures for prerequisite components, excluding the Warehouse Proxy Agent and the Summarization and Pruning Agent	The detailed instructions for installing the prerequisite components are described in Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207. See your database documentations for instructions on how to install a supported database server.
Supported RDBMS versions	For specific information about the supported database platforms for the portal server database and the Tivoli Data Warehouse, see “Hardware and software requirements” on page 138.

Implementing a Tivoli Data Warehouse solution using DB2 on z/OS

Use the instructions in the remainder of this chapter to implement a Tivoli Data Warehouse solution using DB2 on z/OS version 9.1 (or subsequent) as your data warehouse.

Requirements

You must set permissions for both the Warehouse Proxy Agent and the Summarization and Pruning Agent when connecting to a DB2 database on z/OS. In the commands that follow, `USER1` is in place of the user you are connecting to the Warehouse Proxy Agent or the Summarization and Pruning Agent.

To configure the Warehouse Proxy Agent run the following command:

```
db2 "GRANT CREATEDBA TO USER1"
```

To grant the permission that is required to start the Warehouse Proxy Agent run the following command:

```
db2 "GRANT SELECT ON SYSIBM.SYSTABLES TO USER1"
```

Both of the following access permissions must be granted for the Summarization and Pruning Agent:

```
db2 "GRANT SELECT ON SYSIBM.SYSROUTINES TO USER1"
db2 "GRANT SELECT ON SYSIBM.SYSTABLESPACE TO USER1"
```

The Warehouse Proxy Agent and the Summarization and Pruning Agent both use the implicit table creation option when connected to a DB2 database on z/OS. Both agents create tables without specifying a table space or a database in the IN clause of a CREATE statement.

- DB2 on z/OS version 9 creates an implicit database each time a table is created using a name in the range DSN00001 to DSN60000. The characteristics of implicitly created databases are shown in Table 93.

Table 93. Characteristics of implicitly created DB2 on z/OS database

Field	Value
Name	DSNxxxxx, where xxxxx is a number from 00001 to 60000
BUFFERPOOL	BP0, BP8K0, BP16K0, BP32K[1] Default values. Changeable through DSNZPARM update
INDEXBP	IDXBPOOL setting in DSNZPARM
STOGROUP	SYSDEFLT
value in column IMPLICIT of SYSIBM.SYSDATABASE	'Y'
ENCODING_SCHEME	DEFAULT is the DSNHDECP setting (see below)
SBCS_CCSID	DEFAULT is the DSNHDECP setting (see below)
DBCS_CCSID	DEFAULT is the DSNHDECP setting (see below)
MIXEC_CCSID	DEFAULT is the DSNHDECP setting (see below)

Notes:

1. DB2 on z/OS chooses a specific buffer pool during creation of the implicit object, depending on the record size. When the maximum record size reaches approximately 90% of the capacity of the smaller page size, DB2 chooses the next larger page size, as shown in Table 94.

Table 94. Maximum DB2 on z/OS page sizes

Name	Page Size
BP0	4K
BP8KO	8K

Table 94. Maximum DB2 on z/OS page sizes (continued)

Name	Page Size
BP16KO	16KB
BP32K	32KB

2. To ensure databases can be created implicitly, DB2 on z/OS's CREATE IMPLICIT DATABASES installation parameter must be set to YES.

- DB2 on z/OS v9 creates implicit table spaces when implicitly creating a table.
- The Summarization and Pruning Agent creates functions that Tivoli Enterprise Portal user can take advantage of when creating custom history queries. To let the agent create those functions, a default WLM (Workload Manager) should be created.
- Before beginning this procedure, gather the following information about your target DB2 on z/OS database:

Table 95. Required parameters for accessing the DB2 on z/OS database

DB2 on z/OS parameter	Your value
Database name	
Port number	
DB2 userid	
DB2 password	
Fully qualified host name	

Solution steps

To implement your Tivoli Data Warehouse solution using DB2 on z/OS, complete the major steps described in the remaining sections of this chapter, in the order listed:

1. Connect the Warehouse Proxy node to your DB2 on z/OS database.
2. Configure the Tivoli Data Warehouse agents.

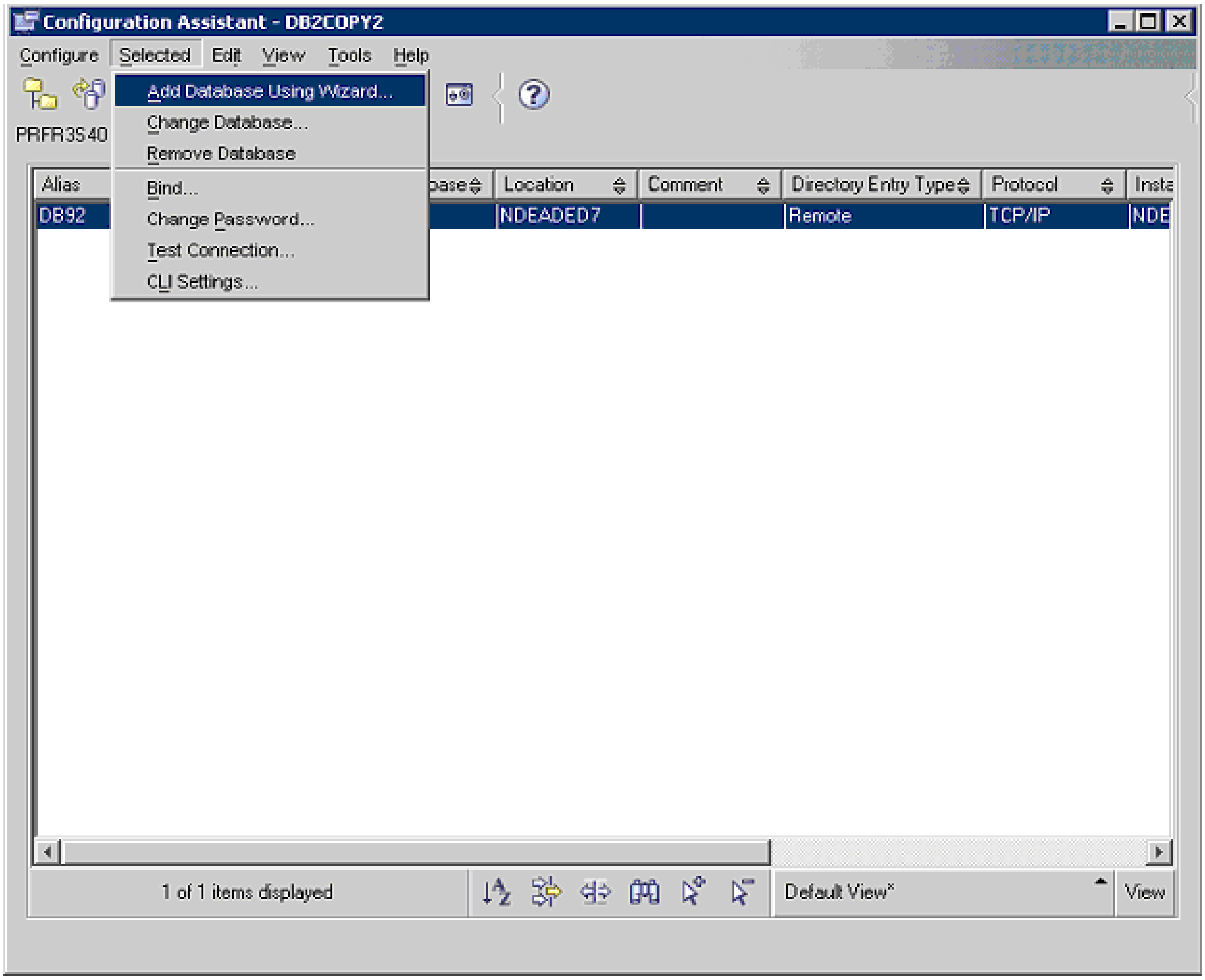
To implement your solution successfully:

- Perform the tasks in the order listed.
- Do not skip a task and move forward to the procedures that follow it.

Step 1: Connect the Warehouse Proxy node to your DB2 on z/OS database

On the IBM Tivoli Monitoring node running the Warehouse Proxy Agent, invoke the DB2 Client Configuration Assistant from the DB2 Control Center.

The Client Configuration Assistant, shown in Figure 119, opens.



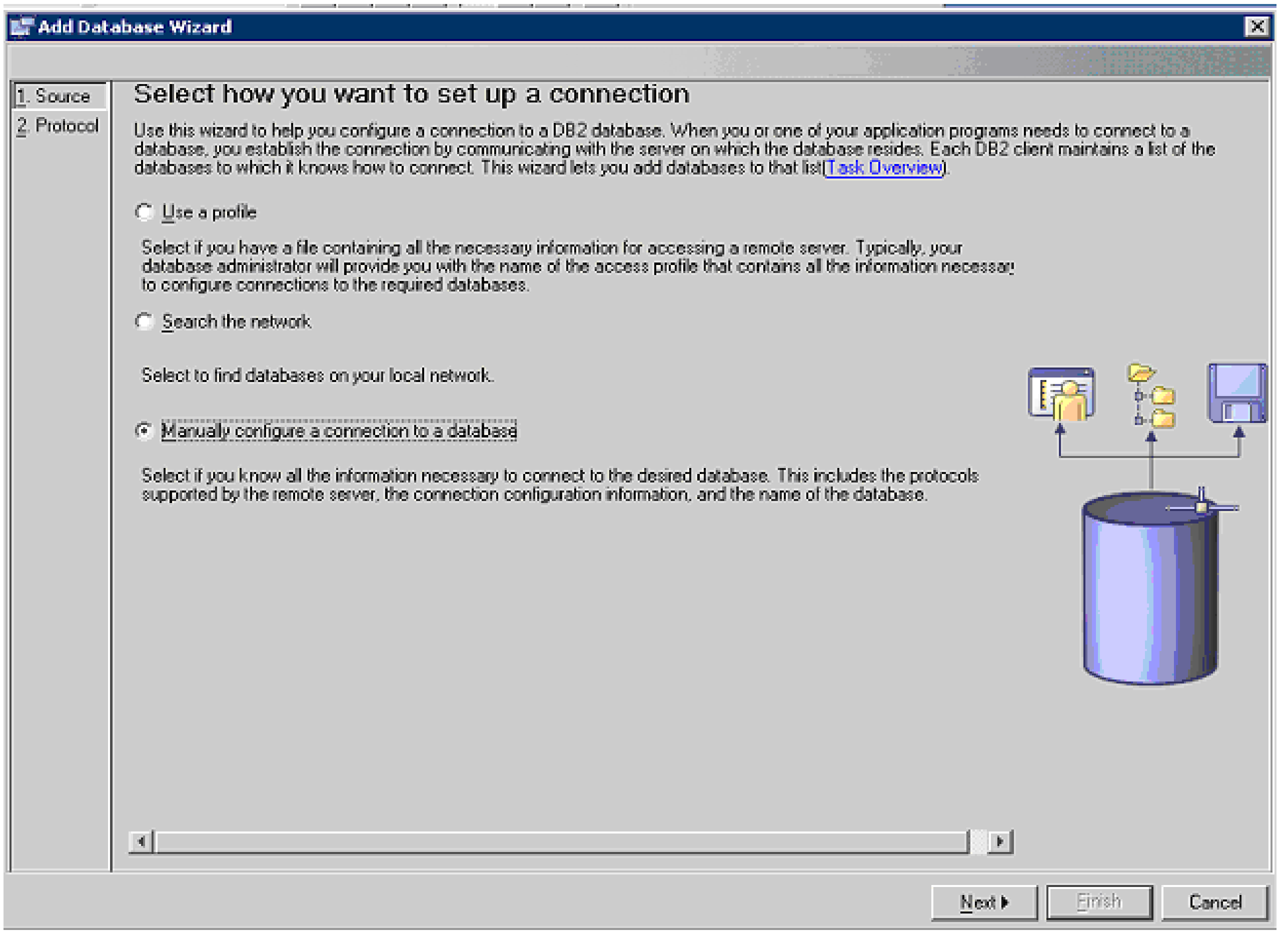
Alias	base	Location	Comment	Directory Entry Type	Protocol	Insta
DB92		NDEADED7		Remote	TCP/IP	NDE

Figure 119. DB2 Client Configuration Assistant screen

Start defining the database connection

From the **Selected** pulldown menu, select the **Add Database Using Wizard** menu option to set up the connection to your DB2 on z/OS database.

The Add Database Wizard notebook opens with the **Source** tab active, as shown in Figure 120.



Select how you want to set up a connection

Use this wizard to help you configure a connection to a DB2 database. When you or one of your application programs needs to connect to a database, you establish the connection by communicating with the server on which the database resides. Each DB2 client maintains a list of the databases to which it knows how to connect. This wizard lets you add databases to that list(Task Overview).

○ Use a profile

Select if you have a file containing all the necessary information for accessing a remote server. Typically, your database administrator will provide you with the name of the access profile that contains all the information necessar to configure connections to the required databases.

○ Search the network

Select to find databases on your local network.

● Manually configure a connection to a database

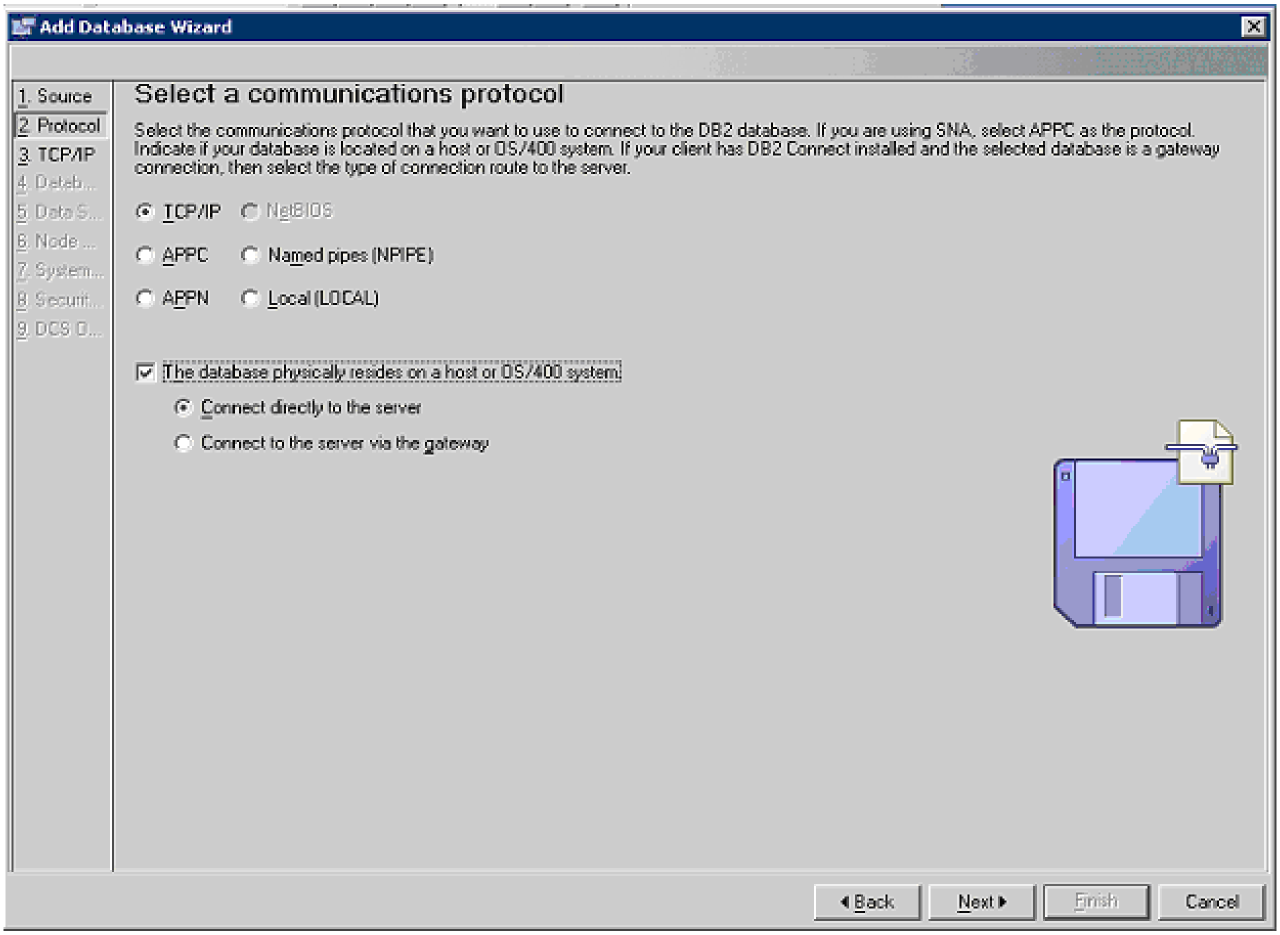
Select if you know all the information necessary to connect to the desired database. This includes the protocols supported by the remote server, the connection configuration information, and the name of the database.

Figure 120. DB2 Add Database Wizard notebook, Source tab

Activate the **Manually configure a connection to a database** radio button; then click **Next**.

Define the communications protocol

The Add Database Wizard notebook reappears with the **Protocol** tab active, as shown in Figure 121.



Select a communications protocol

Select the communications protocol that you want to use to connect to the DB2 database. If you are using SNA, select APPC as the protocol. Indicate if your database is located on a host or OS/400 system. If your client has DB2 Connect installed and the selected database is a gateway connection, then select the type of connection route to the server.

● TCP/IP ○ NetBIOS

○ APPC ○ Named pipes (NPIPE)

○ APPN ○ Local (LOCAL)

☑ The database physically resides on a host or OS/400 system.

● Connect directly to the server

○ Connect to the server via the gateway

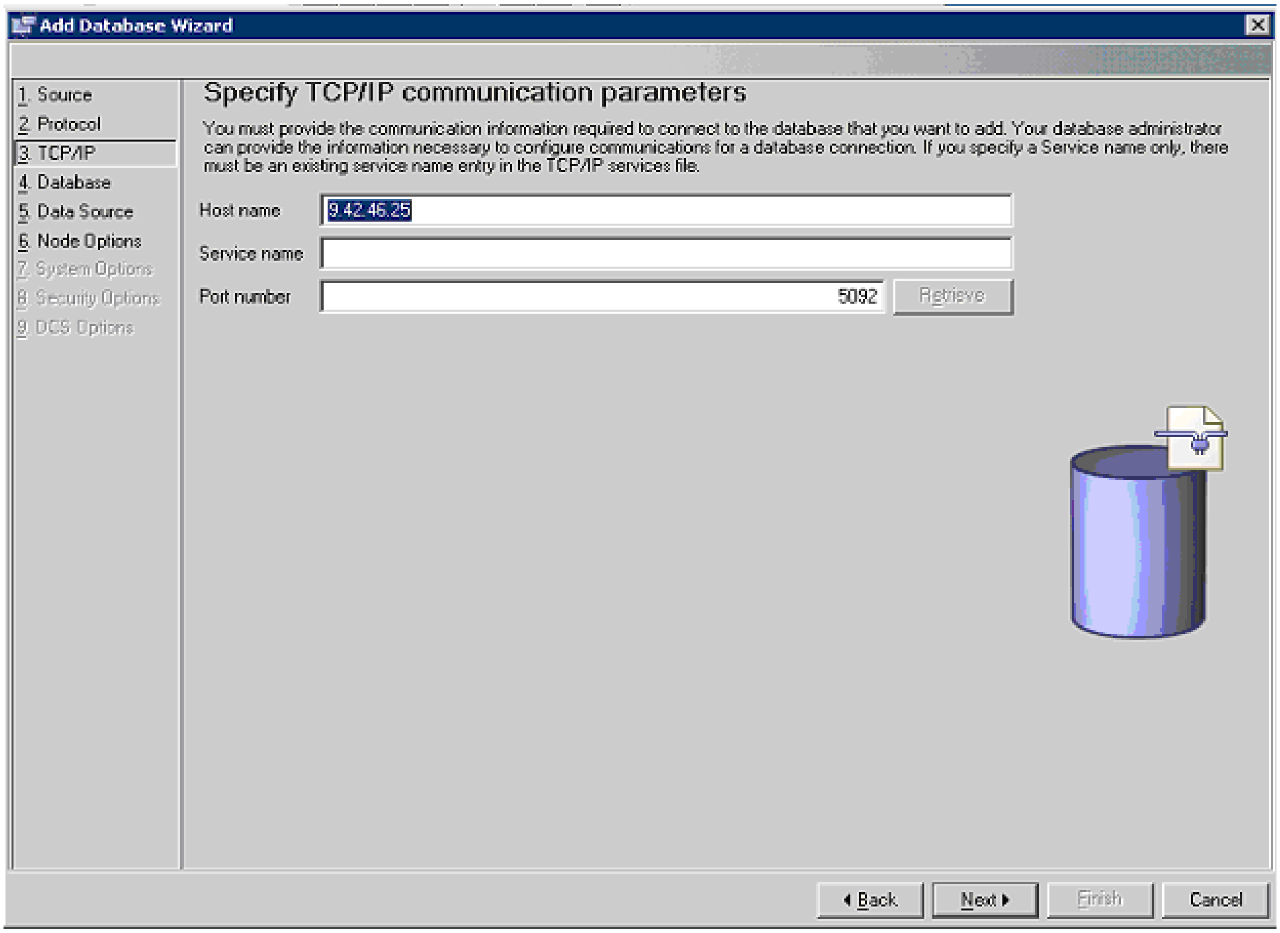
Figure 121. DB2 Add Database Wizard notebook, Protocol tab

On the Protocol notebook page:

1. Activate the **TCP/IP** radio button.
2. Select the **The database physically resides on a host or AS/400 system** option.
3. Activate the **Connect directly to the server** radio button.
4. Click **Next**.

Define the TCP/IP communications parameters

The Add Database Wizard notebook reappears with the **TCP/IP** tab active, as shown in Figure 122.



Add Database Wizard

Specify TCP/IP communication parameters

You must provide the communication information required to connect to the database that you want to add. Your database administrator can provide the information necessary to configure communications for a database connection. If you specify a Service name only, there must be an existing service name entry in the TCP/IP services file.

Host name: 9.42.46.25

Service name:

Port number: 5092

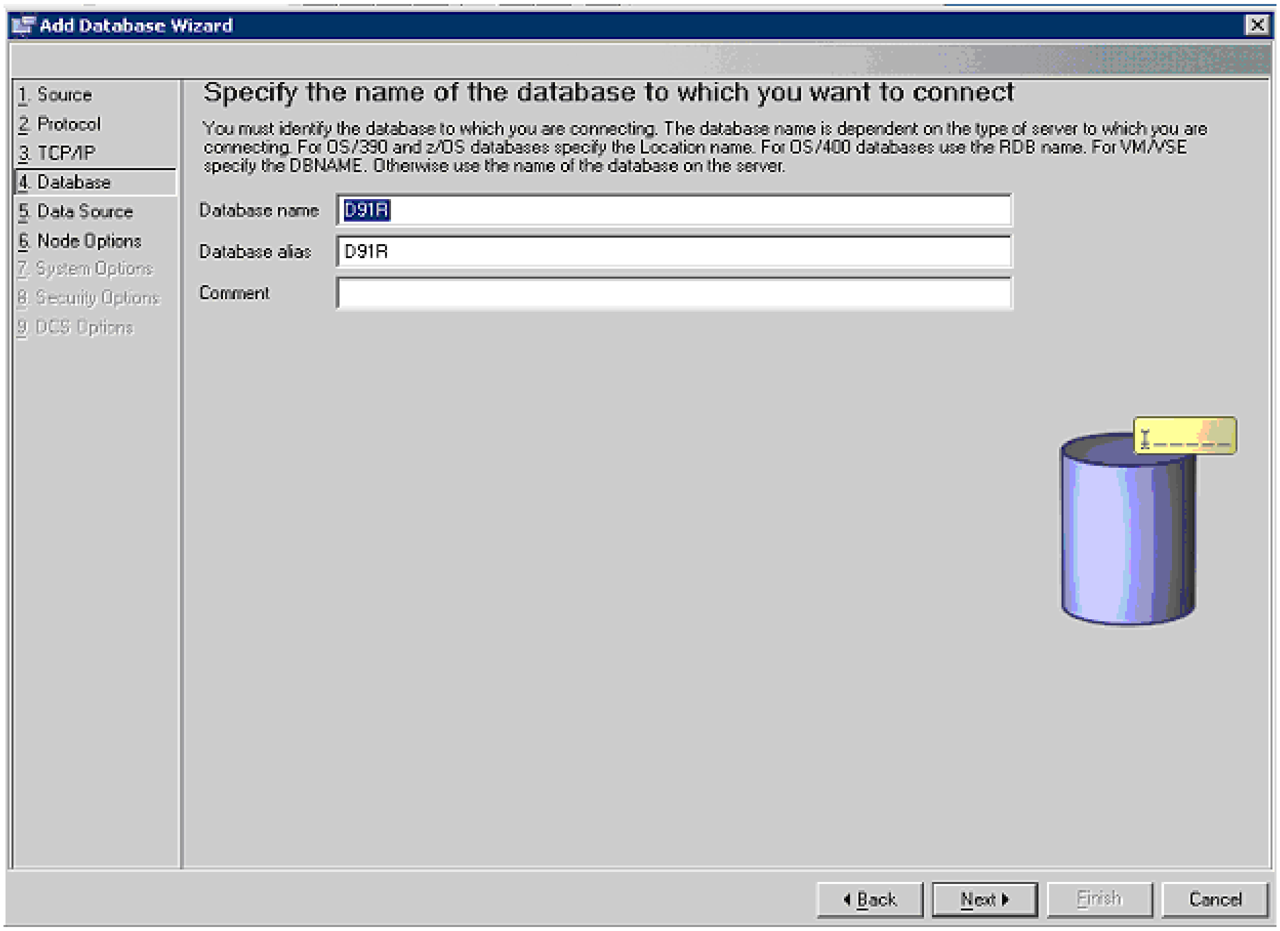
Figure 122. DB2 Add Database Wizard notebook, TCP/IP tab

On the TCP/IP notebook page:

1. Enter the fully qualified IP **Host name**.
2. Specify the **Port number**. The default port for z/OS is 446.
3. Click **Next**.

Identify the DB2 on z/OS database

The **Database** tab opens, as shown in Figure 123.



Add Database Wizard

1. Source
2. Protocol
3. TCP/IP
4. Database
5. Data Source
6. Node Options
7. System Options
8. Security Options
9. DCS Options

Specify the name of the database to which you want to connect

You must identify the database to which you are connecting. The database name is dependent on the type of server to which you are connecting. For OS/390 and z/OS databases specify the Location name. For OS/400 databases use the RDB name. For VM/VSE specify the DBNAME. Otherwise use the name of the database on the server.

Database name: D91R

Database alias: D91R

Comment:

Figure 123. DB2 Add Database Wizard notebook, Database tab

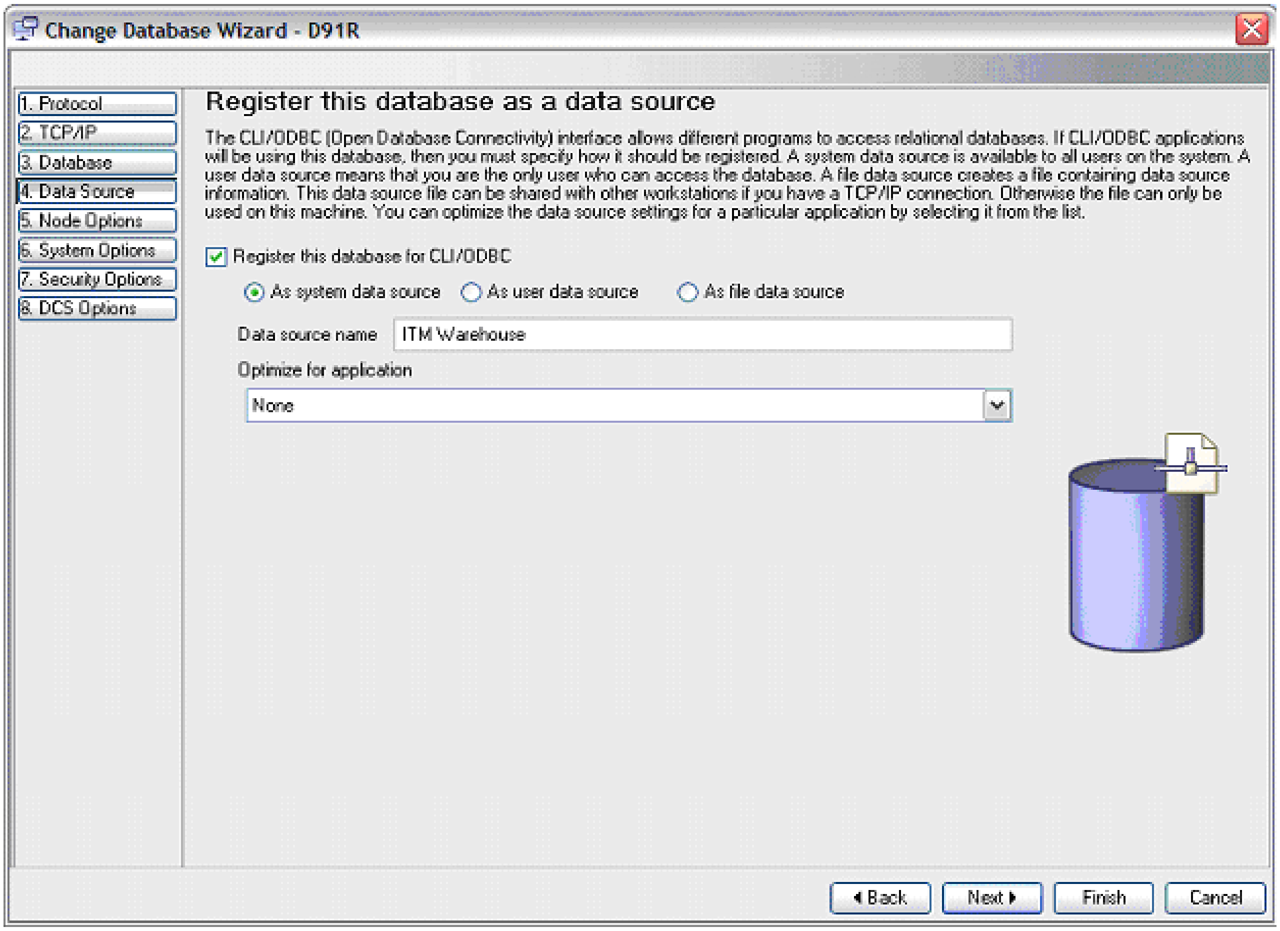
On the Database notebook page:

1. Enter the fully qualified **Database name**. For DB2 for Linux, UNIX, and Windows, this is the database name; however, with DB2 on z/OS, this is the subsystem name.
2. Enter the DB2 on z/OS **Database alias**. This field is required.

   Note: The alias is limited to 8 characters. Do not create duplicate aliases across your Tivoli Monitoring systems (such as WAREHOUS).
3. Click **Next**.

Register the database as an ODBC data source

The **Data Source** tab opens, as shown in Figure 124.



Change Database Wizard - D91R

Register this database as a data source

The CLI/ODBC (Open Database Connectivity) interface allows different programs to access relational databases. If CLI/ODBC applications will be using this database, then you must specify how it should be registered. A system data source is available to all users on the system. A user data source means that you are the only user who can access the database. A file data source creates a file containing data source information. This data source file can be shared with other workstations if you have a TCP/IP connection. Otherwise the file can only be used on this machine. You can optimize the data source settings for a particular application by selecting it from the list.

☑ Register this database for CLI/ODBC

● As system data source ○ As user data source ○ As file data source

Data source name: ITM Warehouse

Optimize for application: None

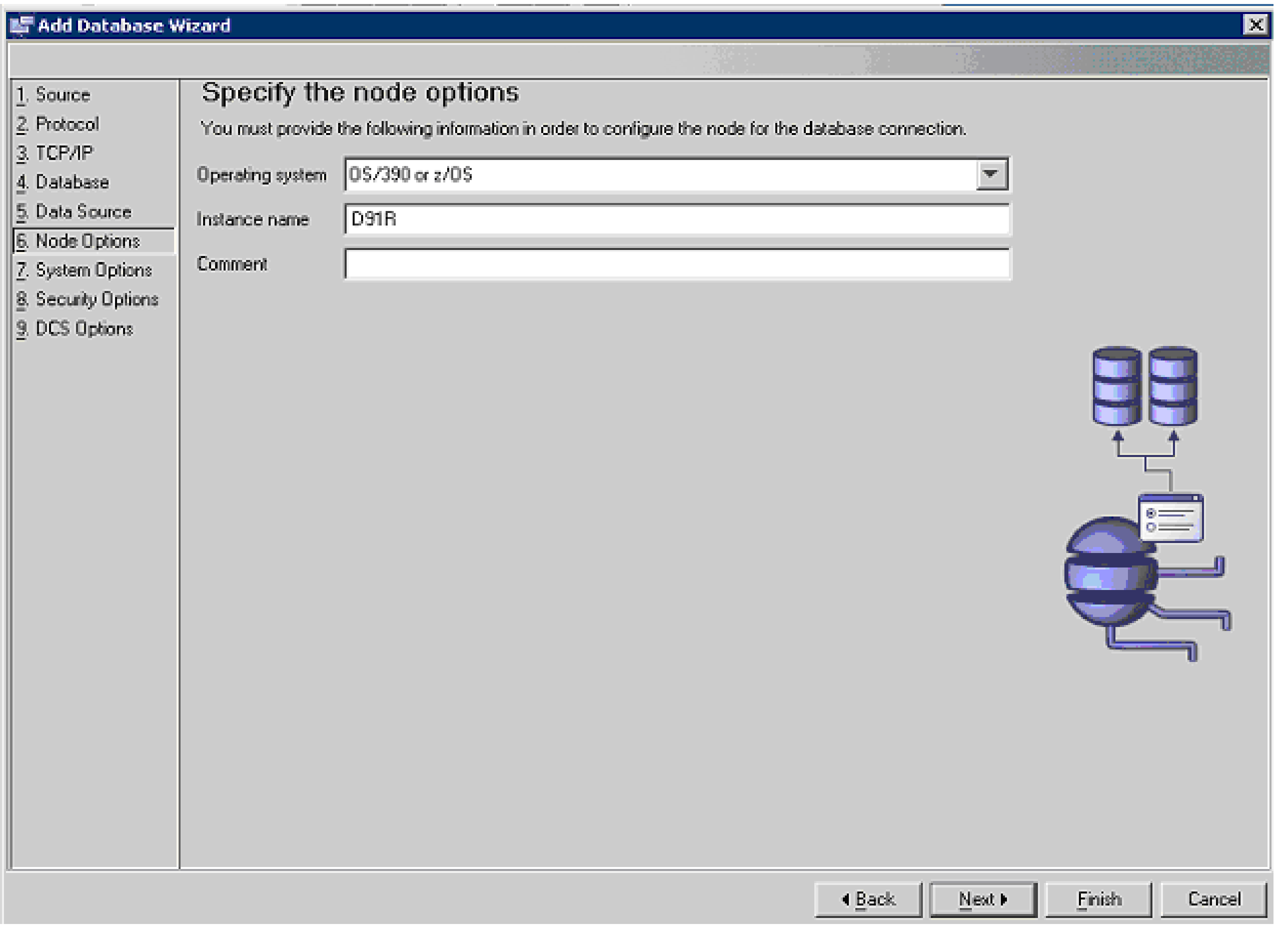
Figure 124. DB2 Add Database Wizard notebook, Data Source tab

On the Data Source notebook page, click **Next**.

1. Ensure the **Register this database for CLI/ODBC** option is selected.
2. Ensure the **As system data source** radio button is active.
3. Enter the fully qualified DB2 on z/OS **Data source name**. This is the name you will specify when configuring the Warehouse Proxy Agent in “Step 2: Configure the Tivoli Data Warehouse agents” on page 537, for example, ITM Warehouse.
4. Click **Next**.

Identify the z/OS server containing the DB2 on z/OS database

The **Node Options** tab opens, as shown in Figure 125.



Specify the node options

You must provide the following information in order to configure the node for the database connection.

Operating system: OS/390 or z/OS

Instance name: D91R

Comment:

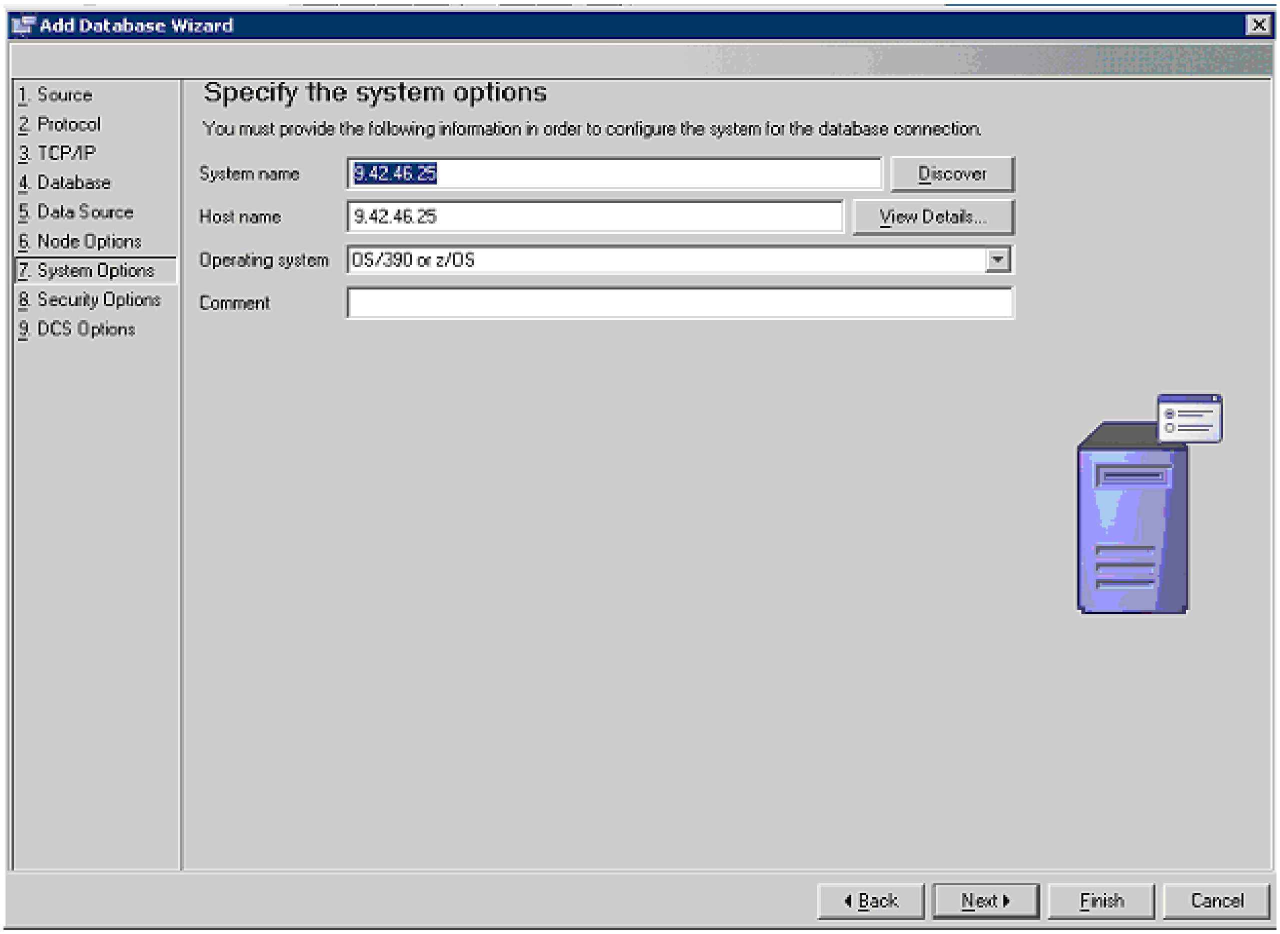
Figure 125. DB2 Add Database Wizard notebook, Node Options tab

On the Node Options notebook page:

1. Pull down the list of **Operating system** settings, and select **OS/390® or z/OS**.
2. For **Instance name**, specify your DB2 on z/OS instance name.
3. Click **Next**.

Define the DB2 on z/OS system options

The **System Options** tab opens, as shown in Figure 126.



Specify the system options

You must provide the following information in order to configure the system for the database connection.

System name: 9.42.46.25

Host name: 9.42.46.25

Operating system: OS/390 or z/OS

Comment:

Figure 126. DB2 Add Database Wizard notebook, System Options tab

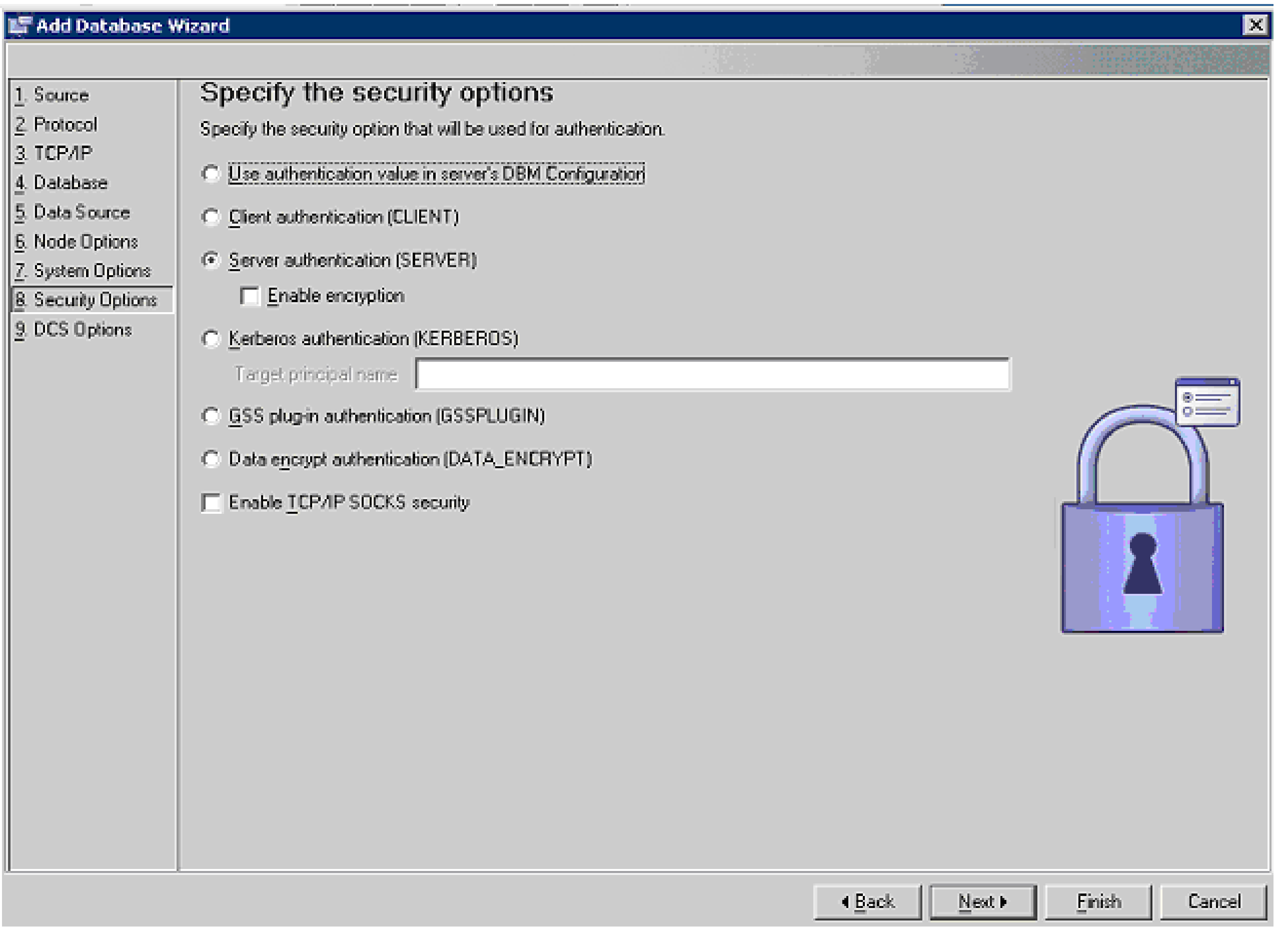
On the System Options notebook page:

1. For **System name**, specify the host name or IP address of the z/OS system where your DB2 on z/OS database is stored.

   The **Host name** field gets filled in automatically with the same value.
2. Pull down the list of **Operating system** settings, and select **OS/390 or z/OS**.
3. Click **Next**.

Define the DB2 on z/OS security options

The **Security Options** tab opens, as shown in Figure 127.



Specify the security options

Specify the security option that will be used for authentication.

○ Use authentication value in server's DBM Configuration

○ Client authentication (CLIENT)

● Server authentication (SERVER)

□ Enable encryption

○ Kerberos authentication (KERBEROS)

Target principal name

○ GSS plug-in authentication (GSSPLUGIN)

○ Data encrypt authentication (DATA_ENCRYPT)

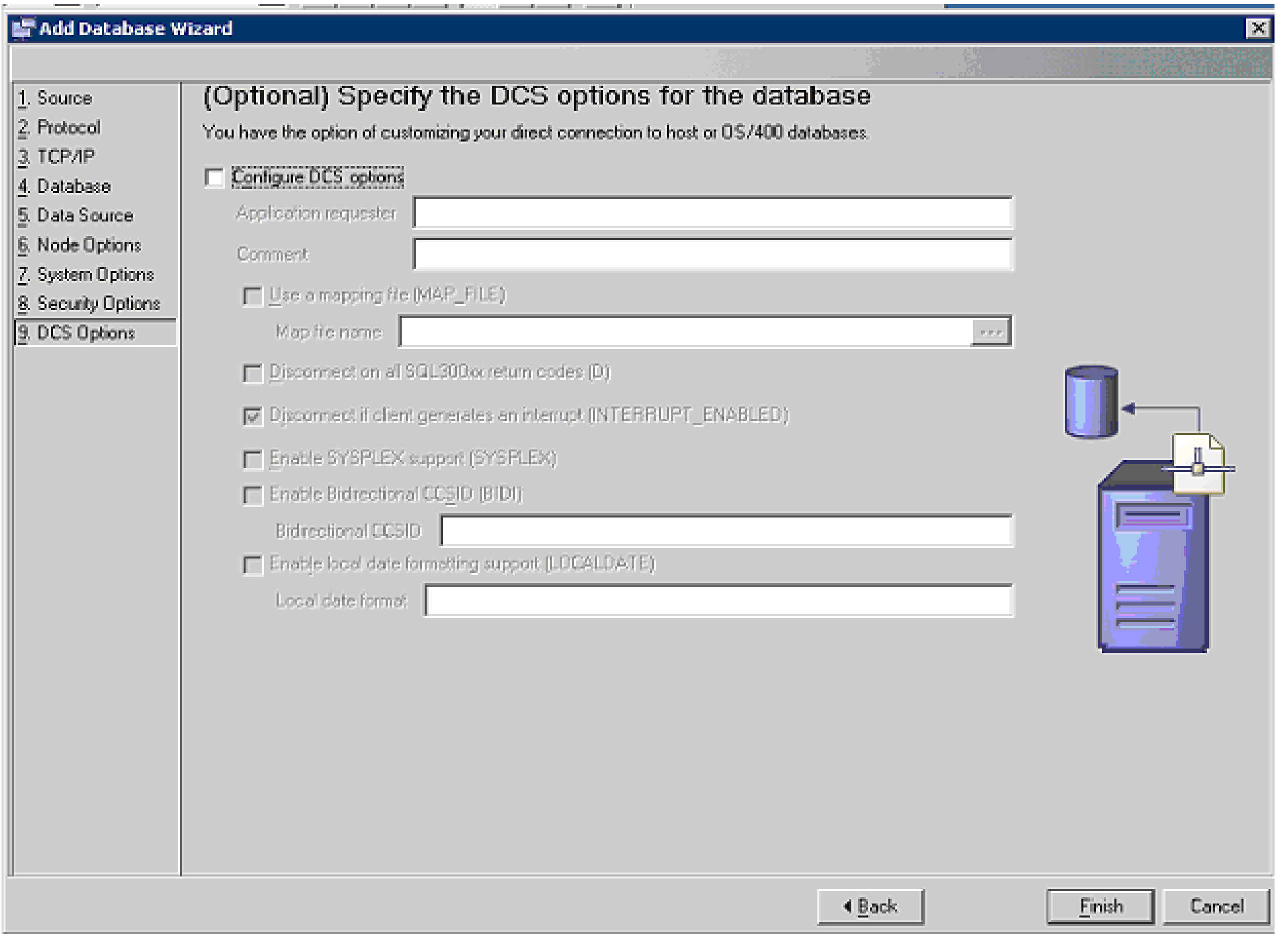
□ Enable TCP/IP SOCKS security

Figure 127. DB2 Add Database Wizard notebook, Security Options tab

On the Security Options notebook page, activate the **Server authentication (SERVER)** radio button, and click **Next**.

Complete the DB2 on z/OS host connection

The **DCS Options** tab opens, as shown in Figure 128.



(Optional) Specify the DCS options for the database

You have the option of customizing your direct connection to host or OS/400 databases.

☐ Configure DCS options

Application requester

Comment

☐ Use a mapping file (MAP_FILE)

Map file name

☐ Disconnect on all SQL300xx return codes (D)

☑ Disconnect if client generates an interrupt (INTERRUPT_ENABLED)

☐ Enable SYSPLEX support (SYSPLEX)

☐ Enable Bidirectional CCSID (BIDI)

Bidirectional CCSID

☐ Enable local date formatting support (LOCALDATE)

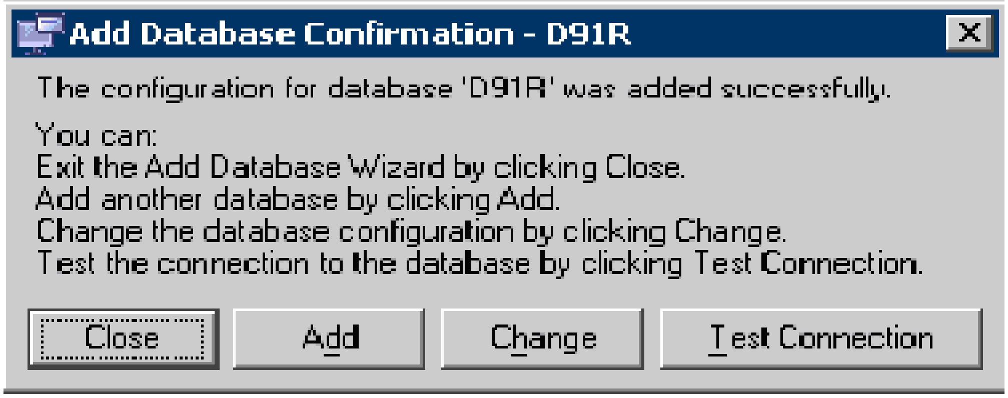
Local date format

Figure 128. DB2 Add Database Wizard notebook, DCS Options tab

On the DCS Options notebook page, click **Finish**.

Verify that the connection can be made

If your database definition is successful, the Add Database Confirmation window shown in Figure 129 is displayed.



Add Database Confirmation - D91R

The configuration for database 'D91R' was added successfully.

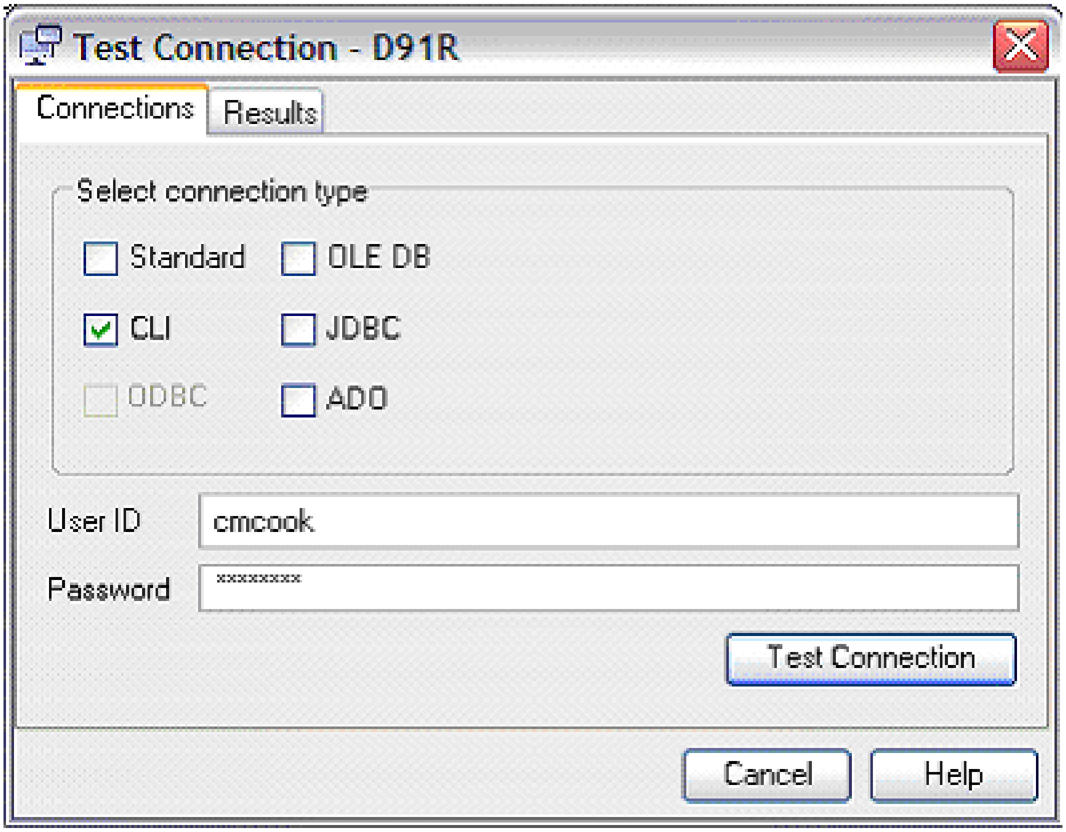
You can:
Exit the Add Database Wizard by clicking Close.
Add another database by clicking Add.
Change the database configuration by clicking Change.
Test the connection to the database by clicking Test Connection.

Figure 129. Connection-confirmation screen

If this window is not displayed, press the **Back** button, verify each notebook page, and correct the information as necessary.

To test the connection to the remote DB2 on z/OS database, press the **Test Connection** button.

The **Connect to DB2 Database** screen, shown in Figure 130, opens.



□ Standard □ OLE DB

☑ CLI □ JDBC

□ ODBC □ ADO

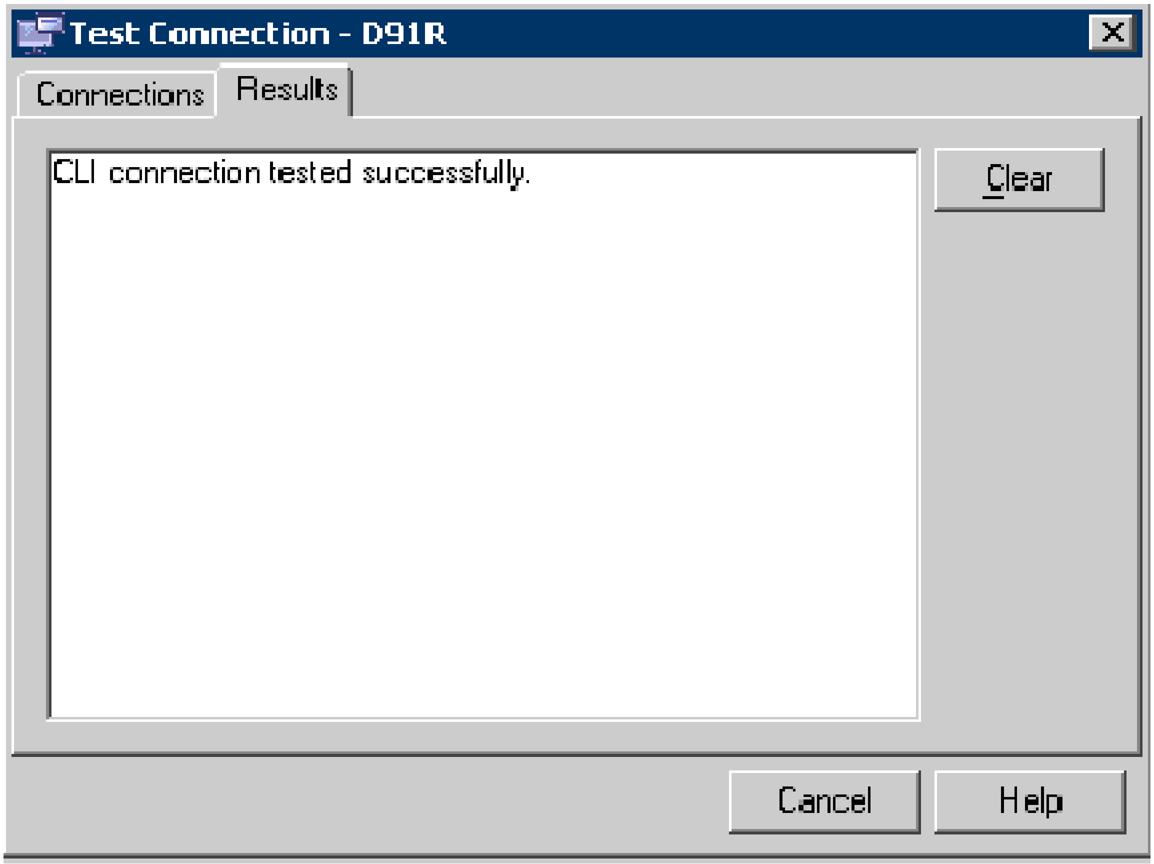
User ID: cmcook

Password: *********

Figure 130. Connect to DB2 Database screen

Enter the user ID and password for the remote DB2 on z/OS database, and press the **Test Connection** button.

If the database connection can be made, the confirmation screen shown in Figure 131 is displayed.



CLI connection tested successfully.

Figure 131. DB2 Connection Confirmation screen

Step 2: Configure the Tivoli Data Warehouse agents

To enable storage of IBM Tivoli Monitoring historical data in your Tivoli Data Warehouse database on z/OS, you need to perform the installation procedures for the Warehouse Proxy Agent and the Summarization and Pruning Agent. Complete these two steps from Chapter 20, “Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows,” on page 489:

- “Step 2: Install and configure communications for the Warehouse Proxy Agent” on page 499

  Note: When specifying the DB2 on z/OS database name on Linux and UNIX, the correct case must be used.
- “Step 4: Install and configure communications for the Summarization and Pruning Agent” on page 513

Testing your DB2 on z/OS database connection

At any time you can invoke either of the following procedures to ensure your DB2 on z/OS database connection is still active.

Testing the database connection using the DB2 Control Center

Invoke the DB2 Control Center, shown in Figure 132.

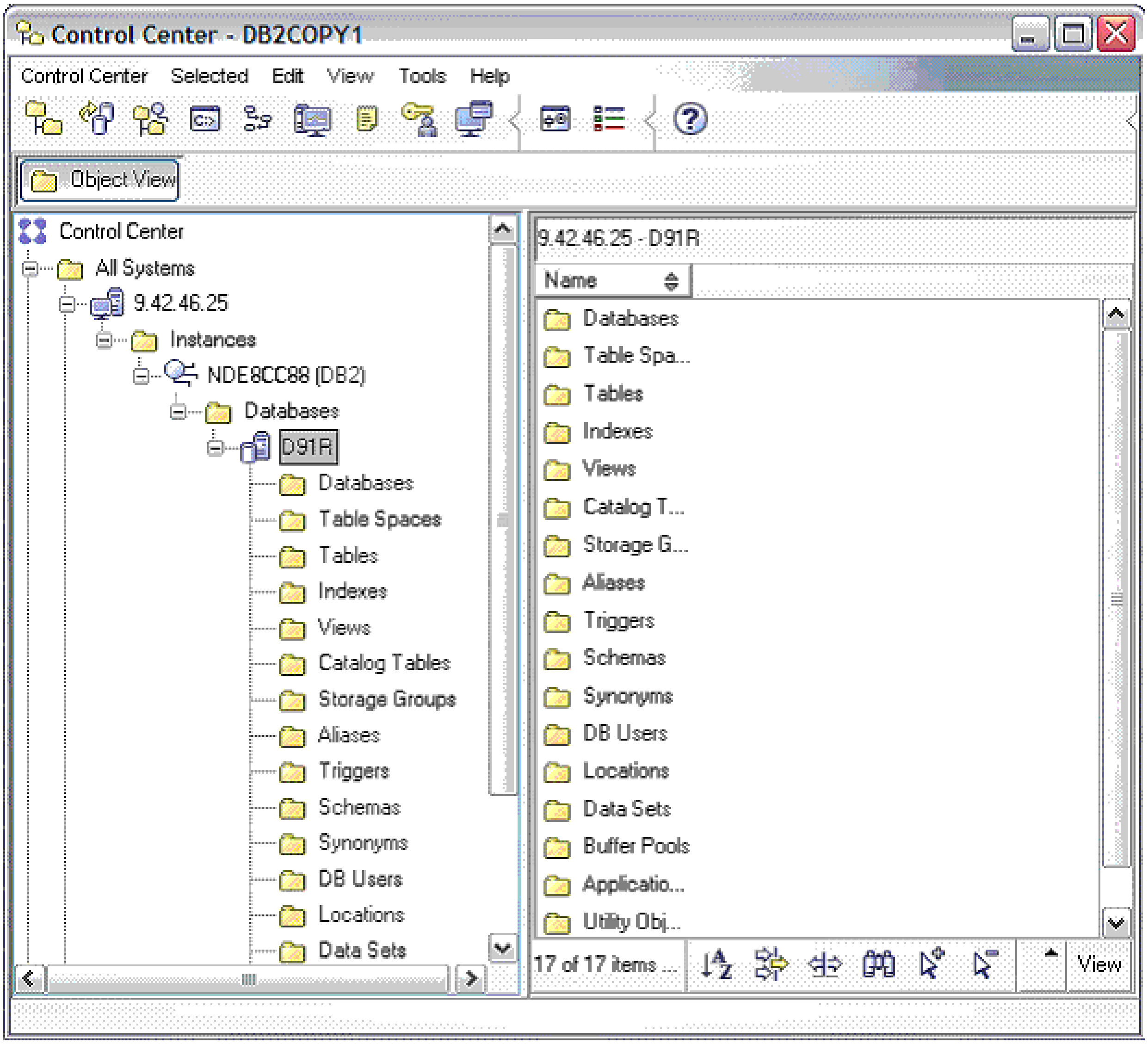


Figure 132. DB2 Control Center

Select the plus sign (⊞) to the left of the z/OS system that is running the DB2 on z/OS subsystem that owns your Tivoli Data Warehouse repository; then expand its list of databases. The list should include the database you're using.

To again check your database connection, right-click your database name, and select **Connect** from the pop-up menu, as shown in Figure 133 on page 539.

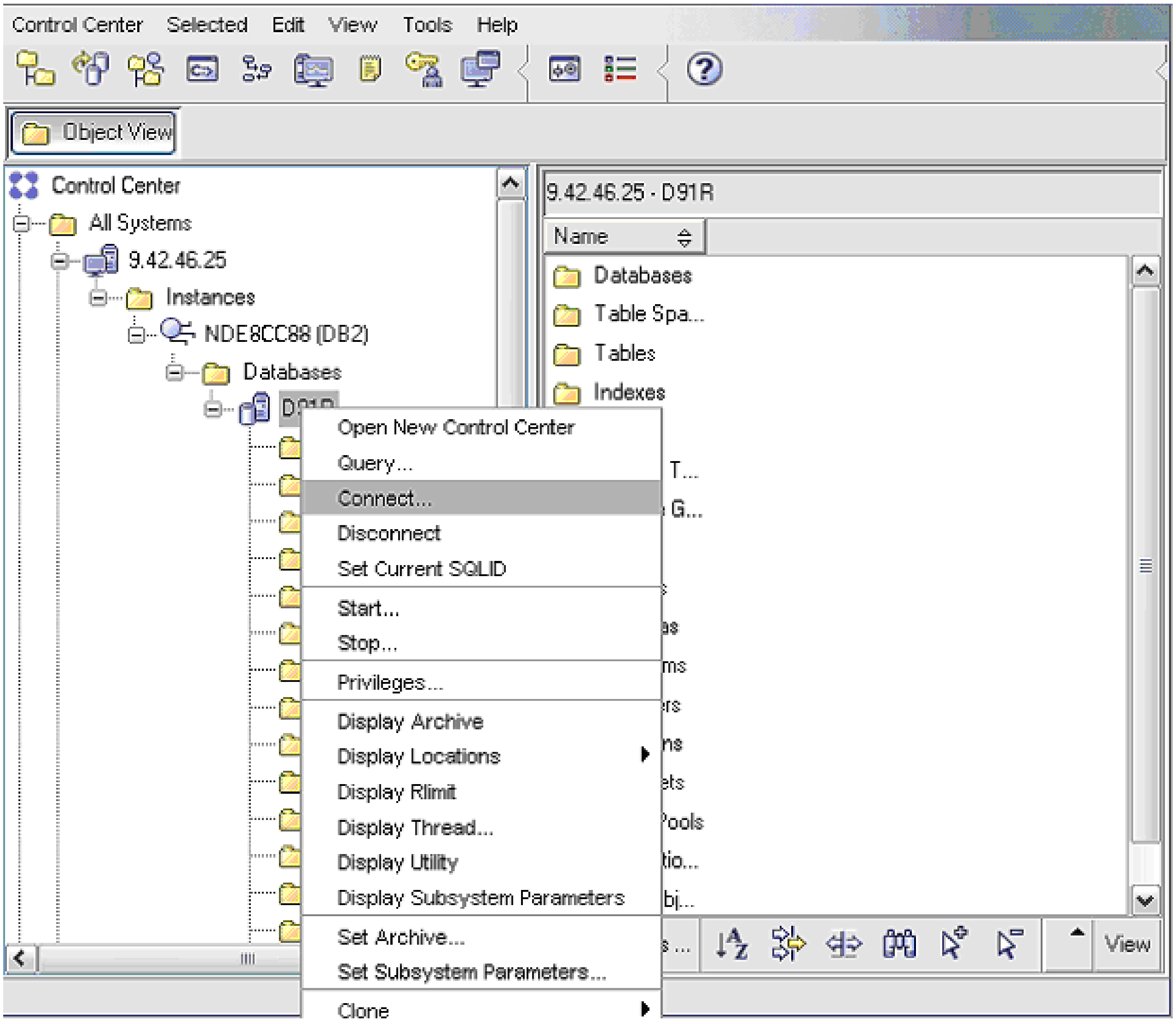


Figure 133. DB2 Control Center right-click action menu

The Connect to the *database* window, shown in Figure 134, opens.

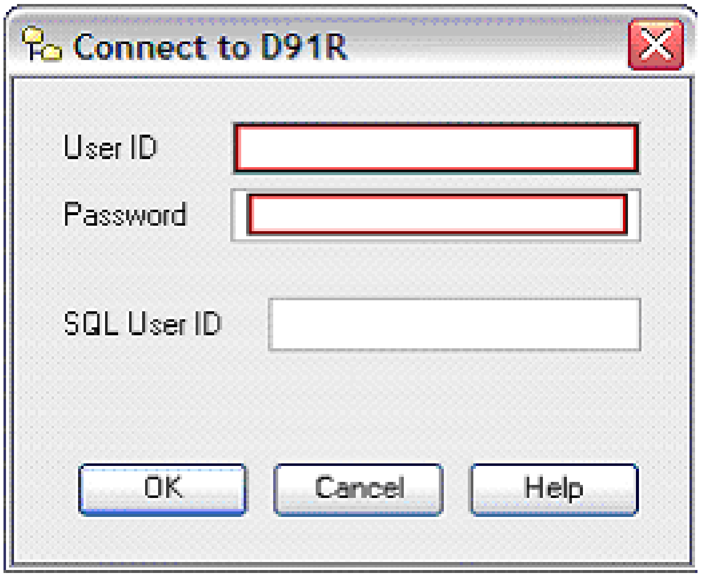
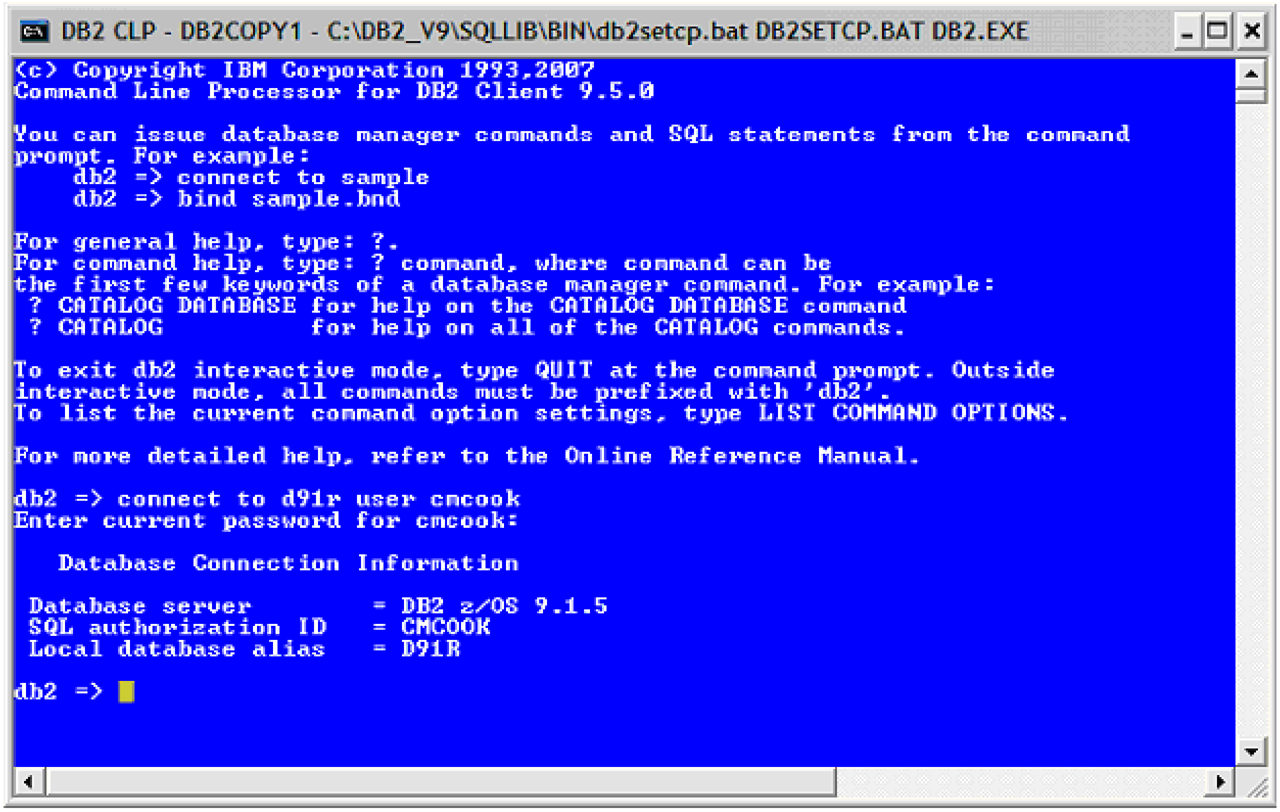


Figure 134. Connect to named database window

Enter the user ID and password for the remote DB2 on z/OS database, and press the **OK** button.

Testing the database connection using the DB2 command-line processor

Invoke the DB2 command-line Processor, shown in Figure 135.


DB2 CLP - DB2COPY1 - C:\DB2_V9\SQLLIB\BIN\db2setcp.bat DB2SETCP.BAT DB2.EXE

```
(c) Copyright IBM Corporation 1993,2007
Command Line Processor for DB2 Client 9.5.0

You can issue database manager commands and SQL statements from the command
prompt. For example:
    db2 => connect to sample
    db2 => bind sample.bnd

For general help, type: ?.
For command help, type: ? command, where command can be
the first few keywords of a database manager command. For example:
 ? CATALOG DATABASE for help on the CATALOG DATABASE command
 ? CATALOG          for help on all of the CATALOG commands.

To exit db2 interactive mode, type QUIT at the command prompt. Outside
interactive mode, all commands must be prefixed with 'db2'.
To list the current command option settings, type LIST COMMAND OPTIONS.

For more detailed help, refer to the Online Reference Manual.

db2 => connect to d91r user cmcook
Enter current password for cmcook:

   Database Connection Information

 Database server        = DB2 z/OS 9.1.5
 SQL authorization ID   = CMCOOK
 Local database alias   = D91R

db2 =>
```

Figure 135. DB2 command-line Processor window. Note that the password is hidden in this figure.

Enter the following command at the db2 prompt:

```
connect to database_name user userid using password
```

where *database_name* is the name you assigned to the DB2 on z/OS database that contains your data warehouse, and *userid* and *password* are the DB2 user ID and password required to access that database.

The database information is displayed, as shown in Figure 135.

Scenario: Connecting a Tivoli Data Warehouse solution using DB2 on z/OS to a Tivoli Enterprise Monitoring Server running z/OS

This scenario covers both Windows and UNIX/Linux machines. All agent configuration windows in this scenario should point to the hub Tivoli Enterprise Monitoring Server on z/OS.

On Windows:

A Warehouse Proxy Agent on Windows communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the DB2 on z/OS client. You must install a DB2 client on the Windows computer where the Warehouse Proxy Agent is located, and then catalog the remote node and database on the local computer.

Note: If you install a version 9 DB2 client, you also must install DB2 Connect Server Edition on the workstation to connect to a DB2 on z/OS data server.

- You must first create an ODBC datasource on the Windows machine that will connect to the DB2 on z/OS database. You can do this using the DB2 Client Configuration Assistant tool, or through the command-line. For instructions on using the DB2 Client Configuration Assistant tool, see “Step 1: Connect the Warehouse Proxy node to your DB2 on z/OS database” on page 524.
- Or run the following commands:

```
db2 catalog tcpip node DBNODE remote DBserverhostname server port number ostype OS390
db2 catalog dcs database db_name as db_name
db2 catalog db databasename-on server as alias-on client-databasename at node DBNODE authentication dcs
```

Use this procedure to reconfigure the Warehouse Proxy with the ODBC datasource you just created:

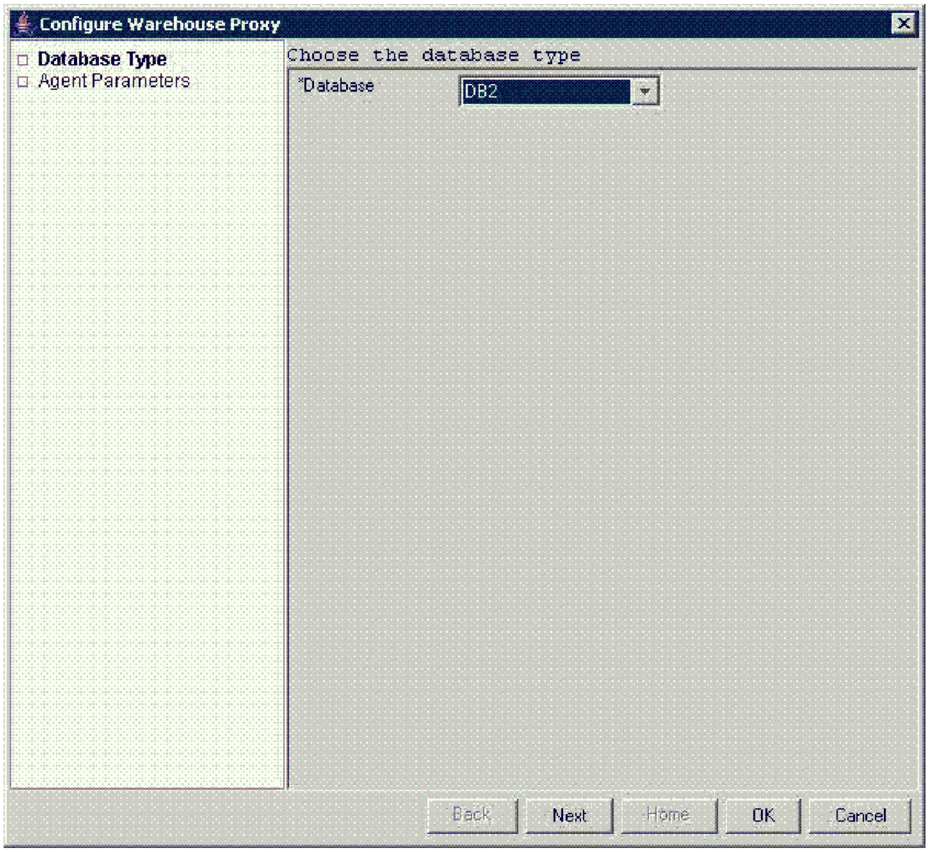
1. Log on to the Windows system where the Warehouse Proxy Agent is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Warehouse Proxy** and click **Configure Using Defaults**.

      If the Warehouse Proxy is installed on the same computer as the portal server, click **Reconfigure**.

   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Warehouse Proxy: Agent Advanced Configuration) contain the settings for the connection between the Warehouse Proxy Agent and the hub monitoring server. These settings were specified when the Warehouse Proxy Agent was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to configure the ODBC data source.
4. Select **DB2** from the list of selectable databases (see Figure 112 on page 503), and click **Next**

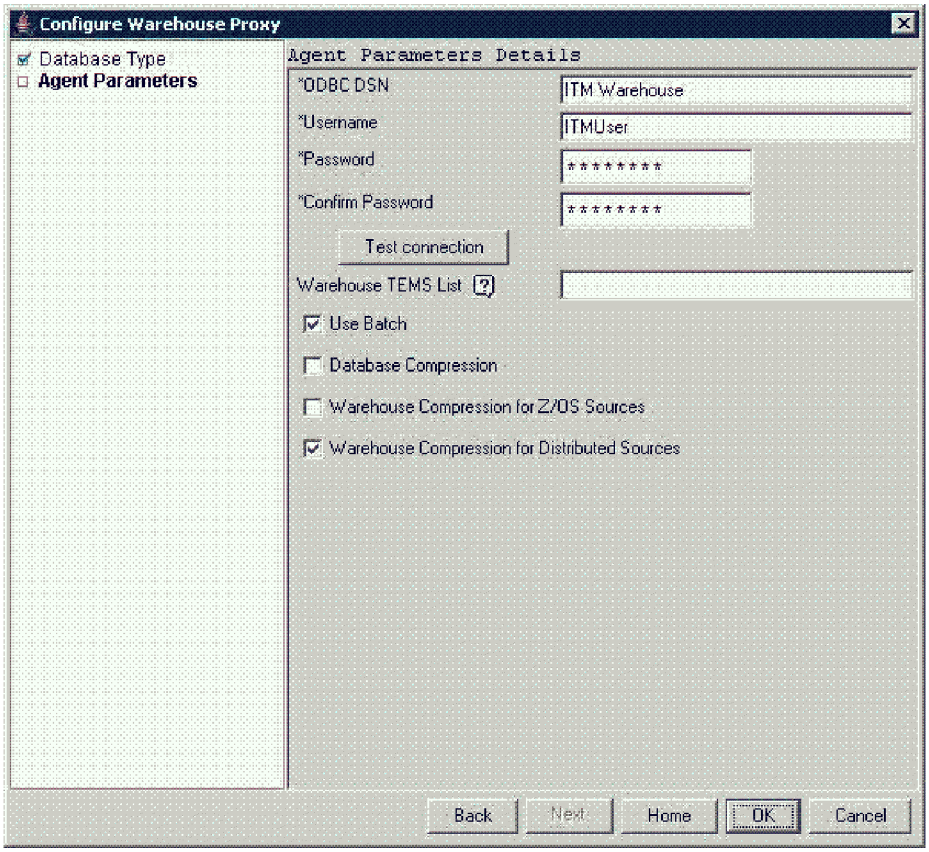


Choose the database type

*Database DB2

Figure 136. Warehouse Proxy Database Selection screen

The following configuration window is displayed.



Configure Warehouse Proxy

☑ Database Type
□ Agent Parameters

Agent Parameters Details

*ODBC DSN: ITM Warehouse
*Username: ITMUser
*Password: ********
*Confirm Password: ********

Warehouse TEMS List:

☑ Use Batch
□ Database Compression
□ Warehouse Compression for Z/OS Sources
☑ Warehouse Compression for Distributed Sources

Figure 137. Configure DB2 Data Source for Warehouse Proxy window

5. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in Table 84 on page 504.

   Note: The values for the data source name, database name, and database user ID and password must match the values that you used when configuring an ODBC connection for the Warehouse Proxy Agent. See “Configuring an ODBC data source for a DB2 data warehouse” on page 501.

Table 96. Configuration information for the Tivoli Data Warehouse database on DB2 running on z/OS

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.
Username	ITMUser	The name of the user that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password for the user that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Confirm Password	itmpswd1	Confirm the password by entering it again.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password.

Table 96. Configuration information for the Tivoli Data Warehouse database on DB2 running on z/OS (continued)

Field	Default value	Description
Warehouse TEMS List		Environment variable containing a space delimited list of TEMS names, which are given during the configuration of the HTEMS or an RTEMS. A Tivoli Enterprise Monitoring Server name in this field indicates that all the agents connected to this Tivoli Enterprise Monitoring Server will have their historical data sent to this Warehouse Proxy Agent. This variable is used when the IBM Tivoli Monitoring environment contains multiple Warehouse Proxy Agents and the workload has to be balanced using specific Warehouse Proxy Agents.
Use Batch		Batch inserts introduced with IBM Tivoli Monitoring V6.2.2 fix pack 2 increase the data insertion rate of the Warehouse Proxy Agent. This is especially true if the proxy and the warehouse are located on different hosts. Batch inserts are supported for both ODBC and JDBC warehouse connections. Using batch inserts is recommended in all configurations, but they increase the load on the data warehouse.
Database Compression		If the database compression mode is supported by the database, the Warehouse Proxy Agent creates all tables and indexes in the Tivoli Data Warehouse with compression enabled. This option reduces the storage costs of the Tivoli Data Warehouse.
Warehouse Compression for Z/OS Sources		Select this option for the Warehouse Proxy server to allow clients installed on z/OS machines to send compressed data.
Warehouse Compression for Distributed Sources		Select this option for the Warehouse Proxy server to allow clients installed on distributed machines (those running Linux/UNIX or Windows) to send compressed data.

6. Click OK.

On UNIX/Linux:

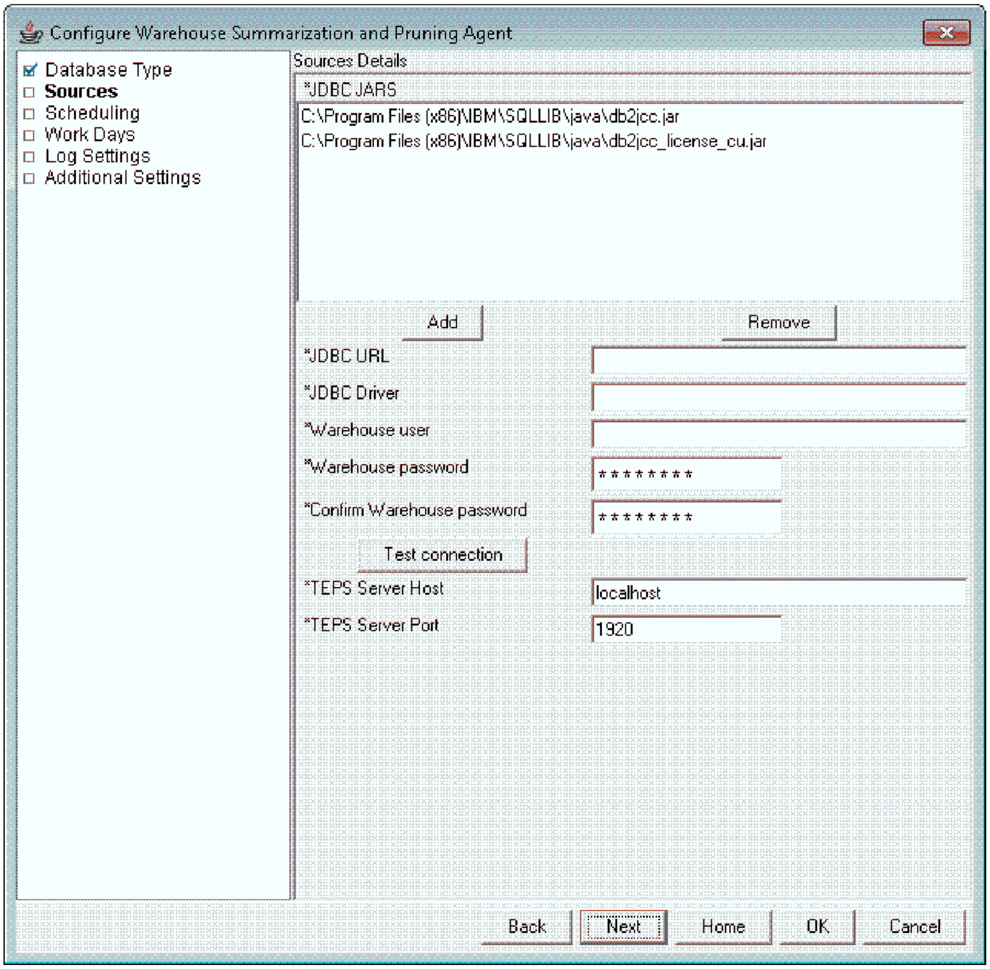
1. Copy the DB2 JAR files from the DB2 server or DB2 connect to the local computer where the Warehouse Proxy Agent is installed. You can copy these files to any directory on the local computer.
   ```
   db2jcc.jar
   db2jcc_license_cisuz.jar
   ```
2. For instructions on configuring a Warehouse Proxy Agent on Linux or UNIX, see “Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)” on page 558.

A Warehouse Proxy Agent on Linux or AIX communicates with the warehouse database through a JDBC connection. Install a Type 4 driver (DB2 on z/OS JDBC Universal Driver) on the computer where the Warehouse Proxy Agent is located. The default DB2 JDBC Driver is: com.ibm.db2.jcc.DB2Driver.

Summarization and Pruning Agent (JDBC connection):

1. The JDBC driver JAR files for your database platform must be located on the computer where you installed the Summarization and Pruning Agent. The files are named `db2jcc.jar` and `db2jcc_license_cisuz.jar`. Use a Type 4 JDBC driver. Do not use the Type 2 driver. Table 119 on page 596 shows where to obtain the driver files for each database platform.
2. Now you can configure the Summarization and Pruning Agent. You must add the names and directory locations of the JDBC driver JAR files to the JDBC Drivers list box. You then verify the default values for the database platform that are displayed in the other text fields on the Sources pane. For instructions on configuring the Summarization and Pruning Agent, see “Configuring the Summarization

and Pruning Agent (JDBC connection)" on page 595.


Configure Warehouse Summarization and Pruning Agent

☑ Database Type
□ **Sources**
□ Scheduling
□ Work Days
□ Log Settings
□ Additional Settings

Sources Details

*JDBC JARS

C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar
C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar

*JDBC URL

*JDBC Driver

*Warehouse user

*Warehouse password: ********

*Confirm Warehouse password: ********

*TEPS Server Host: localhost

*TEPS Server Port: 1920

Figure 138. Sources pane of Configure Summarization and Pruning Agent window

Chapter 22. Tivoli Data Warehouse solution using Microsoft SQL Server

Use the information and instructions in this chapter to implement a Tivoli Data Warehouse solution using Microsoft SQL Server for the warehouse database. The following table lists the goals for achieving a Microsoft SQL solution.

Table 97. Goals for achieving a Tivoli Data Warehouse solution using Microsoft SQL Server

Goal	Where to find information
Review your options, specific to a Microsoft SQL solution, for operating system platforms and communications between warehousing components.	"Supported components" on page 548
Complete prerequisite configuration steps before implementing your Tivoli Data Warehouse solution.	"Prerequisite installation" on page 549
Understand how to use the instructions for implementing your Tivoli Data Warehouse solution.	"Implementing a Tivoli Data Warehouse solution using Microsoft SQL Server" on page 550
Complete the steps for implementing your Tivoli Data Warehouse solution using Microsoft SQL Server for the data warehouse.	"Step 1: Create the Tivoli Data Warehouse database" on page 552 "Step 2: Install and configure communications for the Warehouse Proxy Agent" on page 554 "Step 3: Configure communications between the Tivoli Enterprise Portal Server and the data warehouse" on page 562 "Step 4: Install and configure communications for the Summarization and Pruning Agent" on page 565 "Step 5: Install and configure communications for Tivoli Performance Analyzer" on page 566

Supported components

Figure 139 presents the options for a Tivoli Data Warehouse solution using Microsoft SQL Server for the warehouse database. The diagram summarizes the supported operating system platforms for the various warehousing components, the supported database products, and the connections between components. For more specific information about supported operating systems and database products, including product names and versions, see “Hardware and software requirements” on page 138.

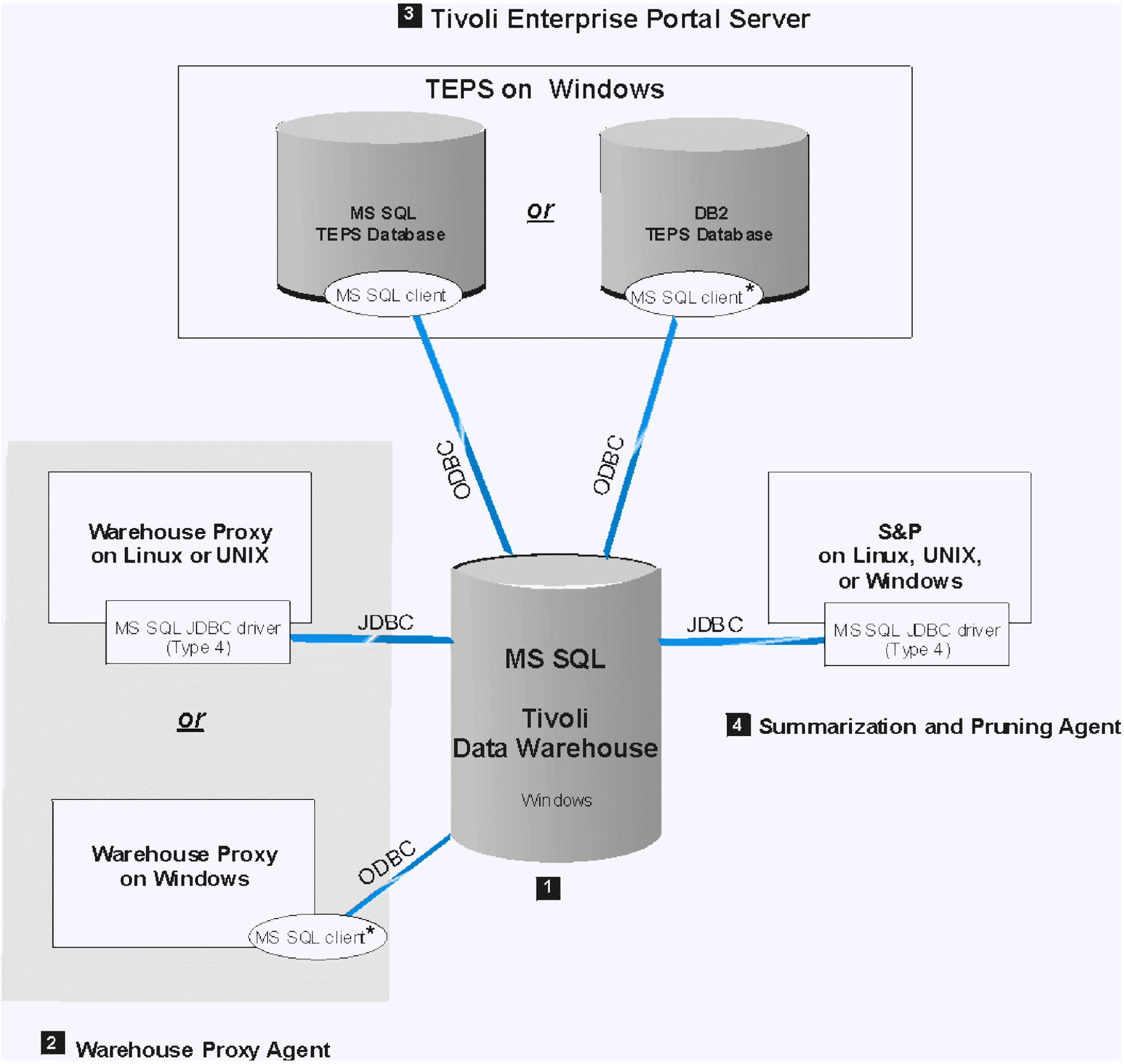

3 Tivoli Enterprise Portal Server
TEPS on Windows
MS SQL TEPS Database
MS SQL client
or
DB2 TEPS Database
MS SQL client*
ODBC
ODBC
Warehouse Proxy on Linux or UNIX
MS SQL JDBC driver (Type 4)
JDBC
or
Warehouse Proxy on Windows
MS SQL client*
ODBC
MS SQL
Tivoli Data Warehouse
Windows
1
JDBC
S&P on Linux, UNIX, or Windows
MS SQL JDBC driver (Type 4)
4 Summarization and Pruning Agent
2 Warehouse Proxy Agent

Figure 139. Tivoli Data Warehouse solution using Microsoft SQL Server

Note: An asterisk (*) next to a database client indicates that you must manually install the client if it does not already exist.

In the following discussion, numbered product components correspond to the numbers on the diagram.

1 Tivoli Data Warehouse on Microsoft SQL Server

A Tivoli Data Warehouse database on Microsoft SQL Server can be installed on supported Windows platforms.

2 **Warehouse Proxy Agent**

A Warehouse Proxy Agent on Linux or AIX communicates with the warehouse database through a JDBC connection. Install a Microsoft SQL Type 4 driver on the computer where the Warehouse Proxy is located.

Important: Use the 2005 SQL driver even if you are connecting to a warehouse database that was created in Microsoft SQL Server 2000.

A Warehouse Proxy Agent on Windows communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the Microsoft SQL Server client. If the Tivoli Data Warehouse is located on a remote computer, install a Microsoft SQL Server client on the local computer where the Warehouse Proxy Agent is located. Also, configure a remote client connection to the Tivoli Data Warehouse.

3 **Tivoli Enterprise Portal Server**

A Tivoli Enterprise Portal Server on Windows can connect to a Microsoft SQL Server data warehouse through a Microsoft SQL Server client installed on the portal server. If the *portal server database* (designated as *TEPS database* in the diagram) uses Microsoft SQL Server, the client already exists. *Manually* install a Microsoft SQL Server client on the portal server only if the portal server database uses DB2 for Linux, UNIX, and Windows.

The portal server communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the Microsoft SQL Server client. Configure a remote client connection to the Tivoli Data Warehouse.

4 **Summarization and Pruning Agent**

The Summarization and Pruning Agent communicates with the warehouse database through a JDBC connection from any supported operating system. Install a Microsoft SQL Type 4 JDBC driver on the computer where the Summarization and Pruning Agent is located.

Important: Use the 2005 SQL driver even if you are connecting to a warehouse database that was created in Microsoft SQL Server 2000.

Prerequisite installation

Before you implement your Tivoli Data Warehouse solution, complete one or more hub installations, *excluding the warehousing components*. Include the following components in each hub installation:

- The hub Tivoli Enterprise Monitoring Server
- *(Optional)* One or more remote monitoring servers
- The Tivoli Enterprise Portal Server, including the prerequisite RDBMS for the portal server database (DB2 for Linux, UNIX, and Windows or Microsoft SQL Server)
- A Microsoft SQL Server instance on the computer where you will create the Tivoli Data Warehouse database. (The Tivoli Data Warehouse database can be shared in a multi-hub installation or dedicated to a single hub.) The SQL Server instance must be patched to the current service pack level.
- *(Optional)* A portal desktop client
- *(Optional)* Monitoring agents, and the application support for the monitoring agents

Note: The term *monitoring agent*, as used here, refers to agents that collect data directly from managed systems, not the Warehouse Proxy Agent or Summarization and Pruning Agent.

- *(Optional)* The Tivoli Performance Analyzer
- *(Optional)* Language packs for all languages other than English

See the following table for related information:

Table 98. Information topics related to installation of prerequisite software for a Tivoli Data Warehouse solution

Topic	**Where to find information**
Single and multiple hub installations	To understand the terminology related to single and multiple hub installations, see “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46.
Installation procedures for prerequisite components	The detailed instructions for installing the prerequisite components are described in Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207. See your database documentations for instructions on how to install a supported database server.
Supported RDBMS versions	For specific information about the supported database platforms for the portal server database and the Tivoli Data Warehouse, see “Hardware and software requirements” on page 138.

Implementing a Tivoli Data Warehouse solution using Microsoft SQL Server

Use the instructions in the remainder of this chapter to implement a Tivoli Data Warehouse solution using Microsoft SQL Server for the data warehouse.

Assumptions

The implementation instructions are based on the following assumptions:

- You will create the Tivoli Data Warehouse database on a different computer from the Tivoli Enterprise Portal Server.
- You will create a single Tivoli Data Warehouse database, to be used either within a single hub installation or to be shared in a multi-hub installation. If you have multiple independent hub installations, repeat the implementation steps for each hub installation. (See “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46 for information about hub installations.)
- No assumption is made about where you will install the Warehouse Proxy Agent and Summarization and Pruning Agent. Either of these agents may be installed on the same computer as the Tivoli Data Warehouse or on a different computer.

Solution steps

To implement your Tivoli Data Warehouse solution using Microsoft SQL Server, complete the four major steps described in the remaining sections of this chapter, in the order listed:

1. Create the Tivoli Data Warehouse database.
2. Install and configure communications for the Warehouse Proxy Agent.
3. Configure communications between the Tivoli Enterprise Portal Server and the data warehouse.
4. Install and configure communications for the Summarization and Pruning Agent.

Except for Step 1, each major step consists of a series of installation and configuration tasks, listed and described in a table. Use the step tables as a road map for implementing your solution. The step tables describe the tasks at a high level, account for variations among configuration options (such as which operating system is used for a component), and reference the appropriate sections for detailed implementation procedures. To implement your solution successfully:

- Perform the tasks in the order listed in the table.
- Do not skip a table to the procedures that follow it.

  Be aware that some of the implementation procedures referenced in a table are included in this chapter and some are documented elsewhere. In some cases, the task is described in the table, without referencing a separate procedure. Read and follow all instructions in the tables.

Step 1: Create the Tivoli Data Warehouse database

This section provides guidelines for creating the Tivoli Data Warehouse database using Microsoft SQL Server 2000, 2005, and 2008. For specific instructions on how to create a Microsoft SQL Server database, see the Microsoft SQL Server documentation or have a database administrator create the database for you.

When you create the warehouse database using Microsoft SQL Server, follow these guidelines:

- Connect to the Microsoft SQL database server and create the Tivoli Data Warehouse database using the system administrator (sa) user.
- Create a database user login name and password that the warehousing components (portal server, Warehouse Proxy Agent, and Summarization and Pruning Agent) can use to access the data warehouse. In these instructions, this user account is referred to as the *warehouse user*.

  You must have SQL Server authentication to create the warehouse user.

  Note: The warehousing components must *not* use the system administrator (sa) user to connect to the data warehouse.
- Consider using the default values shown in the following table for the warehouse name and warehouse user. The default values are used in the configuration procedures for connecting the warehousing components to the warehouse database.

Table 99. Default values for Tivoli Data Warehouse parameters

Parameter	Default value
Tivoli Data Warehouse database name **Note:** When connecting to a DB2 on z/OS database, this value is unnecessary.	WAREHOUS
User name	ITMUser
User password	itmpswd1

- If the Warehouse Proxy and Summarization and Pruning Agents create database objects at runtime, you must give the warehouse user **public** and **db_owner** privileges to the Tivoli Data Warehouse database.

  The warehouse user may have much fewer rights if the schema publication tool is used to create the database objects. If the schema tool is used, the warehouse user needs only the **db_datareader** and **db_datawriter** roles. If the warehouse user has limited privileges, the schema tool must be used to create any additional database objects (using the schema tool's updated mode) if the historical configuration is changed; see Chapter 19, “Schema Publication Tool,” on page 483.
- For Microsoft SQL Server 2005 and 2008, do the following:
  - Create a schema with the same name (and owner) as the database user login name (for example, `ITMUser`) and change the default schema for the user from `dbo` to this login name. (This step is not necessary if you are using Microsoft SQL Server 2000.)
  - Make sure the database is set up to support inbound network TCP/IP connections.

Limiting the authority of the warehouse user

The warehouse user only needs authority for the following activities in the warehouse database:

- Create tables, alter tables, and create indexes on tables.
- Create views.
- Create functions.

To limit the authority of the warehouse user, perform the following steps in either the Microsoft SQL Server Management Studio GUI or the `osql` command-line. You must be logged into the Tivoli Data Warehouse database as a user with administrative privileges:

1. Remove all fixed database and server roles from the warehouse user such as `db_datareader` and `db_datawriter`. If using the command-line, use the `sp_helpuser` stored procedure to see what roles are assigned to the warehouse user and `sp_droprolemember` and `sp_dropsrvrolemember` to remove the warehouse user from roles.
2. Grant the minimal necessary privileges to the warehouse user. Issue the SQL command: `grant create table, create function, create view to <warehouse user>`. Where `<warehouse user>` is the warehouse user name.

Perform these additional steps if your security policy prohibits use of the create table, create view and create function privileges:

1. Use the schema tool to generate the DDL that create the database objects.

   Note: You should create the historical collections that you want first and then configure the Summarization and Pruning Agent so that you can use the schema tool's configured mode.
2. Execute the generated scripts as described in Chapter 19, “Schema Publication Tool,” on page 483. Run the scripts as the Tivoli Data Warehouse user so that the Tivoli Data Warehouse user has sufficient privileges on the tables.
3. Revoke the create table, create view, and create function privileges from the Tivoli Data Warehouse user.

Step 2: Install and configure communications for the Warehouse Proxy Agent

You can install one or more Warehouse Proxy Agents to collect and send historical data to the Tivoli Data Warehouse database. Complete the tasks described in the following table, in the order listed, to install and configure each Warehouse Proxy Agent.

Table 100. Tasks for installing and configuring communications for the Warehouse Proxy Agent

Task	Procedure
Install one or more Warehouse Proxy Agents. If you want to install a Summarization and Pruning Agent on the same computer as one of the Warehouse Proxy Agents, use the referenced procedures to install both agents at the same time. If you are installing more than one Warehouse Proxy Agent, each agent must be installed on a separate computer. The installation procedure for Windows includes steps for configuring the connection between the Warehouse Proxy and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (*Configuring the monitoring agent*). See the information at right. Be sure to perform all of the referenced installation and configuration procedures. **Note for sites setting up autonomous operation::** The installation procedure includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring Server. On Windows operating systems, if you want to run the Warehouse Proxy Agent without a connection to the hub, accept the defaults for the connection information, but specify a nonvalid name for the monitoring server. On UNIX and Linux operating systems, check **No TEMS** on the **TEMS Connection** tab of the configuration window.	To install a Warehouse Proxy Agent on Windows, complete the procedure "Windows: Installing a monitoring agent" on page 253. To install a Warehouse Proxy Agent on Linux or AIX, complete the procedure "Linux or UNIX: Installing a monitoring agent" on page 259, including the following subsections: • *Installing the monitoring agent* • *Configuring the monitoring agent* • *Changing the file permissions for agents* (if you used a non-root user to install the Warehouse Proxy) *Do not complete the procedure for starting the agent*.
(*Warehouse Proxy Agent on Windows only*) • Install a Microsoft SQL Server client on the computer where the Warehouse Proxy Agent is installed if *both* of the following statements are true: – The Warehouse Proxy is installed on Windows, and – The Warehouse Proxy needs to connect to a remote data warehouse. • Configure a remote client connection to the data warehouse server using Microsoft SQL Server tools.	See the Microsoft SQL Server documentation for instructions on how to install a Microsoft SQL client and configure a remote client connection.
(*Warehouse Proxy Agent on Windows only*) On the computer where the Warehouse Proxy Agent is installed, configure an ODBC data source for the data warehouse. Perform this procedure whether or not the Warehouse Proxy Agent and the warehouse database are installed on the same computer.	"Configuring an ODBC data source for a Microsoft SQL data warehouse" on page 555

Table 100. Tasks for installing and configuring communications for the Warehouse Proxy Agent (continued)

Task	Procedure
(*Warehouse Proxy Agent on Linux or AIX only*) Install the most current SQL Server JDBC driver on the computer where the Warehouse Proxy Agent is installed.	Go to the Microsoft Web page at: http://www.microsoft.com and search for **JDBC driver**. Follow the instructions on the Microsoft download page for installing the driver. After you install the driver, the JAR file name and location are as follows: `<mssqlinstalldir>/sqljdbc_1.1/enu/sqljdbc4.jar`
Configure the Warehouse Proxy Agent to connect to the data warehouse. Perform this procedure whether or not the Warehouse Proxy Agent and the warehouse database are installed on the same computer.	For a Warehouse Proxy Agent on Windows, see “Configuring a Warehouse Proxy Agent on Windows (ODBC connection)” on page 556. For a Warehouse Proxy Agent on Linux or AIX, see “Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)” on page 558.
If you are installing more than one Warehouse Proxy Agent within the same hub monitoring server installation, associate each Warehouse Proxy Agent with a subset of monitoring servers (hub or remote) within the installation. Each Warehouse Proxy Agent receives data from the monitoring agents that report to the monitoring servers on the list. Use the environment variable KHD_WAREHOUSE_TEMS_LIST to specify a list of monitoring servers to associate with a Warehouse Proxy Agent.	For instructions about installing and configuring multiple Warehouse Proxy Agents within a single hub monitoring server installation, see “Installing and configuring multiple Warehouse Proxy Agents” on page 608.
(*Optional*) Customize the configuration of the Warehouse Proxy Agent for tuning performance.	“Tuning the performance of the Warehouse Proxy” on page 617
Start the Warehouse Proxy Agent.	“Starting the Warehouse Proxy Agent” on page 562

Configuring an ODBC data source for a Microsoft SQL data warehouse

A Microsoft SQL client on Windows requires an ODBC connection to the data warehouse. For the Warehouse Proxy Agent, you must configure the ODBC connection manually.

Complete the following procedure to set up an ODBC connection for a Warehouse Proxy Agent on Windows to a local or remote Tivoli Data Warehouse.

Note: This procedure uses default values for the data source name and warehouse user ID. (Default values are used in configuration procedures for warehousing components.) Substitute different values if you do not want to use the default values.

1. Open the Control Panel.
2. Click **Administrative Tools → Data Sources (ODBC)**
3. Click **Add** in the **System DSN** tab in the ODBC Data Source Administrator window.
4. Select **SQL Server** and click **Finish**.
5. Enter `ITM Warehouse` in the **Name** field.
6. Select the Microsoft SQL Server where the Tivoli Data Warehouse is located from the drop down list and click **Next**.
7. Select **With SQL Server authentication using a login ID and password entered by the user**.

8. Enter `ITMUser` for the **Login ID**.

   The user ID must match exactly, including case, what was created in the SQL Server database if using database authentication or the ID that was created in the Windows OS. Mixing the case will cause tables to be created with dbo as the owner rather than ITMUser. This will cause many warehousing components to not work correctly.
9. Type a password for the user in the **Password** field. The default password is `itmpswd1`.
10. Click **Next**.
11. Click **Next** again.
12. Click **Finish**.
13. Click **Test Data Source** to test the connection to the database.
14. Click **OK**.

Configuring a Warehouse Proxy Agent on Windows (ODBC connection)

Use this procedure to configure a Warehouse Proxy Agent on Windows to connect to a Tivoli Data Warehouse in Microsoft SQL Server:

1. Log on to the Windows system where the Warehouse Proxy Agent is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

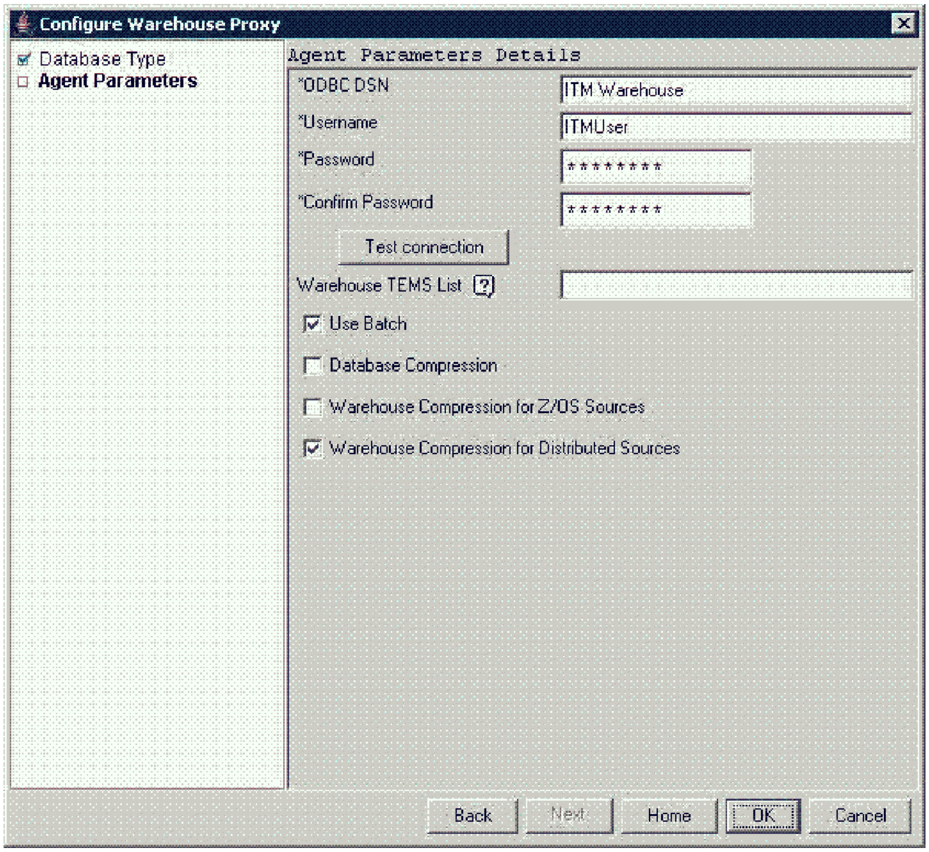
      The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Warehouse Proxy** and click **Configure Using Defaults**.

      Click **Reconfigure** if the Warehouse Proxy is installed on the same computer as the portal server.

   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Warehouse Proxy: Agent Advanced Configuration) contain the settings for the connection between the Warehouse Proxy Agent and the hub monitoring server. These settings were specified when the Warehouse Proxy Agent was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to configure the ODBC data source.
4. Select **SQL Server** from the list of databases and click **Next**.

   The following configuration window is displayed.



*ODBC DSN: ITM Warehouse

*Username: ITMUser

*Password: ********

*Confirm Password: ********

Warehouse TEMS List ?

☑ Use Batch

□ Database Compression

□ Warehouse Compression for Z/OS Sources

☑ Warehouse Compression for Distributed Sources

Figure 140. Configure SQL Data Source for Warehouse Proxy window

5. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in Table 101.

   Note: The values for the data source name, and database user ID and password must match the values that you used when configuring an ODBC connection for the Warehouse Proxy Agent. (See "Configuring an ODBC data source for a Microsoft SQL data warehouse" on page 555.)

Table 101. Configuration information for the Tivoli Data Warehouse database on Microsoft SQL Server

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.
Username	ITMUser	The name of the user that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Confirm Password	itmpswd1	Confirm the password by entering it again.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password. **Note:** Test Connection is not available if configuring remotely from the Tivoli Enterprise Portal.

Table 101. Configuration information for the Tivoli Data Warehouse database on Microsoft SQL Server (continued)

Field	Default value	Description
Warehouse TEMS List		Environment variable containing a space delimited list of TEMS names, which are given during the configuration of HTEMS or RTEMS. A TEMS name in this field indicates that all the agents connected to this TEMS will have their historical data sent to this Warehouse Proxy Agent. This variable is used when the ITM environment contains multiple Warehouse Proxy Agents and the workload has to be balanced using specific Warehouse Proxy Agents.
Use Batch		Batch inserts introduced ITM V6.2.2 fix pack 2, can greatly increase the data insertion rate of the Warehouse Proxy Agent. This is especially true if the proxy and the warehouse are located on different hosts. Batch inserts are supported for ODBC warehouse connections. Using batch inserts is recommended in all configurations, but they will place increased load on the data warehouse.
Database Compression		If the database compression mode is supported by the database, the Warehouse Proxy Agent will create all tables and indexes in the Tivoli Data Warehouse with compression enabled. This option reduces the storage costs of the Tivoli Data Warehouse.
Warehouse Compression for Z/OS Sources		Select this option for the Warehouse Proxy server to allow clients installed on Z/OS machines to send compressed data.
Warehouse Compression for Distributed Sources		Select this option for the Warehouse Proxy server to allow clients installed on distributed machines (Linux/UNIX, Windows) to send compressed data.

6. Click **OK**.

Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)

Use this procedure to configure a Warehouse Proxy Agent on Linux or UNIX to connect to a Microsoft SQL Server data warehouse:

1. Log on to the computer where the Warehouse Proxy Agent is installed and begin the configuration.
   a. Change to the *install_dir*/bin directory and run the following command:

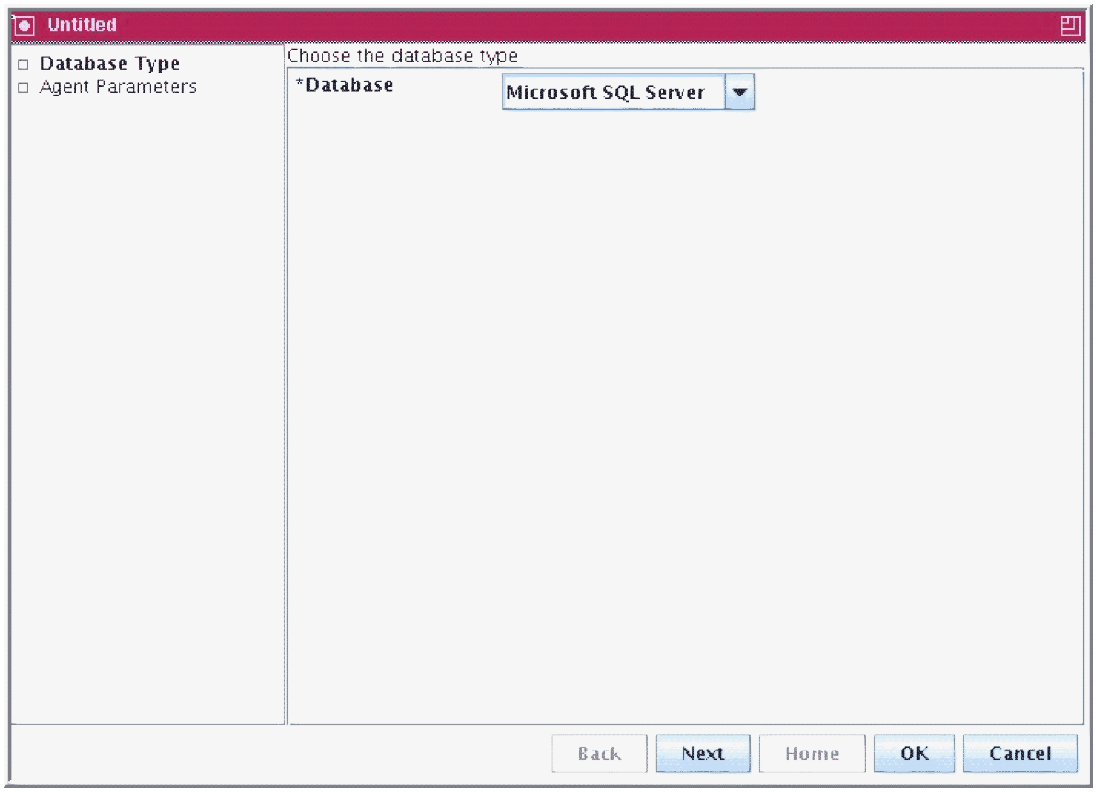
   ```
   ./itmcmd manage [-h install_dir]
   ```

   where *install_dir* is the installation directory for IBM Tivoli Monitoring. The default installation directory is /opt/IBM/ITM.

   The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Warehouse Proxy** and click **Configure**.

   The Configure Warehouse Proxy window is displayed.

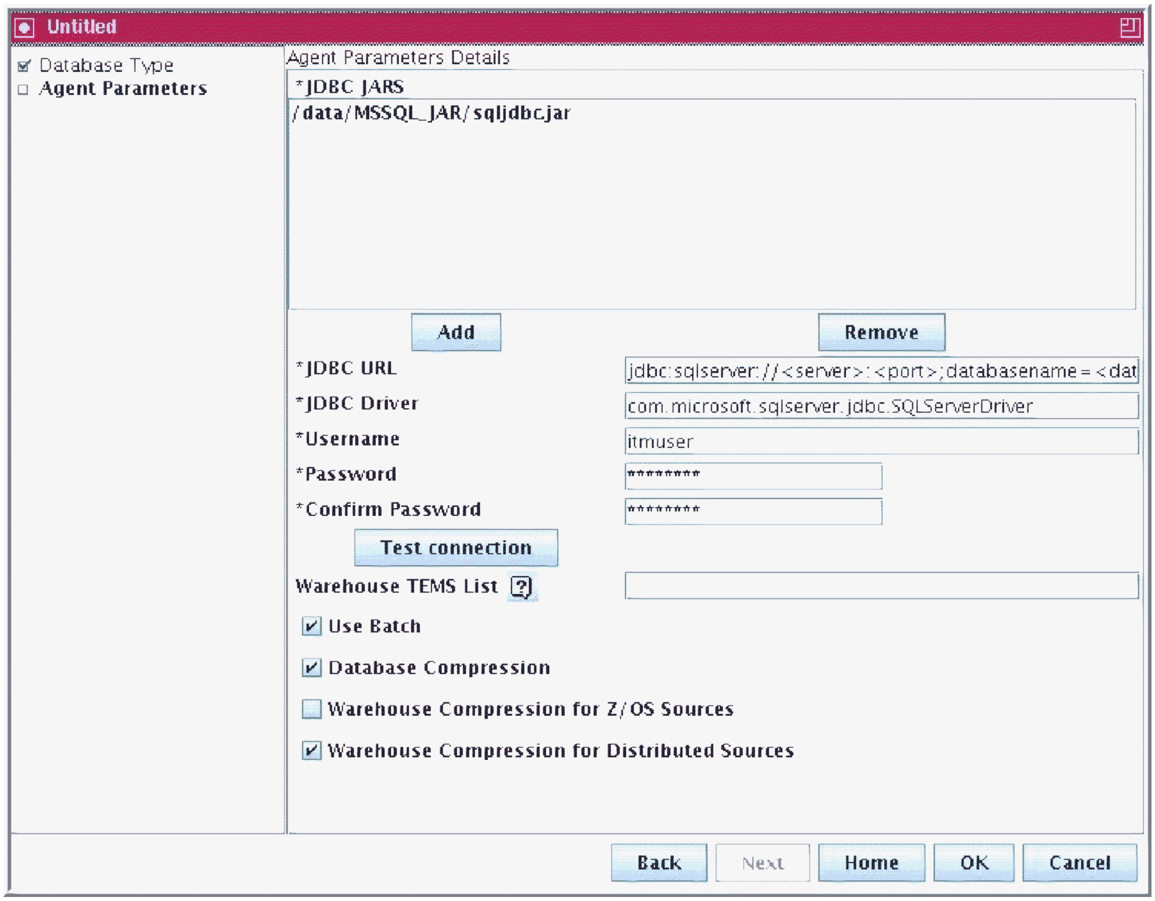


Choose the database type

*Database: Microsoft SQL Server

Figure 141. Configure Warehouse Proxy window (Database Type)

2. Select **Microsoft SQL Server** from the list of selectable databases, and click **Next**. The following configuration window is displayed.



*JDBC JARS
/data/MSSQL_JAR/sqljdbc.jar

*JDBC URL jdbc:sqlserver://<server>:<port>;databasename=<dat

*JDBC Driver com.microsoft.sqlserver.jdbc.SQLServerDriver

*Username itmuser

*Password *********

*Confirm Password *********

Warehouse TEMS List

☑ Use Batch

☑ Database Compression

□ Warehouse Compression for Z/OS Sources

☑ Warehouse Compression for Distributed Sources

Figure 142. Configure Warehouse Proxy window (Agent Parameters tab)

3. Review the settings for the connection between the Warehouse Proxy Agent and the hub monitoring server. Correct the settings if necessary.

   The Warehouse Proxy Agent must use the same protocols used by the application agents and by the hub monitoring. If the proxy agent does not have the same protocol as the hub monitoring server, it cannot register with the hub. If the proxy does not have the same protocol as the application agents, then the application agents cannot communicate with the proxy when they to create a route to it.

4. Add the name and directory location of the JDBC driver JAR file to the **JDBC Drivers** list box:

   a. Click **Add** to display the file browser window. Navigate to the location of the JDBC driver JAR file on this computer and select the sqljdbc4.jar file.

      Important: Use the latest SQL Server JDBC driver from Microsoft, which supports SQL Server 2008, 2005 and 2000.

   b. Click **OK** to close the browser window and add the driver file to the list.

   If you need to delete an entry from the list, select the entry and click **Remove**.

5. Change the default value displayed in the **Warehouse URL** field if it is not correct. The Warehouse URL for Microsoft SQL Server 2005 is as follows:

   jdbc:sqlserver://localhost:1433;databaseName=WAREHOUS;SelectMethod=cursor

   - If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of `localhost`.
   - Change the port number if it is different.

- If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name. (See Table 99 on page 552.)

6. Verify the JDBC driver name, which is displayed in the **JDBC Driver** field. (Note that the **JDBC Driver** field displays the *driver name*, in contrast to the *JDBC JARS* that are listed in the **JDBC JARS** field.)

   The Microsoft SQL Server Driver name is:

   com.microsoft.sqlserver.jdbc.SQLServerDriver
7. If necessary, change the entries in the **Username** and **Password** fields to match the user name and password that were created for the Tivoli Data Warehouse. (See "Step 1: Create the Tivoli Data Warehouse database" on page 552.) The default user name is `ITMUser` and the default password is `itmpswd1`.
8. Click **Test connection** to ensure you can communicate with the Tivoli Data Warehouse database.
9. A Tivoli Enterprise Monitoring Server name in the **Warehouse TEMS List** field indicates that all the agents connected to this Tivoli Enterprise Monitoring Server will have its historical data sent to this Warehouse Proxy Agent. This variable is used when the IBM Tivoli Monitoring environment contains multiple Warehouse Proxy Agents and the workload has to be balanced using specific Warehouse Proxy Agents.
10. Select the **Use Batch** check box if you want the Warehouse Proxy Agent to submit multiple execute statements to the Tivoli Data Warehouse database for processing as a batch.

    In some situations, such as crossing a network, sending multiple statements as a unit is more efficient than sending each statement separately. Batch processing is one of the features provided with the JDBC 2.0 API.
11. Select the **Database Compression** check box for the Warehouse Proxy Agent to create all tables and indexes in the Tivoli Data Warehouse with compression enabled, if the database compression mode is supported by the database. This option reduces the storage costs of the Tivoli Data Warehouse.
12. Select the **Warehouse Compression for Z/OS Sources** check box for the Warehouse Proxy server to allow clients installed on Z/OS machines to send compressed data.
13. Select the **Warehouse Compression for Distributed Sources** check box for the Warehouse Proxy server to allow clients installed on distributed machines to send compressed data.
14. Click **Save** to save your settings and close the window.

Configuring the Warehouse Proxy Agent on Linux or UNIX: command-line procedure

Complete the following steps to configure the Warehouse Proxy Agent from the command-line on Linux or UNIX:

1. Log on to the computer where the Warehouse Proxy Agent is installed.
2. At the command-line change to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the directory where you installed the product.
3. Run the following command to start configuring the Warehouse Proxy Agent:

   ```
   ./itmcmd config -A hd
   ```

   where `hd` is the product code for the Warehouse Proxy Agent.

Here is a sample of Warehouse Proxy Agent configuration from the command-line:

```
itmcmd config -A hd

Database Type
Database [ 1=DB2, 2=Oracle, 3=Microsoft SQL Server ] (default is: 1):

Agent Parameters :
```

```
Fully qualified paths to JDBC JAR files (comma separated)
      JDBC JARs List (default is: ): /data/jdbc/db2jcc.jar,/data/jdbc/db2jcc_license_cu.jar
The Warehouse JDBC URL
     JDBC URL (default is: jdbc:db2://localhost:50000/WAREHOUS):
The Warehouse JDBC Driver
     JDBC Driver (default is: com.ibm.db2.jcc.DB2Driver):
The Warehouse database username
     Username (default is: ITMUSER):
The Warehouse database user password
     Enter Password (default is: ):
     Re-type : Password (default is: ):
Space or comma separated list of Tivoli Enterprise Monitoring Server instances served by
this Warehouse Proxy agent.
*ANY can be specified if this Warehouse Proxy agent will export data of any agents connected
to any TEMS. If the list is left blank, this Warehouse Proxy agent will be the default
Warehouse proxy agent.
    Warehouse TEMS List (default is: ): REMOTE_ITMTDWP12
Batch Database Operations
    Use Batch [ 1=TRUE, 2=FALSE ] (default is: 1):
Database Compression option
     Database Compression [ 1=TRUE, 2=FALSE ] (default is: 2):
Enable the compression of historical data from Z/OS sources before upload to the
Warehouse Proxy Server
    Warehouse Compression for Z/OS Sources [ 1=TRUE, 2=FALSE ] (default is: 2):
Enable the compression of historical data from distributed sources before upload to the
Warehouse Proxy Server
    Warehouse Compression for Distributed Sources [ 1=TRUE, 2=FALSE ] (default is: 1):
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1): 1
     TEMS Host Name (Default is: itmtdwp18):
```

Starting the Warehouse Proxy Agent

Use the following steps to start the Warehouse Proxy Agent:

- To start the Warehouse Proxy Agent from the Manage Tivoli Enterprise Monitoring Services window, right-click **Warehouse Proxy** and select **Start**.
- (*Linux or AIX only*) To start the Warehouse Proxy Agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is /opt/IBM/ITM.

  ```
  ./itmcmd agent start hd
  ```

  where hd is the product code for the Warehouse Proxy Agent.

Step 3: Configure communications between the Tivoli Enterprise Portal Server and the data warehouse

Complete the tasks described in the following table, in the order listed, to configure communications between the portal server and the data warehouse.

Table 102. Tasks for configuring communications between the portal server and a Microsoft SQL Server data warehouse

Task	Procedure
If the *portal server database* was created using DB2 for Linux, UNIX, and Windows, install a Microsoft SQL Server database client on the portal server. If the portal server database was created using Microsoft SQL Server, the Microsoft SQL Server database client already exists on the portal server.	See the Microsoft SQL Server documentation for instructions on how to install a Microsoft SQL Server database client.
Configure a remote client connection to the data warehouse server using Microsoft SQL Server tools.	See the Microsoft SQL Server documentation for instructions on how to configure a remote client connection.

Table 102. Tasks for configuring communications between the portal server and a Microsoft SQL Server data warehouse (continued)

Task	Procedure
Configure the portal server to connect to the data warehouse. The configuration procedure automatically configures an ODBC connection to the data warehouse.	"Configuring the portal server (ODBC connection)"
Restart the portal server.	On the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server** and select **Start**.
Test the connection between the portal server and the Tivoli Data Warehouse by creating a customized query in the Tivoli Enterprise Portal.	"Testing the connection between the portal server and the Tivoli Data Warehouse" on page 614

Configuring the portal server (ODBC connection)

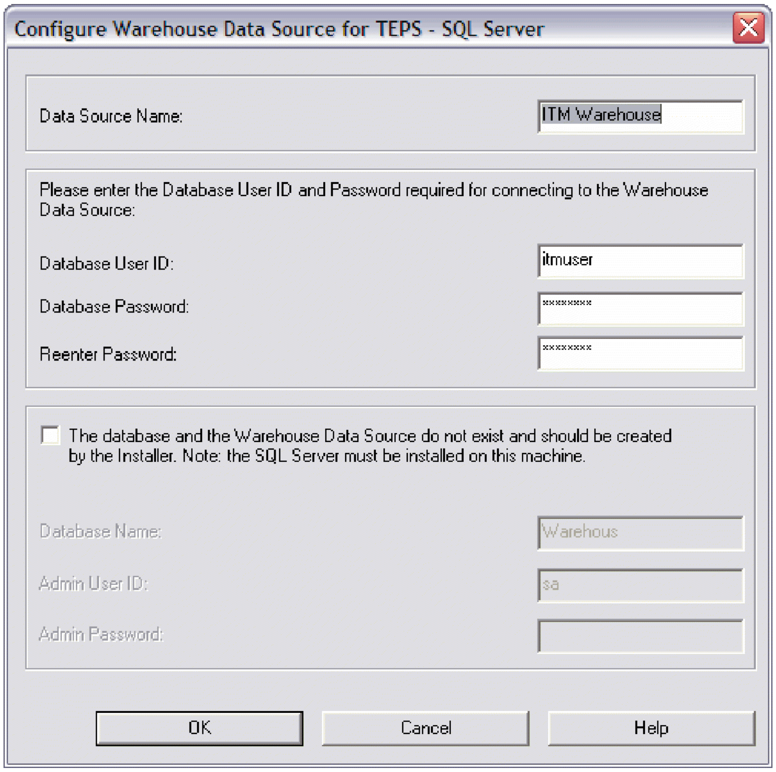
Use this procedure to configure the portal server to connect to a Tivoli Data Warehouse in Microsoft SQL Server:

1. Log on to the Windows system where the portal server is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Tivoli Enterprise Portal Server** and click **Reconfigure**.
2. The next two windows (entitled TEP Server Configuration) contain the settings for the connection between the portal server and the hub monitoring server. These settings were specified when the portal server was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to reconfigure the warehouse information for the Tivoli Enterprise Portal Server.
4. Select **SQL Server** from the list of databases and click **OK**.

   The following configuration window is displayed.



Configure Warehouse Data Source for TEPS - SQL Server

Data Source Name: ITM Warehouse

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

Database User ID: itmuser

Database Password: ********

Reenter Password: ********

□ The database and the Warehouse Data Source do not exist and should be created by the Installer. Note: the SQL Server must be installed on this machine.

Database Name: Warehous

Admin User ID: sa

Admin Password:

Figure 143. Configure SQL Data Source for Warehouse window

5. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in the following table:

Table 103. Configuration information for the Tivoli Data Warehouse database on Microsoft SQL Server

Field	Default value	Description
Data Source Name	ITM Warehouse	The name of the data source.
Database User ID	ITMUser	The login name of the database user that the portal server will use to access the Tivoli Data Warehouse database.
Database Password	(no default)	The password for the database login user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Reenter Password	(no default)	Confirm the password by entering it again.
Database Name	WAREHOUS	The name of the database.
Admin User ID	sa	The database administrator ID.
Admin Password	(no default)	The password for the database administrator.

6. Click **OK**.

Step 4: Install and configure communications for the Summarization and Pruning Agent

Complete the tasks described in the following table, in the order listed, to install and configure the Summarization and Pruning Agent.

Table 104. Tasks for installing and configuring communications for the Summarization and Pruning Agent

Task	Procedure
Install the Summarization and Pruning Agent if you have not already installed it. For best performance, install the Summarization and Pruning Agent on the same computer as the data warehouse. The installation procedure for Windows includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (*Configuring the monitoring agent*). See the information at right. Be sure to perform all referenced installation and configuration procedures. **Note**: The Summarization and Pruning Agent is not automatically started after installation. Do not complete any step or procedure for starting the agent at this point.	To install a Summarization and Pruning Agent on Windows, complete the procedure “Windows: Installing a monitoring agent” on page 253. To install a Summarization and Pruning Agent on Linux or UNIX, complete the procedure “Linux or UNIX: Installing a monitoring agent” on page 259, including the following subsections: • *Installing the monitoring agent* • *Configuring the monitoring agent* • *Changing the file permissions for agents* (if you used a non-root user to install the Warehouse Proxy) *Do not complete the procedure for starting the agent.*
Install the most current SQL Server JDBC driver on the computer where the Summarization and Pruning Agent is installed.	Go to the Microsoft Web page at: http://www.microsoft.com and search for **JDBC driver**. Follow the instructions on the Microsoft download page for installing the driver. After you install the driver, the JAR file name and location are as follows: `<mssqlinstalldir>/sqljdbc_1.1/enu/sqljdbc4.jar`
Configure the Summarization and Pruning Agent. When you configure the Summarization and Pruning Agent, you configure the connection to the Tivoli Data Warehouse and you specify settings that control the operation of the Summarization and Pruning Agent. Perform this procedure whether or not the Summarization and Pruning Agent and the warehouse database are installed on the same computer.	“Configuring the Summarization and Pruning Agent (JDBC connection)” on page 595
Configure the Summarization and Pruning Agent to connect to the Tivoli Enterprise Portal Server. Perform this procedure whether or not the Summarization and Pruning Agent and the warehouse database are installed on the same computer.	See Step 9 of “Configuring the Summarization and Pruning Agent (JDBC connection)” on page 595.
Configure history collection. When you configure history collection, you specify settings for how often to collect, aggregate, and prune data for individual monitoring agents and attribute groups. Configure history collection from the Tivoli Enterprise Portal.	See the *IBM Tivoli Monitoring: Administrator's Guide* for instructions on how to configure history collection.
Start the Summarization and Pruning Agent.	“Starting the Summarization and Pruning Agent” on page 608

Step 5: Install and configure communications for Tivoli Performance Analyzer

You can install Tivoli Performance Analyzer to a server which also has other Tivoli Monitoring components installed, or you can install it to a separate machine. The installation procedure is similar to that for monitoring agents. Complete the tasks described in the following table, in the order listed, to install and configure Tivoli Performance Analyzer.

Table 105. Tasks for installing and configuring communications for the Tivoli Performance Analyzer

Task	Procedure
Install Tivoli Performance Analyzer. The installation procedure for Windows includes steps for configuring the connection between the Tivoli Performance Analyzer and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (Configuring the monitoring agent). See the information opposite. Be sure to perform all of the referenced installation and configuration procedures. **Note:** for sites setting up autonomous operation, the installation procedure includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring Server. On Windows operating systems, if you want to run Tivoli Performance Analyzer without a connection to the hub, accept the defaults for the connection information, but specify a non valid name for the monitoring server. On UNIX and Linux operating systems, select **No TEMS** on the TEMS Connection in the configuration window.	To install Tivoli Performance Analyzer on Windows, complete the procedure “Windows: Installing a monitoring agent” on page 253. To install Tivoli Performance Analyzer agent on Linux or AIX, complete the procedure “Linux or UNIX: Installing a monitoring agent” on page 259, including the following subsections: • Installing the monitoring agent • Configuring the monitoring agent • Changing the file permissions for agents Do not complete the procedure for starting the agent.
Install a Microsoft SQL client and configure a remote client connection.	See the Microsoft SQL Server documentation for instructions on how to install a Microsoft SQL client and configure a remote client connection.
On the computer where Tivoli Performance Analyzer is installed, configure an ODBC data source for the data warehouse.	“Configuring an ODBC data source for a Microsoft SQL data warehouse” on page 567
Install the most current SQL Server JDBC driver on the computer where Tivoli Performance Analyzer is installed.	Go to the Microsoft Web page at: http://www.microsoft.com and search for **JDBC driver**. Follow the instructions on the Microsoft download page for installing the driver. After you install the driver, the JAR file name and location are as follows: `<mssqlinstalldir>/sqljdbc_1.1/enu/sqljdbc4.jar`
Configure Tivoli Performance Analyzer to connect to the data warehouse.	For Tivoli Performance Analyzer on Windows, see “Configuring Tivoli Performance Analyzer on Windows (ODBC connection)” on page 567. For a Tivoli Performance Analyzer on Linux or AIX, see “Configuring Tivoli Performance Analyzer on Linux or UNIX (JDBC connection)” on page 568.
Start Tivoli Performance Analyzer.	“Starting Tivoli Performance Analyzer” on page 569

Configuring an ODBC data source for a Microsoft SQL data warehouse

A Microsoft SQL client on Windows requires an ODBC connection to the data warehouse. For the Tivoli Performance Analyzer, you must configure the ODBC connection manually.

Complete the following procedure to set up an ODBC connection for a Tivoli Performance Analyzer on Windows to a local or remote Tivoli Data Warehouse.

Note: This procedure uses default values for the data source name and warehouse user ID. (Default values are used in configuration procedures for warehousing components.) Substitute different values if you do not want to use the default values.

1. Open the Control Panel.
2. Click **Administrative Tools → Data Sources (ODBC)**
3. Click **Add** in the **System DSN** tab in the ODBC Data Source Administrator window.
4. Select **SQL Server** and click **Finish**.
5. Enter `ITM Warehouse` in the **Name** field.
6. Select the Microsoft SQL Server where the Tivoli Data Warehouse is located from the drop down list and click **Next**.
7. Select **With SQL Server authentication using a login ID and password entered by the user**.
8. Enter `ITMUser` for the **Login ID**.

   The user ID must match exactly, including case, what was created in the SQL Server database if using database authentication or the ID that was created in the Windows OS. Mixing the case will cause tables to be created with dbo as the owner rather than ITMUser. This will cause many warehousing components to not work correctly.
9. Type a password for the user in the **Password** field. The default password is `itmpswd1`.
10. Click **Next**.
11. Click **Next** again.
12. Click **Finish**.
13. Click **Test Data Source** to test the connection to the database.
14. Click **OK**.

Configuring Tivoli Performance Analyzer on Windows (ODBC connection)

Use this procedure to configure Tivoli Performance Analyzer on Windows to connect to a Tivoli Data Warehouse on Microsoft SQL Server:

1. Log on to the Windows system where Tivoli Performance Analyzer is installed and begin the configuration:
   a. Click Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services. The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click Performance Analyzer and click **Configure Using Defaults**.
   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Performance Analyzer: Agent Advanced Configuration) contain the settings for the connection between Tivoli Performance Analyzer and the hub monitoring server. These settings were specified when Tivoli Performance Analyzer was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to configure the ODBC data source.
4. Select **ODBC** from the list of selectable agent database connection types.
5. Set your Database Type to **MSSQL**.

6. Specify the Data Source Name - Agent ODBC DSN (ITM Warehouse is the default).

   Note: Tivoli Performance Analyzer does not create this DSN - it must already exist. If you are installing the agent on the same machine where TEP Server is installed, you can use the existing data source created by Tivoli Monitoring. Otherwise, you must create a new System DSN manually, prior to re-configuring Tivoli Performance Analyzer.

   On 64-bit versions of Windows, data sources created by the default ODBC Data Source Administrator applet available from the Control Panel are not available for 32-bit applications. Therefore you must use the 32-bit version of the ODBC Data Source Administrator applet from `<WINDOWS>\SysWOW64\odbcad32.exe`.

   Values for the data source name, database name, and database user ID and password must match the values that you used when configuring an ODBC connection for Tivoli Performance Analyzer. For more information, see "Configuring an ODBC data source for a DB2 data warehouse" on page 515.
7. Type the Username and Password. The entries in these fields are used to connect to the Tivoli Data Warehouse and are the same credentials as those used by the Tivoli Enterprise Portal Server, the Warehouse Proxy Agent and the Summarization and Pruning Agent to communicate with Tivoli Data Warehouse.
8. Click **Next** to proceed to the Advanced Configuration window.
9. You can enable Advanced Configuration to specify TDW Schema and The TDW database schema. If you do not select **Enable advanced configuration** these options are greyed out.
10. You can also choose whether you want the agent to Initialize PA tables and OS domain tasks.

    Note: Setting Initialize PA tables to YES will remove and recreate all previously created tables deleting all user tasks and reverting each OS task to its default.
11. Use the **Bypass connection tests** option to finish the configuration without running connection tests.
12. Click **OK** to finish the configuration process.

    Note: The values for the data source name, and database user ID and password must match the values that you used when configuring an ODBC connection for Tivoli Performance Analyzer. (See "Configuring an ODBC data source for a Microsoft SQL data warehouse" on page 567).

Table 106. Configuration information for the Tivoli Data Warehouse database on Microsoft SQL

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.
Username	ITMUser	The name of the Windows OS user that the Tivoli Performance Analyzer will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password for the Windows OS user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password.

Configuring Tivoli Performance Analyzer on Linux or UNIX (JDBC connection)

Use this procedure to configure Tivoli Performance Analyzer on Linux or UNIX to connect to a Microsoft SQL Server data warehouse:

1. To begin the configuration, log on to the computer where Tivoli Performance Analyzer is installed.
   a. Change to the `install_dir/bin` directory and run the following command:
      ```
      ./itmcmd manage [-h install_dir]
      ```
      where `install_dir` is the installation directory for IBM Tivoli Monitoring. The default installation directory is `/opt/IBM/ITM`. The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Performance Analyzer** and click **Reconfigure**. The Configure Tivoli Performance Analyzer window is displayed.
2. Set the Database Type to MSSQL.
3. Type the username and the password. The entries in these fields are used to connect to the Tivoli Data Warehouse.
4. Review all the defaults in the Agent Configuration window and change as required.
   a. If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of localhost.
   b. Change the port number if necessary (the default port number for MSSQL is 1433).
   c. If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name. (See “Step 1: Create the Tivoli Data Warehouse database” on page 552.)
5. Specify the JDBC Driver. The default driver name is `com.microsoft.sqlserver.jdbc.SQLServerDriver`.
6. Specify the JDBC Driver Path, which should be provided as a list of JAR files with the full path separated by ":".

   Note: The Microsoft SQL Server 2005 driver JAR file name and default location after downloading from the Web are as follows:
   ```
   mssql2005installdir/sqljdbc_1.1/enu/sqljdbc4.jar
   ```

   Fast path: You can use the Browse button to specify the path. In such a case a file list is added at the end of the JDBC Driver Path text field, separated from the existing content by a path separator.
7. You can use the **Test connection** button to check whether the connection can be initiated.
8. Click **Next** to proceed to the Advanced Configuration window.
   a. You can enable Advanced Configuration to specify TDW Schema and Configuration schema. If you do not select Enable advanced configuration, all of these options are greyed out.
   b. You can also choose whether you want the agent to initialize PA tables.

      Note: Setting Initialize PA tables to YES will remove and recreate all previously created tables deleting all user tasks and reverting each OS task to its default.
   c. Use the **Bypass connection tests** option to finish the configuration without running connection tests.
9. Click **Save** to save your settings and close the window.

Starting Tivoli Performance Analyzer

To start Tivoli Performance Analyzer from the Manage Tivoli Enterprise Monitoring Services window, right-click Tivoli Performance Analyzer and select Start. To start the Tivoli Performance Analyzer agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is `/opt/IBM/ITM`.

```
./itmcmd agent start pa
```

where pa is the product code for Tivoli Performance Analyzer agent.

Chapter 23. Tivoli Data Warehouse solution using Oracle

Use the information and instructions in this chapter to implement a Tivoli Data Warehouse solution using Oracle for the warehouse database. The following table lists the goals for achieving an Oracle solution.

Table 107. Goals for achieving a Tivoli Data Warehouse solution using Oracle

Goal	Where to find information
Review your options, specific to an Oracle solution, for operating system platforms and communications between warehousing components.	"Supported components" on page 572
Install prerequisite software before implementing your Tivoli Data Warehouse solution.	"Prerequisite installation" on page 573
Understand how to use the instructions for implementing your Tivoli Data Warehouse solution.	"Implementing a Tivoli Data Warehouse solution using Oracle" on page 574
Complete the steps for implementing your Tivoli Data Warehouse solution using Oracle for the data warehouse.	"Step 1: Create the Tivoli Data Warehouse database" on page 575 "Step 2: Install and configure communications for the Warehouse Proxy Agent" on page 577 "Step 3: Configure communications between the Tivoli Enterprise Portal Server and the data warehouse" on page 585 "Step 4: Install and configure communications for the Summarization and Pruning Agent" on page 589 "Step 5: Install and configure communications for Tivoli Performance Analyzer" on page 590

Supported components

Figure 144 presents the options for a Tivoli Data Warehouse solution using Oracle for the warehouse database. The diagram summarizes the supported operating system platforms for the various warehousing components, the supported database products, and the connections between components. For more specific information about supported operating systems and database products, including product names and versions, see "Hardware and software requirements" on page 138.

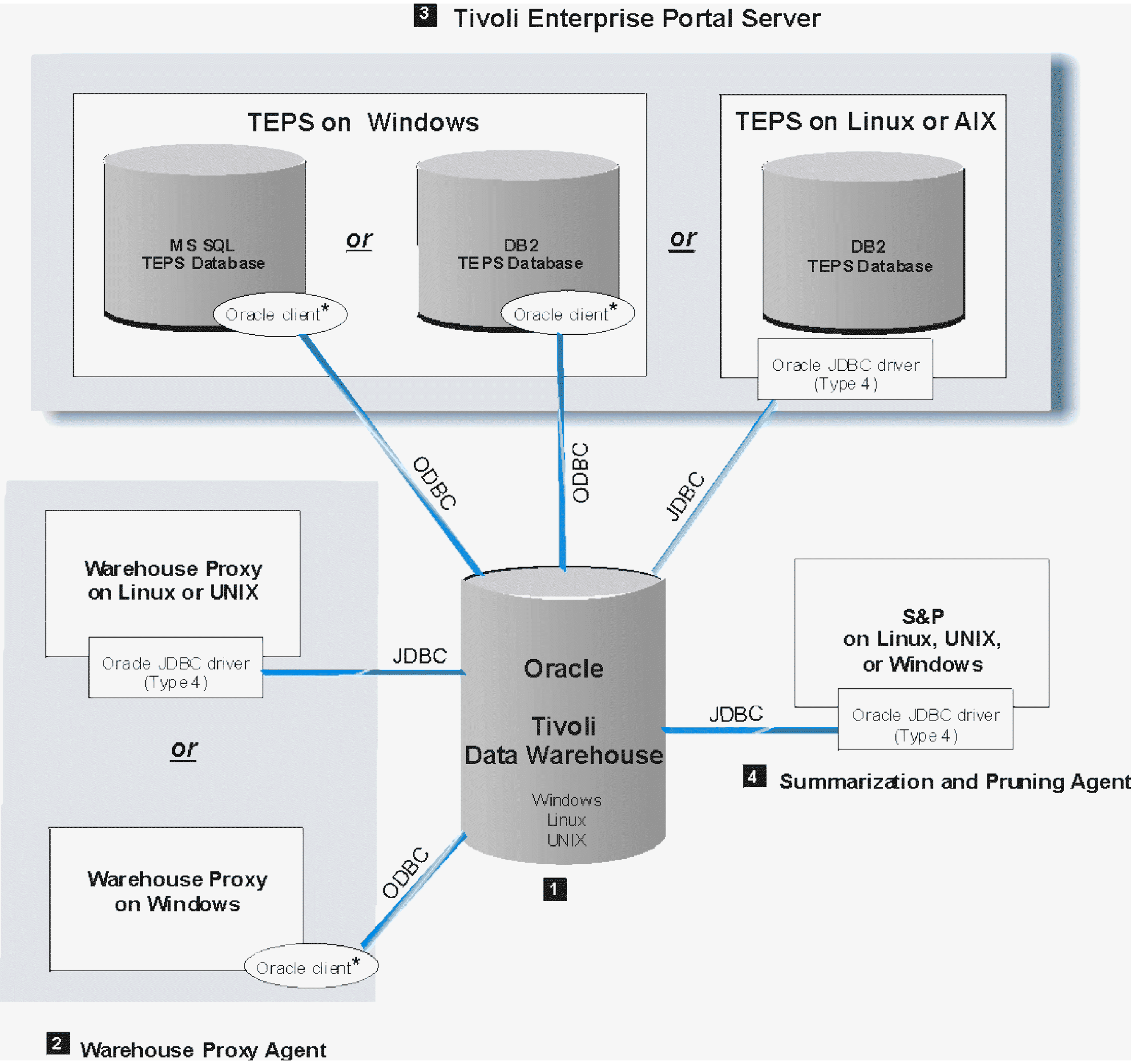

3 Tivoli Enterprise Portal Server
TEPS on Windows
MS SQL TEPS Database
Oracle client*
or
DB2 TEPS Database
Oracle client*
or
TEPS on Linux or AIX
DB2 TEPS Database
Oracle JDBC driver (Type 4)
ODBC
ODBC
JDBC
Warehouse Proxy on Linux or UNIX
Oracle JDBC driver (Type 4)
JDBC
or
Warehouse Proxy on Windows
Oracle client*
ODBC
Oracle
Tivoli Data Warehouse
Windows
Linux
UNIX
1
S&P on Linux, UNIX, or Windows
Oracle JDBC driver (Type 4)
JDBC
4 Summarization and Pruning Agent
2 Warehouse Proxy Agent

Figure 144. Tivoli Data Warehouse solution using Oracle

Note: An asterisk (*) next to a database client indicates that you must manually install the client if it does not already exist.

In the following discussion, numbered product components correspond to the numbers on the diagram.

1 Tivoli Data Warehouse on Oracle

A Tivoli Data Warehouse database on Oracle can be installed on supported Windows, Linux, or any UNIX platform support by Oracle. Ensure that the Oracle listener is active in order to accept connections from an Oracle client or JDBC driver.

2 Warehouse Proxy Agent

A Warehouse Proxy Agent on Linux or AIX communicates with the warehouse database through a JDBC connection. Install an Oracle Type 4 JDBC driver on the computer where the Warehouse Proxy Agent is located. If the Tivoli Data Warehouse is located on a remote computer, create a TNS (Transparent Network Substrate) Service Name on the local computer where the Warehouse Proxy Agent is located.

A Warehouse Proxy Agent on Windows communicates with the warehouse database through an ODBC connection. The ODBC driver is included with the Oracle client. If the Tivoli Data Warehouse is located on a remote computer, install an Oracle client on the local computer where the Warehouse Proxy Agent is located. Also, create a TNS Service Name on the local computer.

3 Tivoli Enterprise Portal Server

A Tivoli Enterprise Portal Server on Windows communicates with an Oracle data warehouse through an Oracle database client using an ODBC connection. You must manually install the Oracle client on the portal server. The ODBC driver is included with the Oracle client.

A portal server on Linux or AIX communicates with the warehouse database through a JDBC connection. Install an Oracle Type 4 JDBC driver on the portal server.

4 Summarization and Pruning Agent

The Summarization and Pruning Agent communicates with the warehouse database through a JDBC connection from any supported operating system. Install an Oracle Type 4 JDBC driver on the computer where the Summarization and Pruning Agent is located.

Prerequisite installation

Before you implement your Tivoli Data Warehouse solution, complete one or more hub installations, *excluding the warehousing components*. Include the following components in each hub installation:

- The hub Tivoli Enterprise Monitoring Server
- *(Optional)* One or more remote monitoring servers
- The Tivoli Enterprise Portal Server, including the prerequisite RDBMS for the portal server database (DB2 for Linux, UNIX, and Windows or Microsoft SQL Server)
- An Oracle database server on the computer where you will create the Tivoli Data Warehouse database. (The Tivoli Data Warehouse database can be shared in a multi-hub installation or dedicated to a single hub.)
- *(Optional)* A portal desktop client
- *(Optional)* Monitoring agents, and the application support for the monitoring agents

  Note: The term *monitoring agent*, as used here, refers to agents that collect data directly from managed systems, not the Warehouse Proxy Agent or Summarization and Pruning Agent.
- *(Optional)* The Tivoli Performance Analyzer
- *(Optional)* Language packs for all languages other than English

See the following table for related information:

Table 108. Information topics related to installation of prerequisite software for a Tivoli Data Warehouse solution

Topic	Where to find information
Single and multiple hub installations	To understand the terminology related to single and multiple hub installations, see “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46.
Installation procedures for prerequisite components	The detailed instructions for installing the prerequisite components are described in Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207. See your database documentations for instructions on how to install a supported database server.
Supported RDBMS versions	For specific information about the supported database platforms for the portal server database and the Tivoli Data Warehouse, see “Hardware and software requirements” on page 138.

Implementing a Tivoli Data Warehouse solution using Oracle

Use the instructions in the remainder of this chapter to implement a Tivoli Data Warehouse solution using Oracle for the data warehouse.

Assumptions

The implementation instructions are based on the following assumptions:

- You will create the Tivoli Data Warehouse database on a different computer from the Tivoli Enterprise Portal Server.
- You will create a single Tivoli Data Warehouse database, to be used either within a single hub installation or to be shared in a multi-hub installation. If you have multiple independent hub installations, repeat the implementation steps for each hub installation. (See “Locating and sizing the hub Tivoli Enterprise Monitoring Server” on page 46 for information about hub installations.)
- No assumption is made about where you will install the Warehouse Proxy Agent and Summarization and Pruning Agent. Either of these agents may be installed on the same computer as the Tivoli Data Warehouse or on a different computer.

Solution steps

To implement your Tivoli Data Warehouse solution using Oracle, complete the four major steps described in the remaining sections of this chapter, in the order listed:

1. Create the Tivoli Data Warehouse database.
2. Install and configure communications for the Warehouse Proxy Agent.
3. Configure communications between the Tivoli Enterprise Portal Server and the data warehouse.
4. Install and configure communications for the Summarization and Pruning Agent.

Each major step consists of a series of installation and configuration tasks, listed and described in a table. Use the step tables as a road map for implementing your solution. The step tables describe the tasks at a high level, account for variations among configuration options (such as which operating system is used for a component), and reference the appropriate sections for detailed implementation procedures. To implement your solution successfully:

- Perform the tasks in the order listed in the table.
- Do not skip a table to the procedures that follow it.

  Be aware that some of the implementation procedures referenced in a table are included in this chapter and some are documented elsewhere. In some cases, the task is described in the table, without referencing a separate procedure. Read and follow all instructions in the tables.

Step 1: Create the Tivoli Data Warehouse database

Complete the tasks described in the following table to create a Tivoli Data Warehouse database using Oracle and to make it accessible to clients.

Table 109. Tasks for creating the Tivoli Data Warehouse database

Task	Procedure
Create the Tivoli Data Warehouse database on one of the supported Windows, Linux, or UNIX operating systems. To comply with the assumptions described in the introduction to this chapter, create the database on a different computer from the Tivoli Enterprise Portal Server.	For guidance on planning the size and disk requirements for the warehouse database, see "Planning considerations for the Tivoli Data Warehouse" on page 469. For information about creating the warehouse database using Oracle, see "Creating the warehouse database on Oracle."
Activate the Oracle listener on the Oracle server where the Tivoli Data Warehouse is installed. To activate the Oracle listener: • Use the Oracle Listener Service on Windows. • Use the **lsnrctl start** command on Linux and UNIX.	See the Oracle documentation for instructions on how to activate the Oracle listener.

Creating the warehouse database on Oracle

This section provides guidelines for creating the Tivoli Data Warehouse database on Oracle. For specific instructions on how to create an Oracle database, see the Oracle documentation or have a database administrator create the database for you.

When you create the warehouse database using Oracle, follow these guidelines:

- Use the Unicode character set (AL32UTF8) when creating the database.
- Create a database user login name and password that the warehousing components (portal server, Warehouse Proxy Agent, and Summarization and Pruning Agent) can use to access the data warehouse. In these instructions, this user account is referred to as the *warehouse user*.
- Use the default values shown in the following table for the warehouse name and warehouse user. The default values are used in the configuration procedures for connecting the warehousing components to the warehouse database.

Table 110. Default values for Tivoli Data Warehouse parameters

Parameter	Default value
Tivoli Data Warehouse database name	WAREHOUS
User name	ITMUser
User password	itmpswd1

- Create an ITM_DW role, and give this role the following permissions:

  ```
  CREATE ROLE role not IDENTIFIED;
  GRANT CREATE SESSION TO role;
  GRANT ALTER SESSION TO role;
  GRANT CREATE PROCEDURE TO role;
  GRANT CREATE TABLE TO role;
  GRANT CREATE VIEW TO role;
  ```

  After you create the warehouse user ID that will be used by the Warehouse Proxy and the Summarization and Pruning Agents to connect to the Tivoli Data Warehouse database, give this user ID the *role* you just created:

```
CREATE USER itmuser PROFILE DEFAULT IDENTIFIED by itmuser_password
   ACCOUNT UNLOCK;
GRANT role TO itmuser;
```

As all the Tivoli Data Warehouse tables are created in the user's default tablespace, you need to allocate enough space quota on the default tablespace to this user to create all the tables, or you can simplify it by allowing unlimited tablespace to this user.

Note: The ITM_DW role needs the connect privilege only if the database objects are created using the schema publication tool. If the historical configuration is changed and the warehouse user has limited privileges, the schema tool must be used to create any additional database objects (using the schema tool's updated mode; see Chapter 19, “Schema Publication Tool,” on page 483).

- Activate the Oracle listener using the Oracle Listener Service on Windows or the **lsnrctl start** command on Linux and UNIX.

Step 2: Install and configure communications for the Warehouse Proxy Agent

You can install one or more Warehouse Proxy Agents to collect and send historical data to the Tivoli Data Warehouse database. Complete the tasks described in the following table, in the order listed, to install and configure each Warehouse Proxy Agent.

Table 111. Tasks for installing and configuring communications for the Warehouse Proxy Agent

Task	Procedure
Install one or more Warehouse Proxy Agents. If you want to install a Summarization and Pruning Agent on the same computer as one of the Warehouse Proxy Agents, use the referenced procedures to install both agents at the same time. If you are installing more than one Warehouse Proxy Agent, each agent must be installed on a separate computer. The installation procedure for Windows includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (*Configuring the monitoring agent*). See the information at right. Be sure to perform all referenced installation and configuration procedures. **Note for sites setting up autonomous operation::** The installation procedure includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring Server. On Windows operating systems, if you want to run the Warehouse Proxy Agent without a connection to the hub, accept the defaults for the connection information, but specify a nonvalid name for the monitoring server. On UNIX and Linux operating systems, check **No TEMS** on the **TEMS Connection** tab of the configuration window.	To install a Warehouse Proxy Agent on Windows, complete the procedure “Windows: Installing a monitoring agent” on page 253. To install a Warehouse Proxy Agent on Linux or AIX, complete the procedure “Linux or UNIX: Installing a monitoring agent” on page 259, including the following subsections: • *Installing the monitoring agent* • *Configuring the monitoring agent* • *Changing the file permissions for agents* (if you used a non-root user to install the Warehouse Proxy) *Do not complete the procedure for starting the agent.*
(*Warehouse Proxy Agent on Windows only*) • Install an Oracle client on the computer where the Warehouse Proxy Agent is installed if *both* of the following statements are true: – The Warehouse Proxy is installed on Windows, and – The Warehouse Proxy needs to connect to a remote data warehouse. • Ensure that the the latest Oracle patches are installed. • Set the following system variable on the computer where the Warehouse Proxy Agent is installed. Restart the computer after setting the variable. The format of the NLS_LANG environment variable is: `NLS_LANG=language_territory.charset` Set the language and territory to appropriate variables. For the United States this is `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`. Perform the last two tasks whether or not the warehouse database is local or remote.	See the Oracle documentation for instructions on how to install an Oracle client. Obtain Oracle ODBC drivers from the following Web site: http://www.oracle.com/technology/software/tech/windows/odbc/htdocs/utilsoft.html
(*Warehouse Proxy Agent on Windows only*) If the Tivoli Data Warehouse is located on a remote computer, create a TNS (Transparent Network Substrate) Service Name on the local computer where the Warehouse Proxy Agent is located.	“Creating a TNS Service Name” on page 578

Table 111. Tasks for installing and configuring communications for the Warehouse Proxy Agent (continued)

Task	Procedure
(*Warehouse Proxy Agent on Windows only*) On the computer where the Warehouse Proxy Agent is installed, configure an ODBC data source for the data warehouse, using the TNS Service Name that you created in the preceding step. Perform this procedure whether or not the Warehouse Proxy Agent and the warehouse database are installed on the same computer.	"Configuring an ODBC data source for an Oracle data warehouse" on page 579
(*Warehouse Proxy Agent on Linux or AIX only*) Install an Oracle Type 4 JDBC driver on the computer where the Warehouse Proxy Agent is installed.	Obtain the Oracle JDBC Driver from the following Web site: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html The Oracle JDBC driver JAR file name and location after installation is as follows: `*oracleinstalldir*/jdbc/lib/ojdbc14.jar` The `ojdbc14.jar` file supports JRE 1.5 or higher, the required Java Runtime Environment for IBM Tivoli Monitoring.
Configure the Warehouse Proxy Agent to connect to the data warehouse. Perform this procedure whether or not the Warehouse Proxy Agent and the warehouse database are installed on the same computer.	For a Warehouse Proxy Agent on Windows, see "Configuring a Warehouse Proxy Agent on Windows (ODBC connection)" on page 580. For a Warehouse Proxy Agent on Linux or AIX, see "Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)" on page 581.
If you are installing more than one Warehouse Proxy Agent within the same hub monitoring server installation, associate each Warehouse Proxy Agent with a subset of monitoring servers (hub or remote) within the installation. Each Warehouse Proxy Agent receives data from the monitoring agents that report to the monitoring servers on the list. Use the environment variable KHD_WAREHOUSE_TEMS_LIST to specify a list of monitoring servers to associate with a Warehouse Proxy Agent.	For instructions about installing and configuring multiple Warehouse Proxy Agents within a single hub monitoring server installation, see "Installing and configuring multiple Warehouse Proxy Agents" on page 608.
(*Optional*) Customize the configuration of the Warehouse Proxy Agent for tuning performance.	"Tuning the performance of the Warehouse Proxy" on page 617
Start the Warehouse Proxy Agent.	"Starting the Warehouse Proxy" on page 584

Creating a TNS Service Name

Create a TNS (Transparent Network Substrate) Service Name (also called a Net Service Name) on a computer where an Oracle client is installed if the Tivoli Data Warehouse exists on a *remote* Oracle server. The TNS Service name is needed to create an ODBC connection between the client and the server. Use this procedure to create a TNS Service name:

- On a Windows computer where a Warehouse Proxy Agent installed.
- On a Windows computer where the Tivoli Enterprise Portal Server is installed.

Do *not* perform this procedure on the computer where the data warehouse (Oracle server) is installed or on a computer where there is no Oracle client (for example, on a computer where a Type 4 Oracle JDBC driver is used to communicate with the remote data warehouse).

Note: This procedure uses the default value for the warehouse name (`WAREHOUS`). Substitute a different value if you do not want to use the default name.

Complete the following steps to create the TNS Service Name. Click **Next** after each step.

1. Enter `dbca` at the Oracle command-line to start the Oracle Net Configuration Assistant tool.
2. On the Welcome window, select **Local Net Service Name configuration**.
3. Select **Add**.
4. Enter `WAREHOUS` in the **Service Name** field. (This is the remote name for the Tivoli Data Warehouse.)
5. Select **TCP** as the network protocol to communicate with the Tivoli Data Warehouse database.
6. Specify the fully qualified host name and port number of the computer where the warehouse database is installed.
7. Perform the connection test to verify the connection to the warehouse database.
8. Optionally change the default name in the **Net Service Name** field.

   This is the TNS Service Name. The default name matches the name that you entered in Step 4. You can change this to a different name. The TNS Service Name can be considered a local alias for the remote Tivoli Data Warehouse name.
9. When prompted to configure another net service name, click **No** to return to the Welcome window.
10. Click **Finish**.

Configuring an ODBC data source for an Oracle data warehouse

An Oracle client on Windows requires an ODBC connection to the data warehouse. For the Warehouse Proxy Agent, you must configure the ODBC connection manually.

Complete the following procedure to set up an ODBC connection for a Warehouse Proxy Agent on Windows to a local or remote Tivoli Data Warehouse.

Note: This procedure uses default values for the data source name (`ITM Warehouse`) and warehouse user ID (`ITMUser`). (Default values are used in configuration procedures for warehousing components.) Substitute different values if you do not want to use the default values.

1. On the computer where the Warehouse Proxy Agent is installed, open the Control Panel.
2. Click **Administrative Tools → Data Sources (ODBC)**
3. Click **Add** in the **System DSN** tab in the ODBC Data Source Administrator window.
4. Select the Oracle ODBC driver:
   - For Oracle 9, the ODBC driver name is **Oracle in Ora9ias_home**.
   - For Oracle 10, the ODBC driver name is **Oracle in OraDb10g_home1**.
5. Enter `ITM Warehouse` in the **Data Source Name** field.
6. Enter the TNS Service Name in the **TNS Service Name** field. This is the name that you specified in Step 8 (for example, `WAREHOUS`).
7. Enter `ITMUser` in the **User ID** field.
8. Click **Test Connection**.
9. In the Oracle ODBC Driver Connect window, enter the TNS Service Name, and the user ID and password of the warehouse user.
10. Click **OK** on the **Connection successful** message.

Configuring a Warehouse Proxy Agent on Windows (ODBC connection)

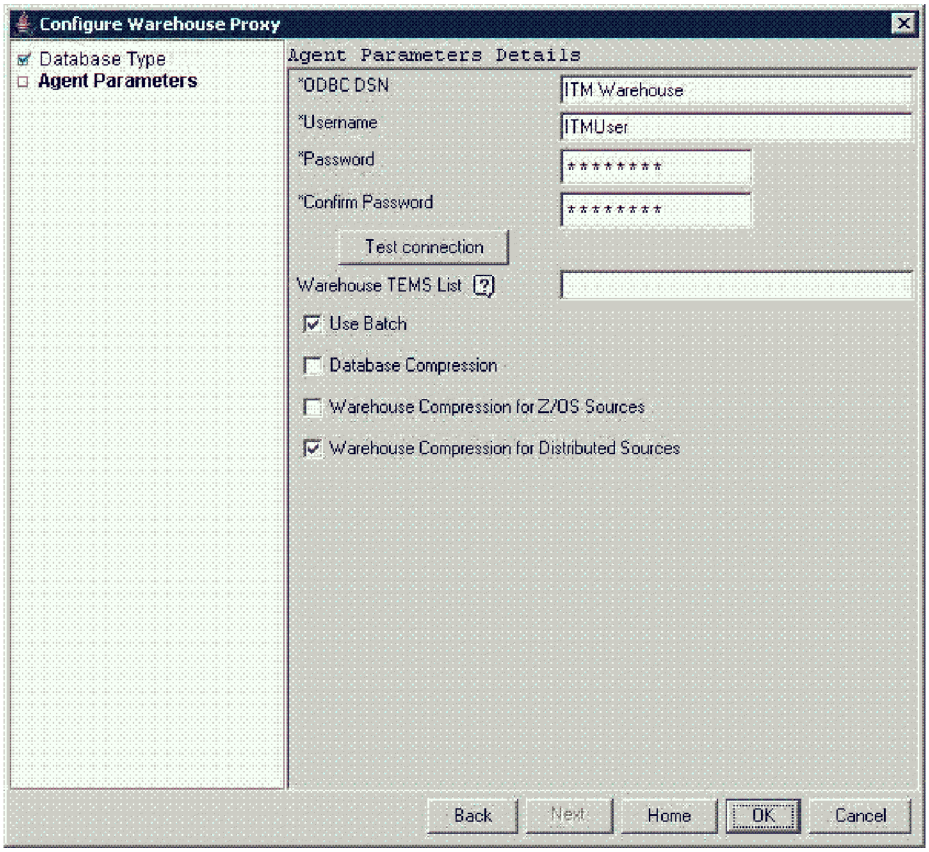
Use this procedure to configure a Warehouse Proxy Agent on Windows to connect to an Oracle Tivoli Data Warehouse:

1. Log on to the Windows system where the Warehouse Proxy Agent is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Warehouse Proxy** and click **Configure Using Defaults**.

      Click **Reconfigure** if the Warehouse Proxy is installed on the same computer as the portal server.
   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Warehouse Proxy: Agent Advanced Configuration) contain the settings for the connection between the Warehouse Proxy Agent and the hub monitoring server. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to configure the ODBC data source.
4. Select **Oracle** from the list of databases and click **OK**.

   The following configuration window is displayed.



*ODBC DSN: ITM Warehouse

*Username: ITMUser

*Password: ********

*Confirm Password: ********

Warehouse TEMS List [?]:

☑ Use Batch

□ Database Compression

□ Warehouse Compression for Z/OS Sources

☑ Warehouse Compression for Distributed Sources

Figure 145. Configure Oracle Data Source for Warehouse Proxy window

5. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in Table 112 on page 581.

Note: The values for the data source name, database name, and database user ID and password must match the values that you used when configuring an ODBC connection for the Warehouse Proxy Agent. (See "Configuring an ODBC data source for an Oracle data warehouse" on page 579.)

Table 112. Configuration information for the Tivoli Data Warehouse database on Oracle

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.
Username	ITMUser	The name of the user that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password that the Warehouse Proxy Agent will use to access the Tivoli Data Warehouse database. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Confirm Password	itmpswd1	Confirm the password by entering it again.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password. **Note:** Test Connection is not available if configuring remotely from the Tivoli Enterprise Portal.
Warehouse TEMS List		Environment variable containing a space delimited list of TEMS names, which are given during the configuration of HTEMS or RTEMS. A TEMS name in this field indicates that all the agents connected to this TEMS will have their historical data sent to this Warehouse Proxy Agent. This variable is used when the ITM environment contains multiple Warehouse Proxy Agents and the workload has to be balanced using specific Warehouse Proxy Agents.
Use Batch		Batch inserts introduced ITM V6.2.2 fix pack 2, can greatly increase the data insertion rate of the Warehouse Proxy Agent. This is especially true if the proxy and the warehouse are located on different hosts. Batch inserts are supported for ODBC warehouse connections. Using batch inserts is recommended in all configurations, but they will place increased load on the data warehouse.
Database Compression		If the database compression mode is supported by the database, the Warehouse Proxy Agent will create all tables and indexes in the Tivoli Data Warehouse with compression enabled. This option reduces the storage costs of the Tivoli Data Warehouse.
Warehouse Compression for Z/OS Sources		Select this option for the Warehouse Proxy server to allow clients installed on Z/OS machines to send compressed data.
Warehouse Compression for Distributed Sources		Select this option for the Warehouse Proxy server to allow clients installed on distributed machines (Linux/UNIX, Windows) to send compressed data.

6. Click **OK**.

Configuring a Warehouse Proxy Agent on Linux or UNIX (JDBC connection)

Use this procedure to configure a Warehouse Proxy Agent on Linux or UNIX to connect to an Oracle data warehouse:

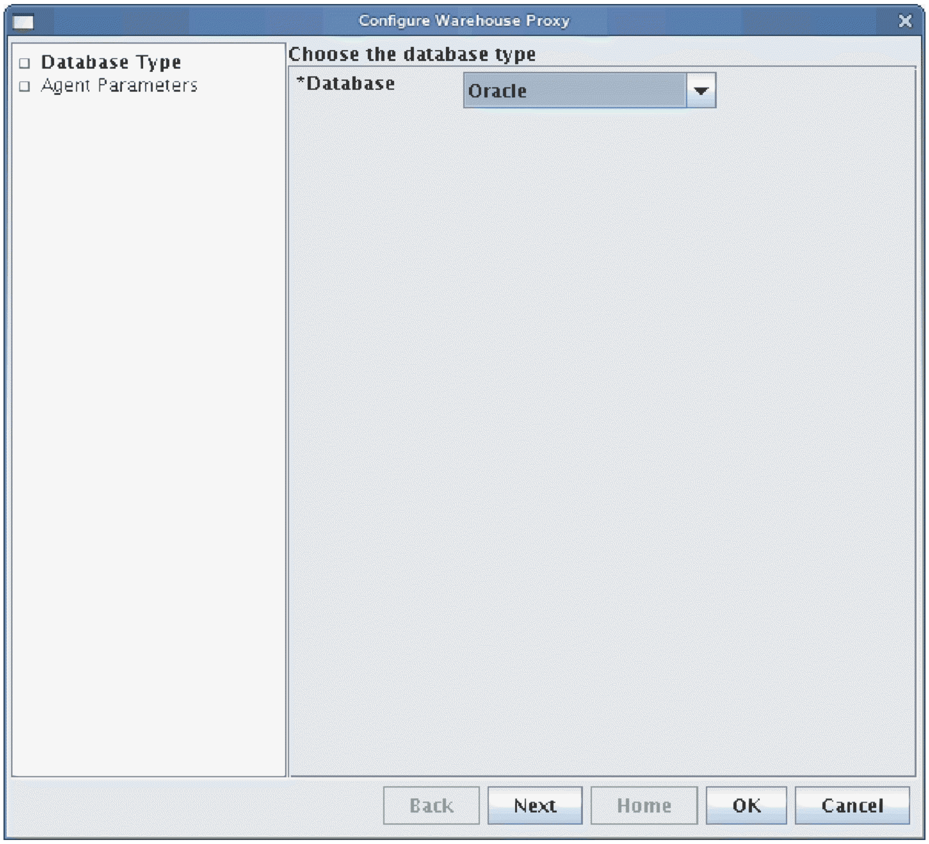
1. Log on to the computer where the Warehouse Proxy Agent is installed and begin the configuration.
   a. Change to the `install_dir/bin` directory and run the following command:

      ```
      ./itmcmd manage [-h install_dir]
      ```

      where *install_dir* is the installation directory for IBM Tivoli Monitoring. The default installation directory is /opt/IBM/ITM.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Warehouse Proxy** and click **Configure**.

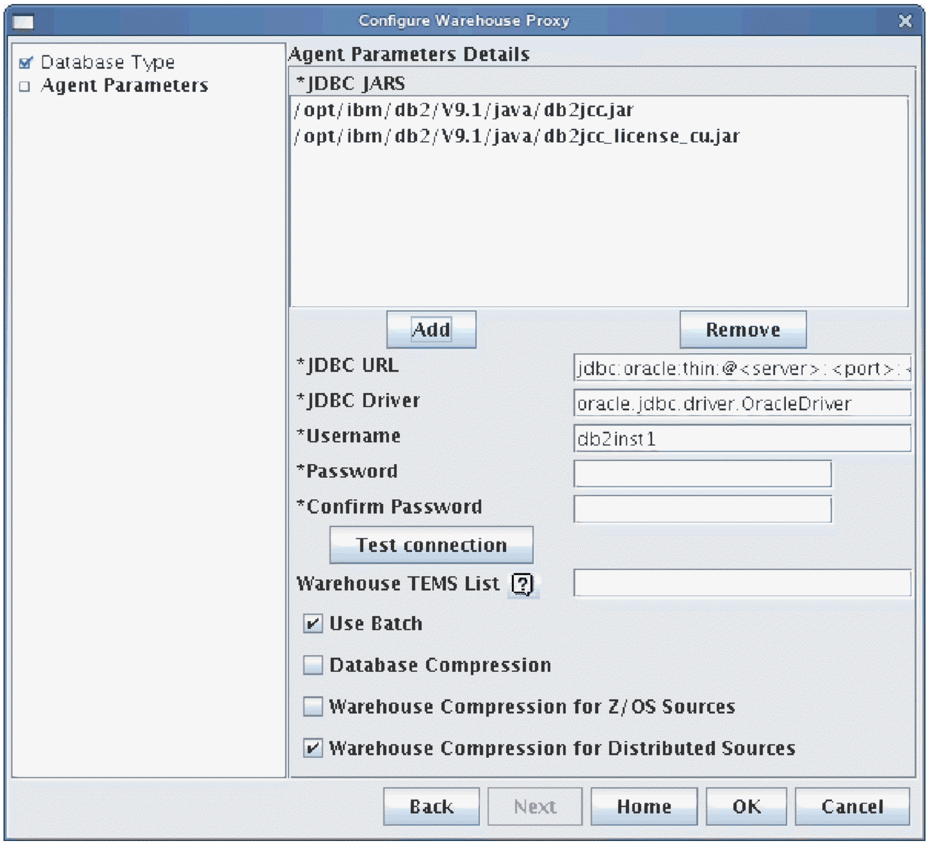
      The Configure Warehouse Proxy window is displayed.



*Database: Oracle

Figure 146. Configure Warehouse Proxy window (Database Type pane)

2. In the **Database** drop-down list, select **Oracle**.
3. Select the **Agent Parameters** check box.



☑ Database Type
□ Agent Parameters

*JDBC JARS

```
/opt/ibm/db2/V9.1/java/db2jcc.jar
/opt/ibm/db2/V9.1/java/db2jcc_license_cu.jar
```

*JDBC URL: jdbc:oracle:thin:@<server>:<port>:

*JDBC Driver: oracle.jdbc.driver.OracleDriver

*Username: db2inst1

*Password:

*Confirm Password:

Warehouse TEMS List ?:

☑ Use Batch

□ Database Compression

□ Warehouse Compression for Z/OS Sources

☑ Warehouse Compression for Distributed Sources

Figure 147. Configure Warehouse Proxy window (Agent Parameters pane)

4. Add the name and directory location of the JDBC driver JAR file to the **JDBC JARS** list box:
   a. Click **Add** to display the file browser window. Navigate to the location of the JDBC driver JAR file on this computer and select the file. The Oracle JDBC driver JAR file name and default location after downloading from the Web are as follows:

      oracleinstalldir/jdbc/lib/ojdbc14.jar

   b. Click **OK** to close the browser window and add the driver file to the list.

   If you need to delete an entry from the list, select the entry and click **Delete**.
5. Change the default value displayed in the **JDBC URL** field if it is not correct. The default Tivoli Data Warehouse URL for Oracle is as follows:

   jdbc:oracle:thin:@localhost:1521:WAREHOUS

   - If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of `localhost`.
   - Change the port number if it is different.
   - If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name. (See “Creating the warehouse database on Oracle” on page 575.)
6. Verify the JDBC driver name, which is displayed in the **JDBC Driver** field. (Note that the **JDBC Driver** field displays the *driver name*, in contrast to the *driver JAR file* that is listed in the **JDBC JARS** field.)

   The Oracle JDBC Type 4 driver name is as follows:

   oracle.jdbc.driver.OracleDriver

7. If necessary, change the entries in the **Username** and **Password** fields to match the user name and password that were created for the Tivoli Data Warehouse. (See "Creating the warehouse database on Oracle" on page 575.) The default user name is `itmuser` and the default password is `itmpswd1`.
8. Check the **Use Batch** check box if you want the Warehouse Proxy Agent to submit multiple execute statements to the Tivoli Data Warehouse database for processing as a batch.

   In some situations, such as crossing a network, sending multiple statements as a unit is more efficient than sending each statement separately. Batch processing is one of the features provided with the JDBC 2.0 API.
9. Click **Test connection** to ensure you can communicate with the Tivoli Data Warehouse database.
10. Click **Save** to save your settings and close the window.

Starting the Warehouse Proxy

- To start the Warehouse Proxy Agent from the Manage Tivoli Enterprise Monitoring Services window, right-click **Warehouse Proxy** and select **Start**.
- (*Linux or AIX only*) To start the Warehouse Proxy Agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is /opt/IBM/ITM.

  ```
  ./itmcmd agent start hd
  ```

  where `hd` is the product code for the Warehouse Proxy Agent.

Step 3: Configure communications between the Tivoli Enterprise Portal Server and the data warehouse

Complete the tasks described in the following table, in the order listed, to configure communications between the portal server and the data warehouse.

Table 113. Tasks for configuring communications between the portal server and an Oracle data warehouse

Task	Procedure
(*Portal server on Windows only*) Install an Oracle database client on the portal server.	See the Oracle documentation for instructions on how to install an Oracle client.
(*Portal server on Windows only*) Create a TNS (Transparent Network Substrate) Service Name on the portal server.	"Creating a TNS Service Name" on page 578
(*Portal server on Linux or AIX only*) Install an Oracle JDBC Type 4 driver on the portal server.	Obtain the Oracle JDBC Type 4 Driver from the following Web site: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html The Oracle JDBC driver JAR file name and location after installation is as follows: `*oracleinstalldir*/jdbc/lib/ojdbc14.jar` The `ojdbc14.jar` file supports JRE 1.5 or higher, the required Java Runtime Environment for IBM Tivoli Monitoring.
Configure the portal server to connect to the data warehouse. The configuration procedure on Windows automatically configures an ODBC connection to the data warehouse.	For a portal server on Windows, see "Configuring a Windows portal server (ODBC connection)." For a portal server on Linux or AIX, see "Configuring a Linux or AIX portal server (JDBC connection)" on page 586.
Restart the portal server.	"Starting the portal server" on page 588
Test the connection between the portal server and the Tivoli Data Warehouse by creating a customized query in the Tivoli Enterprise Portal.	"Testing the connection between the portal server and the Tivoli Data Warehouse" on page 614

Configuring a Windows portal server (ODBC connection)

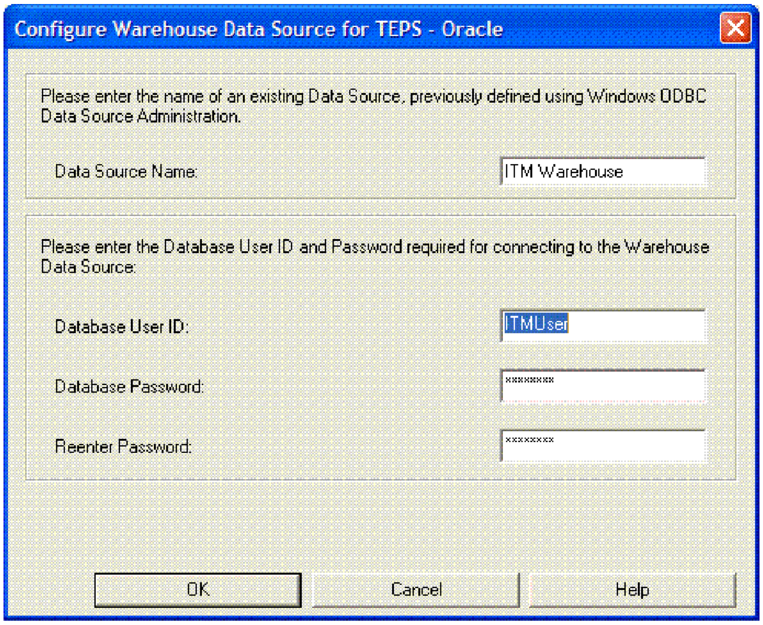
Use this procedure to configure a portal server on Windows to connect to an Oracle Tivoli Data Warehouse:

1. Log on to the Windows system where the portal server is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.
   b. Right-click **Tivoli Enterprise Portal Server** and click **Reconfigure**.
2. The next two windows (entitled TEP Server Configuration) contain the settings for the connection between the portal server and the hub monitoring server. These settings were specified when the portal server was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to reconfigure the warehouse information for the Tivoli Enterprise Portal Server.

4. Select **Oracle** from the list of databases and click **OK**.

   The following configuration window is displayed.



Configure Warehouse Data Source for TEPS - Oracle

Please enter the name of an existing Data Source, previously defined using Windows ODBC Data Source Administration.

Data Source Name: ITM Warehouse

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

Database User ID: ITMUser

Database Password: ********

Reenter Password: ********

Figure 148. Configure Oracle Data Source for Warehouse window

5. Click **OK** to accept all default information on this window, or change one or more default values and then click **OK**. The fields on this window are described in the following table:

Table 114. Configuration information for the Tivoli Data Warehouse database on Oracle

Field	Default value	Description
Data Source Name	ITM Warehouse	The name of the data source.
Database User ID	ITMUser	The login name of the database user that the portal server will use to access the Tivoli Data Warehouse database.
Database Password	itmpswd1	The password for the database login user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Reenter Password	itmpswd1	Confirm the password by entering it again.

6. Click **OK**.

Configuring a Linux or AIX portal server (JDBC connection)

Use this procedure to configure a portal server on Linux or AIX to connect to an Oracle Tivoli Data Warehouse on any operating system:

1. Log on to the computer where the Tivoli Enterprise Portal Server is installed and begin the configuration.
   a. Change to the *install_dir*/bin directory and run the following command:

      ```
      ./itmcmd manage [-h install_dir]
      ```

      where *install_dir* is the installation directory for IBM Tivoli Monitoring. The default installation directory is /opt/IBM/ITM.

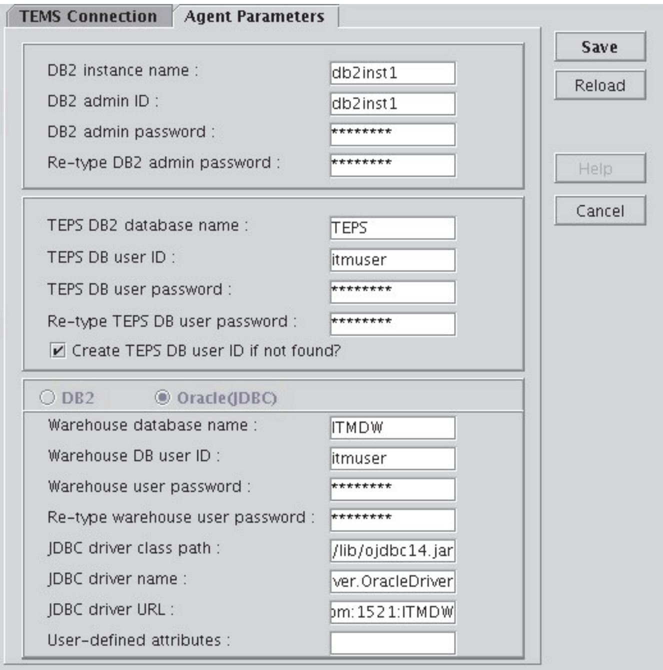
      The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Tivoli Enterprise Portal Server** and click **Configure**.

      The Configure Tivoli Enterprise Portal Server window is displayed.

2. On the **TEMS Connection** tab, review the settings for the connection between the portal server and the hub monitoring server. These settings were specified when the portal server was installed.
3. Click the **Agent Parameters** tab.
4. Select the **Oracle** radio button.

   The fields for configuring the connection to an Oracle data warehouse are displayed at the bottom of the window.



DB2 instance name : db2inst1
DB2 admin ID : db2inst1
DB2 admin password : ********
Re-type DB2 admin password : ********

TEPS DB2 database name : TEPS
TEPS DB user ID : itmuser
TEPS DB user password : ********
Re-type TEPS DB user password : ********
☑ Create TEPS DB user ID if not found?

○ DB2 ● Oracle(JDBC)
Warehouse database name : ITMDW
Warehouse DB user ID : itmuser
Warehouse user password : ********
Re-type warehouse user password : ********
JDBC driver class path : /lib/ojdbc14.jar
JDBC driver name : ver.OracleDriver
JDBC driver URL : om:1521:ITMDW
User-defined attributes :

Figure 149. Configuring the connection to an Oracle data warehouse

5. Fill in the fields in Figure 149 with the configuration values described in Table 115.

Table 115. Configuration information for a Tivoli Data Warehouse database on Oracle

Field	Default value	Description
Warehouse database name	WAREHOUS	The name of the Tivoli Data Warehouse database.
Warehouse DB user ID	ITMUser	The database login user that the portal server uses to access the Tivoli Data Warehouse database. This user is referred to as the *warehouse user*.
Warehouse user password	itmpswd1	The password for the warehouse user.
Re-type Warehouse user password	itmpswd1	The password for the warehouse user.
JDBC driver class path	`*oracleinstalldir*/jdbc/lib/ojdbc14.jar`	The full path name of the Oracle JDBC Type 4 driver JAR file on this computer.
JDBC driver name	oracle.jdbc.driver.OracleDriver	The Oracle JDBC Type 4 driver name.

Table 115. Configuration information for a Tivoli Data Warehouse database on Oracle (continued)

Field	Default value	Description
JDBC driver URL	jdbc:oracle:thin:@localhost:1521:WAREHOUS	The Oracle-defined URL that identifies the Oracle instance used for the remote Tivoli Data Warehouse. Replace `localhost` with the host name of the remote computer where the Tivoli Data Warehouse is installed. Change the default port number (`1521`) and Tivoli Data Warehouse name (`WAREHOUS`) if they are different.
User-defined attributes	(no default)	Enter any user-defined attributes that are used to customize the behavior of the driver connection. Use semi-colons (;) to delimit the attributes.

6. Click **Save** to save your settings and close the window.

Starting the portal server

- To start the portal server from the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server** and select **Start**.
- (*Linux or AIX only*) To start the portal server from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is /opt/IBM/ITM.

  ```
  ./itmcmd agent start cq
  ```

  where cq is the product code for the portal server.

Step 4: Install and configure communications for the Summarization and Pruning Agent

Complete the tasks described in the following table, in the order listed, to install and configure the Summarization and Pruning Agent.

Table 116. Tasks for installing and configuring communications for the Summarization and Pruning Agent

Task	Procedure
Install the Summarization and Pruning Agent if you have not already installed it. For best performance, install the Summarization and Pruning Agent on the same computer as the data warehouse. The installation procedure for Windows includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (*Configuring the monitoring agent*). See the information at right. Be sure to perform all referenced installation and configuration procedures. **Note**: The Summarization and Pruning Agent is not automatically started after installation. Do not complete any step or procedure for starting the agent at this point.	To install a Summarization and Pruning Agent on Windows, complete the procedure "Windows: Installing a monitoring agent" on page 253. To install a Summarization and Pruning Agent on Linux or UNIX, complete the procedure "Linux or UNIX: Installing a monitoring agent" on page 259, including the following subsections: • *Installing the monitoring agent* • *Configuring the monitoring agent* • *Changing the file permissions for agents* (if you used a non-root user to install the Warehouse Proxy) *Do not complete the procedure for starting the agent.*
Install an Oracle Type 4 JDBC driver on the computer where the Summarization and Pruning Agent is installed.	Obtain the Oracle JDBC Driver from the following Web site: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html The Oracle JDBC driver JAR file name and location after installation is as follows: `*oracleinstalldir*/jdbc/lib/ojdbc14.jar` The `ojdbc14.jar` file supports JRE 1.5 or higher, the required Java Runtime Environment for IBM Tivoli Monitoring.
Configure the Summarization and Pruning Agent. When you configure the Summarization and Pruning Agent, you configure the connection to the Tivoli Data Warehouse and you specify settings that control the operation of the Summarization and Pruning Agent. Perform this procedure whether or not the Summarization and Pruning Agent and the warehouse database are installed on the same computer.	"Configuring the Summarization and Pruning Agent (JDBC connection)" on page 595
Configure the Summarization and Pruning Agent to connect to the Tivoli Enterprise Portal Server. Perform this procedure whether or not the Summarization and Pruning Agent and the warehouse database are installed on the same computer.	See Step 9 of "Configuring the Summarization and Pruning Agent (JDBC connection)" on page 595.

Table 116. Tasks for installing and configuring communications for the Summarization and Pruning Agent (continued)

Task	Procedure
Configure history collection. When you configure history collection, you specify settings for how often to collect, aggregate, and prune data for individual monitoring agents and attribute groups. Configure history collection from the Tivoli Enterprise Portal.	See the *IBM Tivoli Monitoring: Administrator's Guide* for instructions on how to configure history collection.
Start the Summarization and Pruning Agent.	"Starting the Summarization and Pruning Agent" on page 608

Step 5: Install and configure communications for Tivoli Performance Analyzer

You can install Tivoli Performance Analyzer to a server which also has other Tivoli Monitoring components installed, or you can install it to a separate machine. The installation procedure is similar to that for monitoring agents. Complete the tasks described in the following table, in the order listed, to install and configure Tivoli Performance Analyzer.

Table 117. Tasks for installing and configuring communications for the Tivoli Performance Analyzer

Task	Procedure
Install Tivoli Performance Analyzer. The installation procedure for Windows includes steps for configuring the connection between the Tivoli Performance Analyzer and the hub Tivoli Enterprise Monitoring server. On Linux or AIX, this step is performed in a separate configuration procedure (Configuring the monitoring agent). See the information opposite. Be sure to perform all of the referenced installation and configuration procedures. **Note:** for sites setting up autonomous operation, the installation procedure includes steps for configuring the connection between the agent and the hub Tivoli Enterprise Monitoring Server. On Windows operating systems, if you want to run Tivoli Performance Analyzer without a connection to the hub, accept the defaults for the connection information, but specify a non valid name for the monitoring server. On UNIX and Linux operating systems, select **No TEMS** on the TEMS Connection in the configuration window.	To install Tivoli Performance Analyzer on Windows, complete the procedure "Windows: Installing a monitoring agent" on page 253. To install Tivoli Performance Analyzer agent on Linux or AIX, complete the procedure "Linux or UNIX: Installing a monitoring agent" on page 259, including the following subsections: • Installing the monitoring agent • Configuring the monitoring agent • Changing the file permissions for agents Do not complete the procedure for starting the agent.
Ensure that the latest Oracle patches are installed. Set the following system variable on the computer where Tivoli Performance Analyzer is installed. Restart the computer after setting the variable. The format of the **NLS_LANG** environment variable is: **NLS_LANG=language_territory.charset**. Set the language and territory to appropriate variables. For the United States this is **NLS_LANG=AMERICAN_AMERICA.AL32UTF8**.	See the Oracle documentation for instructions on how to install an Oracle client. Obtain Oracle ODBC drivers from the following Web site:http://www.oracle.com/technology/software/tech/windows/odbc/htdocs/utilsoft.html.
If the Tivoli Data Warehouse is located on a remote computer, create a TNS (Transparent Network Substrate) Service Name on the local computer where the Tivoli Performance Analyzer is located.	"Creating a TNS Service Name" on page 591
On the computer where the Tivoli Performance Analyzer is installed, configure an ODBC data source for the data warehouse, using the TNS Service Name that you created in the preceding step.	"Configuring an ODBC data source for an Oracle data warehouse" on page 592

Table 117. Tasks for installing and configuring communications for the Tivoli Performance Analyzer (continued)

Task	Procedure
Install an Oracle Type 4 JDBC driver on the computer where the Tivoli Performance Analyzer is installed.	Obtain the Oracle JDBC Driver from the following Web site: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html The Oracle JDBC driver JAR file name and location after installation is as follows: `oracleinstalldir/jdbc/lib/ojdbc14.jar` The `ojdbc14.jar` file supports JRE 1.5 or higher, the required Java Runtime Environment for IBM Tivoli Monitoring.
Configure Tivoli Performance Analyzer to connect to the data warehouse.	For Tivoli Performance Analyzer on Windows, see "Configuring a Tivoli Performance Analyzer on Windows (ODBC connection)" on page 592. For a Tivoli Performance Analyzer on Linux or AIX, see "Configuring Tivoli Performance Analyzer on Linux or UNIX (JDBC connection)" on page 593.
Start Tivoli Performance Analyzer.	"Starting Tivoli Performance Analyzer" on page 594

Creating a TNS Service Name

Create a TNS (Transparent Network Substrate) Service Name (also called a Net Service Name) on a computer where an Oracle client is installed if the Tivoli Data Warehouse exists on a *remote* Oracle server. The TNS Service name is needed to create an ODBC connection between the client and the server. Use this procedure to create a TNS Service name:

- On a Windows computer where Tivoli Performance Analyzer is installed.
- On a Windows computer where the Tivoli Enterprise Portal Server is installed.

Do not perform this procedure on the computer where the data warehouse (Oracle server) is installed or on a computer where there is no Oracle client (for example, on a computer where a Type 4 Oracle JDBC driver is used to communicate with the remote data warehouse).

Note: This procedure uses the default value for the warehouse name (`WAREHOUS`). Substitute a different value if you do not want to use the default name.

Complete the following steps to create the TNS Service Name. Click **Next** after each step.

1. Enter `dbca` at the Oracle command-line to start the Oracle Net Configuration Assistant tool.
2. On the Welcome window, select **Local Net Service Name configuration**.
3. Select **Add**.
4. Enter `WAREHOUS` in the **Service Name** field. (This is the remote name for the Tivoli Data Warehouse.)
5. Select **TCP** as the network protocol to communicate with the Tivoli Data Warehouse database.
6. Specify the fully qualified host name and port number of the computer where the warehouse database is installed.
7. Perform the connection test to verify the connection to the warehouse database.
8. Optionally change the default name in the **Net Service Name** field.

This is the TNS Service Name. The default name matches the name that you entered in Step 4. You can change this to a different name. The TNS Service Name can be considered a local alias for the remote Tivoli Data Warehouse name.

9. When prompted to configure another net service name, click **No** to return to the Welcome window.
10. Click **Finish**.

Configuring an ODBC data source for an Oracle data warehouse

An Oracle client on Windows requires an ODBC connection to the data warehouse. For the Tivoli Performance Analyzer, you must configure the ODBC connection manually.

Complete the following procedure to set up an ODBC connection for a Tivoli Performance Analyzer on Windows to a local or remote Tivoli Data Warehouse.

Note: This procedure uses default values for the data source name (`ITM Warehouse`) and warehouse user ID (`ITMUser`). (Default values are used in configuration procedures for warehousing components.) Substitute different values if you do not want to use the default values.

1. On the computer where the Tivoli Performance Analyzer is installed, open the Control Panel.
2. Click **Administrative Tools → Data Sources (ODBC)**
3. Click **Add** in the **System DSN** tab in the ODBC Data Source Administrator window.
4. Select the Oracle ODBC driver:
   - For Oracle 9, the ODBC driver name is **Oracle in Ora9ias_home**.
   - For Oracle 10, the ODBC driver name is **Oracle in OraDb10g_home1**.
5. Enter `ITM Warehouse` in the **Data Source Name** field.
6. Enter the TNS Service Name in the **TNS Service Name** field. This is the name that you specified in Step 8 on page 579 (for example, `WAREHOUS`).
7. Enter `ITMUser` in the **User ID** field.
8. Click **Test Connection**.
9. In the Oracle ODBC Driver Connect window, enter the TNS Service Name, and the user ID and password of the warehouse user.
10. Click **OK** on the **Connection successful** message.

Configuring a Tivoli Performance Analyzer on Windows (ODBC connection)

Use this procedure to configure Tivoli Performance Analyzer on Windows to connect to an Oracle Tivoli Data Warehouse:

1. Log on to the Windows system where the Tivoli Performance Analyzer is installed and begin the configuration:
   a. Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

      The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Tivoli Performance Analyzer** and click **Reconfigure**.

   c. Click **OK** on the message regarding connection to a hub monitoring server.
2. The next two windows (entitled Performance Analyzer: Agent Advanced Configuration) contain the settings for the connection between the Tivoli Performance Analyzer and the hub monitoring server. These settings were specified when Tivoli Performance Analyzer was installed. Click **OK** on each window to accept the settings.
3. Click **Yes** on the message asking if you want to configure the ODBC data source.
4. Select **ODBC** from the list of selectable agent database connection types.
5. Set your Database Type to Oracle.
6. Specify the Data Source Name - **Agent ODBC DSN** (**ITM Warehouse** by default)

Note:

- Tivoli Performance Analyzer does not create this DSN - it must already exist. If you are installing Performance Analyzer on the same machine where TEP Server is installed, you can use the existing data source created by Tivoli Monitoring. Otherwise, you must create a new System DSN manually, prior to re-configuring Performance Analyzer.
- On 64-bit versions of Windows, data sources created by the default **ODBC Data Source Administrator** applet available from the **Control Panel** are not available for 32-bit applications. Therefore you must use the 32-bit version of the **ODBC Data Source Administrator** applet from `<WINDOWS>\SysWOW64\odbcad32.exe`.

7. Type the **Username** and **Password**. The entries in these fields are used to connect to the Tivoli Data Warehouse and are the same credentials as those used by the Tivoli Enterprise Portal Server, the Warehouse Proxy Agent and the Summarization and Pruning Agent to communicate with Tivoli Data Warehouse.
8. Click **Next** to proceed to the Advanced Configuration window.
9. You can enable Advanced Configuration to specify TDW Schema and The TDW database schema. If you do not select **Enable advanced configuration** these options are greyed out.
10. You can also choose whether you want the agent to Initialize PA tables and OS domain tasks.

    Note: Setting Initialize PA tables to YES will remove and recreate all previously created tables deleting all user tasks and reverting each OS task to its default.
11. Use the **Bypass connection tests** option to finish the configuration without running connection tests.
12. Click **OK** to finish the configuration process.

    Note: The values for the data source name, and database user ID and password must match the values that you used when configuring an ODBC connection for Tivoli Performance Analyzer.

Table 118. Configuration information for the Tivoli Data Warehouse database on Oracle

Field	Default value	Description
ODBC DSN	ITM Warehouse	The name of the data source.
Username	ITMUser	The name of the Windows OS user that the Tivoli Performance Analyzer will use to access the Tivoli Data Warehouse database.
Password	itmpswd1	The password for the Windows OS user. If your environment requires complex passwords (passwords that require both alpha and numeric characters), specify a password that complies with these requirements.
Test Connection		Test the connection to the Tivoli Data Warehouse Database based on the completed fields above: ODBC DSN, Username, and Password.

Configuring Tivoli Performance Analyzer on Linux or UNIX (JDBC connection)

Use this procedure to configure Tivoli Performance Analyzer on Linux or UNIX to connect to an Oracle data warehouse:

1. To begin the configuration, log on to the computer where Tivoli Performance Analyzer is installed.
   a. Change to the `install_dir/bin` directory and run the following command:

   ```
   ./itmcmd manage [-h install_dir]
   ```

   where `install_dir` is the installation directory for IBM Tivoli Monitoring. The default installation directory is `/opt/IBM/ITM`. The Manage Tivoli Enterprise Monitoring Services window is displayed.

   b. Right-click **Performance Analyzer** and click **Configure**. The Configure Tivoli Performance Analyzer window is displayed.
2. Set the Database Type to Oracle.
3. Type the username and the password. The entries in these fields are used to connect to the Tivoli Data Warehouse.
4. Review all the defaults in the Agent Configuration window and change as required.
   a. If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of localhost.
   b. Change the port number if necessary (the default port number for Oracle is 1521).
   c. If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name. (See “Creating the warehouse database on DB2 for Linux, UNIX, and Windows” on page 494.)
5. Specify the JDBC Driver. The Oracle JDBC Type 4 driver name is `oracle.jdbc.driver.OracleDriver`.
6. Specify the JDBC Driver Path, which should be provided as a list of JAR files with the full path separated by ":".

   Note: The Oracle JDBC driver JAR file name and default location after downloading from the Web are as follows:

   ```
   doracleinstalldir/jdbc/lib/ojdbc14.jar
   ```

   Fast path: You can use the Browse button to specify the path. In such a case a file list is added at the end of the JDBC Driver Path text field, separated from the existing content by a path separator.
7. You can use the **Test connection** button to check whether the connection can be initiated.
8. Click **Next** to proceed to the Advanced Configuration window.
   a. You can enable Advanced Configuration to specify TDW Schema and Configuration schema. If you do not select Enable advanced configuration, all of these options are greyed out.
   b. You can also choose whether you want the agent to initialize PA tables.

      Note: Setting Initialize PA tables to YES will remove and recreate all previously created tables deleting all user tasks and reverting each OS task to its default.
   c. Use the **Bypass connection tests** option to finish the configuration without running connection tests.
9. Click **Save** to save your settings and close the window.

Starting Tivoli Performance Analyzer

To start Tivoli Performance Analyzer from the Manage Tivoli Enterprise Monitoring Services window, right-click Tivoli Performance Analyzer and select Start. To start the Tivoli Performance Analyzer agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is `/opt/IBM/ITM`.

```
./itmcmd agent start pa
```

where pa is the product code for Tivoli Performance Analyzer agent.

Chapter 24. Tivoli Data Warehouse solutions: common procedures

This chapter contains information and procedures for Tivoli Data Warehouse solutions that use any of the supported database platforms: IBM DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, and Oracle. Read the chapter pertaining to the database platform you are using for the Tivoli Data Warehouse before performing any of these procedures.

- "Configuring the Summarization and Pruning Agent (JDBC connection)"
- "Starting the Summarization and Pruning Agent" on page 608
- "Installing and configuring multiple Warehouse Proxy Agents" on page 608
- "Running the warehouse agents autonomously" on page 609
- "Testing the connection between the portal server and the Tivoli Data Warehouse" on page 614
- "Tuning the performance of the Warehouse Proxy" on page 617
- "WAREHOUSELOG and WAREHOUSEAGGREGLOG tables" on page 638
- "Configuring the Warehouse Proxy Agent on Linux or UNIX: command-line procedure" on page 605
- "Configuring the Summarization and Pruning Agent on Linux or UNIX: command-line procedure" on page 606

Configuring the Summarization and Pruning Agent (JDBC connection)

Use this procedure to configure the Summarization and Pruning Agent to connect to a Tivoli Data Warehouse database created on any of the supported database platforms and operating systems.

Before you begin

The JDBC driver JAR files for your database platform must be located on the computer where you installed the Summarization and Pruning Agent. Use a Type 4 JDBC driver. Do not use the Type 2 driver.

- If you are using DB2 for Linux, UNIX, and Windows for your Tivoli Data Warehouse database, the JDBC driver files are included with the database platform installation. If the warehouse database and the Summarization and Pruning Agent are installed on the same computer, the JDBC driver files are already present. It is not necessary to move them from their current location. If your warehouse database is located on a remote computer, copy the driver files to the local computer (the computer where you installed the Summarization and Pruning Agent). Copy the files to any directory to which the user the Summarization and Pruning Agent is running as has access to.
- If you are using Oracle or Microsoft SQL Server for your Tivoli Data Warehouse database, download the driver file from the company Web site to the computer where you installed the Summarization and Pruning Agent. Download the files to any directory to which the user the Summarization and Pruning Agent is running as has access to.

Table 119 on page 596 shows where to obtain the driver files for each database platform.

Table 119. Where to obtain the JDBC driver files for the Summarization and Pruning Agent

Database platform	JDBC driver files

IBM DB2 for Linux, Unix and Windows	Use the DB2 for Linux, UNIX, and Windows JDBC Universal Driver (Type 4 driver). The DB2 for Linux, UNIX, and Windows driver files are located with your Tivoli Data Warehouse server installation. The Type 4 driver file names and locations are as follows:
	`db2installdir/java/db2jcc.jar`
	`db2installdir/java/db2jcc_license_cu.jar`
	where *db2installdir* is the directory where DB2 for Linux, UNIX, and Windows was installed. The default DB2 for Linux, UNIX, and Windows Version 9 installation directory is as follows:
	• On Windows: `C:\Program Files\IBM\SQLLIB`
	• On AIX: `/usr/opt/db2_09_01`
	• On Linux and Solaris: `/opt/IBM/db2/V9.1`

Microsoft SQL Server	Use the latest Microsoft SQL Server JDBC driver, which supports SQL Server 2008, 2005, and 2000, to connect to a Tivoli Data Warehouse on either SQL Server 2000 or SQL Server 2005. (The SQL Server 2005 JDBC Driver works with a Tivoli Data Warehouse on SQL Server 2000.) Go to the Microsoft Web page at: http://www.microsoft.com and search for **JDBC driver**. Download and install the driver to the computer where you installed the Summarization and Pruning Agent. Follow the instructions on the Microsoft download page for installing the driver. The SQL Server JAR file name and location after installation is as follows: `<mssqlinstalldir>/sqljdbc_1.1/enu/sqljdbc4.jar`.
Oracle	Obtain the Oracle JDBC Type 4 driver from the following Web site: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html The Oracle JDBC driver JAR file name and location after installation is as follows: `oracleinstalldir/jdbc/lib/ojdbc14.jar` The ojdbc14.jar file supports JRE 1.5 or higher, the required Java Runtime Environment for IBM Tivoli Monitoring.

Procedure

Complete the following steps to configure the Summarization and Pruning Agent.

1. Log on to the computer where the Summarization and Pruning Agent is installed and begin the configuration:
   a. Open the Manage Tivoli Enterprise Monitoring Services window:
      - On Windows, click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.
      - On Linux or UNIX, change to the *install_dir*/bin directory and run the following command:

        ```
        ./itmcmd manage [-h install_dir]
        ```

        where *install_dir* is the installation directory for IBM Tivoli Monitoring. The default installation directory is /opt/IBM/ITM.

   b. Right-click **Summarization and Pruning Agent**.

   c. On Windows, click **Configure Using Defaults**. On Linux or UNIX, click **Configure**. If you are reconfiguring, click **Reconfigure**.
2. Review the settings for the connection between the Summarization and Pruning Agent and the hub Tivoli Enterprise Monitoring server. These settings were specified when the Summarization and Pruning Agent was installed.

- On Windows, perform the following steps:
  a. On the Warehouse Summarization and Pruning Agent: Agent Advanced Configuration window, verify the communications protocol of the hub monitoring server in the **Protocol** drop down list. Click **OK**.
  b. On the next window, verify the host name and port number of the hub monitoring server. Click **OK**.

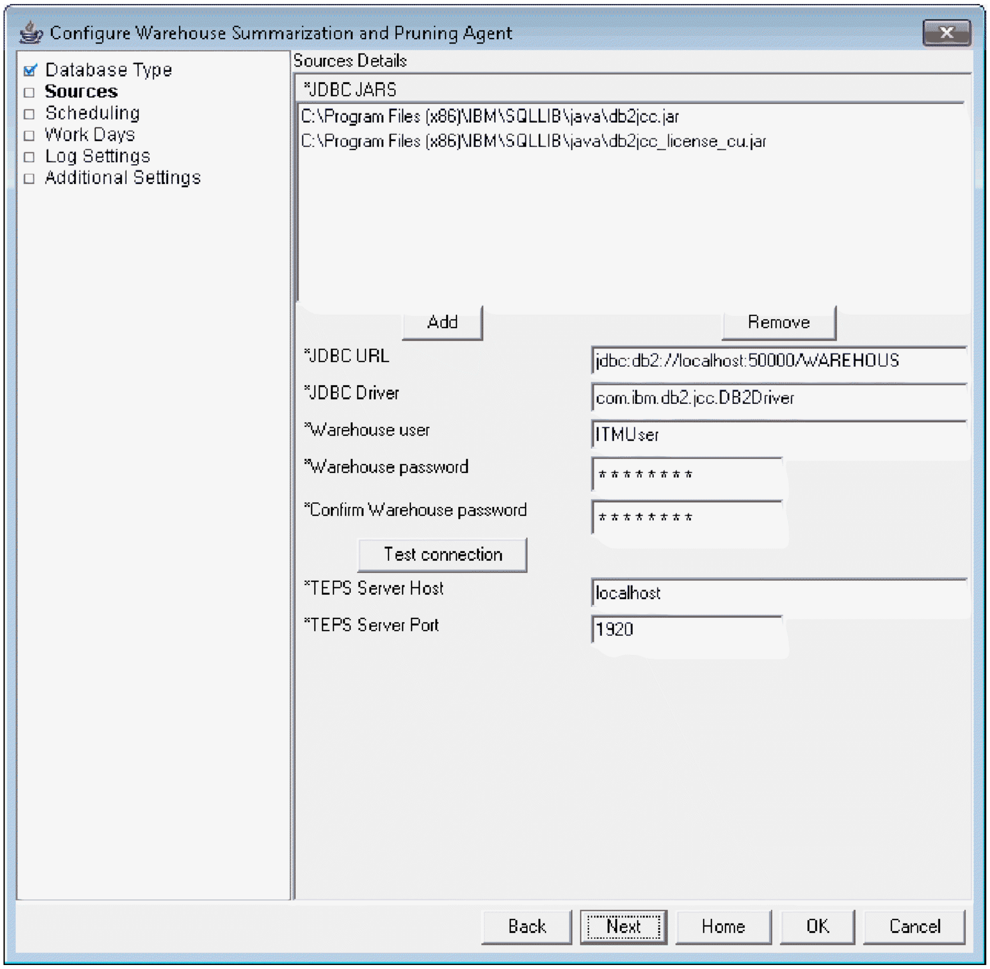
  For information about the different protocols available to the hub monitoring server on Windows, and associated default values, see 12e on page 211.
- On Linux or UNIX, verify the following information on the **TEMS Connection** tab:
  – The hostname of the hub monitoring server in the **TEMS Hostname** field. (If the field is not active, clear the **No TEMS** check box.)
  – The communications protocol that the hub monitoring server uses in the **Protocol** drop down list.
    - If you select IP.UDP, IP.PIPE, or IP.SPIPE, enter the port number of the monitoring server in the **Port Number** field.
    - If you select SNA, enter information in the **Net Name**, **LU Name**, and **LOG Mode** fields.

  For information about the different protocols available to the hub monitoring server on Linux or UNIX, and associated default values, see Table 38 on page 215.

3. When you are finished verifying or entering information about the hub monitoring server:
   - On Windows, click **Yes** on the message asking if you want to configure the Summarization and Pruning Agent.
   - On Linux or UNIX, click the **Agent Parameters** tab.

   A multi-tabbed configuration window is displayed with the **Sources** tab at the front.

   Figure 150 on page 598 shows the configuration window for a Summarization and Pruning Agent on Windows (values displayed are for a DB2 for Linux, UNIX, and Windows warehouse database). The configuration window for a Summarization and Pruning Agent on Linux or UNIX is similar.



*JDBC JARS
C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar
C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar

*JDBC URL: jdbc:db2://localhost:50000/WAREHOUS
*JDBC Driver: com.ibm.db2.jcc.DB2Driver
*Warehouse user: ITMUser
*Warehouse password: ********
*Confirm Warehouse password: ********
*TEPS Server Host: localhost
*TEPS Server Port: 1920

Figure 150. Sources pane of Configure Warehouse Summarization and Pruning Agent window

4. Add the names and directory locations of the JDBC driver JAR files to the **JDBC Drivers** list box:
   a. Click **Add** to display the file browser window. Navigate to the location of the driver files on this computer and select the Type 4 driver files for your database platform. See Table 119 on page 596 for the names and default locations of the driver files to add.
   b. Click **OK** to close the browser window and add the JDBC driver files to the list.

   If you need to delete an entry from the list, select the entry and click **Remove**.
5. The default values for the database platform you selected in the Database Type pane are displayed in the other text fields on the **Sources** pane. Change the default value displayed in the **JDBC URL** field if it is not correct. The following table lists the default Tivoli Data Warehouse URLs for the different database platforms:

Table 120. Tivoli Data Warehouse URLs

Database platform	Warehouse URL
IBM DB2 for Linux, UNIX, and Windows	jdbc:db2://localhost:60000/WAREHOUS
Oracle	jdbc:oracle:thin:@localhost:1521:WAREHOUS
Microsoft SQL Server 2000 or SQL Server 2005	jdbc:sqlserver://localhost:1433;databaseName=WAREHOUS

- If the Tivoli Data Warehouse is installed on a remote computer, specify the host name of the remote computer instead of `localhost`.
- Change the port number if it is different.
- If the name of the Tivoli Data Warehouse database is not WAREHOUS, replace WAREHOUS with the actual name.

6. Verify the JDBC driver name.

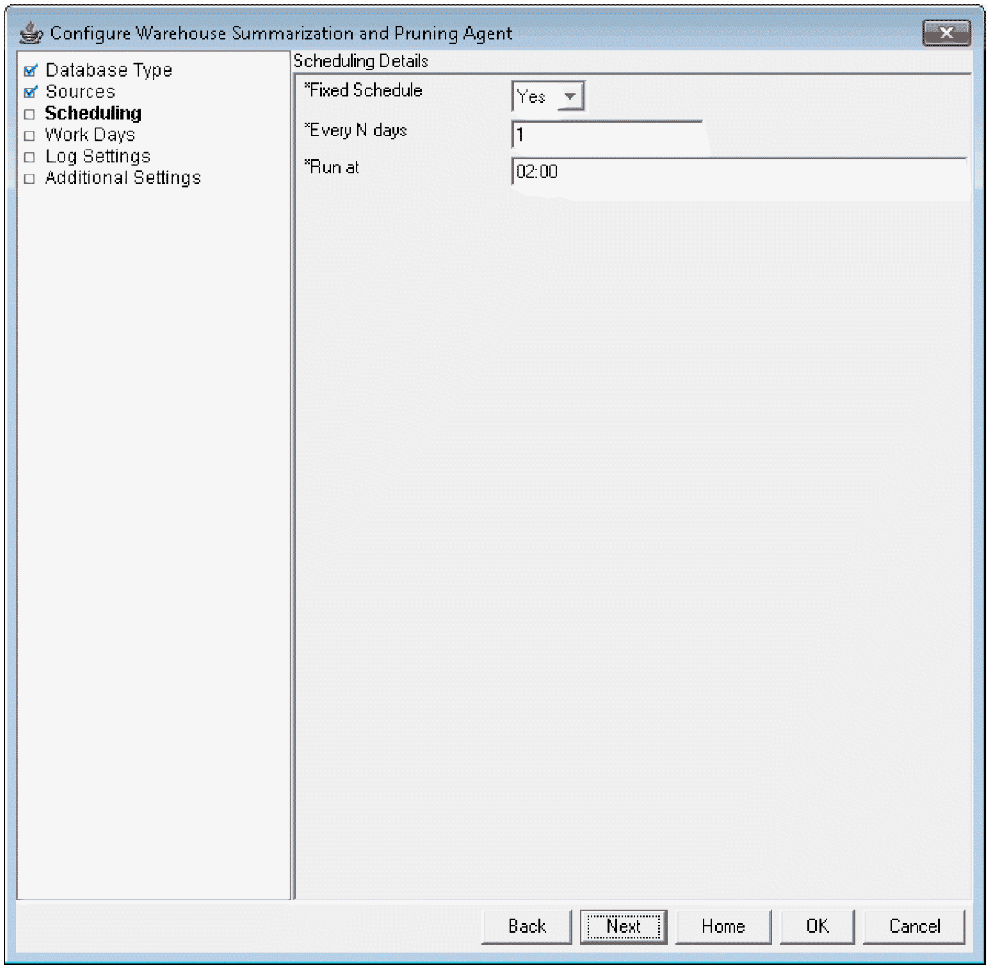
   The following table lists the JDBC Type 4 driver names for each database platform:

Table 121. JDBC driver names

Database platform	JDBC driver name
IBM DB2 for Linux, UNIX, and Windows	com.ibm.db2.jcc.DB2Driver
Oracle	oracle.jdbc.driver.OracleDriver
Microsoft SQL Server	com.microsoft.sqlserver.jdbc.SQLServerDriver **Note:** This is the name of the 2005 SQL Driver. Do not use the SQL Server 2000 JDBC driver, even if the Tivoli Data Warehouse was created in Microsoft SQL 2000. (The name of the 2000 SQL driver was com.microsoft.jdbc.sqlserver.SQLServerDriver. Note the reversal of the string `jdbc.sqlserver`.)

7. If necessary, change the entries in the **Warehouse user** and **Warehouse password** fields to match the user name and password that were created for the Tivoli Data Warehouse. The default user name is `itmuser` and the default password is `itmpswd1`.
8. In the **TEPS Server Host** and **TEPS Server Port** fields, enter the host name of the computer where the Tivoli Enterprise Portal Server is installed and the port number that it uses to communicate with the Summarization and Pruning Agent.

   Note: The default Tivoli Enterprise Portal Server interface port of 15001 is also used after the Summarization and Pruning Agent's initial connection to the portal server over port 1920. Any firewalls between the two need to allow communications on either 15001 or whichever port is defined for any new Tivoli Enterprise Portal Server interface used per the instructions in “Defining a Tivoli Enterprise Portal Server interface on Windows” on page 408.
9. Click **Test connection** to ensure you can communicate with the Tivoli Data Warehouse database.
10. Select the **Scheduling** check box to specify when you want summarization and pruning to take place. You can schedule it to run on a fixed schedule or on a flexible schedule:


Configure Warehouse Summarization and Pruning Agent

☑ Database Type
☑ Sources
□ **Scheduling**
□ Work Days
□ Log Settings
□ Additional Settings

Scheduling Details

*Fixed Schedule: Yes

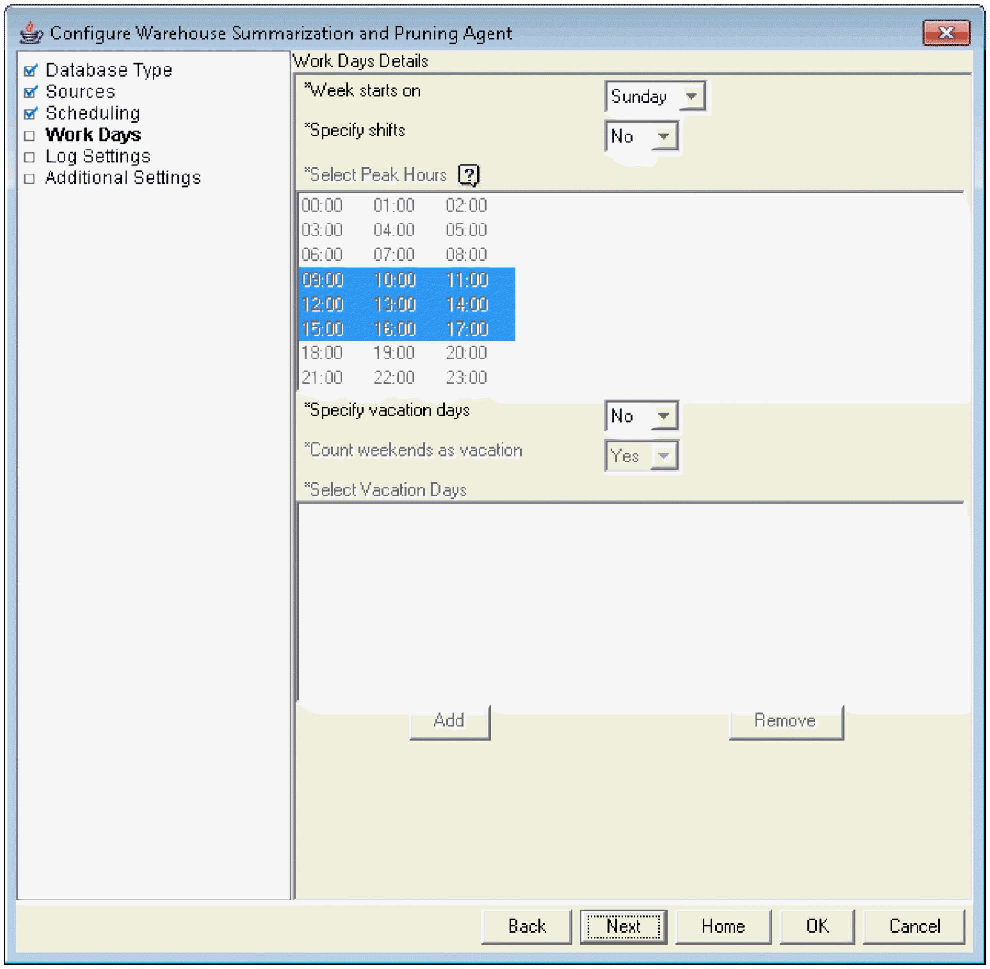
*Every N days: 1

*Run at: 02:00

Figure 151. Scheduling pane of Configure Warehouse Summarization and Pruning Agent window

Note: If you select Fixed, the Summarization and Pruning Agent does not immediately perform any summarization or pruning when it *starts*. It performs summarization and pruning when it *runs*. It runs according to the schedule you specify on the **Scheduling** pane. If you select Flexible, the Summarization and Pruning Agent runs once immediately after it is started and then at the interval you specified except during any blackout times.

11. Specify shift and vacation settings in the **Work Days** pane:



Configure Warehouse Summarization and Pruning Agent

- ☑ Database Type
- ☑ Sources
- ☑ Scheduling
- □ **Work Days**
- □ Log Settings
- □ Additional Settings

Work Days Details

*Week starts on: Sunday

*Specify shifts: No

*Select Peak Hours

00:00	01:00	02:00
03:00	04:00	05:00
06:00	07:00	08:00
09:00	10:00	11:00
12:00	13:00	14:00
15:00	16:00	17:00
18:00	19:00	20:00
21:00	22:00	23:00

*Specify vacation days: No

*Count weekends as vacation: Yes

*Select Vacation Days

Figure 152. Work Days pane of Configure Warehouse Summarization and Pruning Agent window

When you enable and configure shifts, IBM Tivoli Monitoring produces three separate summarization reports:

- Summarization for peak shift hours
- Summarization for off-peak shift hours
- Summarization for all hours (peak and off-peak)

Similarly, when you enable and configure vacations, IBM Tivoli Monitoring produces three separate summarization reports:

- Summarization for vacation days
- Summarization for nonvacation days
- Summarization for all days (vacation and nonvacation)

Complete the following steps to enable shifts, vacations, or both:

- Select when the beginning of the week starts.
- To configure shifts:
  a. Select **Yes** in the **Specify shifts** drop-down list.
  b. Optionally change the default settings for peak and off peak hours by selecting hours in the **Select Peak Hours** box.

Note: Changing the shift information after data has been summarized creates an inconsistency in the data. Data that was previously collected is not summarized again to account for the new shift values.

- To configure vacation settings:
  a. Select **Yes** in the **Specify vacation days** drop-down list to enable vacation days.
  b. Select **Yes** in the drop-down list if you want to specify weekends as vacation days.
  c. Select **Add** to add vacation days.
  d. Select the vacation days you want to add from the calendar.

     On UNIX or Linux, right-click, instead of left-click, to select the month and year.

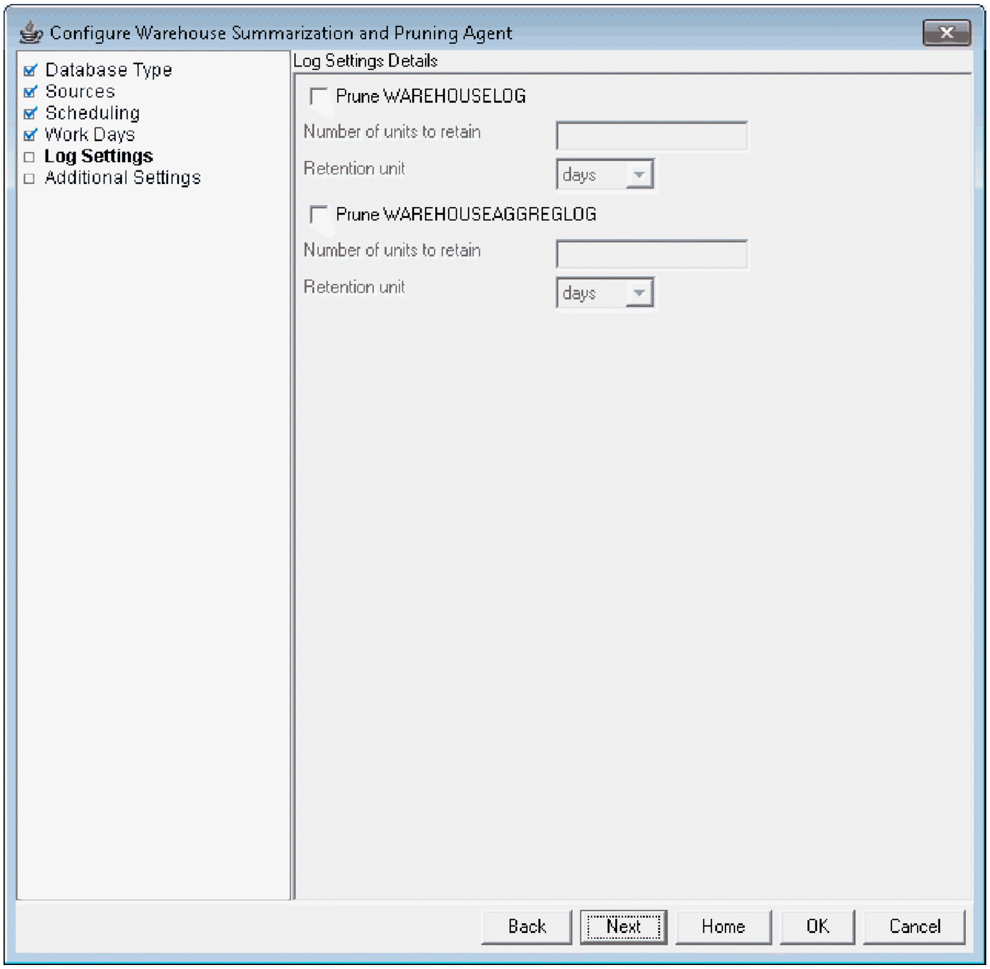
     The days you select are displayed in the list box.

     If you want to delete any days you have previously chosen, select them and click **Delete**.

     Notes:

     1) Add vacation days in the future. Adding vacation days in the past creates an inconsistency in the data. Data that was previously collected is not summarized again to account for vacation days.
     2) Enabling shifts or vacation periods can significantly increase the size of the warehouse database. It will also negatively affect the performance of the Summarization and Pruning Agent.

12. Select the **Log Settings** check box to set the intervals for log pruning:



Log Settings Details

□ Prune WAREHOUSELOG

Number of units to retain

Retention unit: days

□ Prune WAREHOUSEAGGREGLOG

Number of units to retain

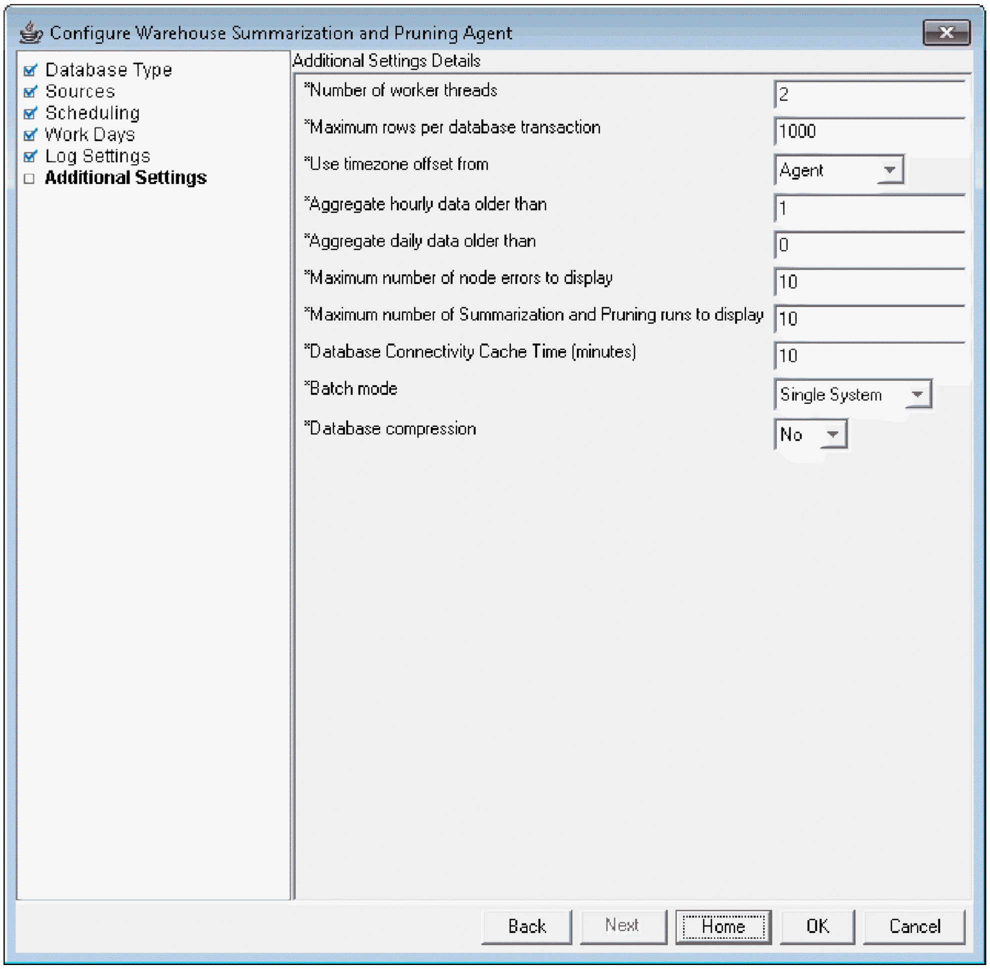
Retention unit: days

Figure 153. Log Settings pane of Configure Warehouse Summarization and Pruning Agent window

- Select Prune WAREHOUSELOG, select the number of units for which data should be kept, and the unit of time (day, month or year).
- Select Prune WAREHOUSEAGGREGLOG, select the number of units for which data should be kept, and the unit of time (day, month or year).

Note: Beginning with V6.2.3, these table inserts are disabled by default. The self-monitoring workspaces provided by the Summarization and Pruning Agent provide sufficient information to determine if the agent is operating correctly. Tivoli Data Warehouse log tables can grow very large and require regular pruning. This new default configuration decreases the Summarization and Pruning Agent processing time, and decreases database resource utilization and contention.

13. Specify additional summarization and pruning settings in the **Additional Settings** pane:



Additional Settings Details

*Number of worker threads: 2

*Maximum rows per database transaction: 1000

*Use timezone offset from: Agent

*Aggregate hourly data older than: 1

*Aggregate daily data older than: 0

*Maximum number of node errors to display: 10

*Maximum number of Summarization and Pruning runs to display: 10

*Database Connectivity Cache Time (minutes): 10

*Batch mode: Single System

*Database compression: No

Figure 154. Additional Settings pane of Configure Warehouse Summarization and Pruning Agent window

a. Specify the number of additional threads you want to use for handling summarization and pruning processing. The number of threads should be 2 * N, where N is the number of processors running the Summarization and Pruning Agent. A higher number of threads can be used, depending on your database configuration and hardware.

b. Specify the maximum rows that can be deleted in a single pruning transaction. Any positive integer is valid. The default value is `1000`. There is no value that indicates you want all rows deleted.

   If you increase the number of threads, you might consider increasing this value if your transaction log allows for it. The effective number of rows deleted per transaction is based on this value divided by the number of worker threads.

c. Indicate a time zone for historical data from the **Use timezone offset from** drop down list.

   This field indicates which time zone to use when a user specifies a time period in a query for monitoring data.

   - Select **Agent** to use the time zone (or time zones) where the monitoring agents are located.
   - Select **Warehouse** to use the time zone where the Summarization and Pruning Agent is located. If the Tivoli Data Warehouse and the Summarization and Pruning Agent are in different time zones, the **Warehouse** choice indicates the time zone of the Summarization and Pruning Agent, not the warehouse.

   Skip this field if the Summarization and Pruning Agent and the monitoring agents that collect data are all in the same time zone.

d. Specify the age of the data you want summarized in the **Aggregate hourly data older than** and **Aggregate daily data older than** fields. The default value is `1` for hourly data and `0` for daily data.

e. The **Maximum number of node errors to display** refers to the node error table in the Summarization and Pruning workspace. It determines the maximum number of rows that workspace is to save and display.

f. The **Maximum number of summarization and pruning runs to display** refers to the Summarization and Pruning Run table in the Summarization and Pruning workspace. It determines the maximum number of rows that workspace is to save and display.

Maximum number of Summarization and Pruning runs to display and Maximum number of node errors to display together determine the number of rows shown in the Summarization and Pruning overall run table and Errors table respectively. There is a minimum value of 10 for each. These equate to keywords KSY_SUMMARIZATION_UNITS and KSY_NODE_ERROR_UNITS in file `KSYENV/sy.ini`.

g. The **Database Connectivity Cache Time** determines how long after a positive check for connectivity that the result will be cached. Longer times may result in inaccurate results in the workspace; however, it saves processing time.

Database Connectivity Cache Time records the number of minutes to cache the database connectivity for MOSWOS reporting purposes. The minimum value is 5 minutes. This equates to keyword KSY_CACHE_MINS in file `KSYENV/sy.ini`.

h. **Batch mode** determines if data from different managed systems are used in the same database batch; this setting also improves performance.

Batch mode controls the batching method used by the Summarization and Pruning Agent. A value of Single Managed System (0) means that data should only be batched for the same system, whereas a value of Multiple Managed System (1) means that data from multiple systems can be batched together; this can lead to higher performance at potentially bigger transaction sizes. The default value is Single Managed System (0). This equates to keyword KSY_BATCH_MODE in file `KSYENV/sy.ini`.

i. Specify if you want to turn **Database compression** on or off.

To change these values, you can either use the Summarization and Pruning configuration window's **Additional settings** tab or update these parameters directly in file `KSYENV/sy.ini`.

14. Save your settings and close the window. Click **Save** to save your settings. On Windows, click **Close** to close the configuration window.
    - On Windows, click **Save** and then click **Close**.
    - On Linux or UNIX, click **Save** and then click **Cancel**.

Configuring the Warehouse Proxy Agent on Linux or UNIX: command-line procedure

Complete the following steps to configure the Warehouse Proxy Agent from the command-line on Linux or UNIX:

1. Log on to the computer where the Warehouse Proxy Agent is installed.
2. At the command-line change to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the directory where you installed the product.
3. Run the following command to start configuring the Warehouse Proxy Agent:

   ```
   ./itmcmd config -A hd
   ```

   where `hd` is the product code for the Warehouse Proxy Agent.

Here is a sample of Warehouse Proxy Agent configuration from the command-line. DB2 is the database used in this example, for the other supported database platforms the responses will be different:

```
itmcmd config -A hd

Database Type
Database [ 1=DB2, 2=Oracle, 3=Microsoft SQL Server ] (default is: 1):

Agent Parameters :
Fully qualified paths to JDBC JAR files (comma separated)
      JDBC JARs List (default is: ): /data/jdbc/db2jcc.jar,/data/jdbc/db2jcc_license_cu.jar
The Warehouse JDBC URL
     JDBC URL (default is: jdbc:db2://localhost:50000/WAREHOUS):
The Warehouse JDBC Driver
     JDBC Driver (default is: com.ibm.db2.jcc.DB2Driver):
The Warehouse database username
     Username (default is: ITMUSER):
The Warehouse database user password
     Enter Password (default is: ):
     Re-type : Password (default is: ):
Space or comma separated list of Tivoli Enterprise Monitoring Server instances served by
this Warehouse Proxy agent.
*ANY can be specified if this Warehouse Proxy agent will export data of any agents connected to
any TEMS. If the list is left blank, this Warehouse Proxy agent will be the default
Warehouse proxy agent.
    Warehouse TEMS List (default is: ): REMOTE_ITMTDWP12
Batch Database Operations
    Use Batch [ 1=TRUE, 2=FALSE ] (default is: 1):
Database Compression option
     Database Compression [ 1=TRUE, 2=FALSE ] (default is: 2):
Enable the compression of historical data from Z/OS sources before upload to the
Warehouse Proxy Server
    Warehouse Compression for Z/OS Sources [ 1=TRUE, 2=FALSE ] (default is: 2):
Enable the compression of historical data from distributed sources before upload to the
Warehouse Proxy Server
    Warehouse Compression for Distributed Sources [ 1=TRUE, 2=FALSE ] (default is: 1):
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1): 1
     TEMS Host Name (Default is: itmtdwp18):
```

Configuring the Summarization and Pruning Agent on Linux or UNIX: command-line procedure

Complete the following steps to configure the Summarization and Pruning Agent from the command-line on Linux or UNIX:

1. Log on to the computer where the Summarization and Pruning Agent is installed.
2. At the command-line change to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the directory where you installed the product.
3. Run the following command to start configuring the Summarization and Pruning Agent:

   ```
   ./itmcmd config -A sy
   ```

   where `sy` is the product code for the Summarization and Pruning Agent.

Here is a sample of Summarization and Pruning Agent configuration from the command-line. DB2 is the database used in this example, for the other supported database platforms the responses will be different:

```
itmcmd config -A sy

Choose the database type
Database Type
    Database [ 1=DB2, 2=Oracle, 3=Microsoft SQL Server ] (default is: 1):

Sources Details

Fully qualified paths to JDBC JAR files (comma separated)
    JDBC JARs List (default is: /data/jdbc/db2jcc.jar, /data/jdbc/db2jcc_license_cu.jar):
The Warehouse JDBC URL
```

```
    JDBC URL (default is: jdbc:db2://localhost:50000/WAREHOUS):
The Warehouse JDBC Driver
    JDBC Driver (default is: com.ibm.db2.jcc.DB2Driver):
The Warehouse user
    Warehouse user (default is: itmuser):
The Warehouse password
     Enter Warehouse password (default is: *):
     Re-type : Warehouse password (default is: *):

The TEPS hostname
     T EPS Server Host (default is: localhost):
The TEPS port (default 1920)
      TEPS Server Port (default is: 1920):

Scheduling Details:

If fixed scheduling is in use
     Fixed Schedule [ 1=No, 2=Yes ] (default is: 2):
The number of days between runs (default is 1)
     Every N days (default is: 1):
The fixed hour to run (valid values are 0-12, default is 2)
     Hour to run (default is: 02):
The fixed minute to run (default is 0)
     Minute to run (default is: 00):
AM or PM
      AM/PM [ 1=AM, 2=PM ] (default is: 1):
Minutes between flexible runs
      Every N minutes (default is: 60):
Exception times in HH:MM-HH:MM format (24 hour clock), comma separated when flexible
scheduling shouldn't run
       Blackout (default is: ):

Log Settings Details

Specify whether the WAREHOUSELOG table will be pruned. Format is nnn.unit where nnn is
the number of units and unit is day, month or year. Specify blank to not prune the table.
      Prune WAREHOUSELOG (default is: ):
Specify whether the WAREHOUSEAGGREGLOG table will be pruned. Format is nnn.unit where nnn is
the number of units and unit is day, month or year. Specify blank to not prune the table.
      Prune WAREHOUSEAGGREGLOG (default is: ):

Additional Settings Details

The number of worker threads to be used
      Number of worker threads (default is: 2):
The maximum number of rows per transaction (effective size is this value divided by number
of worker threads)
      Maximum rows per database transaction (default is: 1000):
Which timezone to use when aggregating the data: agent or warehouse (default is agent)
       Use timezone offset from [ 1=Agent, 2=Warehouse ] (default is: 1):
The minimum age for hourly data to be aggregated (default is 1)
       Aggregate hourly data older than (default is: 1):
The minimum age for daily data to be aggregated (default is 0)
       Aggregate daily data older than (default is: 0):
The number of errors to keep in memory (default is 10)
      Maximum number of node errors to display (default is: 10):
The number of summarization runs to keep in memory (default is 10)
      Maximum number of Summarization and Pruning runs to display (default is: 10):
The number of minutes to cache the database status (default is 10)
       Database Connectivity Cache Time (minutes) (default is: 10):
The type of batching to be used (default is single system)
       Batch mode [ 1=Single System, 2=Multiple System ] (default is: 1):
Enable database compression, if supported (default is no)
        Database compression [ 1=No, 2=Yes ] (default is: 1):
```

```
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
        TEMS Host Name (Default is: itmtdwp18):
```

Starting the Summarization and Pruning Agent

- To start the Summarization and Pruning Agent from the Manage Tivoli Enterprise Monitoring Services window, right-click **Summarization and Pruning** and select **Start**.
- (*Linux or UNIX only*) To start the Summarization and Pruning Agent from the command-line, run the following command from the bin directory of the IBM Tivoli Monitoring installation directory. The default installation directory is /opt/IBM/ITM.

  ```
  ./itmcmd agent start sy
  ```

  where `sy` is the product code for the Summarization and Pruning Agent.

Installing and configuring multiple Warehouse Proxy Agents

IBM Tivoli Monitoring supports multiple Warehouse Proxies within a single hub monitoring server environment. The provision for multiple warehouse proxies provides for greater scalability and performance in historical data collection. More importantly, use multiple proxies enhance reliability via their failover mechanism: if a Warehouse Proxy is unavailable, data can be inserted into the warehouse by a different Warehouse Proxy Agent (if the agents are configured properly for failover).

The support for multiple Warehouse Proxy Agents has the following important features:

- All Warehouse Proxy Agents within a single hub monitoring server environment export data to a single Tivoli Data Warehouse.
- Each Warehouse Proxy Agent is associated with a subset of monitoring server instances that you specify when you configure the proxy agent. Each Warehouse Proxy exports data only for monitoring agents that report to one of the monitoring servers (hub or remote) on the specified list. Alternatively, a given Warehouse Proxy Agent can be configured to service *any* monitoring server. This is important for backward compatibility and for configuring failover.

The following sequence of events explains how the monitoring agents, which collect the data for historical reports, know which Warehouse Proxy Agent to use:

1. When a Warehouse Proxy Agent starts, it registers with the Global Location Broker on the hub monitoring server, sending it the list of monitoring servers that it is configured to serve, or an indication that the Warehouse Proxy Agent can service any monitoring server. This registration process is repeated every hour.
2. Each monitoring server queries the Global Location Broker at regular intervals to determine which Warehouse Proxy it is associated with. The monitoring server then sends the address of this Warehouse Proxy to all of its child monitoring agents to use during historical data exports. You can change the default query interval of 60 minutes to some other value.

When a Warehouse Proxy Agent registers with the Global Location Broker, it is registered as the default proxy agent if no other proxy agent is already configured as the default. When a monitoring server queries the Global Location Broker for its associated Warehouse Proxy, the default proxy agent is used if that monitoring server is not on the list of servers for any proxy agent.

Setting a permanent socket address for a proxy agent

In some network environments, use of the Global Location Broker default registration algorithm is not supportable and a specific monitoring server variable must be used. This applies also when monitoring agents configure their connections as ephemeral (using `KDC_FAMILIES=ip.pipe use:y ephemeral:y` for instance).

KPX_WAREHOUSE_LOCATION is the variable that allows a fixed warehouse route to be delivered to connected agents. KPX_WAREHOUSE_LOCATION is an optional list of fully qualified, semicolon delimited network names and should be added in the monitoring server environment file located in different place depending on the platform:

- Windows: *install_dir*/CMS/KBBENV
- UNIX and Linux: *install_dir*/config/kbbenv.ini

KPX_WAREHOUSE_LOCATION is used instead of the routing string currently derived by the monitoring server from the Location Broker monitoring. If the variable KPX_WAREHOUSE_LOCATION is set on the hub monitoring server, then the Warehouse Proxy Agent is registered in the Global Location Broker. If the variable is set on the remote monitoring server, then the Warehouse Proxy Agent is registered in the Local Location Broker. This variable has a maximum length of 200 bytes. The syntax is:

```
KPX_WAREHOUSE_LOCATION= family_protocol:network_address[port number];
```

Verifying the configuration

Use the following trace settings to verify the configuration:

- To verify that a Warehouse Proxy is registering with the hub monitoring server and placing the correct entries into the Global Location Broker:
  1. Open the environment file for the proxy agent:
     - (Windows) *ITMinstall_dir*\TMAITM6\KHDENV
     - (Linux or AIX) *ITMinstall_dir*/config/hd.ini

     where *ITMinstall_dir* is the directory where you installed the product.
  2. Add the following entry to the KBB_RAS1 trace setting:

     ```
     KBB_RAS1=ERROR(UNIT:khdxrpcr STATE)
     ```

     This setting prints the value of KHD_WAREHOUSE_TEMS_LIST and shows any errors associated with its components.
- To determine which Warehouse Proxy a particular monitoring server uses for its agents:
  1. Open the environment file for the monitoring server:
     - (Windows) *ITMinstall_dir*\CMS\KBBENV
     - (Linux or UNIX) *ITMinstall_dir*/config/*hostname*_ms_*TEMSid*

     where *ITMinstall_dir* is the directory where you installed the product.
  2. Add the following entry to the KBB_RAS1 trace setting:

     ```
     KBB_RAS1=ERROR(UNIT:kpxrwhpx STATE)
     ```

     This setting prints entries in the RAS log of the monitoring server when a registration change occurs. The entry specifies the name and address of the new Warehouse Proxy Agent that the monitoring server is using.

Running the warehouse agents autonomously

Both the Warehouse Proxy Agent and Warehouse Summarization and Pruning Agent can run in autonomous ("unmanaged") mode.

If a Warehouse Proxy Agent is running in connected ("managed") mode, it makes its location available to the monitoring servers and application monitoring agents by registering the location with the Global Location Broker on the hub monitoring server, along with a list of monitoring servers that it supports. Remote monitoring servers and the agents that connect to them obtain the network addresses of the proxy agents to which they report from the hub. However, if the Warehouse Proxy Agent is running in autonomous ("unmanaged") mode, it does not register its location with the hub monitoring server; instead, monitoring agents are configured to obtain the location of their Warehouse Proxy Agents from their local configuration file or from a central configuration server facility. This allows the Warehouse Proxy Agent to support monitoring agents that are running autonomously (that is, without a connection to a monitoring server).

If the Summarization and Pruning Agent is running in connected mode, it obtains the information it needs about the collected historical data from the Tivoli Enterprise Portal Server. If the Summarization and Pruning Agent is configured to run in autonomous mode, it obtains the information about attribute data directly from attribute support files for the monitoring agents, and the summarization and pruning settings from the WAREHOUSESUMPRUNE table in the warehouse database. The location of the application support files is specified in the Summarization and Pruning Agent configuration file (see “Configuring a Summarization and Pruning Agent to run autonomously” on page 611). See “Configuring summarization and pruning without the Tivoli Enterprise Portal Server” on page 612 for more information on the WAREHOUSESUMPRUNE table.

Configuring a Warehouse Proxy Agent to run autonomously

For a Warehouse Proxy Agent to run in autonomous mode, it must be configured to run without registering its location with the hub monitoring server. Every monitoring agent that sends historical data to the Warehouse Proxy Agent must be configured with its location.

To configure the Warehouse Proxy Agent to run without registering, complete the following steps:

1. Install and configure one or more Warehouse Proxy Agents following the procedures in the appropriate chapter:
   - Chapter 20, “Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows,” on page 489
   - Chapter 22, “Tivoli Data Warehouse solution using Microsoft SQL Server,” on page 547
   - Chapter 23, “Tivoli Data Warehouse solution using Oracle,” on page 571

   Note: The installation procedure for Windows operating systems includes steps for configuring the connection between the proxy agent and the hub Tivoli Enterprise Monitoring Server. On Linux or UNIX operating systems, this step is performed in a separate configuration procedure. On Windows operating systems, if you want to run the Warehouse Proxy Agent without a connection to the hub, accept the defaults for the connection information, but specify a nonvalid name for the monitoring server. On UNIX and Linux operating systems, check **No TEMS** on the **TEMS Connection** tab of the configuration window.
2. Add the following variable to the Warehouse Proxy Agent environment file (*install_dir*\TMAITM6\KHDENV on Windows operating systems; *install_dir*/config/hd.ini on UNIX and Linux operating systems):
   ```
   KHD_REGWITHGLB=N
   ```
3. Configure the Warehouse Proxy Agent to use the same IP port number as you chose for the various autonomous agents that will be sending historical data to it; see the *IBM Tivoli Monitoring: Administrator's Guide* for details.
4. Restart the agent.
5. Restart the agent's Tivoli Enterprise Monitoring Server, if necessary.

   If the Warehouse Proxy Agent you reconfigured to run autonomously has previously connected to either a hub monitoring server or a remote monitoring server, the agent has already registered with the monitoring server to which it connected. To clear this registration information now that the agent is running autonomously, recycle the monitoring server. If the monitoring server is a remote monitoring server, also recycle the hub monitoring server to which it connects.

To configure a monitoring agent with the location of the Warehouse Proxy Agent or agents to which it should export historical data, complete the following steps:

1. Install the agent following “Installing monitoring agents” on page 253 or the documentation for the agent.

2. Open the monitoring agent environment file in a text editor:
   - Windows operating systems: `install_dir\TMAITM6\kpcenv`
   - Linux and UNIX operating systems: `install_dir/config/pc.ini`
   - z/OS operating system: `&hilev.&rte.RKANPARU(KPCENV)`

   where *pc* is the two-character product code for the monitoring agent (see Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815).
3. Add the following variable to the file:
   ```
   KHD_WAREHOUSE_LOCATION=family protocol.#network address[port number]
   ```

   The value of the variable can be a semicolon-delimited list of network addresses. For example:
   ```
   KHD_WAREHOUSE_LOCATION=ip.pipe:SYS2-XP[63358];ip:SYS2-XP[63358]
   ```
4. Restart the agent.

Configuring a Summarization and Pruning Agent to run autonomously

For a Summarization and Pruning Agent to run without connecting to a Tivoli Enterprise Portal Server, it must be configured to run autonomously and it must be able to locate the application support files for the monitoring agents.

To configure a Summarization and Pruning Agent to run in autonomous mode, complete the following steps:

1. If you have not already installed done so, install a Tivoli Enterprise Portal Server, and add application support for all types of monitoring agents that will be collecting historical data.
2. Install and configure Summarization and Pruning Agent following the instructions in the appropriate chapter:
   - Chapter 20, “Tivoli Data Warehouse solution using DB2 for Linux, UNIX, and Windows,” on page 489
   - Chapter 22, “Tivoli Data Warehouse solution using Microsoft SQL Server,” on page 547
   - Chapter 23, “Tivoli Data Warehouse solution using Oracle,” on page 571
3. If the Summarization and Pruning Agent is not installed on the same machine as the portal server, copy the required application support files to the machine on which the Summarization and Pruning agent resides.

   These files are named dock*pc*, where *pc* is the two-letter product code for the monitoring agent (see Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815). For Universal Agent attributes, the file is named `cccodicc`.

   On Windows operating systems, the files are located in `install_dir\cnps` directory; on Linux and UNIX operating systems, the files are located in the `installdir/arch/cq/data` directory.

   By default, the Summarization and Pruning Agent looks for the ODI files in the `install_dir\TMAITM6` directory (on Windows) or the `install_dir/arch/cq/data` directory (on UNIX and Linux). If you do not create this directory and copy the files to it, you must add the **KSY_AUTONOMOUS_ODI_DIR** variable to the Summarization and Pruning Agent environment file and specify the alternative location.

   Note: There is no need to copy the `dockcj` file; it is not used when reconfiguring the Summarization and Pruning Agent. If you do copy this file, the following error will occur and can be ignored.
   ```
   Validation failed: Column name exceeds 10 characters: ACKNOWLEDGED.
   ODI File contents not loaded: /install_dir/dockcj
   ```
4. On the machine where the Summarization and Pruning Agent is installed, open its environment file in a text editor:
   - Windows operating systems: `install_dir\TMAITM6\KSYENV`
   - Linux and UNIX operating systems: `install_dir/config/sy.ini`
5. Edit the following variables:

- To enable the Summarization and Pruning Agent to run without connecting to the Tivoli Enterprise Portal Server, set **KSY_AUTONOMOUS=Y**.
- If you did not install the application support files in the default directory (see step 3), set **KSY_AUTONOMOUS_ODI_DIR=***<alternative location of application support files>*.

6. Restart the Summarization and Pruning Agent agent. The WAREHOUSESUMPRUNE table is automatically created when the Summarization and Pruning Agent is started.
7. If you are upgrading from a previous version and already have summarization and pruning settings stored in the Tivoli Enterprise Portal Server database, restart the Tivoli Enterprise Portal Server.

   The first time the portal server is started after the WAREHOUSESUMPRUNE table has been created, any previously existing data collection and summarization and pruning configuration settings are migrated to the WAREHOUSESUMPRUNE table in the warehouse data base. Subsequently, any settings configured using the portal server are stored directly in the warehouse.

Configuring summarization and pruning without the Tivoli Enterprise Portal Server

You can configure historical data collection and summarization and pruning using the Tivoli Enterprise Portal , or you can configure them directly in the warehouse database WAREHOUSESUMPRUNE table using SQL commands.

Table 122 contains descriptions of the columns in the WAREHOUSESUMPRUNE table. Insert one row for each attribute group for which you want to collect historical data, along with the values for any summarization and pruning settings. You do not need to set defaults for unused options; they are built into the table design. Varchar values must be enclosed in single quotes (' ')

Table 122. Descriptions of the columns in the WAREHOUSESUMPRUNE control settings table

Name	Type	Description
TABNAME	VARCHAR (40) NOT NULL PRIMARY KEY	The short table name. In the application support file, this is the value of TABLE. Review the application support file associated with each agent for the TABLE names.
YEARSUM	VARCHAR (8) DEFAULT ‘-16823’	Yearly Summarization on (-16822); off (-16823)
QUARTSUM	VARCHAR (8) DEFAULT ‘-16823’	Quarterly Summarization on (-16822); off (-16823)
MONSUM	VARCHAR (8) DEFAULT ‘-16823’	Monthly Summarization on (-16822); off (-16823)
WEEKSUM	VARCHAR (8) DEFAULT ‘-16823’	Weekly Summarization on (-16822); off (-16823)
DAYSUM	VARCHAR (8) DEFAULT ‘-16823’	Hourly Summarization on (-16822); off (-16823)
HOURSUM	VARCHAR (8) DEFAULT ‘-16823’	Daily Summarization on (-16822); off (-16823)
PYEAR	VARCHAR (8) DEFAULT ‘-16838’	Yearly Pruning on (-16837); off (-16838)
PYEARINT	SMALLINT DEFAULT 1	Number of units (years, months, or days)
PYEARUNIT	VARCHAR (8) DEFAULT ‘-16834’	Type of Unit. Years (-16834), Months (-16835), Days (-16836)
PQUART	VARCHAR (8) DEFAULT ‘-16838’	Quarterly Pruning on (-16837); off (-16838)
PMON	VARCHAR (8) DEFAULT ‘-16838’	Monthly Pruning on (-16837); off (-16838)
PMONINT	SMALLINT DEFAULT 1	Number of units (years, months, or days)

Table 122. Descriptions of the columns in the WAREHOUSESUMPRUNE control settings table (continued)

Name	Type	Description
PMONUNIT	VARCHAR (8) DEFAULT '-16835'	Type of Unit. Years (-16834), Months (-16835), Days (-16836)
PWEEK	VARCHAR (8) DEFAULT '-16838'	Weekly Pruning on (-16837); off (-16838)
PWEEKINT	SMALLINT DEFAULT 1	Number of units (years, months or days)
PWEEKUNIT	VARCHAR (8) DEFAULT '-16835'	Type of Unit. Years (-16834), Months(-16835), Days(-16836)
PDAY	VARCHAR (8) DEFAULT '-16838'	Daily Pruning on (-16837); off (-16838)
PDAYINT	SMALLINT DEFAULT 1	Number of units (years, months, or days)
PDAYUNIT	VARCHAR (8) DEFAULT '-16835'	Type of Unit. Years (-16834), Months (-16835), Days (-16836)
PHOUR	VARCHAR (8) DEFAULT '-16838'	Hourly Pruning on (-16837); off (-16838)
PHOURINT	SMALLINT DEFAULT 1	Number of units (years, months, or days)
PHOURUNIT	VARCHAR (8) DEFAULT '-16836'	Type of Unit. Years (-16834), Months (-16835), Days (-16836)
PRAW	VARCHAR (8) DEFAULT '-16838'	Detailed Pruning on (-16837); off (-16838)
PRAWINT	SMALLINT DEFAULT 1	Number of units (years, months, or days)
PRAWUNIT	VARCHAR (8) DEFAULT '-16836'	Type of Unit. Years (-16834), Months (-16835), Days (-16836)

Examples: Following are examples of basic collection and summarization and pruning configuration.

Configuration and daily/hourly summarization

Collection is configured, and daily and hourly summarizations are set. No pruning has been specified. Use the SQL INSERT command.

Required:

- TABNAME= *Table Code*
- DAYSUM= -16822 (summarize daily)
- HOURSUM=-16822 (summarize hourly)

```
INSERT INTO WAREHOUSESUMPRUNE (TABNAME,DAYSUM,HOURSUM) VALUES ('WTMEMORY','-16822','-16822');
```

Configuration, daily/hourly summarization, and daily/hourly pruning

Collection is configured, and daily and hourly summarizations are set. Pruning is specified for daily 3-month intervals and hourly 2-day intervals. Use the SQL INSERT command.

Required:

- TABNAME=*Table Code*
- DAYSUM=-16822 (summarize daily)
- HOURSUM= -16822 (summarize hourly on)
- PDAY= -16837 (prune daily)
- PDAYINT=3 (prune interval)

- PDAYUNIT= -16835 (prune monthly)
- PHOUR = -16837 (prune hourly on)
- PHOURINT= 2 (prune interval)
- PHOURUNIT= -16836 (prune daily)
- PRAW = -16837 (prune detailed on)
- PRAWINT= 1 (prune interval)
- PRAWUNIT= -16836 (prune daily)

```
INSERT INTO WAREHOUSESUMPRUNE
(TABNAME,DAYSUM,HOURSUM,PDAY,PDAYINT,PDAYUNIT,PHOUR,PHOURINT,PHOURUNIT,PRAW,PRAWINT,PRAWUNIT)
VALUES ('WTMEMORY','-16822','-16822','-16837',3,'-16835',2,'-16836','-16837',1,'-16836');
```

Remove daily summarization and pruning

In this example, collection has been configured, and daily and hourly summarizations have been set. You want to remove daily summarization and pruning. Use the SQL UPDATE command.

Required:

- TABNAME= *Table Code*
- DAYSUM= -16823 (summarize off)
- PDAY= -16838 (prune daily off)
- PDAYINT= 1 (prune interval to default)
- PDAYUNIT= -16836 (prune daily)

```
UPDATE WAREHOUSESUMPRUNE SET DAYSUM='-16823', SET PDAY='-16838', SET PDAYINT=1,
SET PDAYUNIT='-16836' WHERE TABNAME='WTMEMORY';
```

Remove collection entirely

Collection has been configured, and daily and hourly summarizations have been set. Set the collection to unconfigured by removing the entire row from the table, using the SQL DELETE command.

Required: TABNAME= *Table Code*

```
DELETE FROM WAREHOUSESUMPRUNE WHERE TABNAME='WTMEMORY';
```

Testing the connection between the portal server and the Tivoli Data Warehouse

To test the connection between the Tivoli Enterprise Portal Server and the Tivoli Data Warehouse database, use the Tivoli Enterprise Portal to create a custom SQL query to the warehouse database and display the results. The procedure described in this section creates an SQL query to the WAREHOUSELOG table, a status table that is created in the warehouse database when the Warehouse Proxy Agent starts. See “WAREHOUSELOG and WAREHOUSEAGGREGLOG tables” on page 638 for a description of this table.

Before you begin, make sure that the following components are installed and started:

- The Tivoli Enterprise Portal Server
- The Warehouse Proxy Agent
- The Tivoli Data Warehouse RDBMS server

The test procedure consists of the following steps:

1. Create a custom SQL query to the WAREHOUSELOG table.
2. Create a new workspace to display the results.
3. Assign the query to the workspace.

1. Create the query

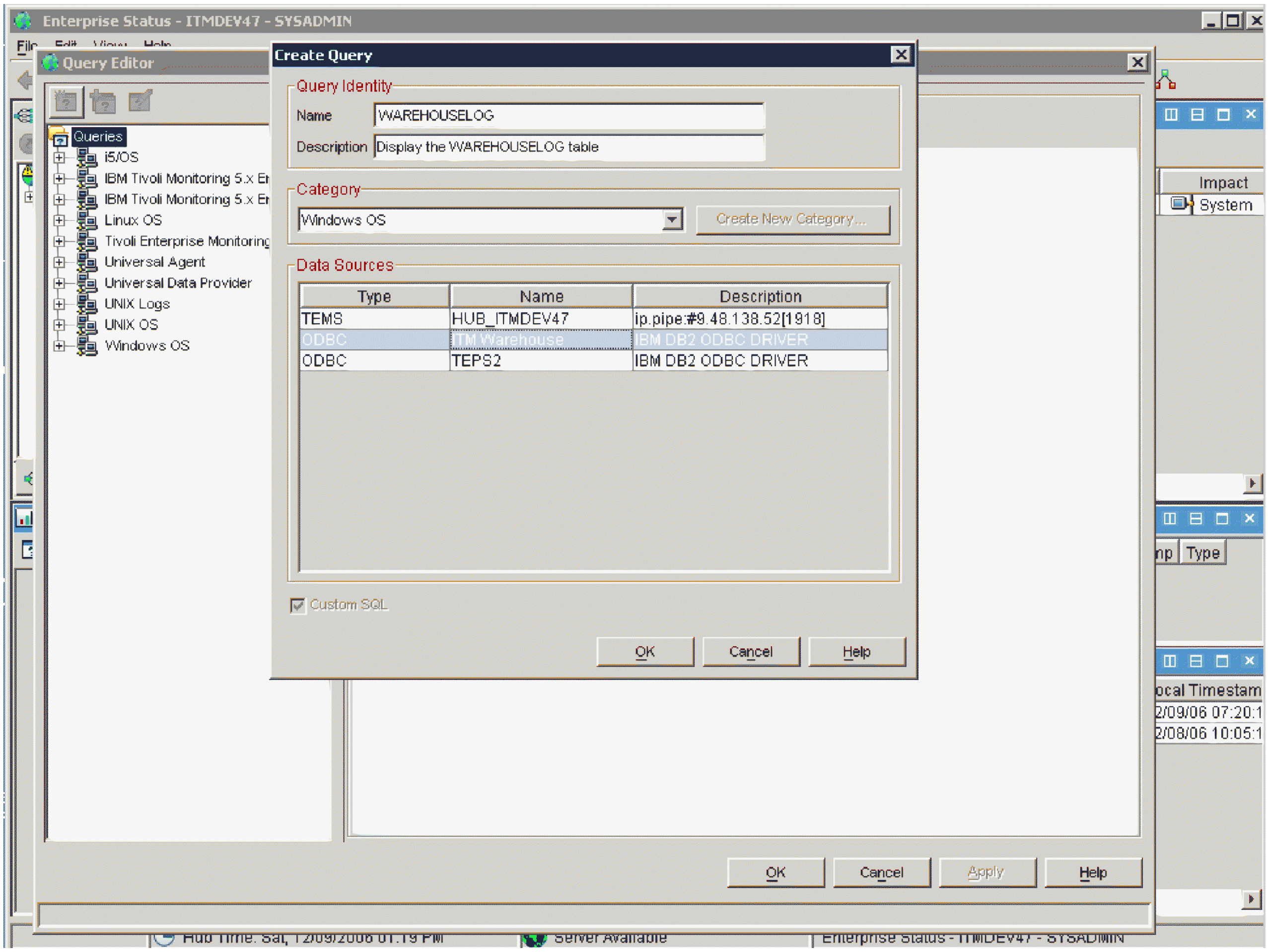
1. Start the Tivoli Enterprise Portal using the desktop or browser client. See “Starting the Tivoli Enterprise Portal client” on page 320 for instructions.

   Log in as the system administrator. (The default system administrator ID is **sysadmin**.)

2. Click **Query Editor** on the Tivoli Enterprise Portal tool bar.

3. In the Query Editor window, click **Create New Query**.

   The Create Query window is displayed.



Name: WAREHOUSELOG

Description: Display the WAREHOUSELOG table

Category: Windows OS

Type	Name	Description
TEMS	HUB_ITMDEV47	ip.pipe:#9.48.138.52[1918]
ODBC	ITM Warehouse	IBM DB2 ODBC DRIVER
ODBC	TEPS2	IBM DB2 ODBC DRIVER

☑ Custom SQL

Figure 155. Create Query window

4. Enter a name and description for the query. For this test, enter `WAREHOUSELOG` for the query name.

5. From the **Category** drop-down list, select the folder where you want the WAREHOUSELOG query to appear in the Query tree.

   For example, select a folder name that corresponds to the operating system (Windows OS or UNIX OS) where the Tivoli Data Warehouse is installed.

6. Select the name of the data source for the Tivoli Data Warehouse in the **Data Sources** list.

7. Click **OK**.

   The WAREHOUSELOG query appears in the Query tree in the Custom_SQL folder under the category that you selected. The **Specification** tab opens with a **Custom SQL** text box for you to enter an SQL command.

8. Enter the following SQL statement in the **Custom SQL** text box to select all columns of the WAREHOUSELOG table:

   ```
   select * from WAREHOUSELOG
   ```

9. Click **OK** to save the query and close the Query Editor window.

2. Create a workspace

1. If the Enterprise Status workspace is not open, click the Enterprise Navigator item.
2. Select **Save Workspace As** from the File menu.
3. Enter the name `WAREHOUSELOG` for the new workspace.
4. Click **OK**.

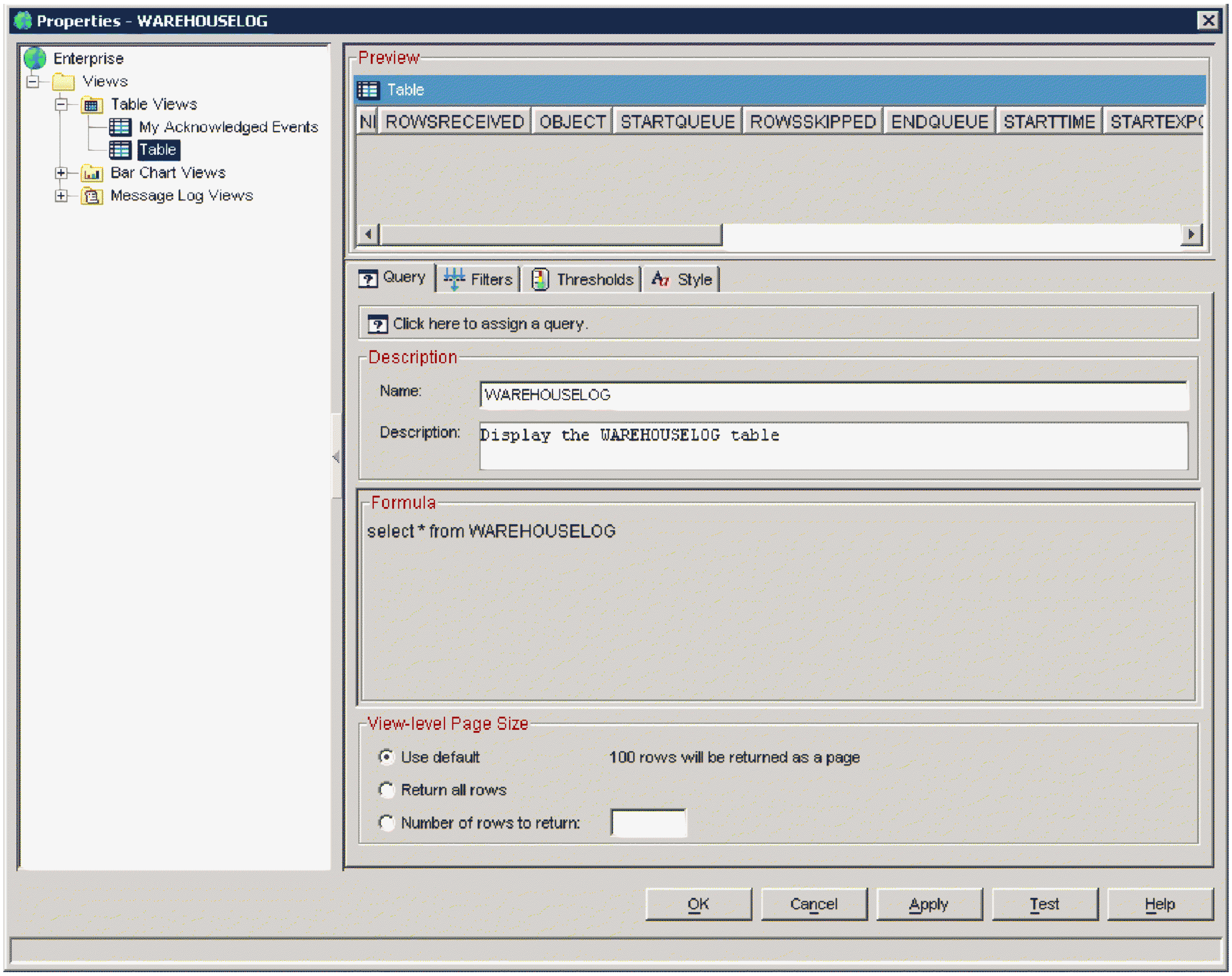
   You now have a duplicate of the Enterprise Status workspace named WAREHOUSELOG that you can modify.

3. Assign the query to the workspace

1. In the WAREHOUSELOG workspace, click **Table** on the tool bar, release the mouse button; then click inside one of the workspace views (for example, the Situation Event Console view).
2. Click **Yes** on the message asking if you want to assign the query now.
3. In the Query tab, select **Click here to assign query**.

   The Query editor opens with a list of all product queries.
4. In the list of queries, expand the category folder you specified when you created the query (for example Windows OS or UNIX OS), and then expand the Custom_SQL folder.
5. Select **WAREHOUSELOG** from the list of queries in the Custom_SQL folder.

   The query definition opens in the right frame.



Description

Name: WAREHOUSELOG

Description: Display the WAREHOUSELOG table

Formula

select * from WAREHOUSELOG

View-level Page Size

● Use default 100 rows will be returned as a page

○ Return all rows

○ Number of rows to return:

Figure 156. Assigning the WAREHOUSELOG query to a workspace

6. Click **OK** to select the WAREHOUSELOG query for the table and close the Query editor.
7. Click **OK** to close the **Properties** editor.

   The columns of the WAREHOUSELOG table are displayed within the table view in the WAREHOUSELOG workspace. The content of the table is also displayed if the table is not empty.

Tuning the performance of the Warehouse Proxy

You can set the following environment variables to control the way that the Warehouse Proxy works:

- "Database initialization"
- "Work queue" on page 618
- "Connection pool" on page 618
- "RPC threads and export requests" on page 619
- "Timeout values" on page 619

You can set these variables in the KHDENV file.

Database initialization

When the Warehouse Proxy starts, the following tests are done:

- Checks that the Warehouse Proxy can connect to the database.

- If the database is Oracle or DB2 for Linux, UNIX, and Windows, checks that the encoding is set to UTF8.
- If the database is DB2 for Linux, UNIX, and Windows, checks that a bufferpool of page size 8KB is created. If it is not, one is created, along with three new tablespaces that use the 8KB bufferpool. The bufferpool is called "ITMBUF8K" and the tablespaces are named "ITMREG8K," "ITMSYS8K," and "ITMBUF8K."
- Creates a database cache that contains a list of all the tables and columns that exist in the database.

If any of these tests fail, a message is written to the log file and messages are displayed in the Event Viewer.

These tests are repeated every 10 minutes.

You can change this default start up behavior by changing the following environment variables:

KHD_CNX_WAIT_ENABLE
Enables the Warehouse Proxy to wait in between attempts to connect to the database. By default, this variable is set to Y. If you do not want the Warehouse Proxy to wait, change the variable to N. However, this can generate a large log file if the connection to the database fails with each attempt.

KHD_CNX_WAIT
Defines the amount of time, in minutes, that the Warehouse Proxy waits between attempts to connect to the database. The default value is 10 minutes.

Work queue

The work queue consists of a single queue instance and a configurable number of worker threads that run work placed on it. There are two primary configuration parameters that you can set. You can set these parameters in the KHDENV file on Windows or the hd.ini file on Linux and UNIX before starting the Warehouse Proxy.

KHD_QUEUE_LENGTH
The length of the KHD work queue. This is an integer that identifies the maximum number of export work requests that can be placed on the work queue before the work queue begins rejecting requests. The default value is 1000. Setting this value to 0 means that the work queue has no limit.

KHD_EXPORT_THREADS
The number of worker threads exporting data to the database. The default value is 10.

Connection pool

The Warehouse Proxy uses several pre-initialized ODBC connections to access the target database. The use of these ODBC connection objects is synchronized through a single connection pool. The connection pool is initialized when the Warehouse Proxy starts.

You can configure the number of connections in the pool by defining the following environment variable in the KHDENV file on Windows or the hd.ini file on Linux and UNIX:

- KHD_CNX_POOL_SIZE: The total number of pre-initialized ODBC connection objects available to the work queue export threads. The default value is 10.

All export worker threads request connections from the connection pool and must obtain a connection before the work of exporting data can continue.

You only see the connections established when a request is active. It is important to set the number of worker threads to greater or equal to the number of ODBC connections. To do this, set `KHD_KHD_EXPORT_THREADS >= KHD_CNX_POOL_SIZE`.

RPC threads and export requests

In addition to the configurable environment variables discussed previously, the standard agent framework provides some control over the scalability and performance profile for the Warehouse Proxy. When the Warehouse Proxy starts, it initializes RPC and registers a group of function pointers that respond to incoming RPC calls.

Use the **CTIRA_NCSLISTEN** variable to set the number of RPC threads.

Timeout values

You can set two environment variables to control the timeout value. One variable is set on the application agent, the other on the Warehouse Proxy.

KHD_STATUSTIMEOUT
The time, in seconds, to wait for a status from the Warehouse Proxy before sending an export request again. The default value is 900 seconds, or 15 minutes.

KHD_SRV_STATUSTIMEOUT
The timeout value, in seconds, for the work queue to perform work. The default value is 600 seconds, or 10 minutes.

Export requests are rejected by the Warehouse Proxy are the following four reasons:

- The time between when an export request is sent to the work queue and when it is extracted from the queue exceeds the timeout. If you have tracing for the Warehouse Proxy set to ERROR, an error similar to the following is logged in the Warehouse Proxy log file:

  ```
  REJECTED: The export for the originnode OriginNodeName, the application
  applicationName and the table tableName has been rejected for timeout
  reason in stage END_QUEUE.
  ```

- The time between when an export request is sent to the work queue and when the work queue starts to do existence checking in the database exceeds the timeout. If you have tracing for the Warehouse Proxy set to ERROR, an error similar to the following is logged in the Warehouse Proxy log file and the WAREHOUSELOG table:

  ```
  Sample data rejected for timeout reason at stage START EXPORT
  ```

- The time between when an export request is sent to the work queue and when the work queue fetches all the rows in the sample exceeds the timeout. If you have tracing for the Warehouse Proxy set to ERROR, a message similar to the following is logged in the Warehouse Proxy log file and the WAREHOUSELOG table:

  ```
  Sample data rejected for timeout reason at stage START SAMPLE
  ```

- The time between when an export request is sent to the work queue and when the work queue commits the rows in the database exceeds the timeout. If you have tracing for the Warehouse Proxy set to ERROR, a message similar to the following is logged in the Warehouse Proxy log file and the WAREHOUSELOG table:

  ```
  Sample data rejected for timeout reason at stage COMMIT
  ```

The KHD_SRV_STATUSTIMEOUT variable should be set less than KHD_STATUSTIMEOUT by at least 60 seconds. Do not change these values unless there is a problem in the environment.

Configuring the Warehouse Proxy Agent and Summarization and Pruning Agent using the new graphical interface

The database administrator must create an empty database that will serve as the Data Warehouse database and provide the database name, username and password to the IBM Tivoli Monitoring administrator. The IBM Tivoli Monitoring administrator creates the ODBC or JDBC Data Source using the credentials provided. Then, the IBM Tivoli Monitoring administrator must configure the Warehouse Proxy Agent and the Summarization and Pruning Agent to use the ODBC or JDBC Data Source. The Warehouse

Proxy Agent uses ODBC on Windows systems. On Unix and Linux, the Warehouse Proxy Agent uses JDBC. The Summarization and Pruning Agent always uses JDBC.

You can confirm a successful configuration when the Warehouse Proxy Agent and Summarization and Pruning Agent can connect to the Data Warehouse database. To verify this, login to the Tivoli Enterprise Portal Desktop and check the state of the Data Warehouse database connectivity for both the agents. These should be displayed as green.

You need to complete the following steps for this configuration option:

1. During installation, when the Configuration Defaults for Connecting to a Tivoli Enterprise Monitoring Server window is displayed, you will see the user interface for the Warehouse Proxy Agent. Select the database type and click **Next**.
2. Enter the database connection information and adjust any other configuration parameters, then click **OK**.

Installing and configuring the Warehouse Proxy Agent and Summarization and Pruning Agent in silent mode, using the silent response file

A silent installation runs on its own and does not require you to provide input to dialog boxes or monitor the installation. This configuration option allows for an unattended and automatic installation and configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent as well as the remote configuration of these two agents. Configuration of the Warehouse Proxy Agent and Summarization and Pruning Agent takes place during the setup of the IBM Tivoli Monitoring infrastructure, or manually later.

The database administrator must create the following before the silent install is run:

- An empty database that will serve as the Data Warehouse database. The default DB2 database is WAREHOUS.
- The username and password that will be used with the database default ITMUser.
- The ODBC connection that will be used to connect the database to the Warehouse Proxy and Summarization and Pruning Agents default ITM Warehouse. The ODBC datasource connection must be created in 32-bit and 64-bit mode if the 64-bit Warehouse Proxy Agent is used to connect to the Data Warehouse database.

After these items are created, the database administrator must provide the database name, username and password, and ODBC connection information to the IBM Tivoli Monitoring administrator. The IBM Tivoli Monitoring administrator takes the information provided by the database administrator and updates all the configuration information in the silent response file that drives the installation and configuration process.

You can confirm a successful configuration when the Warehouse Proxy Agent, Summarization and Pruning Agent, and Performance Analyzer Warehouse Agent show a connection to the Data Warehouse database. To verify this, login to the Tivoli Enterprise Portal Desktop and check the state of the Data Warehouse database connectivity for all three agents. The Summarization and Pruning Agent should also show connectivity to the Tivoli Enterprise Portal. The connections should be displayed as green.

You must complete the following steps for this configuration option:

1. Fill in the silent response file (`silent_server.txt`), providing the configuration information for the Warehouse Proxy Agent and the Summarization and Pruning Agent.

   You can also have response files created automatically on Windows. From the Manage Tivoli Enterprise Monitoring Services screen, right-click the agent whose configuration information you want saved, and select the **Advanced** → **Utilities** → **Generate Response Files** option from the pop-up menu.
2. Start the installation in silent mode, providing the silent response file as a parameter.

a. Start a DOS Command Shell.

b. From the shell, cd to the directory containing this installation (where setup.exe and setup.isn reside).

c. Invoke setup as follows:

```
start /wait setup /z"/sfC:\temp\SILENT_SERVER.txt" /s /f2"C:\temp\silent_setup.log"
```

You must specify the parameters in the same order listed above.

Where:

/z"/sf
: Specifies the name of the installation driver you have customized for your site. This is a required parameter. This file must exist.

/s Specifies that this is a silent install. This causes no responses to be displayed during installation on the installed target workstation.

/f2
: Specifies the name of the InstallShield log file. If you do not specify this parameter, the default is to create Setup.log in the same location as setup.iss. This log is the InstallShield log not the install log. The install log can be found in the install target directory, default c:\IBM\ITM sub-directory InstallITM, or on the Windows Boot drive root directory if the install aborts before the install location has been identified. In either case, the Setup program must be able to create and write to this file.

Installing and configuring the Warehouse Proxy Agent in silent mode on UNIX, using the silent response file

A silent installation runs on its own and does not require you to provide input to dialog boxes or monitor the installation. This reduces the time spent on installation and configuration. To install and configure the Warehouse Proxy Agent on UNIX with a silent response file, you must complete the following steps:

1. Create the silent install response file, see “Warehouse Proxy Agent silent install response file sample” on page 624. Remember that user and password information are not encrypted in the response file.
2. Run the following command from the installation image:

   ```
   ./install.sh -q -h <install_dir> -p <response_file>
   ```

   Note: You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.
3. Edit the silent config response file from the samples directory and change it according to your configuration, see “Warehouse Proxy Agent silent config response file sample” on page 624.
4. Run the silent config command to configure the Warehouse Proxy Agent:

   ```
   itmcmd config –A –h <install_dir> -p <response file> hd
   ```

Note: After running the command in step 2, the Warehouse Proxy Agent should be installed but not yet configured. The install `itm_install_output` log file should display content similar to the following:

```
You selected the following products:
        Warehouse Proxy

 ... installing "Warehouse Proxy for AIX R5.3 (64 bit)  AIX R6.1 (64 bit)";
please wait.

 => installed "Warehouse Proxy for AIX R5.3 (64 bit)  AIX R6.1 (64 bit)".
... Initializing component Warehouse Proxy for AIX R5.3 (64 bit)  AIX R6.1 (64 bit).
... Warehouse Proxy for AIX R5.3 (64 bit)  AIX R6.1 (64 bit) initialized.
```

```
... postprocessing; please wait.
... finished postprocessing.
Installation step complete.
```

Note: After running the command in step 4, the config trace file should show the following:

```
<date> ITMinstall.CandleConfigAgent main [LOG_INFO]
      Agent configuration completed...
```

The config file `hd.ini` should contain the exact information contained in the response file.

After the Warehouse Proxy Agent has been installed and configured, it can be started using the command-line `itmcmd agent start hd`. You should check that the Agent started without any errors.

Installing and configuring the Summarization and Pruning Agent in silent mode on UNIX, using the silent response file

A silent installation runs on its own and does not require you to provide input to dialog boxes or monitor the installation. This reduces the time spent on installation and configuration. To install and configure the Summarization and Pruning Agent on UNIX with a silent response file, you must complete the following steps:

1. Create the silent install response file, see “Summarization and Pruning Agent silent install response file sample” on page 627. Remember that user and password information are not encrypted in the response file.
2. Run the following command from the installation image:

   ```
   ./install.sh -q -h <install_dir> -p <response_file>
   ```

   Note: You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.
3. Edit the silent config response file from the samples directory and change it according to your configuration, see “Summarization and Pruning Agent silent config response file sample” on page 627.
4. Run the silent config command to configure the Summarization and Pruning Agent:

   ```
   itmcmd config –A –h <install_dir> -p <response file> sy
   ```

Note: After running the command in step 2, the Summarization and Pruning Agent should be installed but not yet configured. The install `itm_install_output` log file should display content similar to the following:

```
You selected the following products:
        Summarization and Pruning Agent

 ..installing "Summarization and Pruning Agent for AIX R5.3(64 bit) AIX R6.1
(64 bit)"; please wait.

 => installed "Summarization and Pruning Agent for AIX R5.3 (64 bit)
AIX R6.1 (64 bit)".
..Initializing component Summarization and Pruning Agent for AIX R5.3 (64 bit)
AIX R6.1 (64 bit).
..Summarization and Pruning Agent for AIX R5.3 (64 bit)  AIX R6.1 (64 bit)
initialized.


..postprocessing; please wait.
..finished postprocessing.
Installation step complete.
```

Note: After running the command in step 4, the config trace file should show the following:

```
<date> ITMinstall.CandleConfigAgent main [LOG_INFO]
      Agent configuration completed...
```

The config file hd.ini should contain the exact information contained in the response file.

After the Summarization and Pruning Agent has been installed and configured, it can be started using the command-line itmcmd agent start sy. You should check that the Agent started without any errors.

Upgrading the Warehouse Proxy Agent and Summarization and Pruning Agent in silent mode, using the silent response file

A silent installation runs on its own and does not require you to provide input to dialog boxes or monitor the installation. This reduces the time spent on upgrading. To upgrade the Warehouse Proxy Agent or the Summarization and Pruning Agent in silent mode, you must complete the following steps:

1. The Warehouse Proxy Agent or Summarization and Pruning Agent should already be installed at a lower level before upgrading.
2. All agents should be stopped before upgrading.
3. Create the silent install response file, see “Warehouse Proxy Agent silent install response file sample” on page 624 or“Summarization and Pruning Agent silent install response file sample” on page 627.
4. Run the following command from the installation image 6.2.3:

   ```
   ./install.sh -q -h <install_dir> -p <response_file>
   ```

   Note: You must not specify the path of the directory containing ./install.sh as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.

The Warehouse Proxy Agent or Summarization and Pruning Agent should now be upgraded and should not change the previous configuration. Manage Tivoli Monitoring Services should show the new version of the Warehouse Proxy Agent or Summarization and Pruning Agent.

The config files hd.ini as well as sy.ini should still contain the same information that it had before the upgrade, in addition to the new parameters introduced in release 6.2.3. The Warehouse Proxy Agent or Summarization and Pruning Agent can be restarted without reconfiguration, provided they were configured correctly before upgrade.

Note: The Warehouse Proxy Agent and Summarization and Pruning Agent are not started automatically after the upgrade has completed. Depending on the installed agents and the associated attribute groups enabled for historical collection, and whether limited database permissions are granted to the warehouse user, a database administrator might need to use the schema publication tool to generate a script with the required changes for the database. For more information, see Chapter 19, “Schema Publication Tool,” on page 483.

Upgrading the Warehouse Proxy Agent and Summarization and Pruning Agent using remote deploy

To upgrade the Warehouse Proxy Agent or the Summarization and Pruning Agent, by remotely deploying bundles containing the installation image and other prerequisites, you must complete the following steps:

1. For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely upgrade a Warehouse Proxy Agent or Summarization and Pruning Agent on Linux, you must install the monitoring agent for Linux OS on the same server. To remotely upgrade a Warehouse Proxy Agent or Summarization and Pruning Agent on Windows, you must install the monitoring agent for Windows on the same server.
2. Populate the DEPOT with the latest version of the Warehouse Proxy Agent or the Summarization and Pruning Agent. The DEPOT must be located where a Tivoli Enterprise Monitoring Server is installed.

The CD image contains a directory called deploy where the Warehouse Proxy Agent bundle and the Summarization and Pruning Agent bundle are located. From this directory you can run the following commands:

Warehouse Proxy Agent command-line example:

```
tacmd addbundles -i . -t hd
```

Summarization and Pruning Agent command-line example:

```
tacmd addbundles -i . -t sy
```

Note: You can also use the install command `install.sh` to populate the depot with the latest version of the Warehouse Proxy Agent or the Summarization and Pruning Agent.

3. Connect to the Tivoli Enterprise Monitoring Server using the `tacmd` command:

```
tacmd login –s <TEMS hostname> -u <userid> -p <pw>
```

4. Run the update command to update to the latest version.

Warehouse Proxy Agent:

```
./tacmd updateAgent -n <system name> -t HD
```

Summarization and Pruning Agent:

```
./tacmd updateAgent -n <system name> -t SY
```

The Warehouse Proxy Agent or the Summarization and Pruning Agent should now be updated but not restarted. A new version should appear in the topology workspace, the manage system status, manage monitoring services, and in the trace file.

You should now restart the Warehouse Proxy Agent or the Summarization and Pruning Agent, and check that the configuration has not been modified. The agents should start successfully.

Warehouse Proxy Agent silent install response file sample

```
INSTALL_ENCRYPTION_KEY=IBMTivoliMonitoringEncryptionKey
INSTALL_PRODUCT=hd
```

Warehouse Proxy Agent silent config response file sample

```
################# PRIMARY TEMS CONFIGURATION #################
# Will this agent connect to a Tivoli Enterprise Monitoring Server (TEMS)?
# This parameter is required.
# Valid values are: YES and NO
CMSCONNECT=YES

# What is the hostname of the TEMS to connect to?
# This parameter is NOT required.  (default is the local system hostname)
# The TEMS must be the HUB TEMS.
HOSTNAME=localhost

# Will this agent connect to the TEMS through a firewall?
# This parameter is NOT required.  (default is NO)
# Valid values are: YES and NO
#   - If set to YES the NETWORKPROTOCOL must be ip.pipe
#FIREWALL=NO

# What network protocol is used when connecting to the TEMS?
# This parameter is required.
# Valid values are: ip, sna, ip.pipe, or ip.spipe
NETWORKPROTOCOL=ip.pipe

# What is the first backup network protocol used for connecting to the TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, ip.spipe, or none
#BK1NETWORKPROTOCOL=none
```

```
# What is the second backup network protocol used for connecting to the TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, ip.spipe or none
#BK2NETWORKPROTOCOL=none

# If ip.pipe is one of the three protocols what is the IP pipe port number?
# This parameter is NOT required. (default is 1918)
IPPIPEPORTNUMBER=1918

# If ip.pipe is one of the three protocol what is the IP pipe partition name?
# This parameter is NOT required. (default is null)
#KDC_PARTITIONNAME=null

# If ip.pipe is one of the three protocols what is the KDC partition file?
# This parameter is NOT required. (default is null)
#KDC_PARTITIONFILE=null

# If ip.spipe is one of the three protocols what is the IP pipe port number?
# This parameter is NOT required. (default is 3660)
#IPSPIPEPORTNUMBER=3660

# If ip is one of the three protocols what is the IP port number?
# This parameter is NOT required. (default is 1918)
# A port number and or one or more pools of port numbers can be given.
# The format for a pool is #-# with no embedded blanks.
#PORTNUMBER=1918

# If sna is one of the three protocols what is the SNA net name?
# This parameter is NOT required. (default is CANDLE)
#NETNAME=CANDLE

# If sna is one of the three protocols what is the SNA LU name?
# This parameter is NOT required. (default is LUNAME)
#LUNAME=LUNAME

# If sna is one of the three protocols what is the SNA log mode?
# This parameter is NOT required. (default is LOGMODE)
#LOGMODE=LOGMODE

################# SECONDARY TEMS CONFIGURATION #################

# Would you like to configure a connection for a secondary TEMS?
# This parameter is NOT required. (default is NO)
# Valid values are: YES and NO
#FTO=NO

# If configuring a connection for a secondary TEMS, what is the hostname of the secondary TEMS?
# This parameter is required if FTO=YES
#MIRROR=somehost.somewhere.com

# Will the agent connect to the secondary TEMS through a firewall?
# This parameter is NOT required. (default is NO)
# Valid values are: YES and NO
#FIREWALL2=NO

# What network protocol is used when connecting to the secondary TEMS?
# This parameter is required when FTO=YES and FIREWALL2 is NO
# Valid values are: ip, sna, or ip.pipe
#HSNETWORKPROTOCOL=ip.pipe

# What is the first backup network protocol used for connecting to the secondary TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, or none
#BK1HSNETWORKPROTOCOL=none

# What is the second backup network protocol used for connecting to the secondary TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, or none
#BK2HSNETWORKPROTOCOL=none

# If ip.pipe is one of the three secondary TEMS protocols what is the IP pipe port number?
# This parameter is NOT required. (default is 1918)
#HSIPPIPEPORTNUMBER=1918

# If ip is one of the three secondary TEMS protocols what is the IP port number?
# This parameter is NOT required. (default is 1918)
# A port number and or one or more pools of port numbers can be given.
# The format for a pool is #-# with no embedded blanks.
#HSPORTNUMBER=1918

# If sna is one of the three secondary TEMS protocols what is the SNA net name?
# This parameter is NOT required. (default is CANDLE)
#HSNETNAME=CANDLE

# If sna is one of the three secondary TEMS protocols what is the SNA LU name?
# This parameter is NOT required. (default is LUNAME)
#HSLUNAME=LUNAME

# If sna is one of the three secondary TEMS protocols what is the SNA log mode?
# This parameter is NOT required. (default is LOGMODE)
#HSLOGMODE=LOGMODE

################# OPTIONAL PRIMARY NETWORK NAME CONFIGURATION #################

# If the system is equipped with dual network host adapter cards you can designate
# another network name.  What is the network name?
# This parameter is NOT required. (default is none)
#PRIMARYIP=none


#
#-----------------------------------------------------------------*
# WPA Specific Configuration Values
#-----------------------------------------------------------------*
# Allowed values are DB2, ORACLE, MSSQL
KHD_DBMS=DB2

#Comma separated list of fully qualified paths to JDBC JAR files
KHD_WAREHOUSE_JARS=/data/ana-db2/db2jcc.jar,/data/ana-db2/db2jcc_license_cu.jar

#-----------------------------------------------------------------*
# Specify the URL that corresponds to the warehouse type you selected
#
# Oracle URL
#KHD_ORACLE_JDBCURL=jdbc:oracle:thin:@<server>:<port>:<database>
#
# Microsoft SQL Seerver URL
#KHD_MSSQL_JDBCURL=jdbc:sqlserver://<server>:<port>;databasename=<database>;SelectMethod=cursor
#
# DB2 URL
KHD_DB2_JDBCURL=jdbc:db2://localhost:60000/ITMDW
#-----------------------------------------------------------------*


#-----------------------------------------------------------------*
# Specify the JDBC driver class that corresponds to the warehouse type you selected
#
# DB2 driver class
KHD_DB2_JDBCDRIVER=com.ibm.db2.jcc.DB2Driver
#
# Oracle driver class
#KHD_ORACLE_JDBCDRIVER=oracle.jdbc.driver.OracleDriver
#
```

```
# Microsoft SQL Server driver class
#KHD_MSSQL_JDBCDRIVER=com.microsoft.sqlserver.jdbc.SQLServerDriver
#-----------------------------------------------------------------*


KHD_WAREHOUSE_USER=itmuser
KHD_WAREHOUSE_PASSWORD=itmtdw08

# Should inserts be batched
# allowed values are true/false
KHD_BATCH_USE=true

# Enable Database Compression
# allowed values are true/false
KHD_DB_COMPRESSION=false

# List of TEMS served by this Warehouse Proxy
KHD_WAREHOUSE_TEMS_LIST=HUB_ITMTDWP12

# Enable Warehouse Compression for Z sources
# allowed values are true/false
KHD_SERVER_Z_COMPRESSION_ENABLE=false

# Enable Warehouse Compression for Distributed sources
# allowed values are true/false
KHD_SERVER_DIST_COMPRESSION_ENABLE=true
```

Summarization and Pruning Agent silent install response file sample

```
INSTALL_ENCRYPTION_KEY=IBMTivoliMonitoringEncryptionKey

INSTALL_PRODUCT=sy
```

Summarization and Pruning Agent silent config response file sample

```
# This is a sample silent configuration response file for the summarization
# and pruning agent

# To configure an agent using this silent response file:
#   1) copy this file to another location and modify the necessary parameters
#   2) run itmcmd config -A -p <silent_response_file> sy
#      - give an absolute path for the silent_response_file

# Syntax rules:
# - Comment lines begin with "#"
# - Blank lines are ignored
# - Parameter lines are PARAMETER=value (do not put space before the PARAMETER)
# - Space before or after an equal sign is OK
# - Parameter values should NOT contain the following characters $, =, or |

################# PRIMARY TEMS CONFIGURATION ##################

# Will this agent connect to a Tivoli Enterprise Monitoring Server (TEMS)?
# This parameter is required.
# Valid values are: YES and NO
CMSCONNECT=YES

# What is the hostname of the TEMS to connect to?
# This parameter is NOT required.  (default is the local system hostname)
HOSTNAME=localhost

# Will this agent connect to the TEMS through a firewall?
# This parameter is NOT required.  (default is NO)
# Valid values are: YES and NO
#   - If set to YES the NETWORKPROTOCOL must be ip.pipe
#FIREWALL=NO
```

```
# What network protocol is used when connecting to the TEMS?
# This parameter is required.
# Valid values are: ip, sna, ip.pipe, or ip.spipe
NETWORKPROTOCOL=ip.pipe

# What is the first backup network protocol used for connecting to the TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, ip.spipe, or none
#BK1NETWORKPROTOCOL=none

# What is the second backup network protocol used for connecting to the TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, ip.spipe or none
#BK2NETWORKPROTOCOL=none

# If ip.pipe is one of the three protocols what is the IP pipe port number?
# This parameter is NOT required. (default is 1918)
IPPIPEPORTNUMBER=1918

# If ip.pipe is one of the three protocol what is the IP pipe partition name?
# This parameter is NOT required. (default is null)
#KDC_PARTITIONNAME=null

# If ip.pipe is one of the three protocols what is the KDC partition file?
# This parameter is NOT required. (default is null)
#KDC_PARTITIONFILE=null

# If ip.spipe is one of the three protocols what is the IP pipe port number?
# This parameter is NOT required. (default is 3660)
#IPSPIPEPORTNUMBER=3660

# If ip is one of the three protocols what is the IP port number?
# This parameter is NOT required. (default is 1918)
# A port number and or one or more pools of port numbers can be given.
# The format for a pool is #-# with no embedded blanks.
#PORTNUMBER=1918

# If sna is one of the three protocols what is the SNA net name?
# This parameter is NOT required. (default is CANDLE)
#NETNAME=CANDLE

# If sna is one of the three protocols what is the SNA LU name?
# This parameter is NOT required. (default is LUNAME)
#LUNAME=LUNAME

# If sna is one of the three protocols what is the SNA log mode?
# This parameter is NOT required. (default is LOGMODE)
#LOGMODE=LOGMODE

################## SECONDARY TEMS CONFIGURATION ##################

# Would you like to configure a connection for a secondary TEMS?
# This parameter is NOT required. (default is NO)
# Valid values are: YES and NO
#FTO=NO

# If configuring a connection for a secondary TEMS, what is the hostname of the secondary TEMS?
# This parameter is required if FTO=YES
#MIRROR=somehost.somewhere.com

# Will the agent connect to the secondary TEMS through a firewall?
# This parameter is NOT required. (default is NO)
# Valid values are: YES and NO
#FIREWALL2=NO

# What network protocol is used when connecting to the secondary TEMS?
# This parameter is required when FTO=YES and FIREWALL2 is NO
# Valid values are: ip, sna, or ip.pipe
#HSNETWORKPROTOCOL=ip.pipe

# What is the first backup network protocol used for connecting to the secondary TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, or none
#BK1HSNETWORKPROTOCOL=none

# What is the second backup network protocol used for connecting to the secondary TEMS?
# This parameter is NOT required. (default is none)
# Valid values are: ip, sna, ip.pipe, or none
#BK2HSNETWORKPROTOCOL=none

# If ip.pipe is one of the three secondary TEMS protocols what is the IP pipe port number?
# This parameter is NOT required. (default is 1918)
#HSIPPIPEPORTNUMBER=1918

# If ip is one of the three secondary TEMS protocols what is the IP port number?
# This parameter is NOT required. (default is 1918)
# A port number and or one or more pools of port numbers can be given.
# The format for a pool is #-# with no embedded blanks.
#HSPORTNUMBER=1918

# If sna is one of the three secondary TEMS protocols what is the SNA net name?
# This parameter is NOT required. (default is CANDLE)
#HSNETNAME=CANDLE

# If sna is one of the three secondary TEMS protocols what is the SNA LU name?
# This parameter is NOT required. (default is LUNAME)
#HSLUNAME=LUNAME

# If sna is one of the three secondary TEMS protocols what is the SNA log mode?
# This parameter is NOT required. (default is LOGMODE)
#HSLOGMODE=LOGMODE

################## OPTIONAL PRIMARY NETWORK NAME CONFIGURATION ##################

# If the system is equipped with dual network host adapter cards you can designate
# another network name.  What is the network name?
# This parameter is NOT required. (default is none)
#PRIMARYIP=none


#
#-----------------------------------------------------------------*
# SNP Specific Configuration Values
#-----------------------------------------------------------------*
# Allowed values are DB2, ORACLE, MSSQL
KSY_WAREHOUSE_TYPE=DB2

#Comma separated list of fully qualified paths to JDBC JAR files
KSY_WAREHOUSE_JARS=

#-----------------------------------------------------------------*
# Specify the URL that corresponds to the warehouse type you selected
#
# Oracle URL
#KSY_ORACLE_JDBCURL=jdbc:oracle:thin:@<server>:<port>:<database>
#
# Microsoft SQL Seerver URL
#KSY_MSSQL_JDBCURL=jdbc:sqlserver://<server>:<port>;databasename=<database>;SelectMethod=cursor
#
# DB2 URL
KSY_DB2_JDBCURL=jdbc:db2://<server>:<port>/<database>
#-----------------------------------------------------------------*
```

```
#-----------------------------------------------------------------*
# Specify the JDBC driver class that corresponds to the warehouse type you selected
#
# DB2 driver class
KSY_DB2_JDBCDRIVER=com.ibm.db2.jcc.DB2Driver
#
# Oracle driver class
#KSY_ORACLE_JDBCDRIVER=oracle.jdbc.driver.OracleDriver
#
# Microsoft SQL Server driver class
#KSY_MSSQL_JDBCDRIVER=com.microsoft.sqlserver.jdbc.SQLServerDriver
#-----------------------------------------------------------------*

KSY_WAREHOUSE_USER=
KSY_WAREHOUSE_PASSWORD=

#-----------------------------------------------------------------*
# Warehouse Database Compression                                  *
# If set to Y, database compression will be enabled.              *
#-----------------------------------------------------------------*
KSY_DB_COMPRESSION=N
#
#-----------------------------------------------------------------*
# Timezone indicator                                              *
# AGENT means use the time zone offset of the agent               *
# WAREHOUSE means use the time zone offset of the warehouse       *
#-----------------------------------------------------------------*
KSY_TIMEZONE_IND=AGENT
#
#-----------------------------------------------------------------*
# Start of the week day                                           *
# 0 = Sunday 1 = Monday                                           *
#-----------------------------------------------------------------*
KSY_START_OF_WEEK_DAY=0

#
#-----------------------------------------------------------------*
# Shift periods                                                   *
# Only two shifts are allowed.                                    *
# If shifts are enabled,                                          *
# each hour (0-23) must be specified once                         *
#-----------------------------------------------------------------*
KSY_SHIFTS_ENABLED=N
KSY_SHIFT1_HOURS=0,1,2,3,4,5,6,7,8,18,19,20,21,22,23
KSY_SHIFT2_HOURS=9,10,11,12,13,14,15,16,17
#
#-----------------------------------------------------------------*
# Vacation periods                                                *
# Vacation days are comma delimited and use the YYYYMMDD format   *
#-----------------------------------------------------------------*
KSY_VACATIONS_ENABLED=N
KSY_WEEKENDS_AS_VACATIONS=N
KSY_VACATION_DAYS=
#
#-----------------------------------------------------------------*
# Maximum rows per database transaction                           *
#-----------------------------------------------------------------*
KSY_MAX_ROWS_PER_TRANSACTION=1000
#
#-----------------------------------------------------------------*
# Schedule                                                        *
# Use KSY_FIXED_SCHEDULE to indicate you would like the S&P agent *
# to run at a set time every day or to run periodically           *
# throughout the day.                                             *
# KSY_FIXED_SCHEDULE=Y means the S&P agent will use the           *
#  KSY_EVERY_N_DAYS, KSY_HOUR_TO_RUN, KSY_HOUR_AM_PM and          *
#  KSY_MIUNUTE_TO_RUN variables to determine at which fixed time  *
#  to run.                                                        *
# KSY_FIXED_SCHEDULE=N means the S&P agent will use the           *
#  KSY_EVERY_N_MINS and KSY_BLACKOUT variables to determine       *
#  how often and when to run.                                     *
#                                                                 *
# KSY_BLACKOUT can be used with KSY_FIXED_SCHEDULE=N to list      *
# ranges of time where the S&P agent should not run. The list is  *
# in the format HH:MM-HH:MM with multiple values separated by a   *
# comma. The start time must be lower than the end time and the   *
# hour values must be between 0 and 23. To have S&P agent run     *
# every hour except between 12 AM and 2 AM, use the following:    *
# KSY_BLACKOUT=00:00-01:59                                        *
#-----------------------------------------------------------------*
KSY_FIXED_SCHEDULE=Y
KSY_EVERY_N_DAYS=1
KSY_HOUR_TO_RUN=2
KSY_HOUR_AM_PM=AM
KSY_MINUTE_TO_RUN=0
KSY_EVERY_N_MINS=60
KSY_BLACKOUT=

#-----------------------------------------------------------------*
# KSY_SUMMARIZATION_UNITS and KSY_NODE_ERROR_UNITS determine the  *
# number of rows that will be shown in the Summarization and      *
# Pruning overall run table and Errors table respectively. There  *
# is a minimum value of 10 for each                               *
#-----------------------------------------------------------------*
KSY_SUMMARIZATION_UNITS=10
KSY_NODE_ERROR_UNITS=10
#
#-----------------------------------------------------------------*
# KSY_BATCH_MODE controls the batching method used by the         *
# Summarization and Pruning agent.                                *
# A value of 0 means that data should only be batched for the same*
# system while a value of 1 means that data from multiple systems *
# can be batched together which can lead to higher performance at *
# potentially bigger transaction sizes. Default is 0 (single).    *
#-----------------------------------------------------------------*
KSY_BATCH_MODE=0
#
#-----------------------------------------------------------------*
# WAREHOUSELOG and WAREHOUSEAGGREGLOG pruning settings            *
# The entries are in the format <number>.<unit> where <number>    *
# is the number of units to keep and <unit> is one of:            *
# day, month or year. For example, keep data for 14 days, set the *
# value to 14.day                                                 *
#-----------------------------------------------------------------*
KSY_WAREHOUSELOG_PRUNE=
KSY_WAREHOUSEAGGREGLOG_PRUNE=
#
#-----------------------------------------------------------------*
# CNP Server connection                                           *
# default host = localhost                                        *
# default port = 1920                                             *
#-----------------------------------------------------------------*
KSY_CNP_SERVER_HOST=localhost
KSY_CNP_SERVER_PORT=1920
#
#-----------------------------------------------------------------*
# Age Units                                                       *
# Defines the minimum age of data before aggregation is done      *
# Only applies to the lowest aggregation unit for a product       *
# Only HOUR and DAY are supported                                 *
#-----------------------------------------------------------------*
KSY_HOUR_AGE_UNITS=1
KSY_DAY_AGE_UNITS=0
```

```
#---------------------------------------------------------------*
# Maximum number of simultaneous worker threads                 *
# Default is 2                                                  *
# Recommended value: number of processors on your server times 2 *
#---------------------------------------------------------------*
KSY_MAX_WORKER_THREADS=2

#---------------------------------------------------------------*
# KSY_CACHE_MINS records the number of minutes to cache the     *
# database connectivity for MOSWOS reporting purposes. The      *
# minimum value is 5 minutes.                                   *
#---------------------------------------------------------------*
KSY_CACHE_MINS=10
```

Configuring the Warehouse Proxy Agent using the Tivoli Enterprise Portal

This function allows you to remotely configure the Warehouse Proxy Agent using the Tivoli Enterprise Portal. You can use this function anytime the Warehouse Proxy Agent configuration needs to be changed.

For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely manage a Warehouse Proxy Agent on Linux systems, you must install the Monitoring Agent for Linux OS on the same server. To remotely manage a Warehouse Proxy Agent on Windows systems, you must install the Monitoring Agent for Windows on the same server.

Also, to remotely configure the agent, the Warehouse Proxy Agent itself must be added to the remote depot. The CD image contains a deploy directory where the Warehouse Proxy Agent bundle is located. From this directory you can run the following command:

```
tacmd addbundles -i . -t hd
```

Notes:

1. The **Test Connection** button is not available when configuring through the Tivoli Enterprise Portal.
2. The Warehouse Proxy Agent for the target platform being configured must have already been added to the Tivoli Enterprise Monitoring Server depot for the configuration to succeed.
3. Browsing for JDBC JAR files is not supported since they are on a remote system. You must enter the comma separated, fully qualified list of JDBC JAR files. The same as you do in the CLI.

The Deployment Status Detail workspace should show a SUCCESSFUL Transaction for the Configure command against the Warehouse Proxy Agent. The configuration file on the remote system where Warehouse Proxy Agent is installed should contain the corresponding changes executed on the configuration panel. Configuration parameters for the Warehouse Proxy Agent are set in the following files according to operating system:

Windows

ITM_HOME\TMAITM6\khdenv

For example: `C:\IBM\ITM\TMAITM6\khdenv`

Linux and UNIX

ITM_HOME/config/hd.ini

For example: `/opt/IBM/ITM/config/hd.ini`

Afterwards, the Warehouse Proxy Agent operates as usual when configuration is successful. Connectivity of the Warehouse Proxy Agent can be checked on the Warehouse Proxy Agent Status workspace.

To enable this function, complete the following steps:

1. Open the Tivoli Enterprise Portal.

2. Right-click on the Warehouse Proxy Agent in the navigator tree.
3. Choose **Configure**. The configuration panel for the Warehouse Proxy Agent is displayed.

The log file can be found in the `%CANDLEHOME%/cnp/logs` directory.

Configuring the Summarization and Pruning Agent using the Tivoli Enterprise Portal

This function allows you to remotely configure the Summarization and Pruning Agent using the Tivoli Enterprise Portal. You can use this function anytime the Summarization and Pruning Agent configuration needs to be changed.

For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely manage a Summarization and Pruning Agent on Linux systems, you must install the Monitoring Agent for Linux OS on the same server. To remotely manage a Summarization and Pruning Agent on Windows systems, you must install the Monitoring Agent for Windows on the same server.

Also, to remotely configure the agent, the Summarization and Pruning Agent itself must be added to the remote depot. The CD image contains a deploy directory where the Summarization and Pruning Agent bundle is located. From this directory you can run the following command:

```
tacmd addbundles -i . -t sy
```

The Deployment Status Detail workspace should show a SUCCESSFUL Transaction for the Configure command against the Summarization and Pruning Agent. The configuration file (`sy.ini` or `KSYENV`) on the remote system where the Summarization and Pruning Agent is installed should contain the corresponding changes executed on the configuration panel. Afterwards, the Summarization and Pruning Agent operates as usual when configuration is successful. Connectivity of the Summarization and Pruning Agent can be checked on the Summarization and Pruning Status workspace.

To enable this function, complete the following steps:

1. Open the Tivoli Enterprise Portal.
2. Select the Summarization and Pruning Agent that needs to be remotely configured.
3. Right-click on the Summarization and Pruning Agent in the navigator tree.
4. Choose **Configure**. The configure panel for the Summarization and Pruning Agent is displayed.

Notes:

1. The **Test Connection** button is not available when configuring through the Tivoli Enterprise Portal.
2. The Summarization and Pruning Agent for the target platform being configured must have already been added to the Tivoli Enterprise Monitoring Server depot for the configuration to succeed.
3. Browsing for JDBC JAR files is not supported since they are on a remote system. You must enter the comma separated, fully qualified list of JDBC JAR files. The same as you do in the CLI.

The Tivoli Enterprise Portal log file can be found in the `%CANDLEHOME%\cnp\logs` directory for Windows systems, or the `installer/CANDLEHOME/logs` directory for UNIX systems. Remote deploy tracing can also be set: ERROR(UNIT: KDY ALL) on the OS agent, ERROR(UNIT: KDY ALL) on the monitoring server.

Remotely starting and stopping the Warehouse Proxy Agent using the Tivoli Enterprise Portal

This function allows you to remotely manage the Warehouse Proxy Agent using the Tivoli Enterprise Portal. You can use this function anytime the Warehouse Proxy Agent needs to be started or stopped.

For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely manage a Warehouse Proxy Agent on Linux systems, you must install the Monitoring Agent for Linux OS on the same server. To remotely manage a Warehouse Proxy Agent on Windows systems, you must install the Monitoring Agent for Windows on the same server.

The Deployment Status Detail workspace should show a SUCCESSFUL Transaction for the Configure command against the Warehouse Proxy Agent. The Warehouse Proxy Agent in the navigator tree should be grayed out if the agent is stopped, and visible if it is started. Afterwards, the Warehouse Proxy Agent operates as usual when configuration is successful. Connectivity of the Warehouse Proxy Agent can be checked on the Warehouse Proxy Agent Status workspace.

To enable this function, complete the following steps:

1. Open the Tivoli Enterprise Portal.
2. Select the Warehouse Proxy Agent that needs to be remotely managed.
3. Right-click on the Warehouse Proxy Agent in the navigator tree.
4. Choose **Start** or **Stop**. The configure panel for the Warehouse Proxy Agent is displayed.

The Tivoli Enterprise Portal log file can be found in the `%CANDLEHOME%\cnp\logs` directory for Windows systems, or the `installer/CANDLEHOME/logs` directory for UNIX systems. Remote deploy tracing can also be set: ERROR(UNIT: KDY ALL) on the OS agent, ERROR(UNIT: KDY ALL) on the monitoring server.

Remotely starting and stopping the Summarization and Pruning Agent using the Tivoli Enterprise Portal

This function allows you to remotely manage the Summarization and Pruning Agent using the Tivoli Enterprise Portal. You can use this function anytime the Summarization and Pruning Agent needs to be started or stopped.

For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely manage a Summarization and Pruning Agent on Linux systems, you must install the Monitoring Agent for Linux OS on the same server. To remotely manage a Summarization and Pruning Agent on Windows systems, you must install the Monitoring Agent for Windows on the same server.

The Deployment Status Detail workspace should show a SUCCESSFUL Transaction for the Configure command against the Summarization and Pruning Agent. The Summarization and Pruning Agent in the navigator tree should be grayed out if the agent is stopped, and visible if it is started. Afterwards, the Summarization and Pruning Agent operates as usual when configuration is successful. Connectivity of the Summarization and Pruning Agent can be checked on the Summarization and Pruning Agent Status workspace.

To enable this function, complete the following steps:

1. Open the Tivoli Enterprise Portal.
2. Select the Summarization and Pruning Agent that needs to be remotely managed.
3. Right-click on the Summarization and Pruning Agent in the navigator tree.
4. Choose **Start** or **Stop**. The configure panel for the Summarization and Pruning Agent is displayed.

The Tivoli Enterprise Portal log file can be found in the `%CANDLEHOME%\cnp\logs` directory for Windows systems, or the `installer/CANDLEHOME/logs` directory for UNIX systems. Remote deploy tracing can also be set: ERROR(UNIT: KDY ALL) on the OS agent, ERROR(UNIT: KDY ALL) on the monitoring server.

Remotely deploying the Warehouse Proxy Agent

This function allows you to remotely deploy the Warehouse Proxy Agent. You can use this function anytime the Warehouse Proxy Agent needs to be installed remotely.

For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely manage a Warehouse Proxy Agent on Linux systems, you must install the Monitoring Agent for Linux OS on the same server. To remotely manage a Warehouse Proxy Agent on Windows systems, you must install the Monitoring Agent for Windows on the same server.

Also, to remotely deploy the Warehouse Proxy Agent, you must populate the remote depot with the necessary bundles (the Warehouse Proxy Agent bundle and its prerequisites) for the operating systems where the Warehouse Proxy Agent needs to be deployed. This can be done using the product installer or the **tacmd addBundles** command.

During this process of adding bundles, IBM Tivoli Monitoring reads the descriptor file (dsc) of the specified bundle you are adding, and identifies (among all of the files in the installation media) which files need to be copied on the agent's depot in your monitoring server. The files loaded are copied onto the path specified in the DEPOTHOME environment variable, defined in the `KBBENV` environment file. The **tacmd viewDepot** command allows you to display the types of agents in the deployment depot. The CD image contains a deploy directory where the Warehouse Proxy Agent bundle is located. When the depot is populated, you must deploy an OS agent on the remote computer, using the **tacmd createNode** command.

After the node has been created and OS agent is running, you can install other non-OS agents. This installation can be done by using the Tivoli Enterprise Portal or by using the **tacmd createNode** command.

Run the following command to complete enabling this function:

```
tacmd addSystem –t hd –n Os_node_name
```

Properties supported:

```
 KHD_DB_TYPE.KHD_DBMS
Allowed values are DB2, ORACLE,MSSQL
 KHD_PARMS.KHD_BATCH_USE
Controls if batching is enabled. Allowed values are true, false. Default is true.
 KHD_PARMS.KHD_DB_COMPRESSION
Controls if database compression should be used when creating tables and indexes.
Allowed values are true, false. Default is false.
 KHD_PARMS.KHD_WAREHOUSE_TEMS_LIST
Comma-separated list of TEMS that this Warehouse Proxy is responsible for.
 KHD_PARMS.KHD_WAREHOUSE_USER
The user ID used to connect to TDW
 KHD_PARMS.KHD_WAREHOUSE_PASSWORD
The password used to connect to TDW.
 KHD_PARMS.KHD_SERVER_Z_COMPRESSION_ENABLE
Allowed values are true, false. Default is false.
 KHD_PARMS.KHD_SERVER_DIST_COMPRESSION_ENABLE
Allowed values are true, false. Default is true.
```

Windows Only:

```
 KHD_PARMS.KHD_ODBC_DSN
The ODBC data source name to be used.
```

Unix/Linux only:

```
 KHD_PARMS.KHD_DB2_JDBCURL
 The JDBC URL to connect to DB2. Default is jdbc:db2://localhost:50000/WAREHOUS
 KHD_PARMS.KHD_DB2_JDBCDRIVER
The DB2 JDBC driver. Default is com.ibm.db2.jcc.DB2Driver
```

```
 KHD_PARMS.KHD_ORACLE_JDBCURL
The JDBC URL to connect to URL. Default is jdbc:oracle:thin:@<server>:<port>:<database>
 KHD_PARMS.KHD_ORACLE_JDBCDRIVER
The Oracle JDBC drive. Default is oracle.jdbc.driver.OracleDriver
 KHD_PARMS.KHD_MSSQL_JDBCURL
The JDBC URL to connect to Microsoft SQL Server. Default is
jdbc:sqlserver://<server>:<port>;databasename=<database>;
SelectMethod=cursor
 KHD_PARMS.KHD_MSSQL_JDBCDRIVER
The Microsoft SQL Server JDBC driver. The default is
com.microsoft.sqlserver.jdbc.SQLServerDriver
 KHD_PARMS.KHD_WAREHOUSE_JARS
The comma separated list of JDBC JARs that are needed to connect to the database.
```

A Warehouse Proxy Agent should be installed and configured on the remote system once the transaction has completed successfully. The Warehouse Proxy Agent operates as usual when it is started. Connectivity of the Warehouse Proxy Agent can be checked on the Warehouse Proxy Agent Status workspace.

The configureSystem command allows CLI configuration of a remote agent. If KHD_PARMS.KHD_WAREHOUSE_TEMS_LIST is not provided, the value will be set to an empty value.

It is the OS Agent that is used by remote deploy to do the work. The OS Agent log on the endpoint shows all the remote deploy activity (kdy component) processing. If you enable ERROR(UNIT:KDY ALL) on the endpoint's OS Agent tracing, you will see the remote deploy processing . To redeploy the same agent, run the **tacmd updateagent** command using the –v option.

Note: Remote deploy will start the agent after installation which might cause situations to fire due to lack of default configuration. The configuration supplied during the remote deploy is applied after the install and start is done. The agent will be restarted after the configuration is complete.

Remotely deploying the Summarization and Pruning Agent

This function allows you to remotely deploy the Summarization and Pruning Agent. You can use this function anytime the Summarization and Pruning Agent needs to be installed remotely.

For remote management of monitoring agents in general, you must install the appropriate OS monitoring agent. For instance, to remotely manage a Summarization and Pruning Agent on Linux systems, you must install the Monitoring Agent for Linux OS on the same server. To remotely manage a Summarization and Pruning Agent on Windows systems, you must install the Monitoring Agent for Windows on the same server.

Also, to remotely deploy the Summarization and Pruning Agent, you must populate the remote depot with the necessary bundles (the Summarization and Pruning Agent bundle and its prerequisites) for the operating systems where the Summarization and Pruning Agent needs to be deployed. This can be done using the product installer or the **tacmd addBundles** command.

During this process of adding bundles, IBM Tivoli Monitoring reads the descriptor file (dsc) of the specified bundle you are adding, and identifies (among all of the files in the installation media) which files need to be copied on the agent's depot in your monitoring server. The files loaded are copied onto the path specified in the DEPOTHOME environment variable, defined in the KBBENV environment file. The **tacmd viewDepot** command allows you to display the types of agents in the deployment depot. The CD image contains a deploy directory where the Summarization and Pruning Agent bundle is located. When the depot is populated, you must deploy an OS agent on the remote computer, using the **tacmd createNode** command.

After the node has been created and OS agent is running, you can install other non-OS agents. This installation can be done by using the Tivoli Enterprise Portal or by using the **tacmd createNode** command.

Run the following command to complete enabling this function:

```
tacmd addSystem –t sy –n Os_node_name
```

Properties supported:

```
 DBTYPE.KSY_WAREHOUSE_TYPE
The database type used for TDW. Allowed values are DB2, ORACLE, MSSQL
 SOURCES.KSY_WAREHOUSE_JARS
The comma-separated list of JDBC JAR files needed to connect to the database.
 SOURCES.KSY_DB2_JDBCURL
The JDBC URL used to connect to DB2 database. Default is jdbc:db2://localhost:50000/WAREHOUS
 SOURCES.KSY_DB2_JDBCDRIVER
The DB2 JDBC driver. Default is com.ibm.db2.jcc.DB2Driver
 SOURCES.KSY_ORACLE_JDBCURL
The JDBC URL used to connect to Oracle. Default is jdbc:oracle:thin:@<server>:<port>:<database>
 SOURCES.KSY_ORACLE_JDBCDRIVER
The Oracle JDBC driver. Default is oracle.jdbc.driver.OracleDriver
 SOURCES.KSY_MSSQL_JDBCURL
The JDBC URL used to connect to Microsoft SQL Server. Default is
jdbc:sqlserver://<server>:<port>;databasename=<database>;SelectMethod=cursor
 SOURCES.KSY_MSSQL_JDBCDRIVER
The Microsoft SQL Server JDBC Driver. Default is com.microsoft.sqlserver.jdbc.SQLServerDriver
 SOURCES.KSY_WAREHOUSE_USER
The user ID used to connect to the TDW database.
 SOURCES.KSY_WAREHOUSE_PASSWORD
The password used to connect to the TDW database.
 SOURCES.KSY_CNP_SERVER_HOST
The TEPS hostname. Default is localhost
 SOURCES.KSY_CNP_SERVER_PORT
The TEPS port number. Default is 1920
 SCHEDULING.KSY_FIXED_SCHEDULE
The scheduling to be used by S&P. Y for fixed schedule or N for flexible schedule.
Default is Y
 SCHEDULING.KSY_EVERY_N_MINS
The flexible scheduling interval in minutes. Default is 15
 SCHEDULING.KSY_EVERY_N_DAYS
The fixed scheduling interval in days. Default is 1
 SCHEDULING.KSY_HOUR_TO_RUN
The fixed scheduling time to run. Default is 2
 SCHEDULING.KSY_HOUR_AM_PM
The fixed scheduling AM or PM of time to run. Allowed values: AM,
PM Default is AM
 SCHEDULING.KSY_MINUTE_TO_RUN
The fixed scheduling minute to run. Default is 0
 SCHEDULING.KSY_BLACKOUT
Comma-separated list of blackout periods in the format HH:MM-HH:MM using
24 hour format. The flexible scheduling will not start during the listed blackout periods.
Note that the end time should be greater than the start time of the blackout period.
 WORK.KSY_START_OF_WEEK_DAY
The start of the week day, used when computing weekly aggregates.
Allowed values are 0 (Sunday), 1 (Monday). Default is 0 (Sunday)
 WORK.KSY_SHIFTS_ENABLED
Controls whether shifts are used or not. Allowed  values are Y or N Default is N
 WORK.KSY_SHIFT1_HOURS
Comma-separated list of off peak shift hours. Default is
0,1,2,3,4,5,6,7,8,18,19,20,21,22,23
 WORK.KSY_SHIFT2_HOURS
Comma-separated list of peak shift hours. Default is 9,10,11,12,13,14,15,16,17
 WORK.KSY_VACATIONS_ENABLED
Controls whether vacation days are used or not.  Allowed values are Y or N Default is N
 WORK.KSY_WEEKENDS_AS_VACATIONS
Controls whether weekends are treated as vacation days. Allowed values are Y or N.
```

```
Default is N
 WORK.KSY_VACATION_DAYS
Comma separated list of vacation days in the format YYYYMMDD where YYYY is the year,
MM is the month (1-12) and DD is the day (1-31).
 LOG.KSY_WAREHOUSELOG_PRUNE
Controls the pruning of the WAREHOUSELOG table. The format is nn.uuu where nn is
the number of units to retain and uuu is one of day, month, or year.
Example: 5.day to retain data for 5 days.
 LOG.KSY_WAREHOUSEAGGREGLOG_PRUNE
Controls the pruning of the WAREHOUSEAGGREGLOG table. The format is nn.uuu where nn is
the number of units to retain and uuu is one of day, month, or year.
Example: 5.day to retain data for 5 days.
 ADDITIONAL.KSY_MAX_WORKER_THREADS
The number of threads that will be used for summarization and pruning. Default is 2
 ADDITIONAL.KSY_MAX_ROWS_PER_TRANSACTION
The number of rows per database transaction. Default is 1000
 ADDITIONAL.KSY_TIMEZONE_IND
Controls which time zone offset should be used. Allowed values are AGENT or
WAREHOUSE Default is AGENT
 ADDITIONAL.KSY_HOUR_AGE_UNITS
Controls the age in hours of data to be summarized. Default is 1
 ADDITIONAL.KSY_DAY_AGE_UNITS
Controls the age in days of data to be summarized. Default is 0
 ADDITIONAL.KSY_NODE_ERROR_UNITS
The number of node errors to keep for the self-monitoring workspace. Default is 10
 ADDITIONAL.KSY_SUMMARIZATION_UNITS
The number of summarization runs to keep for the self monitoring workspace.
Default is 10
 ADDITIONAL.KSY_CACHE_MINS
The time in minutes that connectivity information is cached. Default is 15
 ADDITIONAL.KSY_BATCH_MODE
Controls whether batching of data should be used to improve performance.
Allowed values are 0 (single system) or 1 (multiple systems). Default is 0
 ADDITIONAL.KSY_DB_COMPRESSION
Controls whether tables and indexes should be created with database compression
enabled. Allowed values are N or Y. Default is N.
```

A Summarization and Pruning Agent should be installed and configured on the remote system once the transaction has completed successfully. The Summarization and Pruning Agent operates as usual when it is started. Connectivity of the Summarization and Pruning Agent can be checked on the Summarization and Pruning Agent Status workspace.

The configureSystem command allows CLI configuration of a remote agent. If SCHEDULING.KSY_BLACKOUT, WORK.KSY_VACATION_DAYS, LOG.KSY_WAREHOUSELOG_PRUNE or LOG.KSY_WAREHOUSEAGGREGLOG_PRUNE are not provided, their value will be set to an empty value.

It is the OS Agent that is used by remote deploy to do the work. The OS Agent log on the endpoint shows all the remote deploy activity (kdy component) processing. If you enable ERROR(UNIT:KDY ALL) on the endpoint's OS Agent tracing, you will see the remote deploy processing . To redeploy the same agent, run the **tacmd updateagent** command using the –v option.

Note: Remote deploy will start the agent after installation which may cause situations to fire due to lack of default configuration. The configuration supplied during the remote deploy is applied after the install and start is done. The agent will be restarted after the configuration is complete.

WAREHOUSELOG and WAREHOUSEAGGREGLOG tables

The WAREHOUSELOG table lets you know how many exports succeed and how many failed because of an ODBC error or a timeout value issue.

The WAREHOUSELOG table has the following columns. All times are the local time on the machine where the Warehouse Proxy instance is running.

ORIGINNODE
The name of the computer that made the request. This name is the node name for the agent. For example, Primary::box1:NT.

OBJECT
The attribute group that submitted the request. For example, NT_System.

STARTQUEUE
The time when the request was inserted in the work queue. For example, 10508201154000000.

ENDQUEUE
The time when the request exited the work queue. For example, 10508201155000000.

STARTEXPORT
The amount of time that elapsed before the first row of the sample request was retrieved. For example, 105082011562000000.

EXPORTTIME
The amount of time after the export request transaction was committed. For example, 10508201157000000.

ROWSINSERTED
The number of row inserted in the database for the request. For example, 1000.

ROWSRECEIVED
The number of rows retrieved from the RPC source. For example, 1000.

ROWSSKIPPED
This column is not used.

STARTTIME
The start time of the collection for that sample. For example, 10508150920000000.

ENDTIME
The end time of the collection for that sample. For example, 1050815092000000.

ERRORMSG
An error message when no rows are inserted in the database. The error message can indicate an ODBC error or a TIMEOUT error. For example:

```
Sample data rejected for timeout reason at stage COMMIT EXPORT
```

WPSYSNAME
The name of the Warehouse Proxy Agent that inserted the rows into the database.

The WAREHOUSEAGGREGLOG table logs the progress of the Summarization and Pruning Agent as it is processing data. Each time the Summarization and Pruning Agent executes, it adds an entry for each attribute group (OBJECT column) and origin node (ORIGINNODE column) that was processed. The WAREHOUSEAGGREGLOG table contains the following columns:

ORIGINNODE
The name of the computer that is being summarized. This name is the node name for the agent. For example, Primary::box1:NT. -

OBJECT
The attribute group that was processed.

LOGTMZDIFF
The time zone difference for the Summarization and Pruning Agent.

MINWRITETIME
The minimum WRITETIME value that was read from the sample data for the specified ORIGINNODE and OBJECT.

MAXWRITETIME
The maximum WRITETIME value that was read from the sample data for the specified ORIGINNODE and OBJECT

STARTTIME
The time that the Summarization and Pruning Agent processing began for the specified ORIGINNODE and OBJECT.

ENDTIME
The time that the Summarization and Pruning Agent processing ended for the specified ORIGINNODE and OBJECT.

ROWSREAD
The number of sample data rows read for the specified ORIGINNODE and OBJECT in the time interval MINWRITETIME and MAXWRITETIME.

Part 6. Integrating event management systems

If you are using either Tivoli Enterprise Console or Netcool/OMNIbus in addition to IBM Tivoli Monitoring to manage events in your enterprise, you can integrate and manage events from a single console. You can integrate event management by forwarding events reported by Tivoli Enterprise Monitoring Agents to either event system for correlation and management—changes in event status made on the event system are reflected back to the hub monitoring server that forwarded them. Or you can enable events reported by a Tivoli Enterprise Monitoring Agent to be passed directly to OMNIbus for processing, thereby bypassing the monitoring server entirely.

To enable forwarding of situation events, the Tivoli Enterprise Console or Netcool/OMNIbus server (the event server) must be configured to receive the events, a situation update forwarding process must be installed on the event server, situation forwarding must be enabled on either the hub monitoring server or the monitoring agent, and a default event integration facility (EIF) destination must be defined.

Using the Tivoli Enterprise Portal, you can define monitoring specifications, called *situations*, to detect the occurrence of specific conditions on managed systems. When conditions that match the specifications are detected by monitoring agents, events are reported to the monitoring servers. In environments where IBM Tivoli Enterprise Console or Netcool/OMNIbus are also being used for event management, hub Tivoli Enterprise Monitoring Servers or the agents themselves can be configured to forward these situation events to either of these event servers for further correlation and management. Changes in the status of events made on the event server are reported back to the forwarding monitoring server so that events are synchronized on the two event management systems.

The chapters in this section provide instructions for implementing event integration by forwarding situation events to Tivoli Enterprise Console and Netcool/OMNIbus.

In addition you can view events within the Tivoli Enterprise Portal. The *Common Event Console* is a Tivoli Enterprise Portal view that provides an integrated display of events from multiple event systems. The Common Event Console presents normalized events from the supported event systems in a single table. By default, the view displays events reported to a hub Tivoli Enterprise Monitoring Server by Tivoli Enterprise Monitoring Agents (*situation events*); it can also be configured to present events from Tivoli Enterprise Console and Netcool/OMNIbus event systems.

A common event connector (frequently called simply a connector) enables the integrated display of events from an event system in the Common Event Console. The connector retrieves event data from a supported event system and sends user-initiated actions to be run in that event system. To have the events from a specific event system displayed in the Common Event Console, you must configure a connector for that event system. Because the connector for Tivoli Enterprise Monitoring Agents is configured when you install the portal, the Common Event Console includes all situation events by default. However, to have Tivoli Enterprise Console or Netcool/OMNIbus events included in the Common Event Console, you must configure a connector for each of these event systems after you install the Tivoli Enterprise Portal. You might also want to change some of the configuration values for the default connector. For information on configuring the display of events from Tivoli Enterprise Console and Netcool/OMNIbus, see the *IBM Tivoli Monitoring: Administrator's Guide*.

After situation forwarding is enabled, by default all situation events are forwarded to the specified event server. However, you can customize which situation events are forwarded and to which event server. For information on specifying which situation events to forward, see the Tivoli Enterprise Portal online help and the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.

Event integration scenarios

The scenarios in this section illustrate how event integration can be implemented with either Tivoli Enterprise Console or Netcool/OMNIbus by forwarding situation events. To implement these scenarios, hub monitoring servers must be configured to forward events to at least one event server, and the event server must be configured to interpret the events and send updates to the originating monitoring server. A special event-synchronization component, the Situation Update Forwarder, must be installed on the event server. You can find complete instructions for implementing these scenarios in the next two chapters, Chapter 25, "Setting up event forwarding to Tivoli Enterprise Console," on page 643 and Chapter 26, "Setting up event forwarding to Netcool/OMNIbus," on page 675.

As of IBM Tivoli Monitoring V6.2.2, a new type of Tivoli Management Services agent, the System Monitor Agent, allows you to send OS monitoring data directly to Netcool/OMNIbus without first passing the data to a Tivoli Enterprise Monitoring Server. In this way these agents can run in agent-only environments that lack the standard Tivoli Monitoring servers (the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server). These monitoring agents, which run on Windows and on Linux/UNIX, effectively replace the OMNIbus System Service Monitors for monitoring of desktop operating systems. Chapter 11, "Monitoring your operating system via a System Monitor Agent," on page 349 provides complete information on installing, configuring, and uninstalling a System Monitor Agent on either Windows or Linux.

Note: As of fix pack 2 for V6.2.2, for sites running x86_64 CPUs, both 32-bit and 64-bit Windows environments (Windows 2003, Vista, 2008) are supported for the System Monitor Agents.

An enhancement provided with the first fix pack for version 6.2.2 enables the System Monitor Agents to send event data directly to OMNIbus, thus making a monitoring server unnecessary even for event processing.

- EIF events generated by autonomous agents (including the System Monitor Agents) can be sent directly to either Tivoli Enterprise Console or OMNIbus for private situations only.
- SNMP alerts generated by autonomous agents can be forwarded to any SNMP trap receiver, including OMNIbus's MTTRAPD probe for both enterprise and private situations.

For more information, see "Event forwarding from autonomous agents" on page 65.

For a complete list of operating systems that are supported for IBM Tivoli Monitoring components, see "Supported operating systems" on page 139. For event integration with Netcool/OMNIbus software prerequisites, see "Required software for event integration with Netcool/OMNIbus" on page 158.

Chapter 25. Setting up event forwarding to Tivoli Enterprise Console

Tivoli Enterprise Console events can be forwarded from Tivoli Monitoring V6.2.3 to Tivoli Enterprise Console Version 3.9. The events are forwarded from the hub monitoring server to the Tivoli Enterprise Console server or Tivoli Enterprise Console gateway. Make sure that firewall ports are opened between the hub monitoring server and Tivoli Enterprise Console servers. By default, the Tivoli Enterprise Console uses port 5529.

To view updates to the forwarded situation events in the Tivoli Enterprise Portal, you need to install the event synchronization component on the event server. The event synchronization component enables changes in the event status made on the Tivoli Enterprise Console to be reflected on the Tivoli Enterprise Portal.

The following table provides an overview of the tasks required to configure situation event forwarding and synchronization:

Table 123. Tivoli Enterprise Console event synchronization installation and configuration steps

Task	Where to find information
Plan the deployment of your integration.	"Event integration with Tivoli Enterprise Console" on page 644
Gather information required during the installation and configuration processes.	"Information to gather for event forwarding" on page 126
Install the event synchronization component on your event server.	"Installing event synchronization on your event server" on page 648
Install monitoring agent .baroc files on the event server.	"Installing monitoring agent .baroc files on the event server" on page 661
Configure your monitoring server to forward events to Tivoli Enterprise Console.	"Configuring your monitoring server to forward events" on page 662
Start and stop event synchronization on the event server.	"Starting and stopping the Situation Update Forwarder process" on page 664
Modify the configuration of event synchronization as your monitoring needs change.	"Changing the configuration of the event synchronization component on the event server" on page 664
Define additional monitoring servers to the event server.	"Defining additional monitoring servers to the event server" on page 664
Customizing event status processing behavior when agent switching is used or the agent goes offline.	"Customizing event status processing behavior when agent switching is used or the agent goes offline" on page 665
Change the default TCP/IP settings if it is taking a long time to forward events back to the monitoring server.	"Changing the TCP/IP timeout setting on your event server" on page 668

If you are upgrading from a previous IBM Tivoli Monitoring release and already have event synchronization installed for Tivoli Enterprise Console, see "Upgrading to Tivoli Event Synchronization version 2.3.0.0" on page 669.

Event integration with Tivoli Enterprise Console

The scenarios in this section illustrate the various ways to forward situation events from one or more monitoring servers to one or more Tivoli Enterprise Console event servers:

- "One or more hub monitoring servers and a single event server"
- "A single hub monitoring server and multiple event servers" on page 645
- "Multiple hub monitoring servers and multiple event servers in a hub and spoke configuration" on page 646

The Tivoli Enterprise Console product significantly reduces the number of events displayed, enabling you to focus on the most critical, relevant events and manage even the largest, most complex environments. The Tivoli Enterprise Console product gives you the extensive control and flexibility that you need to manage and maintain availability across your enterprise. Managing situation events with the Tivoli Enterprise Console product gives you the following advantages:

- Aggregation of event information from a variety of different sources including those from other Tivoli software applications, Tivoli partner applications, custom applications, network management platforms, and relational database systems
- Pre-configured rules that automatically provide best-practices event management
- Persistence and processing of a high volume of events in an IT environment by:
  - Prioritizing events by their level of importance
  - Filtering redundant or low priority events
  - Correlating events with other events from different sources
  - Root cause analysis and resolution
  - Initiating automatic corrective actions, when appropriate, such as escalation
- Unified system and network management by automatically performing the following event management tasks:
  - Correlating the status of a system or application to the status of the network that it uses
  - Determining if the root cause of a system or application problem is an underlying network failure

Note: If you already have policies that contain emitter activities that send events to the Tivoli Enterprise Console, turning on Tivoli Event Integration event forwarding will result in duplicate events. You can deactivate the emitter activities within policies so you do not have to modify all your policies when you activate Tivoli Event Integration Facility forwarding by using **Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding** when you configure the monitoring server.

Using policies gives you more control over which events are sent and may not want to lose this granularity. Moreover, it is likely the policies that are invoking the Tivoli Enterprise Console emitter are doing little else. If you deactivate these activities, there is no point in running the policy. You may delete policies that are longer required, instead of disabling them.

One or more hub monitoring servers and a single event server

You can configure one or more monitoring servers to forward situation events to an event server. Figure 157 on page 645 shows multiple hub monitoring servers that are configured to forward situation events to the same event server. The event server sends situation updates based on Tivoli Enterprise Console rules and operator actions back to the hub monitoring server that is associated with that situation.

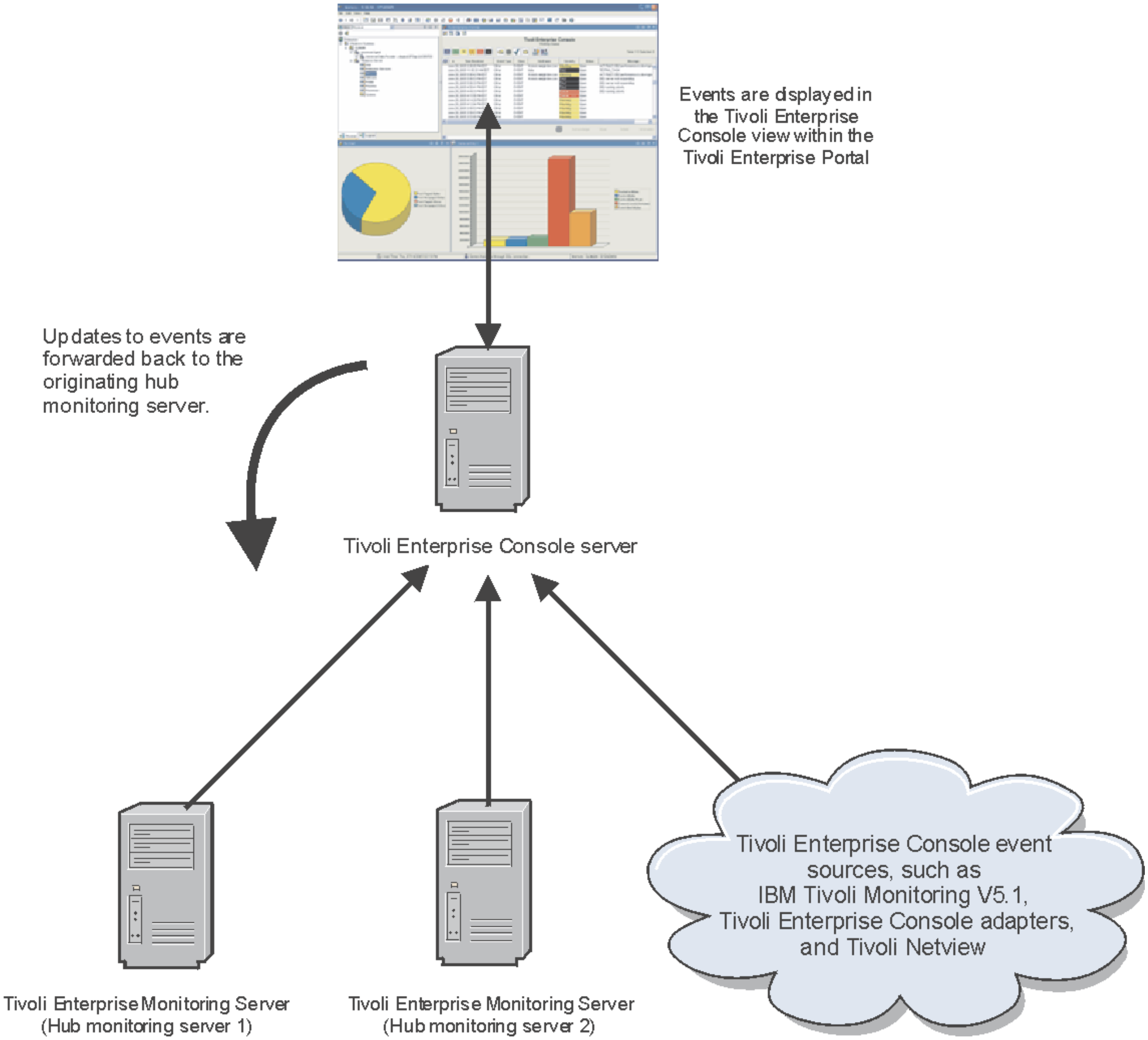

Events are displayed in the Tivoli Enterprise Console view within the Tivoli Enterprise Portal
Updates to events are forwarded back to the originating hub monitoring server.
Tivoli Enterprise Console server
Tivoli Enterprise Console event sources, such as IBM Tivoli Monitoring V5.1, Tivoli Enterprise Console adapters, and Tivoli Netview
Tivoli Enterprise Monitoring Server (Hub monitoring server 1)
Tivoli Enterprise Monitoring Server (Hub monitoring server 2)

Figure 157. One or more hub monitoring servers connecting to a single event server

A single hub monitoring server and multiple event servers

Figure 158 on page 646 shows a single hub monitoring server that is configured to forward situation events to multiple event servers. The event servers send situation updates based on Tivoli Enterprise Console rules and operator actions back to the hub monitoring server.

For this configuration, you must install the Tivoli Enterprise Console event synchronization component on each event server, and, for each situation, you must specify the event server to which the situation event is forwarded (see the *IBM Tivoli Monitoring: Administrator's Guide*).

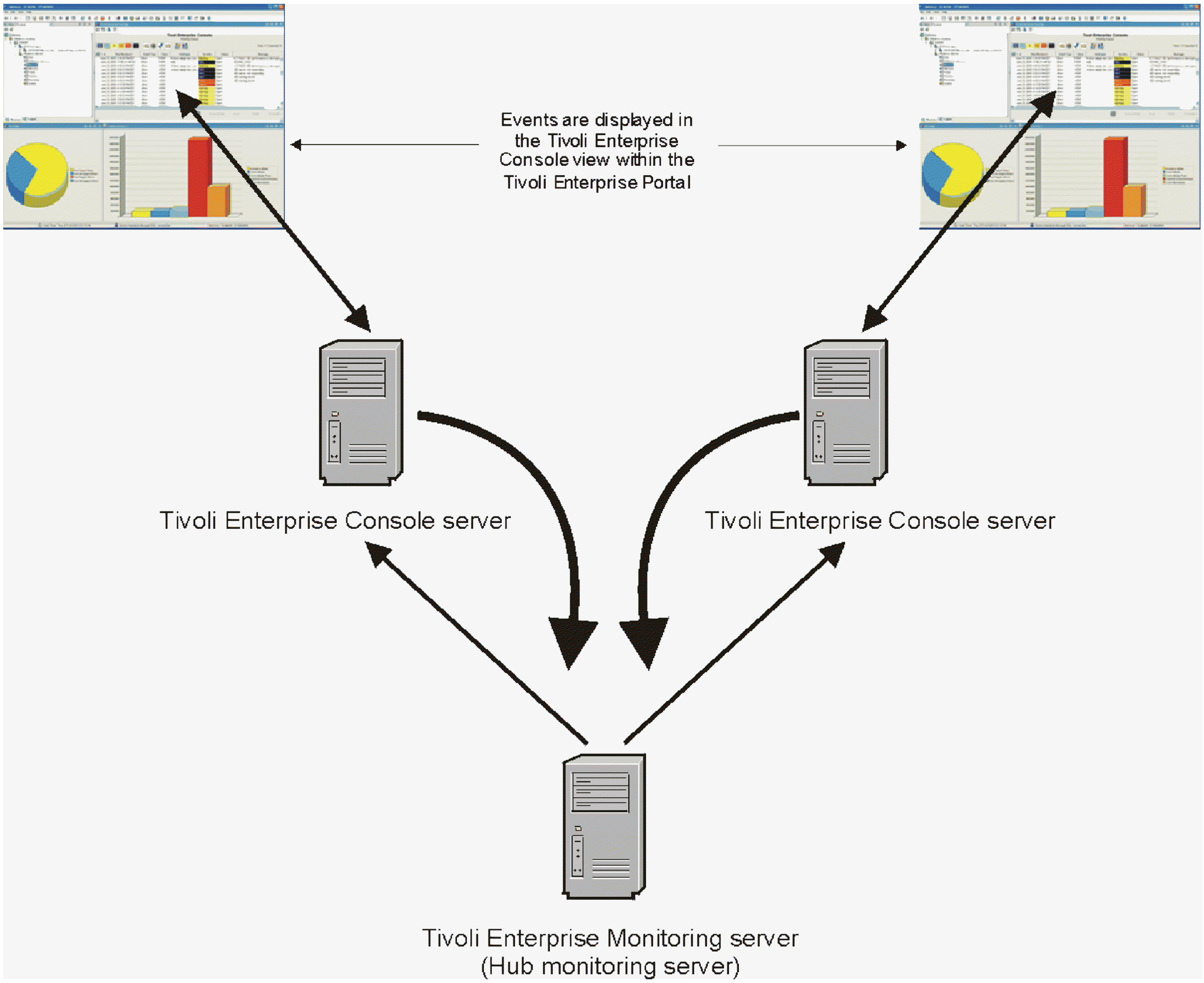

Events are displayed in the Tivoli Enterprise Console view within the Tivoli Enterprise Portal
Tivoli Enterprise Console server
Tivoli Enterprise Console server
Tivoli Enterprise Monitoring server (Hub monitoring server)

Figure 158. Single hub monitoring server and multiple event servers

Multiple hub monitoring servers and multiple event servers in a hub and spoke configuration

Figure 159 on page 647 shows multiple hub monitoring servers that are configured to forward situation events to an event server that is connected to a hub event server. The hub event server sends situation updates based on Tivoli Enterprise Console rules and operator actions back to the hub monitoring server that is associated with that situation.

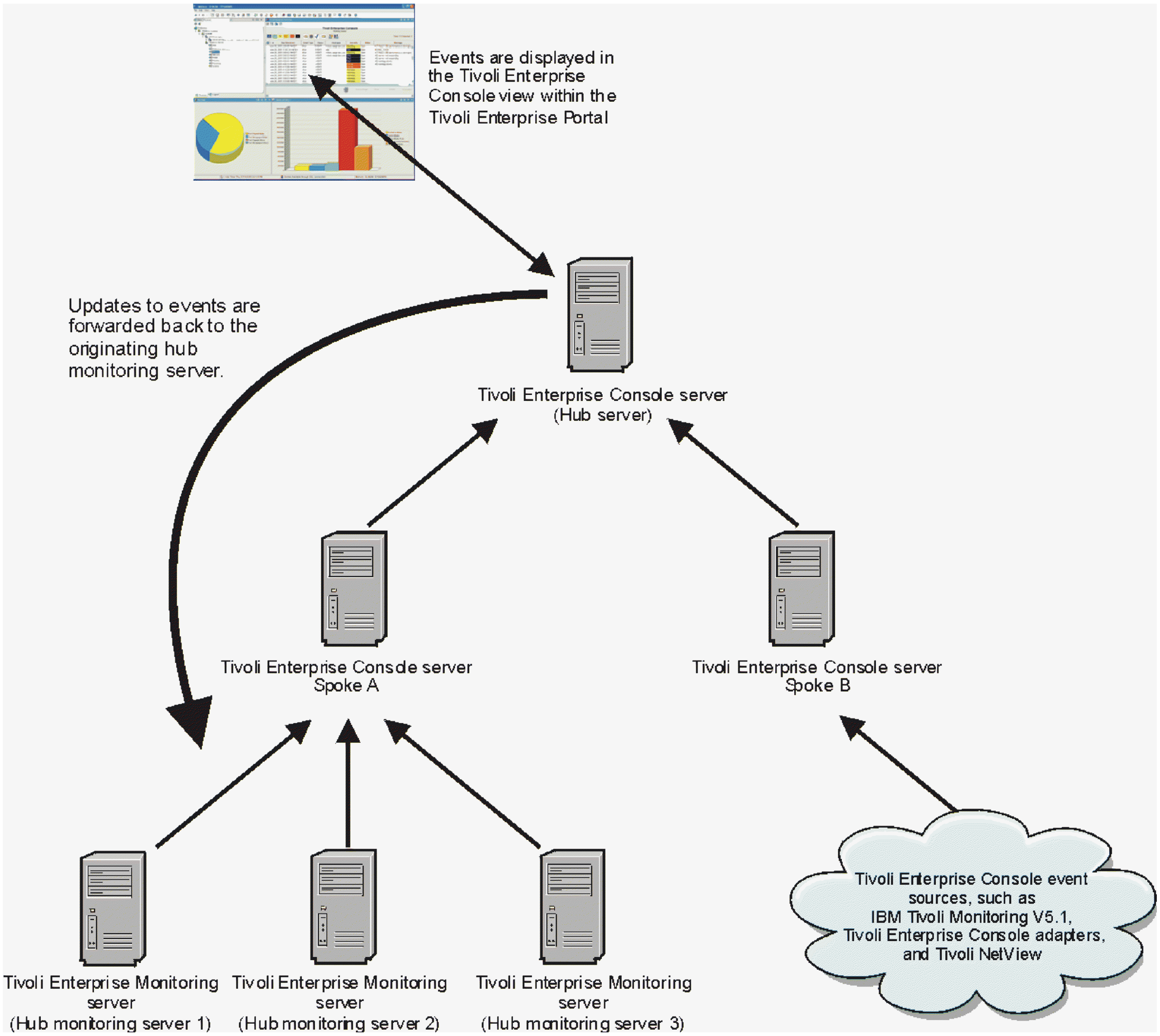

Events are displayed in the Tivoli Enterprise Console view within the Tivoli Enterprise Portal
Updates to events are forwarded back to the originating hub monitoring server.
Tivoli Enterprise Console server (Hub server)
Tivoli Enterprise Console server Spoke A
Tivoli Enterprise Console server Spoke B
Tivoli Enterprise Monitoring server (Hub monitoring server 1)
Tivoli Enterprise Monitoring server (Hub monitoring server 2)
Tivoli Enterprise Monitoring server (Hub monitoring server 3)
Tivoli Enterprise Console event sources, such as IBM Tivoli Monitoring V5.1, Tivoli Enterprise Console adapters, and Tivoli NetView

Figure 159. Multiple hub monitoring servers and multiple event servers in a hub and spoke configuration

Note: This graphic is intended to be an example of one possible scaled configuration for the IBM Tivoli Monitoring and Tivoli Enterprise Console integration. The procedures in this chapter do not provide all of the information needed to set up this sort of configuration.

For this configuration, you must install the Tivoli Enterprise Console event synchronization component on the hub event server. You must also load the omegamon.baroc and Sentry.baroc files on the spoke event servers, as described in “Modifying an existing rule base” on page 661. In addition, you must load each .baroc file for any monitoring agent generating situations that are forwarded to spoke event servers, as described in “Installing monitoring agent .baroc files on the event server” on page 661.

Determining when to use the IBM Tivoli Enterprise Console

IBM Tivoli Enterprise Console is a key element of a monitoring environment. Because of its capabilities, the Tivoli Enterprise Console is commonly used in a Manager of Managers role, providing an enterprise level solution for aggregation, consolidation, correlation, and management of events across the enterprise. It is also used as an integration point with which other management applications interact. Tivoli Enterprise Console provides the following capabilities:

- Aggregation of event information from a large number and variety of different sources including those from other Tivoli software applications, Tivoli partner applications, custom applications, network management platforms, and relational database systems
- Pre-configured rules that automatically provide best-practices event management and root cause determination from an end to end perspective
- Persistence, processing, and access to a high volume of events in an IT environment
- Unified system and network management by automatically performing the following event management tasks:
  - Correlating the status of a system or application to the status of the network that it uses
  - Determining if the root cause of a system or application problem is an underlying network failure

If you are monitoring fewer than 1000 active events and you want to view only situation events (not the other types of events that IBM Tivoli Enterprise Console can monitor), you can use the Situation Event Console in the Tivoli Enterprise Portal. If you are monitoring more than 1000 active events, consider moving to IBM Tivoli Enterprise Console for your event aggregation, and use the Tivoli Enterprise Console view within the Tivoli Enterprise Portal to display the event information. The response time for the Tivoli Enterprise Console view is better than the Situation Event Console view when a large number of events is displayed.

For additional information about the integration with Tivoli Enterprise Console, see “Event synchronization component” on page 8. For additional information about Tivoli Enterprise Console itself, see the Tivoli Enterprise Console information center. For additional information about using the Situation Event Console in the Tivoli Enterprise Portal, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.

Installing event synchronization on your event server

The installer for the event synchronization component is located in the `/tec` directory of the *IBM Tivoli Monitoring V6.2.3 Tools* DVD.

There are three methods for installing the component:

- “Installing from a wizard” on page 649
- “Installing from the command-line” on page 652
- “Installing from the command-line using a silent installation” on page 656

When you install the event synchronization component on your event server, the following changes are made:

- If you select to use an existing rule base, the event synchronization .baroc class files (omegamon.baroc and Sentry.baroc [if not present]) and the omegamon.rls rule set file are imported into your existing rule base. If you do not want to modify your existing rule base during the installation, you can choose to manually perform rule base modifications after installation is complete. See “Manually importing the event synchronization class files and rule set” on page 659 for more information.
- For all rule bases that have Sentry.baroc imported, the Sentry2_0_Base class is updated to define additional integration attributes for the situation events received from IBM Tivoli Monitoring.
- A new process, Situation Update Forwarder, is installed along with its supporting binary and configuration files. This process is used to forward updates to the situation events back to the monitoring server. On Windows, a Tivoli Situation Update Forwarder service is also created.

Notes:

1. If your IBM Tivoli Enterprise Console event server is running on Windows 2003 and you are planning to install the event synchronization remotely (using a program such as Terminal Services to connect to that Windows 2003 computer), you need to run the **change user /install** command before you run the installation. This puts the computer into the required "install" mode. After the installation, run the **change user /execute** command to return the computer to its previous mode.

2. If you have a monitoring server on an operating system like UNIX or Linux, you must configure your TCP/IP network services in the `/etc/hosts` file to return the fully qualified host name. See "Host name for TCP/IP network services" on page 133 for more information.
3. For a Windows event server, any existing rule base that you use must indicate a relative drive letter (such as C:\) as part of its associated path. To verify that your existing rule base contains a relative drive letter, run the following command from a bash environment on your event server:

   ```
   wrb -lsrb -path
   ```

   If the returned path includes something like *hostname*:*rulebase_directory*, with no drive letter (such as C:\), copy the `ESync2300Win32.exe` file from the \TEC subdirectory of the IBM Tivoli Monitoring installation image to the drive where the rule base exists and run the installation from that file.
4. If you are using a Windows event server, if you have any rule base with an associated path that does not contain a relative drive letter and that has the Sentry2_0_Base class imported, copy the `ESync2300Win32.exe` file from the \TEC subdirectory of the IBM Tivoli Monitoring installation image to the drive where the rule base exists and run the installation from that file.

   To verify if you have any rule bases that have an associated path containing no relative drive letter, run the **wrb -lsrb -path** command as described in the previous note.

   To determine if your rule bases have the Sentry2_0_Base class imported, run the following command against all of your rule bases:

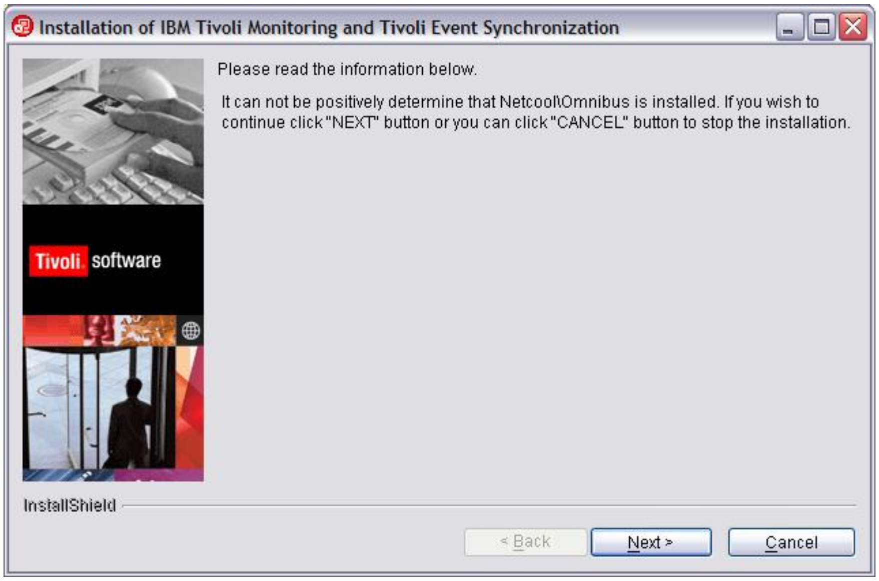
   ```
   wrb -lsrbclass rule_base
   ```

   where *rule_base* is the name of the rule base.

Installing from a wizard

Use the following steps to install event synchronization using the installation wizard:

Note: The appearance of the window shown in Figure 160 indicates that the installer did not find an IBM Tivoli Enterprise Console event server installed on the system, so it expects to install event synchronization for Netcool/OMNIbus. If you see this window when you launch the installation, cancel the installation and install the event server on the system before restarting the installer, or install the synchronization component on another system where an event server is installed.



Installation of IBM Tivoli Monitoring and Tivoli Event Synchronization

Please read the information below.

It can not be positively determine that Netcool\Omnibus is installed. If you wish to continue click "NEXT" button or you can click "CANCEL" button to stop the installation.

Figure 160. Window shown when no Tivoli Enterprise Console event server is found.

1. On the host of the event server, launch the event synchronization installer from the installation media:

On Windows, double-click the `ESync2300Win32.exe` file in the `\tec` subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.

On Linux or UNIX, change to the `\tec` subdirectory of the IBM Tivoli Monitoring V6.2.3 Tools DVD and run the following command:

```
ESync2300operating_system.bin
```

where *operating_system* is the operating system you are installing on (`aix`, `HP11`, `Linux`, `linux390`, `or Solaris`). For example, run the following command on an AIX computer:

```
ESync2300Aix.bin
```

2. Click **Next** on the Welcome window.
3. Select **I accept the terms in the license agreement** and click **Next**.
4. Complete the following fields and click **Next**:

Table 124. IBM Tivoli Enterprise Console event synchronization configuration fields

Field	Description
Name of configuration file	The name of the file where event synchronization configuration information is stored. The default name is situpdate.conf.
Number of seconds to sleep when no new situation updates	The polling interval, in seconds. The minimum value is 1, while the default value is 3. If there are no situation events, the Situation Update Forwarder rests for 3 seconds.
Number of bytes to use to save last event	Number of bytes that the long-running process uses when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) value is 50.
URL of the CMS SOAP Server	The URL for the SOAP Server configured on the computer where the monitoring server is running. The default value is `cms/soap`. This value is used to create the URL to which IBM Tivoli Enterprise Console sends event information. For example, http://*hostname*:*port*///cms/soap, where *hostname* is the host name of the monitoring server and *port* is the port.
Rate for sending SOAP requests to CMS from TEC via web services	The maximum number of event updates sent to the monitoring server at one time. The minimum (and default) value is 10 events.
Level of debug detail for log	The level of information for event synchronization that will be logged. You have the following choices: • Low (default) • Medium • Verbose

5. Complete the following information about the files where events will be written and click **Next**:

Table 125. IBM Tivoli Enterprise Console event synchronization configuration fields, continued

Field	Description
Maximum size of any single cache file, in bytes	The maximum permitted size, in bytes, for any one event cache file. The minimum (and default) value is 50000. Do not use commas when specifying this value (specify 50000 instead of 50,000).

Table 125. IBM Tivoli Enterprise Console event synchronization configuration fields, continued (continued)

Field	Description
Maximum number of caches files	The maximum number of event caches files at any given time. The minimum value is 2, while the default value is 10. When this value is reached, the oldest file is deleted to make room for a new file.
Directory for cache files to be located	The location where event cache files are located. The default locations are as follows: • On Windows: C:\tmp\TME\TEC\OM_TEC\persistence. • On UNIX: /var/TME/TEC/OM_TEC/persistence

6. Type the following information for each monitoring server with which you want to synchronize events and click **Add**. You must specify information for at least one monitoring server.

   Host name
   The fully qualified host name for the computer where the monitoring server is running. This name must match the information that will be in events that are issued from this monitoring server.

   User ID
   The user ID to access the computer where the monitoring server is running.

   Password
   The password to access the computer.

   Confirmation
   The same password, for confirmation.

   You can add information for up to 10 monitoring servers in this wizard. If you want to add additional monitoring servers, add them after you install them by using the steps provided in "Defining additional monitoring servers to the event server" on page 664.

7. When you have provided information about all of the monitoring servers, click **Next**.

   You are presented with the options of having the installer automatically perform rule base modifications, or manually performing the modifications after installation is complete (see Table 126).

Table 126. Options for rule base modification

Option	Description
Automatically install rules and classes	The installation wizard will ask for the rule base into which event synchronization class files and rule set will be imported, and automatically execute the rule base commands to do this.
Manually install rules and classes	The installation wizard will not create or update any rule base with event synchronization files. **Important:** You will have to manually create or update the rule base with event synchronization files after the installation is complete. See "Manually importing the event synchronization class files and rule set" on page 659.

   If you select the automatic option, continue with step 8. If you select the manual option, skip to step 11.

8. Specify the rule base that you want to use to synchronize events. You have two choices:
   - **Create a new rulebase**
   - **Use existing rulebase**

   If you select to use an existing rule base, the event synchronization .baroc class files (omegamon.baroc and Sentry.baroc [if not present]) and the omegamon.rls rule set file are imported into your existing rule base. Also, if Sentry.baroc has already been imported into the existing rule base, the Sentry2_0_Base class is extended to define additional integration attributes for the situation events from IBM Tivoli Monitoring.

- If you are creating a new rule base, type the name for the rule base you want to create and the path to where the new rule base will be located. There is no default location; you must specify a location.
- If you are using an existing rule base, type the name of the rule base.
- If you want to import an existing rule base *into* a new rule base, type the name of the existing rule base in the **Existing rulebase to import** field.

  Note: This step is only available if you are creating a new rule base.

9. Click **Next**.
10. If you indicated in the previous step that the installer uses an existing rule base to import the event synchronization class files and rule set, a window is displayed that allows you to specify whether you want the installer to back up the rule base before updating it. If you request a backup, specify both the backup rule base name and backup rule base path. If you leave these fields blank, no backup is made. Click **Next** to proceed to the pre-installation summary panel.
11. Verify the installation location, then click **Next**.

    The installation begins.
12. When the installation and configuration steps are finished, a message telling you to stop and restart the event server is displayed.

    If you chose to have the installer automatically update the rule base, you are offered the option of having the installer restart the event server for you. Check the box to have the installer restart the server. If you want to restart the event server yourself, leave the box unchecked.
13. Click **OK**.
14. Click **Finish** on the Summary Information window.

    Note: If any configuration errors occurred during installation and configuration, you are directed to a log file that contains additional troubleshooting information.

Perform the following tasks after the installation is finished:
- Stop and restart the event server for the configuration changes to take effect.
- Install the monitoring agent .baroc files on the event server as described in “Installing monitoring agent .baroc files on the event server” on page 661.
- Configure the monitoring server to forward events to the event server as described in “Configuring your monitoring server to forward events” on page 662.
- If you did not choose to have the rule base updated automatically, update the rule base as described in “Manually importing the event synchronization class files and rule set” on page 659.

Installing from the command-line

Use the following steps to install the event synchronization from the command-line on your event server:
1. Change to the `tec` directory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
2. Run the following command to launch the installation:

   On Windows:
   ```
   ESync2300Win32.exe -console
   ```
   On UNIX or Linux:
   ```
   ESync2300operating_system.bin -console
   ```
   where *operating_system* is the operating system you are installing on (`aix`, `HP11`, `Linux`, `linux390`, `or Solaris`). For example, run the following command on an AIX computer:
   ```
   ESync2300Aix.bin -console
   ```
   The following prompt is displayed:
   ```
   Press 1 for Next, 3 to Cancel or 4 to Redisplay [3]
   ```

3. Type 1 to start the installation and press Enter.

   The following prompt is displayed:

   ```
   Software Licensing Agreement:
   Press Enter to display the license agreement on your screen. Please
   read the agreement carefully before installing the Program. After
   reading the agreement, you will be given the opportunity to accept it
   or decline it. If you choose to decline the agreement, installation
   will not be completed and you will not be able to use the Program.
   ```
4. Press Enter to display the software license agreement.
5. Type 1 and press Enter to accept the license.

   The following prompt is displayed:

   ```
   Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
   ```
6. Type 1 and press Enter to continue.

   The following prompt is displayed:

   ```
   Name of configuration file [situpdate.conf]
   ```
7. Press Enter to use the default configuration file, `situpdate.conf`. If you want to use a different configuration file, type the name and press Enter.

   The following prompt is displayed:

   ```
   Number of seconds to sleep when no new situation updates [3]
   ```
8. Type the number of seconds that you want to use for the polling interval. The default value is 3, while the minimum value is 1. Press Enter.

   The following prompt is displayed:

   ```
   Number of bytes to use to save last event [50]
   ```
9. Type the number of bytes to use to save the last event and press Enter. The default and minimum value is 50.

   The following prompt is displayed:

   ```
   URL of the CMS SOAP server [cms/soap]
   ```
10. Type the URL for the monitoring server SOAP server and press Enter. The default value is cms/soap (which you can use if you set up your monitoring server using the defaults for SOAP server configuration).

    The following prompt is displayed:

    ```
    Rate for sending SOAP requests to CMS from TEC via Web Services [10]
    ```
11. Supply the maximum number of event updates to send to the monitoring server at one time and press Enter. The default and minimum value is 10.

    The following prompt is displayed:

    ```
    Level of debug for log

    [x] 1 low
    [ ] 2 med
    [ ] 3 verbose

    To select an item enter its number, or enter 0 when you are finished: [0]
    ```
12. Type the level of debugging that you want to use and press Enter. The default value is Low, indicated by an `x` next to `Low`.
13. Type 0 when you have finished and press Enter.

    The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```
14. Type 1 and press Enter to continue.

    The following prompt is displayed:

    ```
    Maximum size of any single cache file, in bytes [50000]
    ```

15. Type the maximum size, in bytes, for the cache file and press Enter. The default value is 50000. Do not use commas (,) when specifying this value.

    The following prompt is displayed:

    ```
    Maximum number of cache files [10]
    ```

16. Type the maximum number of cache files to have at one time and press Enter. The default value is 10, while the minimum is 2.

    On Windows, the following prompt is displayed:

    ```
    Directory for cache files to reside [C:/tmp/TME/TEC/OM_TEC/persistence]
    ```

    On UNIX, the following prompt is displayed:

    ```
    Directory for cache files to reside [/var/TME/TEC/OM_TEC/persistence]
    ```

17. Type the directory for the cache files and press Enter. The default directory on Windows is C:\tmp\TME\TEC\OM_TEC\persistence; on UNIX, /var/TME/TEC/OM_TEC/persistence.

    The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```

18. Type `1` and press Enter to continue.
19. The following prompt is displayed:

    ```
    --- Tivoli Enterprise Monitoring Server 1 ---

    Host name []
    ```

    Type the fully qualified host name for the computer where the monitoring server is running. This name must match the information that is in events issued by this monitoring server. Press Enter.

    The following prompt is displayed:

    ```
    User ID []
    ```

20. Type the user ID to use to access the computer where the monitoring server is running and press Enter.

    The following prompt is displayed:

    ```
    Password:
    ```

21. Type the password to access the computer and press Enter.

    The following prompt is displayed:

    ```
    Confirmation:
    ```

22. Type the password again to confirm it and press Enter.

    The following prompt is displayed:

    ```
    --- Tivoli Enterprise Monitoring Server 2 ---

    Host name []
    ```

23. Repeat steps 19 to 22 for each monitoring server for which you want to receive events on this event server.

    When you have provided information for all the monitoring servers *and* you specified information for less than 10 monitoring servers, press Enter to move through the remaining fields defining additional monitoring servers. Do not specify any additional monitoring server information.

    The following prompt is displayed:

    ```
    [X] 1 – Automatically install rules and classes (recommended)
    [ ] 2 – Manually install rules and classes (advanced users)

    To select an item enter its number, or 0 when you are finished: [0]
    ```

24. If you want to have the installer automatically install the rules and classes, enter `1` and continue with step 25. If you manually install the rules and classes, enter `2` and proceed to step 35.
25. When you see the following prompt, type `1` and press Enter to continue:

    ```
    Press 1 for Next, 2 for Previous, 3 to cancel or 4 to Redisplay [1]
    ```

The following prompt is displayed:

```
[x] 1 - Create a new rulebase (name and path required)
[ ] 2 - Use Existing Rulebase (path is optional)

To select an item, enter its number, or press 0 when you are finished: [0]
```

26. Type 1 to create a new rule base or 2 to use an existing rule base. Press Enter.
27. Type 0 when you are finished and press Enter.
28. If you are creating a new rule base, the following prompt is displayed:

    ```
    Rulebase Name []
    ```

    type the name for the rule base and press Enter.

    The following prompt is displayed:

    ```
    Rulebase Path []
    ```

29. If you are creating a new rule base, type the path for the new rule base and press Enter.
30. If you are using an existing rule base, the following prompt is displayed:

    ```
    Rulebase Name []
    ```

    Type the name of the rule base and press Enter.
31. If you are creating a new rule base, the following prompt is displayed:

    ```
    Existing rulebase name to import: []
    ```

    If you want to import an existing rule base into the new rule base, type the name of the existing rule base and press Enter.

    The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```

32. Type 1 and press Enter to continue.

    The following prompt is displayed:

    ```
    You indicated on the previous dialog that you want to modify an existing
    rule base. If you want this installer to back up the existing rule base
     before modifying it please provide a backup rule base name. Leave backup
     rule base name blank and click Next if you do not want a backup made.

    Backup rule base name. []
    ```

33. Type the name for the backup rule base and press Enter to continue. If you do not want the installer to back up the existing rule base, press Enter without providing a backup rule base name.

    The following prompt is displayed:

    ```
    If you have provided a backup rule base name you must provide a backup
    rule base path. NOTE: We append the backup rule base name to the backup
    rule base path for clarity and easy lookup.

    Backup rule base path. []
    ```

34. Type the path for the backup rule base and press Enter to continue. If you did not provide a name for a backup rule base, press Enter without providing a rule base path.

    The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
    ```

35. Type 1 and press Enter to continue.

    The event synchronization is installed.

    The following prompt is displayed:

    ```
    Installation and Configuration has completed. Please stop and restart the
    Tivoli Enterprise Console Server.

    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```

36. Type 1 and press Enter to continue.

    The following prompt is displayed:

    ```
    Installation and configuration has completed.
    Please restart the Tivoli Enterprise Console server for the changes
    to take effect.
    Mark appropriately below to restart the Tivoli Enterprise Console
    server.
    [ ] 1 - Restart the Tivoli Enterprise Console server to make changes
    effective
    To select an item enter its number, or 0 when you are finished: [0]
    ```

    The option to automatically restart the Tivoli Enterprise Console is presented only if you chose to have the installer automatically update the rules and classes.
37. If you want the installer to stop and restart the Tivoli Enterprise Console server, type 1 and press Enter. If you want to stop and restart yourself, type 0 and press Enter to continue. The following prompt is displayed:

    ```
    Press 3 to Finish, or 4 to Redisplay [1]
    ```
38. Type `3` to finish and press Enter.

Perform the following tasks after the installation is finished:

- If you did not have the installer do it, stop and restart the event server for the configuration changes to take effect.
- Install the monitoring agent .baroc files on the event server as described in “Installing monitoring agent .baroc files on the event server” on page 661.
- Configure the monitoring server to forward events to the event server as described in “Configuring your monitoring server to forward events” on page 662.
- If you did not choose to have the rule base updated automatically, update the rule base as described in “Manually importing the event synchronization class files and rule set” on page 659.

Installing from the command-line using a silent installation

Use the following steps to install the event synchronization using a silent installation from the command-line on your event server. This installation method runs silently, so you will not see status messages during the actual installation.

1. Change to the `tec` directory of the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
2. Run the following command to generate the configuration file:

   On Windows:

   ```
   ESync2300Win32.exe -options-template filename
   ```

   where *filename* is the name of the configuration file to create, for example, `es_silentinstall.conf`.

   On UNIX:

   ```
   ESync2300operating_system.bin -options-template filename
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, `or Solaris`). For example, run the following command on an AIX computer:

   ```
   ESync2300Aix.bin -options-template filename
   ```
3. Edit the output file to specify the values shown in Table 127 on page 657.

   Notes:

   a. Remove the pound signs (###) from the beginning of any value that you want to specify.

   b. Do not enclose any values in quotation marks (").

   c. You must specify the following values:

      - **configInfoPanel2.fileLocn**

- Information for at least one monitoring server (the **cmdSvrsPnlNotGuiMode.hostname1**, **cmdSvrsPnlNotGuiMode.userID1**, **cmdSvrsPnlNotGuiMode.pswd1**, and **cmdSvrsPnlNotGuiMode.retypePswd1** values)
- **rulebasePanel.chooseNewOrExistingRB**
- **rulebasePanel.rbName**
- **rbInstallTypePanel.rbInstallType**

If you are creating a new rule base, **rulebasePanel.rbPath** is also required. If "manual" is specified for rbInstallTypePanel.rbInstallType, all the other rulebasePanel.* options are ignored.

If you do not specify any of the other values, the default values are used.

d. The following values are specified only when installing event synchronization on a Netcool/Omnibus ObjectServer:

```
### -P installLocation
### -W configInfoPanel3.filesize
### -W configInfoPanel3.fileNumber
### -W configInfoPanel3.fileLocn
```

e. If you specify values, ensure that the value you specify meets the minimum required values. Otherwise, the installation stops and an error is written to the log file.

Table 127. IBM Tivoli Enterprise Console event synchronization configuration values

Value	Description
installLocation	The directory where you want the product to be installed. If the directory path contains spaces, enclose it in double quotes (" "). For example, to install the product to C:\Program Files\My Product, use `-P installLocation="C:\Program Files \My Product"` Any specified value is ignored if an event server is detected.
configInfoPanel.filename	The name of the file where event synchronization configuration information is stored. The default file name is situpdate.conf.
configInfoPanel.pollingInt	The polling interval, in seconds. The minimum value is 1, while the default value is 3. If there are no situation events, the process that forwards events to IBM Tivoli Enterprise Console rests for 3 seconds.
configInfoPanel.crcByteCnt	Number of bytes that the long running process will use when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) value is 50.
configInfoPanel.cmsSoapURL	The URL for the SOAP Server configured on the computer where the monitoring server is running. The default value is cms/soap. This value is used to create the URL to which IBM Tivoli Enterprise Console sends event information. For example, http://*hostname*:*port*/// cms/soap, where *hostname* is the host name of the monitoring server and *port* is the port.
configInfoPanel.bufFlushRate	The maximum number of event updates sent to the monitoring server at one time. The minimum (and default) value is 10 events.

Table 127. IBM Tivoli Enterprise Console event synchronization configuration values (continued)

Value	Description
configInfoPanel.logLevel	The level of debug information for event synchronization that is logged. You have the following choices: • Low (default) • Medium • Verbose
configInfoPanel2.filesize	The maximum permitted size, in bytes, for any one event cache file. The minimum (and default) value is 50000. Do not use commas when specifying this value (50,000 instead of 50000).
configInfoPanel2.fileNumber	The maximum number of event caches files at any given time. The minimum value is 2, while the default value is 10. When this value is reached, the oldest file is deleted to make room for a new file.
configInfoPanel2.fileLocn	The location where event cache files are located. The default locations are as follows: • On Windows: C:\tmp\TME\TEC\OM_TEC\persistence. • On UNIX: /var/TME/TEC/OM_TEC/persistence
cmsSvrsPnlNotGuiMode.hostname*#* **Note:** The pound sign (#) stands for a number between 1 and 10. For example, "hostname1".	The host name of each monitoring server that will send events to the event server. Specify up to 10 monitoring servers.
cmsSvrsPnlNotGuiMode.userID*#*	The user ID for the monitoring server, identified in hostname*#*, to use to access the computer where the monitoring server is running.
cmsSvrsPnlNotGuiMode.pswd*#*	The password for the user ID used to access the computer where the monitoring server is running.
cmsSvrsPnlNotGuiMode.retypePswd*#*	The password confirmation for the user ID.
rbInstallTypePanel.rbInstallType	Specifies whether the installer will automatically update a specified rule base, or if a rule base must be manually modified after installation is complete. Specify either automatic or manual. If manual is specified, all other rulebasePanel.* options are ignored.
rulebasePanel.chooseNewOrExistingRB	Specifies whether you are going to create a new rule base or use an existing rule base. Specify either `new` or `existing`.
rulebasePanel.rbName	The name of the rule base (existing or new).
rulebasePanel.rbPath	The path for the new rule base. There is no default location. You must specify a path.
rulebasePanel.fromRB	If you are creating a new rule base, identify any existing rule bases that you want to import into the new rule base.
bckupERB.backupName	If you want the install to back up an existing rule base before modifying it, specify the name of the backup rule base.
rulebasePanel.backupPath	Specify the directory where the backup rule base should be created.
restartTECQ.restartTEC	Specify whether or not the installer should recycle the event server to implement the changes. Specify `Yes` to have the installer restart the server; specify `No` if you want to restart the server yourself after installation is completed.

4. Save the file.
5. Run the following command:

   On Windows:

   ```
   ESync2300Win32.exe -options filename -silent
   ```

   where *filename* is the name of your configuration file.

   On UNIX:

   ```
   ESync2300operating_system.bin -options filename -silent
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, `Solaris`). For example, on AIX, run the following command:

   ```
   ESync2300Aix.bin -options filename -silent
   ```

You must stop and restart the event server for these changes to take effect.

When installation is complete, the results are written to the itm_tec_event_sync_install.log file. On UNIX, this log file is always created in the /tmp directory. For Windows, this file is created in the directory defined by the %TEMP% environment variable. To determine where this directory is defined for the current command-line window, run the following command:

```
echo %TEMP%
```

If you specified the monitoring servers in the silent installation configuration file, you might consider deleting that file after installation, for security reasons. The passwords specified in the files are not encrypted.

If you want to define additional monitoring servers (in addition to the one required monitoring server), run the sitconfsvruser.sh command as described in "Defining additional monitoring servers to the event server" on page 664. Repeat this command for each monitoring server.

If you specified your monitoring servers after the installation, you must stop and restart the Situation Update Forwarder process manually. See "Starting and stopping the Situation Update Forwarder process" on page 664 for information.

Perform the following tasks after the installation is finished:

- Install the monitoring agent .baroc files on the event server as described in "Installing monitoring agent .baroc files on the event server" on page 661.
- Configure the monitoring server to forward events to the event server as described in "Configuring your monitoring server to forward events" on page 662.
- If you did not choose to have the rule base updated automatically, update the rule base as described in "Manually importing the event synchronization class files and rule set."

Manually importing the event synchronization class files and rule set

If you do not want to permit the installation program to modify your rule base, you can choose the manual rule base modification option during the installation and then use one of the following methods to manually modify your rule base:

- "Creating a new rule base" on page 660
- "Creating a new rule base and importing an existing rule base into it" on page 660
- "Modifying an existing rule base" on page 661

Before you can run any of the commands in the following sections, you must source your Tivoli environment by running the following command:

On Windows, run the following command from a command prompt:

```
C:\Windows\system32\drivers\etc\Tivoli\setup_env.cmd
```

On Linux or UNIX, run the following command from a shell environment:

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

See the *IBM Tivoli Enterprise Console Command and Task Reference* for more information about the **wrb**, **wstopesvr**, and **wstartesvr** commands.

Creating a new rule base

Use the following steps to create a new rule base after you install the event synchronization component:

1. Create the new rule base by running the following command:

   ```
   wrb -crtrb -path newrb_path newrb_name
   ```

   where *newrb_path* is the path to where you want to create the new rule base, and *newrb_name* is the name for the new rule base.
2. Import the event synchronization class and rule files into the new rule base from the `$BINDIR/TME/TEC/OM_TEC/rules` directory created during the installation of the event synchronization component. Run the following commands:

   ```
   wrb -imprbclass path_to_Sentry_baroc_file newrb_name

   wrb -imprbclass path_to_omegamon_baroc_file newrb_name

   wrb -imprbrule path_to_omegamon_rls_file newrb_name

   wrb -imptgtrule omegamon EventServer newrb_name
   ```
3. Compile and load the new rule base by running the following commands:

   ```
   wrb -comprules newrb_name
   wrb -loadrb newrb_name
   ```
4. Stop and restart the event server by running the following commands:

   ```
   wstopesvr
   wstartesvr
   ```

Creating a new rule base and importing an existing rule base into it

Use the following steps to create a new rule base and import an existing rule base into it:

1. Create the new rule base by running the following command:

   ```
   wrb -crtrb -path newrb_path newrb_name
   ```

   where *newrb_path* is the path to where you want to create the new rule base and *newrb_name* is the name for the new rule base.
2. Import the existing rule base into the new rule base by running the following commands:

   ```
   wrb -cprb -overwrite existing_rbname newrb_name
   ```

   where *existing_rbname* is the name of the existing rule base that you want to import.
3. If the existing rule base is an older rule base, you must upgrade the tec.baroc file to include the TEC_Generic class. Run the following command:

   ```
   perl $BINDIR/TME/TEC/OM_TEC/bin/upg_tec_baroc.pl newrb_name
   ```
4. If the rule base already contains a Sentry.baroc file, you must upgrade it with the event synchronization event class attributes. Run the following command:

   ```
   perl $BINDIR/TME/TEC/OM_TEC/bin/upg_sentry_baroc.pl
   ```
5. If the rule base does not contain a Sentry.baroc file, you must import it from the $BINDIR/TME/TEC/OM_TEC/rules directory created during event synchronization installation. Run the following command:

   ```
   wrb -imprbclass path_to_Sentry_baroc_file newrb_name
   ```

6. Import the omegamon.baroc and rules file into the rule base from the $BINDIR/TME/TEC/OM_TEC/rules directory created during event synchronization installation. Run the following commands:

   ```
   wrb -imprbclass path_to_omegamon_baroc_file newrb_name
   wrb -imprbrule path_to_omegamon_rls_file newrb_name
   wrb -imptgtrule omegamon EventServer newrb_name
   ```

7. Compile and load the new rule base by running the following commands:

   ```
   wrb -comprules newrb_name
   wrb -loadrb newrb_name
   ```

8. Stop and restart the event server by running the following commands:

   ```
   wstopesvr
   wstartesvr
   ```

Modifying an existing rule base

Use the following steps to modify an existing rule base to include the class files and rule set for the event synchronization component:

1. If the existing rule base is an older rule base, you must upgrade the tec.baroc file to include the TEC_Generic class. Run the following command:

   ```
   perl $BINDIR/TME/TEC/OM_TEC/bin/upg_tec_baroc.pl newrb_name
   ```

2. If the rule base already contains a Sentry.baroc file, you must upgrade it with the event synchronization event class attributes. Run the following command:

   ```
   perl $BINDIR/TME/TEC/OM_TEC/bin/upg_sentry_baroc.pl
   ```

3. If the rule base does not contain a Sentry.baroc file, you must import it from the $BINDIR/TME/TEC/OM_TEC/rules directory created during event synchronization installation. Run the following command:

   ```
   wrb -imprbclass path_to_Sentry_baroc_file newrb_name
   ```

4. Import the omegamon.baroc and rules file into the rule base from the $BINDIR/TME/TEC/OM_TEC/rules directory created during event synchronization installation. Run the following commands:

   ```
   wrb -imprbclass path_to_omegamon_baroc_file newrb_name
   wrb -imprbrule path_to_omegamon_rls_file newrb_name
   wrb -imptgtrule omegamon EventServer newrb_name
   ```

5. Compile and load the new rule base by running the following commands:

   ```
   wrb -comprules newrb_name
   wrb -loadrb newrb_name
   ```

6. Stop and restart the event server by running the following commands:

   ```
   wstopesvr
   wstartesvr
   ```

Installing monitoring agent .baroc files on the event server

The monitoring server generates Tivoli Enterprise Console events with classes that are unique to each monitoring agent. Each monitoring agent provides a .baroc file with the Tivoli Enterprise Console classes that are generated by IBM Tivoli Monitoring. In order to view this event data in the event console, you must install these monitoring agent .baroc files on the event server.

After you have added application support for each agent to the monitoring server, the monitoring agent .baroc files are located in the following directory:

- On Windows, in the *itm_installdir*\cms\TECLIB directory, where *itm_installdir* is the directory where you installed IBM Tivoli Monitoring.
- On Linux and UNIX, in the *itm_installdir*/tables/*ms_name*/TECLIB directory, where *itm_installdir* is the directory where you installed IBM Tivoli Monitoring and *ms_name* is the name of the monitoring server.
- z/OS users, see the *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS* guide.

Use the following steps to install the monitoring agent .baroc files on the event server:

1. Copy the monitoring agent .baroc files from the computer where the monitoring server is installed to a temporary directory on the event server computer (for example, /tmp). The location of the agent .baroc files is described above. Do not copy the om_tec.baroc file; this file contains classes that are duplicates of classes in the omegamon.baroc file.
2. Set up the Tivoli Management Framework environment by running the following command:

   On Windows, run the following command:

   ```
   C:\WINDOWS\system32\drivers\etc\Tivoli\setup_env.cmd
   ```

   On Linux and UNIX, run the following command from a shell environment:

   ```
   . /etc/Tivoli/setup_env.sh
   ```
3. For each monitoring agent .baroc file to load into the rule base, run the following command from the same command prompt:

   ```
   wrb -imprbclass /tmp/agent_baroc_file rb_name
   ```

   where:

   /tmp/*agent_baroc_file*
   : Specifies the location and name of the monitoring agent .baroc file. The example above uses the /tmp directory as the location.

   rb_name
   : Is the name of the rule base that you are using for event synchronization.
4. Compile and load the rule base by running the following commands

   ```
   wrb -comprules rb_name
   wrb -loadrb rb_name
   ```
5. Stop and restart the event server by running the following commands:

   ```
   wstopesvr
   wstartesvr
   ```

When you have loaded each of the agent .baroc files into the rule base and restarted the event server, the event server is ready to receive and correctly parse any events it receives from the monitoring server from one of the installed monitoring agents.

See the *IBM Tivoli Enterprise Console Command and Task Reference* for more information about the **wrb**, **wstopesvr**, and **wstartesvr** commands.

Configuring your monitoring server to forward events

Before the monitoring server forwards any situation events to Tivoli Enterprise Console, you have to enable that forwarding of events. Use the following steps to enable event forwarding on your monitoring server.

Note: z/OS users, see the *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS* guide.

For Windows monitoring servers *only*, do the following:

1. Open Manage Tivoli Enterprise Monitoring Services.
2. Right-click the monitoring server and click **Reconfigure**.
3. On the configuration options window, select **Tivoli Event Integration Facility**.
4. Click **OK** and **OK**.
5. Complete the following fields on the Event Server: Location and Port Number window and click **OK**:

   Server or EIF Probe Location
   : Type the host name or IP address for the computer where the IBM Tivoli Enterprise Console event server is installed.

Port Number

Type the port number for the event server. If the event server is using port mapping, set this value to 0. If the event server was configured to use a specific port number, specify that number.

To determine the port number that the event server is using, search for the `tec_recv_agent_port` parameter in the .tec_config file in the *$BINDIR*/TME/TEC directory on the event server. If the parameter is commented out with a pound sign (#), the event server is using port mapping. If it is not, the event server is using the port number specified by this parameter.

For Linux and UNIX monitoring servers: You configured the TEC Server and TEC Port information for the Linux/UNIX monitoring server during installation, if you installed the monitoring server using the configuration instructions in this installation guide. However, if you did not configure this information, see "Configuring the hub monitoring server" on page 215 for the procedure.

Note: If you already have policies that contain emitter activities that send events to the Tivoli Enterprise Console, turning on Tivoli Event Integration event forwarding will result in duplicate events. You can deactivate the emitter activities within policies so you do not have to modify all your policies when you activate Tivoli Event Integration Facility forwarding by specifying **Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding** when you enable forwarding using the Event Integration Facility.

Using policies gives you more control over which events are sent and you may not want to lose this granularity. Moreover, it is likely the policies that are invoking the TEC emitter are doing little else. If you deactivate these activities, there is no point in running the policy. You may delete policies that are longer required, instead of disabling them. Note that events forwarded via the TEC event emitter are not eligible for event synchronization (that is, changes to these events on the TEC side will not be sent back to the monitoring server).

Controlling event forwarding

The `om_tec.config` file controls the forwarding of events to IBM Tivoli Enterprise Console. It contains this parameter:

`BufferFlushRate=`*`events_per_minute`*

Specifies the number of events that are sent per minute when the adapter has reestablished its connection to Tivoli Enterprise Console. After the adapter has recovered the lost connection, if there are events in the buffer, the events are sent at this rate per minute. The default value is 0, meaning all events are sent in one burst.

If your environment can have a large number of open situation events, you may want to adjust this parameter to control the rate at which events are sent to the event server. To edit this file and change this parameter:

- On Windows:
  1. Open Manage Tivoli Enterprise Monitoring Services.
  2. Right-click Tivoli Enterprise Monitoring Server, and click Advanced → Edit EIF Configuration.
  3. Once you have completed your reconfiguration, recycle the Tivoli Enterprise Portal Server and each Tivoli Enterprise Portal client.
- On Linux or UNIX:
  1. Edit file *`install_dir`*`/tables/`*`hostname`*`/TECLIB/ om_tec.config`, where *install_dir* is your Tivoli Monitoring installation directory and *hostname* is the name of the host running this monitoring server.
  2. Once you have saved your configuration updates, recycle the Tivoli Enterprise Portal Server and each Tivoli Enterprise Portal client.

Starting and stopping the Situation Update Forwarder process

To send event updates to a monitoring server, you must start the Situation Update Forwarder. This process is started automatically when the event server starts. To start the process manually, change to the `*$BINDIR*/TME/TEC/OM_TEC/bin` directory (where *$BINDIR* is the location of the Tivoli Management Framework installation) and run the following command:

On Windows:

```
startSUF.cmd
```

On UNIX:

```
startSUF.sh
```

To stop the process, run the following command:

On Windows:

```
stopSUF.cmd
```

On UNIX:

```
stopSUF.sh
```

On Windows, you can also start and stop the Tivoli Situation Update Forwarder service to start or stop the forwarding of event updates. You can start and stop this service either from the Windows Service Manager utility or with the following commands:

```
net start situpdate
net stop situpdate
```

Changing the configuration of the event synchronization component on the event server

If you want to change any of the settings for the event synchronization on the event server, use the **sitconfig.sh** command. You have two options for running this command:

- Manually modify the configuration file for event synchronization (named situpdate.conf by default and located in the /etc/TME/TEC/OM_TEC directory on UNIX and Linux, and the %SystemDrive%\Program Files\TME\TEC\OM_TEC\etc directory on Windows), and then run the following command:

  ```
  sitconfig.sh update filename=config_filename
  ```

- Run the **sitconfig.sh** command directly, specifying only those settings that you want to change. See the *IBM Tivoli Monitoring: Command Reference* for the full syntax of this command.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process. See "Starting and stopping the Situation Update Forwarder process" for information.

Defining additional monitoring servers to the event server

To add additional monitoring servers to the list that can receive event status updates from Tivoli Enterprise Console, run the sitconfsvruser.sh command as follows:

1. Source your Tivoli environment by running the following command:
   - On Windows, run the following command from a command prompt:

     ```
     C:\Windows\system32\drivers\etc\Tivoli\setup_env.cmd
     ```

   - On operating systems like UNIX and Linux, run the following command from a shell environment:

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

2. On Windows, type BASH to invoke the bash shell.

3. Change to the *$BINDIR*/TME/TEC/OM_TEC/bin directory (where *$BINDIR* is the location of the Tivoli Management Framework installation) and enter the following command:

   ```
   sitconfsvruser.sh add serverid=server userid=user password=password
   ```

   where:

   server Is the fully qualified host name of the monitoring server.

   user Is the user ID to access the computer where the monitoring server is running.

   password
   Is the password to access the computer.

Repeat this command for each monitoring server.

You can also delete monitoring servers. See the *IBM Tivoli Monitoring: Command Reference* for the full syntax of this command.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process. See "Starting and stopping the Situation Update Forwarder process" on page 664 for information.

Customizing event status processing behavior when agent switching is used or the agent goes offline

The variables in Table 128 on page 666 can be added to the monitoring server's environment file to customize the behavior of event status processing when agent switching is used or when the agent goes offline. The first two variables help ensure that events are not closed by the agent's primary monitoring server after the agent has switched to its secondary monitoring server. You can find the monitoring server's environment file in these locations:

- On Windows systems:

  ```
  ITM_HOME\cms\KBBENV
  For example: C:\IBM\ITM\cms\KBBENV
  ```

- On Linux/UNIX systems:

  ```
  ITM_HOME/config/tems_hostname_ms_tems_name.config
  For example: /opt/IBM/ITM/config/edinburg_ms_labtems.config
  ```

  For Linux/UNIX systems, you must add the variables to the `.config` and the `.ini` files. The name and location of the `.ini` file is `ITM_HOME/config/ms.ini`.

- On z/OS systems:

  ```
  &shilev.&rtename.RKANPARU(KDSENV)
  For example: ITM.SYP1.RKANPARU(KDSENV)
  ```

  Note: The `&shilev` and `&rtename` are variables that correspond to high level qualifiers of the RKANPARU(KDSENV) partitioned dataset. These variables can take 1 to 8 characters.

You must recycle the Tivoli Enterprise Monitoring Server after modifying the environment file for your changes to be picked up.

Table 128. Variables to customize the behavior of event status processing when agent switching is used.

Variable	Architecture Type	Details	Administrator
IRA_MIN_NO_DATA_WAIT_TIME	Unidirectional and bidirectional	The minimum time to wait before the monitoring server closes a situation event. This parameter is defined in number of seconds. The default value is zero. By default, after an agent is disconnected from a Tivoli Enterprise Monitoring Server, situations already open will remain open for three situation polling intervals. For example, take two sampled situations, S1 and S2, with intervals of 30 seconds and 15 minutes respectively. Both situations are open when the agent loses connection. Situation S1 closes after at least one minute and 30 seconds. Situation S2 closes after at least 45 minutes. With agent switching, if a situation closes too soon it might generate duplicate events because the agent did not have sufficient time to connect to the backup monitoring server before the primary server closes the original event. This is particularly true for situations with very short polling intervals. In such a scenario you can use the **IRA_MIN_NO_DATA_WAIT_TIME** variable to set the minimum wait time before a situation is closed. Using the example above, if IRA_MIN_NO_DATA_WAIT_TIME is set to 600 (5 minutes), S1 will close after 5 minutes not 90 seconds. S2 is unaffected and will close after 45 minutes as before. **Note:** You should set this variable in the environment file for all of your monitoring servers.	IBM Tivoli Monitoring

Table 128. Variables to customize the behavior of event status processing when agent switching is used. (continued)

Variable	Architecture Type	Details	Administrator
CMS_SIT_TIME_VALIDATION	Unidirectional and bidirectional	Valid entries are Y or N. The default is N. By default, the monitoring server handles situation events on a first-come-first-serve basis. In a scenario where agent switching is enabled an agent might send events through two different monitoring servers. The events that arrive first might not necessarily be the earlier events if one of the monitoring servers encountered connection issues. This generally has little impact on situation event processing, except when a monitoring server is shutdown and some situations might be closed prematurely even though the agent is already connected to a different monitoring server. You must perform two actions to avoid this scenario: 1. All monitoring server hosts should synchronize time, preferable through Internet Time Protocol (ITP) clients. 2. You should add the **CMS_SIT_TIME_VALIDATION=Y** variable to all monitoring server environment files. This switches the LCLTMSTMP column in situation events to use UTC time instead of local time, which is then used to determine event order.	IBM Tivoli Monitoring

Table 128. Variables to customize the behavior of event status processing when agent switching is used. (continued)

Variable	Architecture Type	Details	Administrator
CMS_SIT_CHECK_NODESTS	Unidirectional and bidirectional	Valid entries are Y or N. The default is N. This variable is only applicable for users of event integration with Tivoli Enterprise Console and Omnibus. When the **CMS_SIT_CHECK_NODESTS** variable is set to Y in the environment file, the hub monitoring servers check the agent status whenever a close status update event is forwarded to Netcool/Omnibus. The CMS_SIT_CHECK_NODESTS variable should only be added to the hub monitoring server environment file. If the agent is offline the close status update event is tagged with a special *OFFLINE* indicator in the `situation_eventdata` EIF slot. If you do not want events to be closed in Netcool/OMNIbus when an agent goes offline, you can customize the EIF probe rules to ignore close events where the situation_status slot is set to **N** and the situation_eventdata EIF slot is set to **OFFLINE**. See “Customizing the rules file” on page 742 for details on how to add customizations to the EIF probe rules. You should also consider setting the IRA_MIN_NO_DATA_WAIT_TIME environment variables described in this table so that close status events are not sent to Netcool/OMNIbus until after the agent offline condition has been detected. (If the default agent heartbeat interval is used, it can take between 10 to 20 minutes before the monitoring server detects that the agent is no longer online.) **Note:** You should only set the CMS_SIT_CHECK_NODESTS variable in the environment file of your hub monitoring server.	IBM Tivoli Monitoring

Changing the TCP/IP timeout setting on your event server

If the Situation Update Forwarder cannot reach a monitoring server to send an update, depending on the TCP/IP settings for the computer where your event server is running, you might have to wait up to 15 minutes before the Situation Update Forwarder tries to connect to the monitoring server again. This might occur if your event server is running on an AIX, Solaris, or HP-UX computer.

Use the following steps to change the TCP/IP timeout value for your computer.

On AIX, run the following command:

```
no -o tcp_keepinit=timeout_value
```

where *timeout_value* is the length of the timeout period, in half seconds. To configure a timeout of 30 seconds, set the *timeout_value* value to 60.

On Solaris and HP-UX, run the following command:

```
ndd -set /dev/tcp tcp_ip_abort_cinterval timeout_value
```

where *timeout_value* is the length of the timeout period, in milliseconds. To configure a timeout of 30 seconds, set the *timeout_value* value to 30000.

Upgrading to Tivoli Event Synchronization version 2.3.0.0

Upgrading replaces the omegamon.rls file. Any changes you have made to this file will be lost. However, a backup file named omegamon.rls.bac is created so you can recover your changes. You must load the noncurrent rule base and restart the Tivoli Enterprise Console event server after updating either the noncurrent or the current rule base.

Note: If you have already installed IBM Tivoli Monitoring and Tivoli Event Synchronization, you must upgrade to version 2.3.0.0.

You install the upgrade from the IBM Tivoli Monitoring installation media.

There are three methods for upgrading event synchronization:

- "Upgrading from a wizard"
- "Upgrading from the command-line" on page 670
- "Upgrading from the command-line using a silent installation" on page 672

If you have multiple rule bases that are using IBM Tivoli Monitoring and Tivoli Event Synchronization, you can run the upgrade installation to update each rule base. After you finish the first rule base, restart the upgrade installer and supply the targeted next rule base you want to update.

Note: You cannot uninstall just the upgrade. You must uninstall the entire product. For instructions on uninstalling, see "Uninstalling the event synchronization component" on page 862

Upgrading from a wizard

Use the following steps to upgrade event synchronization from the installation wizard:

1. On the host of the event server, launch the event synchronization upgrade installation:

   On Windows, double-click the `ESUpgrade23Win32.exe`file in the `tec` subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.

   On Linux or UNIX, change to the `tec` subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image and run the following command:

   ```
   ESUpgrade23operating_system.bin
   ```

   where *operating_system* is the operating system you are installing on (`aix`, `HP11`, `Linux`, `linux390`, `or Solaris`). For example, run the following command on an AIX computer:

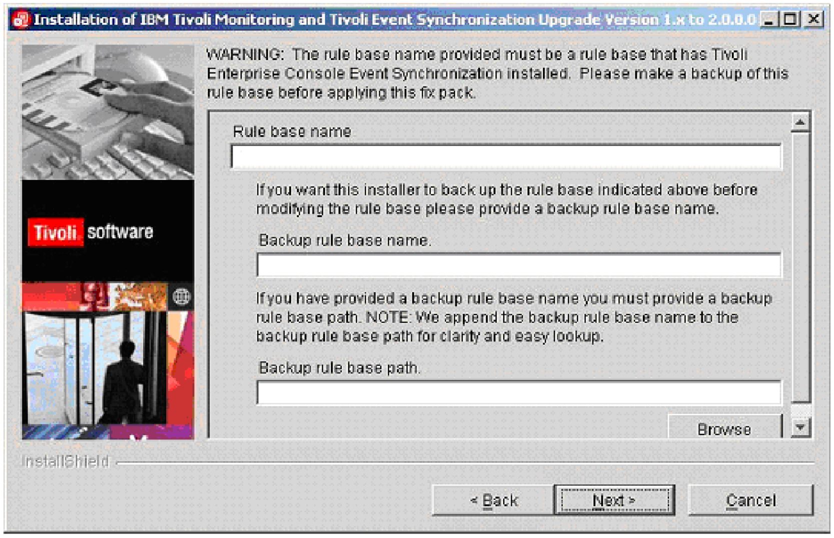
   ```
   ESUpgrade23Aix.bin
   ```

2. Click **Next** on the Welcome window.
3. Select **I accept the terms in the license agreement** and click **Next**.

   You are presented with the option of having the installer automatically update the rule base for you.

4. Select whether you want to have the installer update the rule base or whether you update it manually after the installation is complete, then click **Next**.

   If you chose to manually update the rule base, you will see a progress indicator for the installation. Proceed to step 9. If you chose to have the installer automatically update the rule base, the data collection window shown in Figure 161 is displayed:



Installation of IBM Tivoli Monitoring and Tivoli Event Synchronization Upgrade Version 1.x to 2.0.0.0

WARNING: The rule base name provided must be a rule base that has Tivoli Enterprise Console Event Synchronization installed. Please make a backup of this rule base before applying this fix pack.

Rule base name

If you want this installer to back up the rule base indicated above before modifying the rule base please provide a backup rule base name.

Backup rule base name.

If you have provided a backup rule base name you must provide a backup rule base path. NOTE: We append the backup rule base name to the backup rule base path for clarity and easy lookup.

Backup rule base path.

Figure 161. Upgrade data collection window

Proceed to step 5.

5. Specify the name of the rule base to be upgraded. The rule base must be one that has event synchronization previously installed.
6. If you want the installer to back up the rule base before it is modified, specify a name and a path for the backup rule base.
7. Click **Next** to continue.

   A window is displayed that summarizes the information you entered.
8. If the information is correct, click **Next** to proceed with the installation. If the information is not correct, click **Back** and correct the fields as necessary; then click **Next** and **Next** again to proceed.

   A progress indicator shows the progress of the installation and configuration.
9. When the installation completes successfully, you will see a message that reminds you to restart the TEC server. If the updated rule base is not the currently loaded rule base, you are reminded to load the rule base and restart the server. Click **OK** to dismiss the message.

   A window is displayed that reminds you to restart the Tivoli Enterprise Console server.
10. If you want the installer to restart the server for you, check Restart the Tivoli Enterprise Console server to make changes effective; then click **Next**. If you do not want the installer to restart the server, leave the option unchecked and click **Next**.
11. Click **Finish** to exit the installer.

Important:: If you chose the manual update option, you must copy the files in `$BINDIR/TME/TEC/OM_TEC/rules` directory to the rule base, recompile and reload the rule base, and restart the Tivoli Enterprise Console. See “Manually importing the event synchronization class files and rule set” on page 659 for the commands to use to do this.

Upgrading from the command-line

Use the following steps to upgrade event synchronization from the command-line on your event server:

1. Change to the `tec` directory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
2. Run the following command to launch the installation:

   On Windows:

   ```
   ESUpgrade23Win32.exe -console
   ```

On UNIX or Linux operating systems:

```
ESUpgrade23operating_system.bin -console
```

where *operating_system* is the operating system you are installing on (Aix, HP11, Linux, linux390, or Solaris). For example, run the following command on an AIX computer:

```
ESUpgrade23Aix.bin -console
```

The following prompt is displayed:

```
Press 1 for Next, 3 to Cancel or 4 to Redisplay [1]
```

3. Type 1 to start the installation and press Enter.

   The following prompt is displayed:

   ```
   Software Licensing Agreement:
   Press Enter to display the license agreement on your screen. Please
   read the agreement carefully before installing the Program. After
   reading the agreement, you will be given the opportunity to accept it
   or decline it. If you choose to decline the agreement, installation
   will not be completed and you will not be able to use the Program.
   ```

4. Press Enter to display the software license agreement.
5. Type 1 and press Enter to accept the license.

   The following prompt is displayed:

   ```
   [X] 1 - Automatically install rules and classes (recommended)
   [ ] 2 - Manually install rules and classes (advanced users)

   To select an item enter its number, or 0 when you are finished: [0]
   ```

6. If you want to have the installer automatically install the rules and classes, enter 1. The following prompt is displayed:

   ```
   Rule base Name []
   ```

   Continue with step 7. If you manually install the rules and classes, enter 2 and proceed to step 11.
7. Type 1 and press Enter to continue.
8. Type the name of the rule base to upgrade then press Enter.

   The rule base must be one in which event synchronization was previously installed.

   The following prompt is displayed:

   ```
   Backup rule base name. []
   ```

9. If you want the installer to back up the rule base before modifying it, type a name for the backup rule base, then press Enter. If you do not want the installer to create a backup rule base, press Enter without typing any name. If you provide a name for the backup rule base, you are prompted for the path for the backup rule base:

   ```
   If you have provided a backup rule base name you must provide a backup
   rule base path. NOTE: We append the backup rule base name to the
   backup rule base path for clarity and easy look-up.

   Backup rule base path. []
   ```

10. Type the path and press Enter. The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay []
    ```

11. Type 1 and press Enter to continue.

    The following prompt is displayed:

    ```
    IBM Tivoli Monitoring

    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```

12. Type 1 and press Enter to continue.

    The event synchronization is installed. The following prompt is displayed:

```
Installation and Configuration has completed. Please stop
and restart the Tivoli Enterprise Console Server.

Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
```

13. Type `1` and press Enter to continue.

    The following prompt is displayed:

    ```
    Installation and configuration has completed.
    Please restart the Tivoli Enterprise Console server for the changes
    to take effect.

    Mark appropriately below to restart the Tivoli Enterprise Console
    server.
    [ ] 1 - Restart the Tivoli Enterprise Console server to make changes
    effective

    To select an item enter its number, or 0 when you are finished: [0]
    ```

    If you did not choose to have the installer automatically update the rule base, you will not be offered the option to restart the Tivoli Enterprise Console automatically.
14. If you want the installer to stop and restart the Tivoli Enterprise Console server, type `1` and press Enter. If you want to stop and restart the console yourself, type `0` and press Enter. The following prompt is displayed:

    ```
    Press 3 to Finish, or 4 to Redisplay [1]
    ```
15. Type `3` to finish and press Enter.

You must stop and restart the event server for these changes to take effect.

Important: If you chose the manual update option, you must copy the files in `$BINDIR/TME/TEC/OM_TEC/rules` directory to the rule base, recompile and reload the rule base, and restart the Tivoli Enterprise Console. See “Manually importing the event synchronization class files and rule set” on page 659 for the commands to use to do this.

Upgrading from the command-line using a silent installation

If you have already installed Fix Pack 1, be aware that if you define the rulebasePanel.rbName, this silent installation will update the rule base. If you do not want to apply the updated rules file to a rule base leave the rulebasePanel.rbName commented out (do not delete the pound signs ### preceding the parameter).

Use the following steps to install the event synchronization using a silent installation from the command-line on your event server. This installation method runs silently, so you will not see status messages during the actual installation.

1. Change to the `tec` directory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
2. Run the following command to generate the configuration file:

   On Windows:

   ```
   ESUpgrade23Win32.exe -options-template filename
   ```

   where *filename* is the name of the configuration file to create, for example, `es_silentinstall.conf`.

   On UNIX:

   ```
   ESUpgrade23operating_system.bin -options-template filename
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, `or Solaris`). For example, run the following command on an AIX computer:

   ```
   ESUpgrade23Aix.bin -options-template filename
   ```
3. Edit the output file to specify the following values.

rbInstallTypePanel.rbInstallType
Specifies whether the installer will automatically update a specified rule base, or if a rule base must be manually modified after installation is complete. Specify either automatic or manual. If manual is specified, all other rulebasePanel.* options are ignored.

rulebasePanel.rbName
The name of the rule base to be updated.

rulebasePanel.backupName
The name of the backup rule base.

rulebasePanel.backupPath
The path for the backup rule base.

restartTEC.startTEC
Indicates that you want the installer to restart the Tivoli Enterprise Console event server.

Notes:

a. Remove the pound signs (###) from the beginning of any value that you want to specify.

b. Do not enclose any values in quotation marks (").

c. Specify the following value only if you want the indicated rule base updated:

```
rulebasePanel.rbName
```

d. If you specify values, ensure that the value you specify meets the minimum required values. Otherwise, the installation stops and an error is written to the log file.

4. Save the file.
5. Run the following command:

   On Windows:

   ```
   ESUpgrade23Win32.exe -options filename -silent
   ```

   where *filename* is the name of your configuration file.

   On UNIX:

   ```
   ESUpgrade23operating_system.bin -options filename -silent
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, `Solaris`). For example, on AIX, run the following command:

   ```
   ESUpgrade23Aix.bin -options filename -silent
   ```

The rule base that is updated during silent installation is made current.

Important: If you chose the manual update option, you must copy the files in `$BINDIR/TME/TEC/OM_TEC/rules` directory to the rule base, recompile and reload the rule base, and restart the Tivoli Enterprise Console. See “Manually importing the event synchronization class files and rule set” on page 659 for the commands to use to do this.

Chapter 26. Setting up event forwarding to Netcool/OMNIbus

If you are already using Netcool/OMNIbus to monitor events from other sources in your enterprise, you can also view and manage situation events from a hub Tivoli Enterprise Monitoring Server or monitoring agent in the Netcool/OMNIbus Event List user interfaces. Event integration requires Netcool/OMNIbus V7.2 or later and Netcool/OMNIbus Probe for Tivoli EIF version 10 or later. For all software prerequisites for event integration with Netcool/OMNIbus, see "Required software for event integration with Netcool/OMNIbus" on page 158.

Architecture overview

By using the Tivoli Enterprise Portal or `tacmd` commands, you can create monitoring specifications called *situations* to detect when specific conditions or events occur in your environment and raise an event that is sent to Netcool/OMNIbus. Each situation is assigned (or distributed) to one or more managed systems that are to be monitored for a specific condition or a set of conditions.

Two types of events might be triggered by a situation: pure or sampled. When the determination of the event must be made based on observations made at specific intervals, the event is known as a *sampled event*. When the event is based on a spontaneous occurrence, the event is known as a *pure event*. Therefore, situations for sampled events have an interval associated with them, while those for pure events do not. In sampled events, the condition that caused the event can change, causing it to be no longer true. When a situation condition becomes true for a sampled situation, an event with open status is sent to Netcool/OMNIbus. When a situation condition is no longer true for a sampled situation, a status update event with the closed status is sent to Netcool/OMNIbus so that the event is cleared. Because pure events represent a spontaneous occurrence, an event with open status is sent to Netcool/OMNIbus each time the situation condition is true. No status update event is sent when the situation condition is not true. Therefore, pure events can be left open indefinitely unless you close them by using one of the following options:

- Close them by using the Tivoli Enterprise Portal Situation Event Console.
- Clear them in Netcool/OMNIbus.
- Define a situation Until modifier condition that automatically closes the pure event if a configured time interval passes or another situation condition becomes true.

Event integration between a hub monitoring server (monitoring server) and Netcool/OMNIbus (event server) can be unidirectional or bidirectional. Events can also be sent directly to Netcool/OMNIbus from IBM Tivoli Monitoring agents by using either SNMP or EIF. OMNIbus operators can use the Tivoli Netcool/OMNIbus WebGUI or native desktop environment to view events. By using the Netcool/OMNIbus Event List UI support in the WebGUI and native desktop, operators can acknowledge events, view event journals, take ownership of events, and run event management tools.

In a *unidirectional* architecture, hub monitoring servers use the Tivoli Event Integration Facility (EIF) interface to forward situation events to OMNIbus. The events are received by the Netcool/OMNIbus probe for Tivoli EIF, which maps them to OMNIbus events and then inserts them into the OMNIbus ObjectServer. If the status of a situation event changes, the hub monitoring server also sends status update events to Netcool/OMNIbus. The unidirectional architecture is similar when an agent is configured to send events to Netcool/OMNIbus. The situation events of the agent are forwarded to a Netcool/OMNIbus probe for Tivoli EIF, or to the Netcool/OMNIbus SNMP probe that maps them to OMNIbus events. The agents also forward status update events to Netcool/OMNIbus when a sampled event condition is no longer true.

In a *bidirectional* architecture, when an OMNIbus operator acknowledges, deacknowledges, deletes, or clears a forwarded event, OMNIbus sends those status changes back to the hub monitoring server that forwarded them by using the IBM Tivoli Monitoring Situation Update Forwarder. (Severity changes, other than clearing an event, are not sent back to the hub monitoring server.) Bidirectional updates from

Netcool/OMNIbus are supported only for events that originate from the hub monitoring server. Bidirectional updates are not supported for situation events sent directly from monitoring agents to SNMP, or EIF to Netcool/OMNIbus probes.

Use the bidirectional architecture in the following scenarios:

- If you want your Tivoli Enterprise Portal operators to see the same event status as your Netcool/OMNIbus operators and to acknowledge or deacknowledge events and to close pure events.
- If you want your Netcool/OMNIbus operators to be notified that an event condition has not been resolved when a sampled event is cleared or deleted by using the Netcool/OMNIbus Event List UI or event automations.

If you use the unidirectional architecture, the following conditions apply:

- Your Tivoli Portal Enterprise operators should not acknowledge, deacknowledge or close events.
- Pure events that are cleared in Netcool/OMNIbus remain open in the hub Tivoli Enterprise Monitoring Server. As a result, you should have a situation Until modifier configured for your pure events to close them automatically.
- Sampled events that are cleared in Netcool/OMNIbus are not re-opened in Netcool/OMNIbus until IBM Tivoli Monitoring detects that the situation condition has become false and then true again.

The event integration also supports single-tier, multitier and high availability Netcool/OMNIbus architectures. In a single-tier architecture without high availability, there is a single Netcool/OMNIbus ObjectServer. In a multitier architecture, there are multiple sets of ObjectServers for scalability purposes. You can add high availability to each of these architectures by adding a primary and backup ObjectServer to each tier. You can also configure EIF probes for peer-to-peer failover mode. For more information on configuring the event integration in multitier and high-availability architectures, see “Netcool/OMNIbus Multitiered and High-availability Architecture” on page 709.

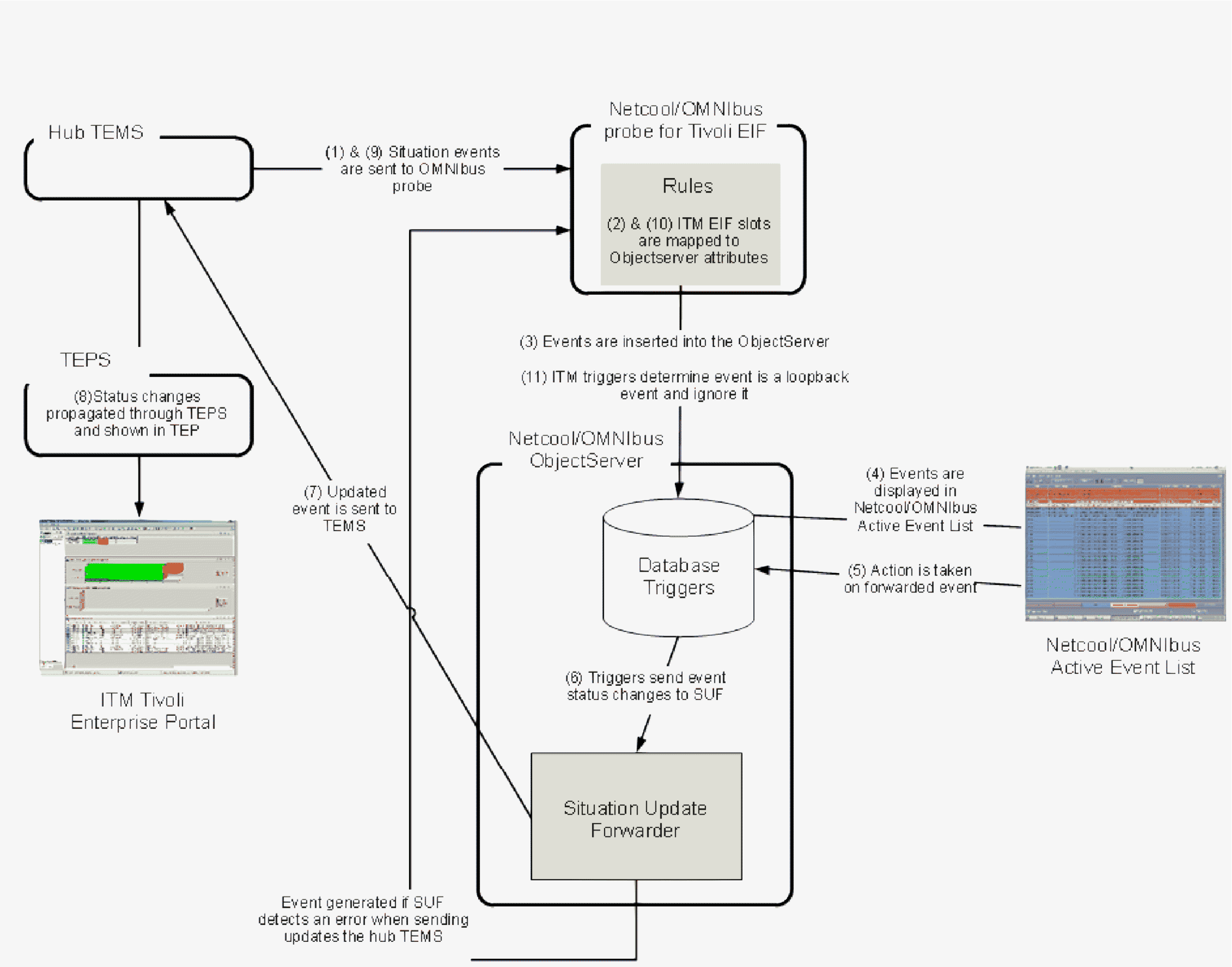

Hub TEMS
(1) & (9) Situation events are sent to OMNIbus probe
Netcool/OMNIbus probe for Tivoli EIF
Rules
(2) & (10) ITM EIF slots are mapped to Objectserver attributes
(3) Events are inserted into the ObjectServer
(11) ITM triggers determine event is a loopback event and ignore it
TEPS
(8)Status changes propagated through TEPS and shown in TEP
ITM Tivoli Enterprise Portal
(7) Updated event is sent to TEMS
Netcool/OMNIbus ObjectServer
Database Triggers
(4) Events are displayed in Netcool/OMNIbus Active Event List
(5) Action is taken on forwarded event
Netcool/OMNIbus Active Event List
(6) Triggers send event status changes to SUF
Situation Update Forwarder
Event generated if SUF detects an error when sending updates the hub TEMS

Figure 162. A typical event flow showing both IBM Tivoli Monitoring and Netcool/OMNIbus environments for a single-tier architecture

The following steps outline the event flow for a typical bidirectional or unidirectional flow between IBM Tivoli Monitoring and Netcool/OMNIbus, with actions taken by Netcool/OMNIbus operators. Steps 1 to 5 are common to both bidirectional and unidirectional event flows. Steps 6 to 11 are specific to bidirectional event flows only.

1. IBM Tivoli Monitoring generates situation events that are sent to the Netcool/OMNIbus probe for Tivoli EIF. These events are also displayed in the Tivoli Enterprise Portal.
2. The probe maps a subset of the IBM Tivoli Monitoring EIF slots to the Netcool/OMNIbus ObjectServer attributes and creates an OMNIbus event.
3. The OMNIbus events are inserted into the Netcool/OMNIbus ObjectServer by using IBM Tivoli Monitoring provided triggers.
4. Events in the Netcool/OMNIbus ObjectServer are displayed in the Netcool/OMNIbus Native Event List or WebGUI and in the Tivoli Enterprise Portal.
5. Forwarded events are acknowledged, deacknowledged, deleted, or cleared by Netcool/OMNIbus operators or by automation within OMNIbus or Impact. Event status can also be updated by an integrated trouble ticket system or by integration with another Event Server.
6. IBM Tivoli Monitoring triggers in the Netcool/OMNIbus ObjectServer database forward the event status changes to the IBM Tivoli Monitoring Situation Update Forwarder.
7. The IBM Tivoli Monitoring Situation Update Forwarder sends SOAP requests to the hub monitoring server.

8. Status changes are propagated through the Tivoli Enterprise Portal Server and shown in the Tivoli Enterprise Portal. The complete list of open, acknowledged, and deacknowledged events is shown in the Situation Event Console workspace of the Tivoli Enterprise Portal.
9. The event status change is also sent back to Netcool/OMNIbus and to any other event destinations that are configured for the situation.
10. The probe maps a subset of the IBM Tivoli Monitoring EIF slots to the Netcool/OMNIbus ObjectServer attributes and creates an OMNIbus event.
11. IBM Tivoli Monitoring triggers determine the event is a loopback event and ignores it.

A Tivoli Enterprise Portal operator can also acknowledge and deacknowledge events or close pure events for the bidirectional architecture. The Tivoli Enterprise Portal Server processes the event status changes from the Tivoli Enterprise Portal and notifies the hub Tivoli Enterprise Monitoring Server of the changes. The event status updates are sent by the hub Tivoli Enterprise Monitoring Server to Netcool/OMNIbus by using the flows shown in Figure 162 on page 677.

Event behavior

How events are handled depends on several criteria, including the architecture type (unidirectional versus bidirectional) and the event type (pure versus sampled). Table 129 describes the behavior for events sent from the Hub monitoring server to Netcool/OMNIbus. Table 130 on page 694 describes the behavior for events sent directly from the monitoring agents to Netcool/OMNIbus.

Table 129. Behavior of events originating from a hub monitoring server

Action	Event type	Unidirectional behavior	Bidirectional behavior
Situation condition becomes true.	Pure and sampled events.	**Summary:** A new event is opened in the hub monitoring server and in the Netcool/OMNIbus ObjectServer if it does not deduplicate an existing event. **Details:** The hub monitoring server opens a new situation event and sends an event with the Open status to Netcool/OMNIbus ObjectServer using flows 1 to 4 as shown in Figure 162 on page 677 and a new event is opened in Netcool/OMNIbus if it does not deduplicate an existing event. **Note:** The hub monitoring server sends an event with open status to Netcool/OMNIbus each time a condition is true for a pure situation. For sampled situations, an event with open status is sent when the situation condition transitions from not true to true. Until the sampled situation condition becomes false, another event is not sent unless the status of the event is changed, for example to acknowledged.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Sampled event situation condition is no longer true or a pure sampled situation UNTIL modifier condition becomes true.	Pure and sampled events.	**Summary:** Event is closed in the hub monitoring server and cleared in the Netcool/OMNIbus ObjectServer. **Details:** After the event is closed in the hub monitoring server, a closed status update event is sent to Netcool/OMNIbus ObjectServer by using flows 1 to 4 as shown in Figure 162 on page 677. When the IBM Tivoli Monitoring triggers process the status update event they clear the event in the Netcool/OMNIbus ObjectServer. **Note:** For more information about creating a situation UNTIL modifier and details on when sampled events are closed if they have an UNTIL modifier, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.	Same as unidirectional behavior.
Event acknowledged using the Netcool/ OMNIbus Event List UI.	Pure and sampled events.	**Summary:** Event status is changed to acknowledged in the Netcool/OMNIbus ObjectServer. However, the event status is not updated in the hub monitoring server and Tivoli Enterprise Portal.	**Summary:** The event status is changed to acknowledged in the Netcool/OMNIbus ObjectServer, the hub monitoring server, and Tivoli Enterprise Portal. **Details:** After the event status is changed to acknowledged in the Netcool/OMNIbus ObjectServer, the hub monitoring server is notified about the status change by flows 5 through 8 in Figure 162 on page 677. In flow 7, the IBM Tivoli Monitoring Situation Update Forwarder sends a CT_Acknowledge SOAP request to the hub monitoring server. The hub monitoring server changes the event status to Acknowledged when it processes the SOAP request and sends a status update event back to OMNIbus using flows 9 through 11.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Event cleared or deleted using the Netcool/ OMNIbus Event List UI.	Pure events.	**Summary:** The pure event is cleared or deleted in the Netcool/OMNIbus ObjectServer. However, the event remains open in the hub monitoring server and Tivoli Enterprise Portal until the UNTIL modifier condition of the situation becomes true.	**Summary:** The pure event is cleared or deleted in the Netcool/OMNIbus ObjectServer and closed in the hub monitoring server and Tivoli Enterprise Portal. **Details:** After the event is cleared or deleted in the Netcool/OMNIbus ObjectServer, the hub monitoring server is notified about the status change by flows 5 through 8 in Figure 162 on page 677. In flow 7, the IBM Tivoli Monitoring Situation Update Forwarder sends a CT_Reset SOAP request to the hub monitoring server. The hub monitoring server closes the event when it processes the SOAP request and sends a status update event back to OMNIbus using flows 9 through 11.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Event cleared or deleted using the Netcool/ OMNIbus Event List UI.	Sampled events.	**Summary:** The sampled event is cleared or deleted in the Netcool/OMNIbus ObjectServer. However, the event remains open in the hub monitoring server and Tivoli Enterprise Portal until the situation condition is no longer true. No further status updates are sent to Netcool/OMNIbus until the situation condition becomes false and then true again. Therefore, the Netcool/OMNIbus operator is not notified that the event condition has not been resolved.	**Summary:** The sampled event is cleared or deleted in the Netcool/OMNIbus ObjectServer but its status is changed to *Acknowledged* in the hub monitoring server and Tivoli Enterprise Portal for a specified time. If the situation condition is still true after the specified time, a status update event is sent to Netcool/OMNIbus and an event is opened. This status update notifies the Netcool/OMNIbus operator that the event condition is not resolved. **Details:** When the sampled event is cleared or deleted, the event data is cached by the ObjectServer in an IBM Tivoli Monitoring table. Then the hub monitoring server is notified of the status change by flows 5 through 8 in Figure 162 on page 677. In flow 7, the IBM Tivoli Monitoring Situation Update Forwarder sends a CT_Acknowledge SOAP request with a configurable timeout to the hub monitoring server. The hub monitoring server changes the event status to *Acknowledged* and starts an expiration timer. A status update event is sent back to OMNIbus by using flows 9 through 11. The events are marked as *Acknowledged* in the hub monitoring server and Tivoli Enterprise Portal because a sampled event cannot be closed unless the situation condition is no longer true. By leaving the situation as *Acknowledged*, Netcool/OMNIbus is notified if the situation condition is still true after the timeout expires.If the situation condition is still true when the timeout expires, the hub monitoring server sends an *Acknowledgement Expired* status update event to Netcool/OMNIbus ObjectServer by using flows 1 to 4 as shown in Figure 162 on page 677. If the event has already been removed from the Netcool/OMNIbus ObjectServer alerts.status table, a new event is opened in the ObjectServer. Because status update events contain only base ITM EIF slots and no agent-specific slots, the event is re-opened using data that was cached when the event was cleared or deleted from the Netcool/OMNIbus Event List UI. However, if the event is still in the Netcool/OMNIbus ObjectServer alerts.status table when the status update event is processed, the event is deduplicated by the ITM triggers. The event is then reopened and contains event attribute settings from the original event.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	**Event type**	**Unidirectional behavior**	**Bidirectional behavior**
Event deacknowledged by using the Netcool/ OMNIbus Event List UI.	Pure and sampled events.	**Summary:** The event status is changed to deacknowledged in the Netcool/OMNIbus ObjectServer. However, the event status is not updated in the hub monitoring server and Tivoli Enterprise Portal.	**Summary:** The event status is changed to deacknowledged in the Netcool/OMNIbus ObjectServer and to resurfaced in the hub monitoring server and Tivoli Enterprise Portal. **Details:** After the event status is changed to deacknowledged in the Netcool/OMNIbus ObjectServer, the hub monitoring server is notified about the status change by flows 5 through 8 in Figure 162 on page 677. In flow 7, the IBM Tivoli Monitoring Situation Update Forwarder sends a CT_Resurface request to the hub monitoring server. The hub monitoring server changes the event status to Resurfaced when it processes the SOAP request and sends a status update event back to OMNIbus using flows 9 through 11.
Event acknowledged without a timeout by using the Tivoli Enterprise Portal Situation Event Console.	Pure and sampled events.	Tivoli Enterprise Portal operators should not change the event status when the unidirectional architecture is being used.	**Summary:** The event status is changed to acknowledged in both the hub monitoring server and Netcool/OMNIbus ObectServer. **Details:** After the event status is changed to acknowledged in the hub monitoring server, an Acknowledged status update event is sent to Netcool/OMNIbus using flows 1 to 4 as shown in Figure 162 on page 677 and the event status is changed to Acknowledged in the Netcool/OMNIbus ObjectServer and Netcool/OMNIbus Event List UI.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Event acknowledged with timeout by using the Tivoli Enterprise Portal Situation Event Console.	Pure and sampled events.	Tivoli Enterprise Portal operators should not change the event status when the unidirectional architecture is being used.	**Summary:** The event status is changed to acknowledged in both the hub monitoring server and Netcool/OMNIbus ObjectServer and you can configure how Netcool/OMNIbus handles the timeout notification event. See the detailed description for more information. **Details:** After the event status is changed to acknowledged in the hub monitoring server, an Acknowledged status update event is sent to Netcool/OMNIbus using flows 1 to 4 as shown in Figure 162 on page 677 and the event status is changed to Acknowledged in the Netcool/OMNIbus ObjectServer and Netcool/OMNIbus Event List UI. When the timeout expires and the situation event is still open in the hub monitoring server, the hub monitoring server sets the event status to Acknowledgement Expired and sends an Acknowledgement Expired status update event to Netcool/OMNIbus using flows 1 to 4 as shown in Figure 162 on page 677. By default, the IBM Tivoli Monitoring triggers reject the Acknowledgement Expired status update event and use flows 5 to 8 to send a request to the hub monitoring server to set the event status to Acknowledged. (In flow 7, the Situation Update Forwarder sends a CT_Acknowledge request to the hub monitoring server.) The hub monitoring server sets the event status to Acknowledged when it processes the SOAP request and sends a status update event back to OMNIbus using flows 9 through 11. You can override the default IBM Tivoli Monitoring trigger behavior by setting the *sit_ack_expired_def_action* variable to ACCEPT using the procedure described in “Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers” on page 754. If you set the variable to ACCEPT, the IBM Tivoli Monitoring triggers deacknowledge the event in the Netcool/OMNIbus ObjectServer but the event still has the acknowledgment expired status in the hub monitoring server.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Event deacknowledged using the Tivoli Enterprise Portal Situation Event Console.	Pure and sampled events.	Tivoli Enterprise Portal operators should not change the event status when the unidirectional architecture is being used.	**Summary:** Behavior is configurable. See the detailed description for the two types of behaviors that are supported. **Details:** After the event status is changed to Resurfaced in the hub monitoring server, a Resurfaced status update event is sent to Netcool/OMNIbus using flows 1 to 4 as shown in Figure 162 on page 677. By default the IBM Tivoli Monitoring triggers accept the Resurfaced status update event and deacknowledge the event in the Netcool/OMNIbus ObjectServer. You can override the default trigger behavior by setting the `sit_resurface_def_action` variable to REJECT using the procedure described in "Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754. If you set the variable to REJECT then the IBM Tivoli Monitoring triggers use flows 5 to 8 in Figure 162 on page 677 to send a CT_ACKNOWLEDGE SOAP request to the hub monitoring server. The hub monitoring server sets the event status to Acknowledged when it processes the SOAP request and sends a status update event back to OMNIbus using flows 9 through 11.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Situation is stopped. **Note:** Situations are stopped if an operator initiates the situation stop action from the Tivoli Enterprise Portal or modifies the situation definition. However, a situation is not stopped if its distribution list is modified.	Pure events.	**Summary:** The hub monitoring server closes all pure events for the situation. These same events are cleared in the Netcool/OMNIbus ObjectServer unless you configure a different behavior in the IBM Tivoli Monitoring triggers. **Details:** As the hub monitoring server closes all pure events for the situation, it sends a situation stop event to Netcool/OMNIbus for each remote monitoring server that was monitoring the situation. The situation stop event specifies the remote monitoring server in the `situation_thrunode` EIF slot. (This slot is mapped to the `ITMThruNode`OMNIbus attribute by the Tivoli Netcool/OMNIbus EIF Probe.) When the IBM Tivoli Monitoring triggers in Netcool/OMNIbus process a situation stop event, they clear all events for the situation that are detected by the remote monitoring server specified by the `ITMThruNode` OMNIbus attribute. However, you can configure the IBM Tivoli Monitoring triggers to ignore situation stop events for pure events. For more information, see "Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Situation is stopped. **Note:** Situations are stopped if an operator initiates the situation stop action from the Tivoli Enterprise Portal or modifies the situation definition. However, a situation is not stopped if its distribution list is modified.	Sampled events.	**Summary:** The hub monitoring server closes all sampled events for the situation. These same events are cleared in the Netcool/OMNIbus ObjectServer. **Details:** As the hub monitoring server closes all sampled events for the situation, it sends a situation stop event to Netcool/OMNIbus for each remote monitoring server that was monitoring the situation. The situation stop event specifies the remote monitoring server in the `situation_thrunode` EIF slot. (This slot is mapped to the `ITMThruNode` OMNIbus attribute by the Tivoli Netcool/OMNIbus EIF Probe.) When the IBM Tivoli Monitoring triggers in Netcool/OMNIbus process a situation stop event, they clear all events for the situation that are detected by the remote monitoring server specified by the `ITMThruNode` OMNIbus attribute.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Agent stopped.	Pure events.	**Summary:** Stopping the agent has no effect on pure situation event status. An MS_Offline situation event is also sent to Netcool/OMNIbus to indicate that the monitoring agent is not being monitored. **Details:** After the monitoring server of the agent detects that the agent has not responded for three situation sampling intervals, the situation's event is closed in the hub monitoring server. A Closed status update event is sent to Netcool/OMNIbus ObjectServer by using flows 1 to 4 as shown in Figure 162 on page 677. When the IBM Tivoli Monitoring triggers process the status update event, they clear the event in the Netcool/OMNIbus ObjectServer. If you do not want events to be closed in Netcool/OMNIbus after an agent is stopped, you can customize this behavior. For more information, see "Customizing event status processing behavior when agent switching is used or the agent goes offline" on page 760.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Agent stopped.	Sampled events.	**Summary:** The sampled events from the agent are closed in the hub monitoring server if the agent is stopped for long enough. Its events are also cleared in the Netcool/OMNIbus ObjectServer. An MS_Offline situation event is also sent to Netcool/OMNIbus to indicate that the monitoring agent is not being monitored. **Details:** After the monitoring server of the agent detects that the agent has not responded for three situation sampling intervals, the situation's event is closed in the hub monitoring server. A Closed status update event is sent to Netcool/OMNIbus ObjectServer by using flows 1 to 4 as shown in Figure 162 on page 677. When the IBM Tivoli Monitoring triggers process the status update event, the triggers clear the event in the Netcool/OMNIbus ObjectServer. If you do not want events to be closed in Netcool/OMNIbus after an agent is stopped, you can customize this behavior. For more information, see “Customizing event status processing behavior when agent switching is used or the agent goes offline” on page 760.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Agent loses connectivity to the primary monitoring server and switches to the secondary monitoring server.	Pure and sampled events.	When an agent has a situation that has triggered and the situation is true, and the agent subsequently loses connectivity to the Tivoli Enterprise Monitoring Server and switches to a different monitoring server, the event may be closed by the original monitoring server if it detects the agent is no longer responding to it before the new monitoring server determines that the situation event is still true. There are environment variables that can be added to the monitoring server environment file to customize the behavior of event status processing when agent switching occurs to help ensure that events are not closed by the original monitoring server. For a complete description of these variables, see "Customizing event status processing behavior when agent switching is used or the agent goes offline" on page 760. **Note:** If an agent switches to another monitoring server because the original monitoring server was stopped, all sampled events for the agent will be closed by the original monitoring server. This behavior is not configurable. However, you can configure whether pure events are closed in this scenario. For more information, see "Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754.	Same as unidirectional behavior.
Hub monitoring server stopped.	Pure and sampled events.	**Summary:** No flows occur between the hub monitoring server and Netcool/OMNIbus when the hub monitoring server is stopped.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Hub monitoring server started.	Pure events.	**Summary:** The pure events in the Netcool/OMNIbus ObjectServer are unaffected. However, the hub monitoring server and Netcool/OMNIbus might not have the same status for pure events. **Details:** When the hub monitoring server is started, it sends a `master_reset` event to Netcool/OMNIbus using flows 1 to 4 as shown in Figure 162 on page 677. The IBM Tivoli Monitoring triggers in the Netcool/OMNIbus ObjectServer do not update the status of pure events when the master reset event is processed. However, Netcool/OMNIbus and the hub monitoring server (and Tivoli Enterprise Portal) might have a different status for the pure events after the hub is restarted. • The hub monitoring server does not have any event status for pure events that were opened or acknowledged prior to the hub restart if the events were for agents connected directly to the hub monitoring server. These events might still be open or acknowledged in the Netcool/OMNIbus ObjectServer. • The hub monitoring server has a status of open for pure events that were open or acknowledged prior to the hub restart if the events were for agents connected to the remote Tivoli Enterprise Monitoring Server. However, these events may be acknowledged, cleared, or deleted in Netcool/OMNIbus.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Hub monitoring server started.	Sampled events.	**Summary:** The sampled events in the Netcool/OMNIbus ObjectServer from this hub monitoring event are cleared. **Details:** When the hub monitoring server is started, it sends a `master_reset` event to Netcool/OMNIbus using flows 1 to 4 as shown in Figure 162 on page 677. The IBM Tivoli Monitoring triggers in the Netcool/OMNIbus ObjectServer clear all sampled events from this hub monitoring server when the master reset event is processed. The master reset handling ensures that events are cleared in Netcool/OMNIbus if the situation condition became false while the hub monitoring server was stopped. The events whose situation conditions are still true will be reopened in Netcool/OMNIbus after the master reset event is sent.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	Event type	Unidirectional behavior	Bidirectional behavior
Remote monitoring server stopped.	Pure events.	**Summary:** The hub monitoring server closes all of the pure events for agents connected to the remote monitoring server. The same events are cleared in Netcool/OMNIbus Object unless you configure a different behavior. An MS_Offline situation event is sent to Netcool/OMNIbus for the remote monitoring server to indicate that these managed systems are not being monitored. You will not see an MS_Offline message for each of the monitoring agents when the remote monitoring server goes offline. **Details:** As the hub monitoring server closes the pure events for each situation being monitored by the remote monitoring server, it sends a situation stop event to Netcool/OMNIbus and specifies the remote monitoring server in the `situation_thrunode` EIF slot. (This slot is mapped to the `ITMThruNode` OMNIbus attribute by the Tivoli Netcool/OMNIbus EIF Probe.) When the IBM Tivoli Monitoring triggers in Netcool/OMNIbus process a situation stop event, they clear all events for the situation that are detected by the remote monitoring server specified by the `ITMThruNode` OMNIbus attribute. However, you can configure the IBM Tivoli Monitoring triggers to ignore situation stop events for pure events. For more information, see “Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers” on page 754.	Same as unidirectional behavior.

Table 129. Behavior of events originating from a hub monitoring server (continued)

Action	**Event type**	**Unidirectional behavior**	**Bidirectional behavior**
Remote monitoring server stopped.	Sampled events.	**Summary:** The hub monitoring server closes all of the sampled events for agents connected to the remote monitoring server. The same events are cleared in Netcool/OMNIbus Object. An MS_Offline situation event is sent to Netcool/OMNIbus for the remote monitoring server and each of the monitoring agents connected to the monitoring server to indicate that these managed systems are not being monitored. You will not see an MS_Offline message for each of the monitoring agents when the remote monitoring server goes offline. **Details:** As the hub monitoring server closes the sampled events for each situation being monitored by the remote monitoring server, it sends a situation stop event to Netcool/OMNIbus and specifies the remote Tivoli Enterprise Monitoring Server in the `situation_thrunode` EIF slot. (This slot is mapped to the `ITMThruNode` OMNIbus attribute by the Tivoli Netcool/OMNIbus EIF Probe.) When the IBM Tivoli Monitoring triggers in Netcool/OMNIbus process a situation stop event, they clear all events for the situation that are detected by the remote monitoring server specified by the `ITMThruNode` OMNIbus attribute.	Same as unidirectional behavior.
Remote monitoring server started.	Pure and sampled events.	Same as unidirectional behavior when a remote monitoring server is stopped.	Same as bidirectional behavior when a remote monitoring server is stopped.

Table 130. Behavior of events originating from IBM Tivoli Monitoring agents

Action	Event Type	Behavior
Situation condition becomes true.	Pure and sampled events.	A new event is opened in the agent and in the Netcool/OMNIbus ObjectServer if it does not deduplicate an existing event. An event with open status is sent from the agent to Netcool/OMNIbus each time the situation condition is true, if the following conditions are met: • The situation generates pure events, or • The situation generates sampled events, the events are sent to Netcool/OMNIbus using SNMP, and the situation mode is set to Rising Continuous.
Sampled event situation condition is no longer true.	Sampled events.	Event is closed in the agent and cleared in the Netcool/OMNIbus ObjectServer.
Event acknowledged or deacknowledged using Netcool/OMNIbus Event List UI in OMNIbus.	Pure and sampled events.	The event status is changed to acknowledged or deacknowledged in Netcool/OMNIbus but the event status maintained by the agent is unaffected.
Event cleared or deleted using Netcool/OMNIbus Event List UI.	Pure events.	The event is cleared or deleted in the Netcool/OMNIbus ObjectServer. However, the event status maintained by the agent is not affected.
Event cleared or deleted using Netcool/OMNIbus Event List UI.	Sampled events.	The event is cleared or deleted in the Netcool/OMNIbus ObjectServer. However, the event status maintained by the agent is unaffected and the Netcool/OMNIbus operator is not notified if the event condition has not been resolved, unless the following conditions are met: • The agent has been configured to send SNMP events to Netcool/OMNIbus, and • The situation mode is set to Rising Continuous so that an event is sent to Netcool/OMNIbus each sampling interval that the situation event evaluates to true. With this mode, the event is reopened in Netcool/OMNIbus if the event condition is still true.

Table 130. Behavior of events originating from IBM Tivoli Monitoring agents (continued)

Action	Event Type	Behavior
Situation is stopped using the Agent Service Interface.	Pure and sampled events.	If the agent is configured to send lifecycle events when a situation is stopped, a `EE_SIT_STOPPED` event is sent to Netcool/OMNIbus. For agents that send SNMP events to Netcool/OMNIbus, the events in Netcool/OMNIbus are not affected by this lifecycle event. For agents that send EIF events to Netcool/OMNIbus, the events from the agent for the stopped situation are cleared in Netcool/OMNIbus.
Agent is stopped.	Pure and sampled events.	No events are sent by the agent when it is stopped.
Agent is started.	Pure and sampled events.	If the agent is configured to send SNMP events to Netcool/OMNIbus, no events other than lifecycle traps are sent by the agent when it is started. If the agent is configured to send EIF events to Netcool/OMNIbus, by default the agent does not send any events (other than lifecycle events) to Netcool/OMNIbus when it is started. However, you can change this behavior and configure the agent to send a master reset event to Netcool/OMNIbus when the agent is started. When the IBM Tivoli Monitoring triggers in Netcool/OMNIbus process this event, they clear all events for the agent. This ensures that events are cleared in Netcool/OMNIbus if the situation condition became false while the agent was stopped. The events whose situation conditions are still true will be reopened in Netcool/OMNIbus after the master reset event is sent. **Note:** The agent does not maintain event status across restarts.

Agents can also send lifecycle and heartbeat events to Netcool/OMNIbus. For more details on lifecycle and heartbeat events, see the Agent Autonomy chapter in the *IBM Tivoli Monitoring: Administrator's Guide*.

IBM Tivoli Monitoring Event Synchronization component

You must install the IBM Tivoli Monitoring Event Synchronization component regardless of the architectural solution you choose for event integration between the hub monitoring server and Netcool/OMNIbus. This component consists of the following three parts:

`IBM Tivoli Monitoring rules file for the Netcool/OMNIbus Probe for Tivoli EIF.`
Used to update the probe so it can understand IBM Tivoli Monitoring situation events.

SQL files to update the Netcool/OMNIbus ObjectServer database.
Used to update the Netcool/OMNIbus ObjectServer database schema that contain attributes specific to Tivoli Monitoring, and create or update the triggers that process IBM Tivoli Monitoring events. The SQL files also forward events to the IBM Tivoli Monitoring Situation Update Forwarder when using bidirectional communication.

IBM Tivoli Monitoring Situation Update Forwarder executable.
The program that forwards the updates from OMNIbus back to IBM Tivoli Monitoring.

The IBM Tivoli Monitoring rules file for the probe maps a subset of the IBM Tivoli Monitoring EIF slots to OMNIbus ObjectServer attributes to create an OMNIbus event. The SQL files update the Netcool/OMNIbus ObjectServer database schema to contain IBM Tivoli Monitoring specific attributes and create or update the triggers that process IBM Tivoli Monitoring events as well as forward events to the IBM Tivoli Monitoring Situation Update Forwarder when using bidirectional communication. The IBM Tivoli Monitoring Situation Update Forwarder is used to forward updates to the situation events, back to the originating hub monitoring server via SOAP messages.

If you choose to send EIF events from agents to Netcool/OMNIbus, you must also install the IBM Tivoli Monitoring Event Synchronization component because it contains the IBM Tivoli Monitoring rules file for the Netcool/OMNIbus Probe for EIF, and the SQL files to update the Netcool/OMNIbus ObjectServer database.

If you are sending only SNMP events from agents to Netcool/OMNIbus, you do not need to install the synchronization component. The Tivoli Monitoring rules file for the Netcool/OMNIbus Probe for SNMP, and the SQL files to update the Netcool/OMNIbus ObjectServer database, are packaged separately.

If IBM Tivoli Monitoring situations have custom slots, you must update OMNIbus to add those custom slots. For more information about making this change, see “Configuring the Netcool/OMNIbus EIF probe” on page 740.

Architecture scenarios

The scenarios in this section illustrate ways to forward situation events from one or more Tivoli Enterprise Monitoring Servers to one or more Netcool/OMNIbus ObjectServers.

These scenarios use the bidirectional architecture, as described in the previous sections. If unidirectional architecture is applied to these scenarios, the IBM Tivoli Monitoring Situation Update Forwarder is removed and Netcool/OMNIbus does not send event status updates back to IBM Tivoli Monitoring.

One hub Tivoli Enterprise Monitoring Server and one Netcool/OMNIbus ObjectServer

In this scenario, a single hub Tivoli Enterprise Monitoring Server is forwarding events to a single Netcool/OMNIbus ObjectServer. With bidirectional architecture, as changes are made to those events in the Netcool/OMNIbus ObjectServer, the updates are forwarded back to the hub monitoring server.

By default, event forwarding is not enabled for a new situation unless you base the new situation definition on an existing situation that already has event forwarding enabled or you explicitly enable event forwarding. When you enable event forwarding for a situation, the hub monitoring server sends the situation events to the EIF probe that was specified when event forwarding was enabled for the hub monitoring server.

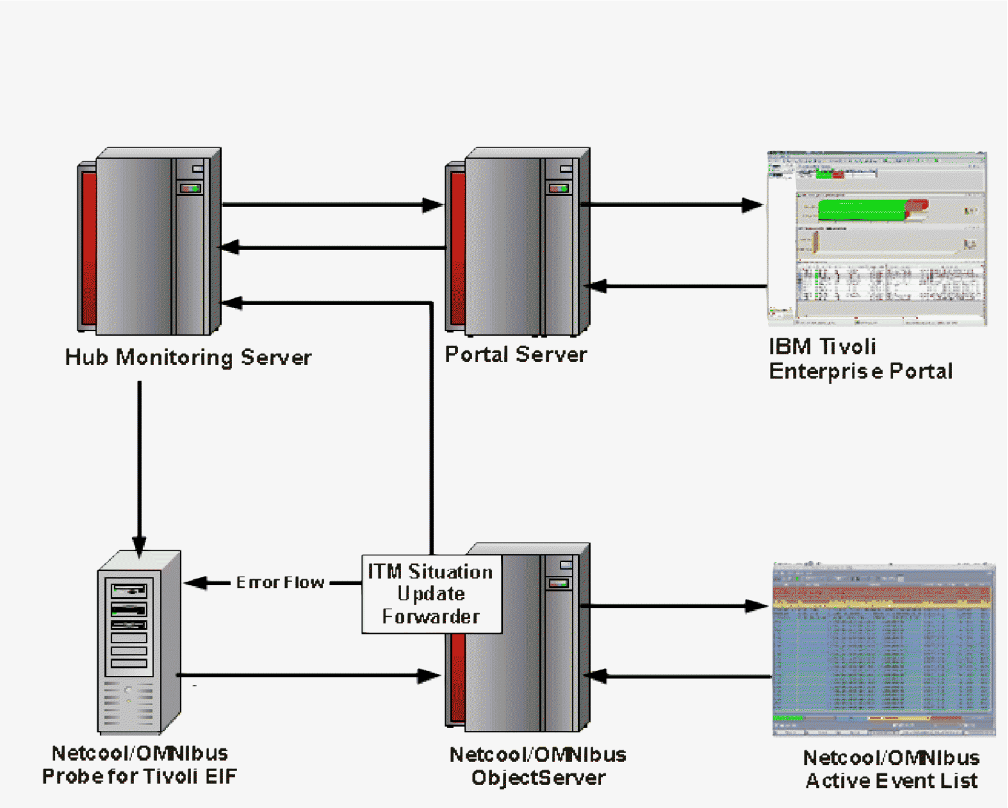

Hub Monitoring Server
Portal Server
IBM Tivoli Enterprise Portal
Error Flow
ITM Situation Update Forwarder
Netcool/OMNIbus Probe for Tivoli EIF
Netcool/OMNIbus ObjectServer
Netcool/OMNIbus Active Event List

Figure 163. One hub Tivoli Enterprise Monitoring Server and one Netcool/OMNIbus ObjectServer

Uses

This scenario describes the most basic of event synchronization architectures. This type of architecture is useful for small environments. Other uses include proof of concept or test environments. You can install all components on the same server. However, for performance reasons, install components on separate servers when setting up production environments.

Installation and configuration

If you are installing IBM Tivoli Monitoring and Netcool/OMNIbus event integration for the first time, complete the tasks in the following table. However, if you are upgrading an existing event integration environment, see “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746.

Table 131. Installation and configuration: one hub Tivoli Enterprise Monitoring Server and one Netcool/OMNIbus ObjectServer

	Task	**Architecture Type**	**Administrator**
1.	Install Netcool/OMNIbus ObjectServer if it is not already installed or upgrade it to the fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
2.	Install the Netcool/OMNIbus Probe for Tivoli EIF if it is not already installed or upgrade it to the release required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
3.	"Installing the IBM Tivoli Monitoring Event Synchronization Component" on page 720.	Unidirectional and bidirectional	Netcool/OMNIbus
4.	"Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers" on page 733.	Unidirectional and bidirectional	Netcool/OMNIbus
5.	"Changing the default deduplication trigger" on page 735.	Unidirectional and bidirectional	Netcool/OMNIbus
6.	"Configuring the OMNIbus server for program execution from scripts" on page 742.	Bidirectional	Netcool/OMNIbus
7.	Start the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see "Starting and stopping the IBM Tivoli Monitoring Situation Update Forwarder" on page 739.	Bidirectional	Netcool/OMNIbus
8.	"Configuring the Netcool/OMNIbus EIF probe" on page 740.	Unidirectional and bidirectional	Netcool/OMNIbus

Table 131. Installation and configuration: one hub Tivoli Enterprise Monitoring Server and one Netcool/OMNIbus ObjectServer (continued)

	Task	Architecture Type	Administrator
9.	"Configuring the hub monitoring server to forward events" on page 743.	Unidirectional and bidirectional	IBM Tivoli Monitoring
10.	"Verifying installation and configuration" on page 745.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus
11.	Determine if additional configuration tasks should be performed. See "Customizing Event Integration" on page 750.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Multiple hub Tivoli Enterprise monitoring servers and one Netcool/OMNIbus ObjectServer

In this scenario, multiple hub Tivoli Enterprise Monitoring Servers are forwarding events to a single Netcool/OMNIbus ObjectServer. As changes are made to those events in the Netcool/OMNIbus ObjectServer, the updates are forwarded back to the hub monitoring server that is associated with that situation. Event forwarding must be enabled on each monitoring server, and the EIF probe associated with the ObjectServer must be defined as the default EIF receiver for each hub monitoring server.

By default, event forwarding is not enabled for a new situation unless you base the new situation definition on an existing situation that already has event forwarding enabled or you explicitly enable event forwarding. When you enable event forwarding for a situation, the hub monitoring server sends the situation's events to the EIF probe that was specified when event forwarding was enabled for the hub monitoring server.

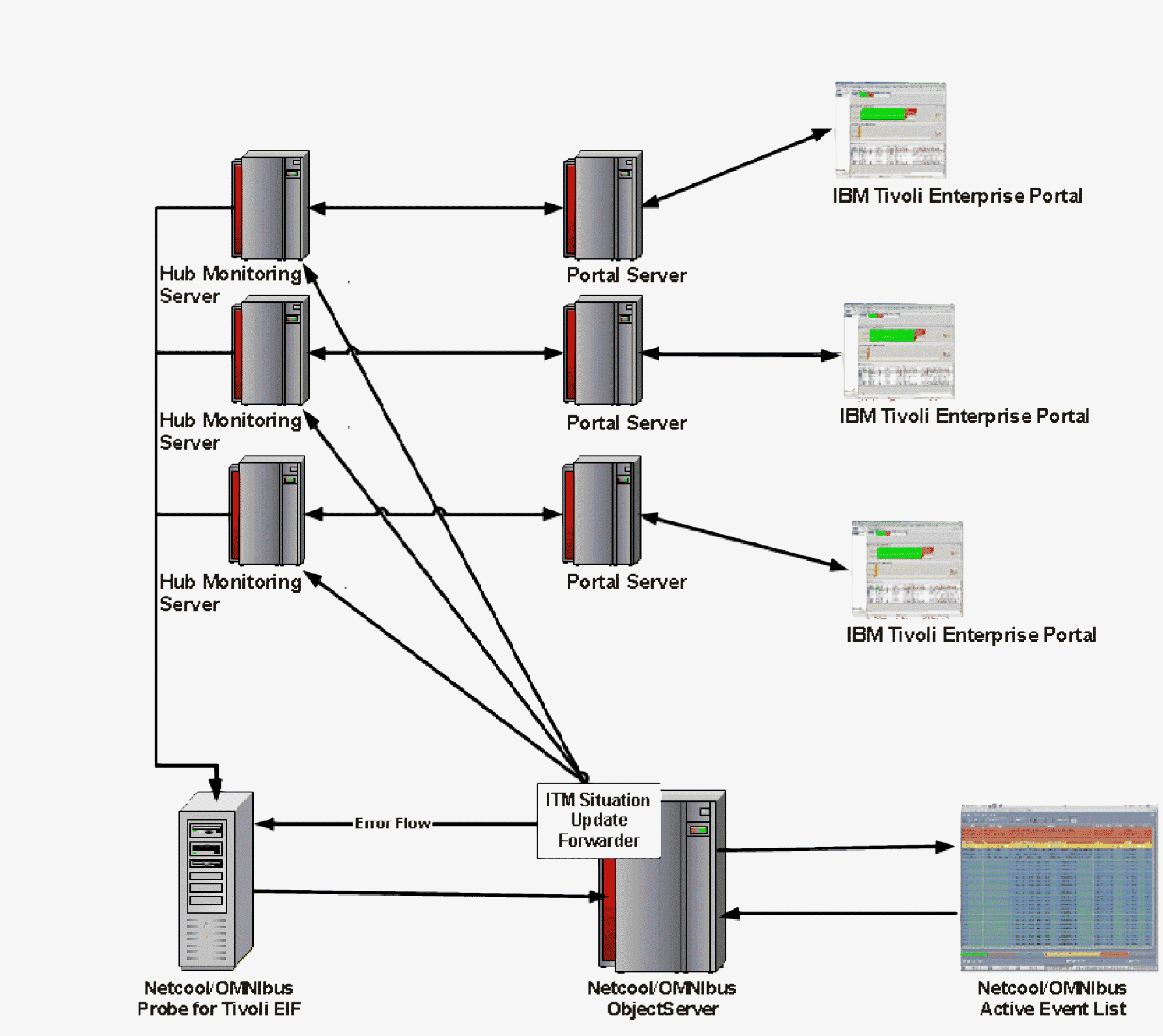

Hub Monitoring Server
Portal Server
IBM Tivoli Enterprise Portal
Hub Monitoring Server
Portal Server
IBM Tivoli Enterprise Portal
Hub Monitoring Server
Portal Server
IBM Tivoli Enterprise Portal
Error Flow
ITM Situation Update Forwarder
Netcool/OMNIbus Probe for Tivoli EIF
Netcool/OMNIbus ObjectServer
Netcool/OMNIbus Active Event List

Figure 164. Multiple hub Tivoli Enterprise Monitoring Servers and one Netcool/OMNIbus ObjectServer

Uses

This scenario demonstrates how you can consolidate all your IBM Tivoli Monitoring events by forwarding them to the same Netcool/OMNIbus ObjectServer through the Netcool/OMNIbus Probe for Tivoli EIF.

Installation and configuration

If you are installing IBM Tivoli Monitoring and Netcool/OMNIbus event integration for the first time, perform the tasks in the following table. However, if you are upgrading an existing event integration environment, see “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746.

Table 132. Installation and configuration: multiple hub Tivoli Enterprise Monitoring Servers and one Netcool/OMNIbus ObjectServer

	Task	Architecture Type	Administrator
1.	Install Netcool/OMNIbus ObjectServer if it is not already installed or upgrade it to the fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
2.	Install the Netcool/OMNIbus Probe for Tivoli EIF if it is not already installed or upgrade it to the release required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
3.	"Installing the IBM Tivoli Monitoring Event Synchronization Component" on page 720.	Unidirectional and bidirectional	Netcool/OMNIbus
4.	"Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers" on page 733.	Unidirectional and bidirectional	Netcool/OMNIbus
5.	"Changing the default deduplication trigger" on page 735.	Unidirectional and bidirectional	Netcool/OMNIbus
6.	"Configuring the OMNIbus server for program execution from scripts" on page 742.	Bidirectional	Netcool/OMNIbus
7.	Start the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see "Starting and stopping the IBM Tivoli Monitoring Situation Update Forwarder" on page 739.	Bidirectional	Netcool/OMNIbus

Table 132. Installation and configuration: multiple hub Tivoli Enterprise Monitoring Servers and one Netcool/OMNIbus ObjectServer (continued)

	Task	Architecture Type	Administrator
8.	Define additional monitoring servers to the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see "Updating the IBM Tivoli Monitoring Situation Forwarder to forward event status updates to additional monitoring servers" on page 752.	Bidirectional	Netcool/OMNIbus
9.	"Configuring the Netcool/OMNIbus EIF probe" on page 740.	Unidirectional and bidirectional	Netcool/OMNIbus
10.	*For each hub monitoring server:* "Configuring the hub monitoring server to forward events" on page 743.	Unidirectional and bidirectional	IBM Tivoli Monitoring
11.	"Verifying installation and configuration" on page 745.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus
12.	Determine if additional configuration tasks should be performed. See "Customizing Event Integration" on page 750.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

One hub Tivoli Enterprise Monitoring Server and multiple Netcool/OMNIbus ObjectServers

In this scenario, one hub Tivoli Enterprise Monitoring Server is forwarding events to multiple Netcool/OMNIbus ObjectServers where each ObjectServer is configured for the Netcool/OMNIbus single-tier architecture. Each event server to which events are forwarded must have an associated EIF probe and an installed IBM Tivoli Monitoring Situation Update Forwarder. For each situation, you must specify the event server to which the situation event should be forwarded.

By default, event forwarding is not enabled for a new situation unless you base the new situation definition on an existing situation that already has event forwarding enabled, or you explicitly enable event forwarding for the situation. When you enable event forwarding for a situation, by default the hub monitoring server sends the situation's events to the EIF probe that was specified when event forwarding was enabled for the hub monitoring server. If you want to forward events to multiple Netcool/OMNIbus ObjectServers, you must use the `tacmd createEventDest` command to create additional event server destination definitions and then select the appropriate event destination for each situation that has event forwarding enabled.

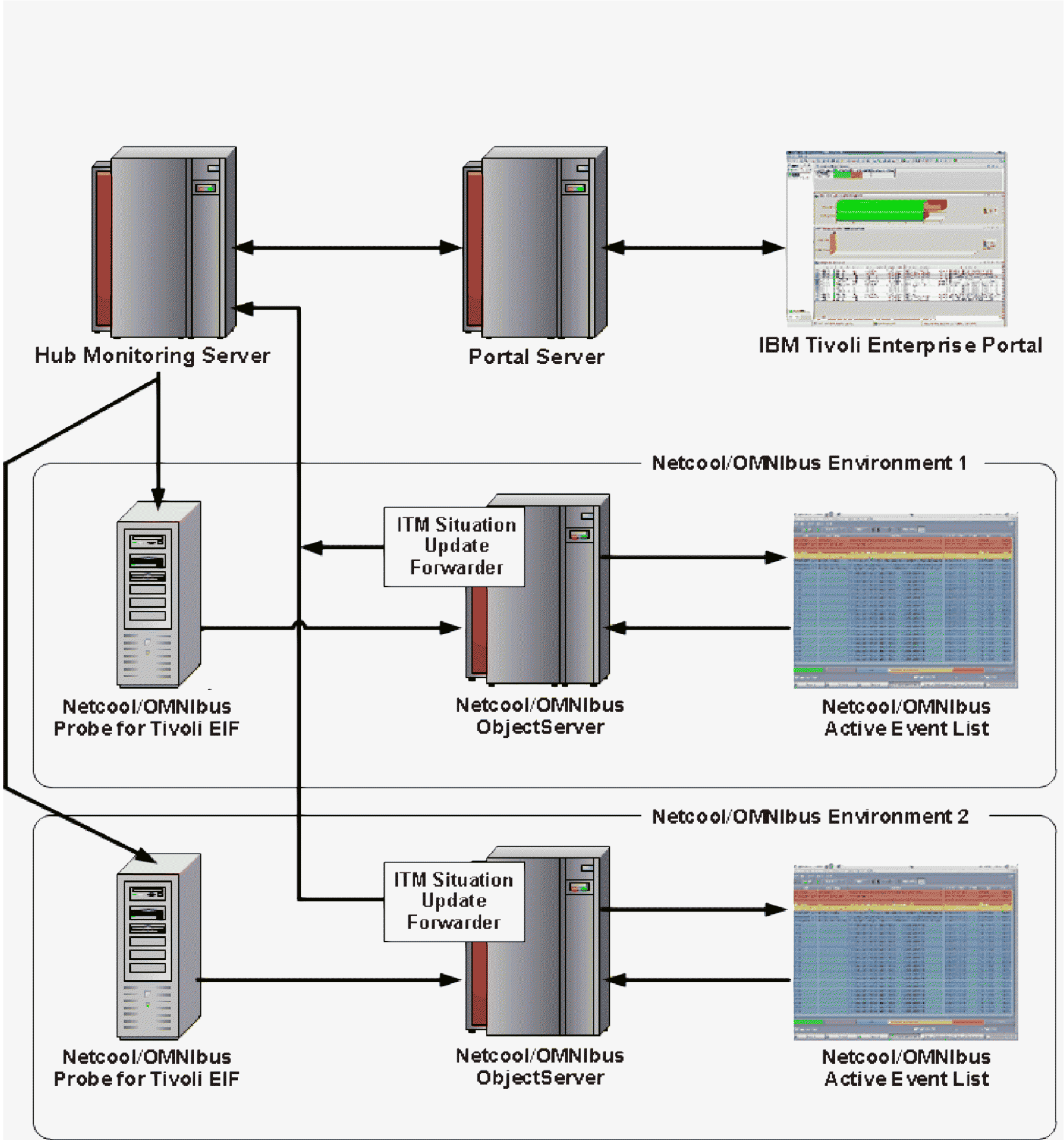

Hub Monitoring Server
Portal Server
IBM Tivoli Enterprise Portal
Netcool/OMNIbus Environment 1
ITM Situation Update Forwarder
Netcool/OMNIbus Probe for Tivoli EIF
Netcool/OMNIbus ObjectServer
Netcool/OMNIbus Active Event List
Netcool/OMNIbus Environment 2
ITM Situation Update Forwarder
Netcool/OMNIbus Probe for Tivoli EIF
Netcool/OMNIbus ObjectServer
Netcool/OMNIbus Active Event List

Figure 165. One hub Tivoli Enterprise Monitoring Server and multiple Netcool/OMNIbus ObjectServer

Uses

By configuring the Hub monitoring server to send all events to multiple Netcool/OMNIbus ObjectServers, you can introduce redundancy into your event synchronization environment. Use this method to increase availability by having your event data on multiple Netcool/OMNIbus ObjectServers.

This scenario can also be used to organize event data by functional area. For example, operating system and virtualization situation events can be configured to forward to one Netcool/OMNIbus event server, while WebSphere Application Server, DB2, and other middleware situation events can be configured to forward to another Netcool/OMNIbus event server.

Installation and configuration

If you are installing IBM Tivoli Monitoring and Netcool/OMNIbus event integration for the first time, complete the tasks in the following table. However, if you are upgrading an existing event integration environment, see "Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration" on page 746.

Table 133. Installation and configuration: one hub Tivoli Enterprise Monitoring Server and multiple Netcool/OMNIbus ObjectServers

	Task	Architecture Type	Administrator
1.	Install Netcool/OMNIbus ObjectServer if it is not already installed or upgrade it to the fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
2.	*For each Netcool/OMNIbus ObjectServer:* Install the Netcool/OMNIbus Probe for Tivoli EIF if it is not already installed or upgrade it to the release required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
3.	*For each Netcool/OMNIbus ObjectServer:* "Installing the IBM Tivoli Monitoring Event Synchronization Component" on page 720.	Unidirectional and bidirectional	Netcool/OMNIbus
4.	*For each Netcool/OMNIbus ObjectServer:* "Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers" on page 733.	Unidirectional and bidirectional	Netcool/OMNIbus
5.	*For each Netcool/OMNIbus ObjectServer:* "Changing the default deduplication trigger" on page 735.	Unidirectional and bidirectional	Netcool/OMNIbus

Table 133. Installation and configuration: one hub Tivoli Enterprise Monitoring Server and multiple Netcool/OMNIbus ObjectServers (continued)

	Task	Architecture Type	Administrator
6.	*For each Netcool/OMNIbus ObjectServer:* "Configuring the OMNIbus server for program execution from scripts" on page 742.	Bidirectional	Netcool/OMNIbus
7.	*For each Netcool/OMNIbus ObjectServer:* Start the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see "Starting and stopping the IBM Tivoli Monitoring Situation Update Forwarder" on page 739.	Bidirectional	Netcool/OMNIbus
8.	*For each Netcool/OMNIbus ObjectServer:* "Configuring the Netcool/OMNIbus EIF probe" on page 740.	Unidirectional and bidirectional	Netcool/OMNIbus
9.	"Configuring the hub monitoring server to forward events" on page 743 to one of the EIF Probes. This probe is the default EIF receiver.	Unidirectional and bidirectional	IBM Tivoli Monitoring
10.	*For each additional Netcool/OMNIbus ObjectServer:* Use the `tacmd createEventDest` command to create an event destination for the EIF probe associated with the ObjectServer. See the *IBM Tivoli Monitoring: Command Reference* for a description.	Unidirectional and bidirectional	IBM Tivoli Monitoring
11.	For each situation, configure which event destinations the events for the situation should be sent to.	Unidirectional and bidirectional	IBM Tivoli Monitoring Operator
12.	"Verifying installation and configuration" on page 745.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus
13.	Determine if additional configuration tasks should be performed. See "Customizing Event Integration" on page 750.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Multiple IBM Tivoli Monitoring agents and one Netcool/OMNIbus ObjectServer

In this scenario, IBM Tivoli Monitoring agents are configured to forward events directly to either the Netcool/OMNIbus Probe for Tivoli EIF or the Netcool/OMNIbus Probe for SNMP. This architecture eliminates the interaction with the hub monitoring server. If your environment uses firewalls, you must ensure any IBM Tivoli Monitoring agents and Netcool/OMNIbus probes can communicate with each other and are not restricted by the firewall. Because events are forwarded directly to Netcool/OMNIbus by the IBM Tivoli Monitoring agents, no bidirectional communication is required. Any actions taken on events through Netcool/OMNIbus are not forwarded back to the IBM Tivoli Monitoring agent.

The Netcool/OMNIbus Probe for Tivoli EIF supports both SSL and non-SSL connections with monitoring agents. If you want to use an SSL connection, you must have version 12.0 or later of the probe.

You can configure the agents to send heartbeat events at regular intervals to Netcool/OMNIbus. You can enable heartbeat automation in the Netcool/OMNIbus ObjectServer. By using heartbeat automation, a "Heartbeat Missing" event is opened if a heartbeat event is not received from an agent in the expected time frame. By enabling heartbeat automation, the Netcool/OMNIbus operator knows when a monitoring agent is not available. The heartbeat interval is configurable.

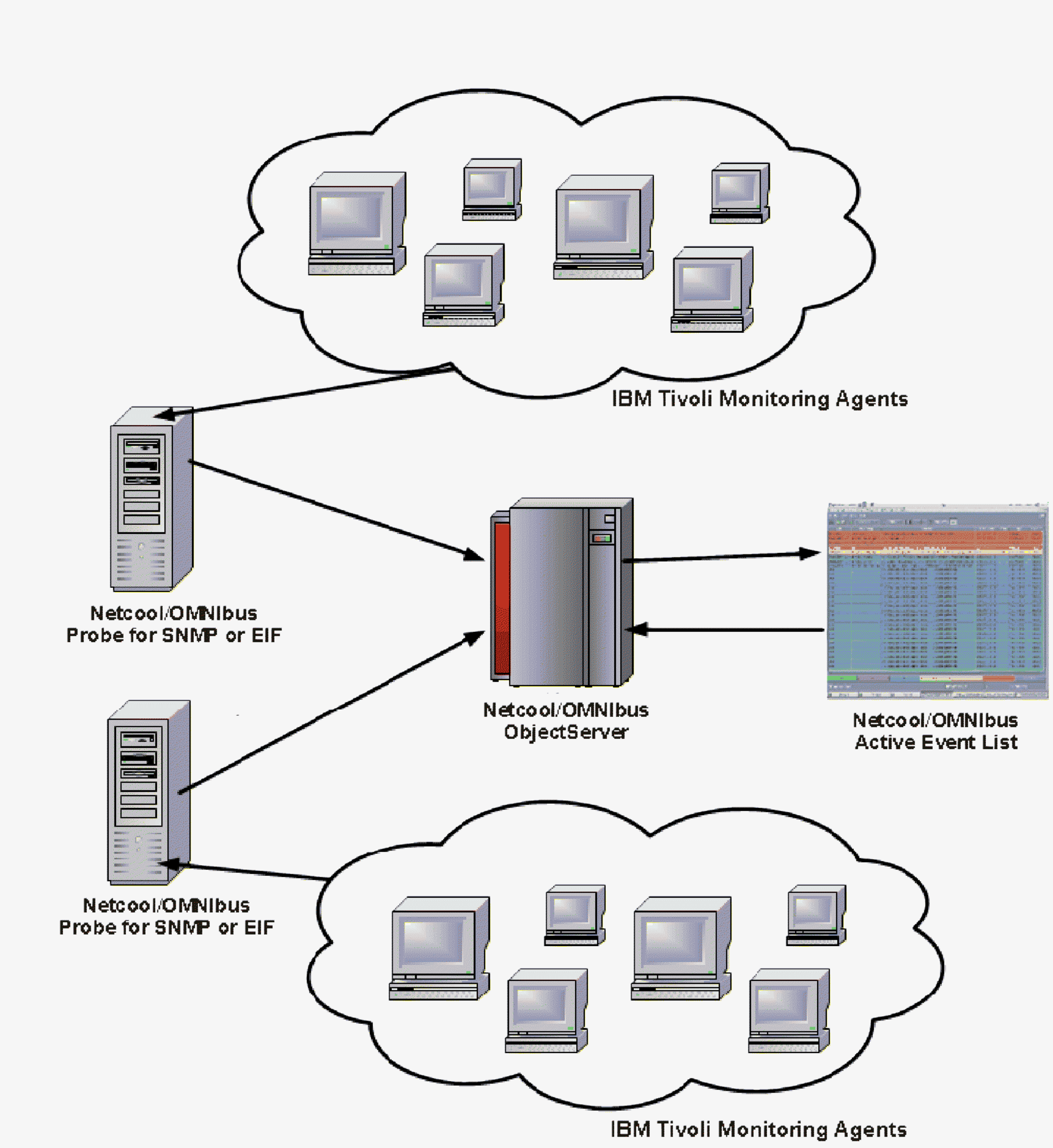

IBM Tivoli Monitoring Agents
Netcool/OMNIbus Probe for SNMP or EIF
Netcool/OMNIbus ObjectServer
Netcool/OMNIbus Active Event List
Netcool/OMNIbus Probe for SNMP or EIF
IBM Tivoli Monitoring Agents

Figure 166. Multiple IBM Tivoli Monitoring Agents and one Netcool/OMNIbus ObjectServer

Uses

Use this scenario when you have agents that are not connected to IBM Tivoli Monitoring Server. You can also use this architecture if you want monitoring server connected agents to send their critical events to the Netcool/OMNIbus Objectserver. The agents can send events directly to the OMNIbus server through the Netcool/OMNIbus Probe for Tivoli EIF or the Probe for SNMP.

Installation and configuration for agents sending events to the Netcool/OMNIbus Probe for Tivoli EIF

If you are installing IBM Tivoli Monitoring and Netcool/OMNIbus event integration for the first time, perform the tasks in the following table. However, if you are upgrading an existing event integration environment, see “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746.

Note: If agents and a hub monitoring server are sending events to the same Netcool/OMNIbus ObjectServer and probe, you must only perform tasks 1 to 7 once.

Table 134. Installation and configuration for agents sending events to the Netcool/OMNIbus Probe for Tivoli EIF

	Task	Architecture type	Administrator
1.	Install Netcool/OMNIbus ObjectServer if it is not already installed or upgrade it to the fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also “Required software for event integration with Netcool/OMNIbus” on page 158.	Unidirectional	Netcool/OMNIbus
2.	Install the Netcool/OMNIbus Probe for Tivoli EIF if it is not already installed or upgrade it to the release required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also “Required software for event integration with Netcool/OMNIbus” on page 158.	Unidirectional	Netcool/OMNIbus
3.	“Installing the IBM Tivoli Monitoring Event Synchronization Component” on page 720.	Unidirectional	Netcool/OMNIbus
4.	“Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers” on page 733.	Unidirectional	Netcool/OMNIbus
5.	“Changing the default deduplication trigger” on page 735.	Unidirectional	Netcool/OMNIbus

Table 134. Installation and configuration for agents sending events to the Netcool/OMNIbus Probe for Tivoli EIF (continued)

	Task	Architecture type	Administrator
6.	Enable heartbeat automation in OMNIbus. For instructions on enabling OMNIbus heartbeat automation, see the *Agent Autonomy* chapter in the *IBM Tivoli Monitoring: Administrator's Guide*.	Unidirectional	Netcool/OMNIbus
7.	"Configuring the Netcool/OMNIbus EIF probe" on page 740. **Note:** If an SSL connection will be used between the probe and a monitoring agent, ensure that you have version 12.0 or later of the probe.	Unidirectional	Netcool/OMNIbus
8.	For each IBM Tivoli Monitoring Agent: configure the IBM Tivoli Monitoring Agent to forward events to the EIF Probe. See the *Agent autonomy* chapter in the *IBM Tivoli Monitoring: Administrator's Guide* for information on configuring agents to send EIF events to OMNIbus using either SSL or non-SSL connections.	Unidirectional	IBM Tivoli Monitoring
9.	"Verifying installation and configuration" on page 745.	Unidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Installation and configuration if agents are sending events to the Netcool/OMNIbus Probe for SNMP

Table 135. Installation and configuration if agents are sending events to the Netcool/OMNIbus Probe for SNMP

	Task	Architecture Type	Administrator
1.	Install Netcool/OMNIbus ObjectServer if it is not already installed or upgrade it to the fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional	Netcool/OMNIbus

Table 135. Installation and configuration if agents are sending events to the Netcool/OMNIbus Probe for SNMP (continued)

	Task	Architecture Type	Administrator
2.	Install the Netcool/OMNIbus Probe for SNMP if it is not already installed or upgrade it to the release required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional	Netcool/OMNIbus
3.	Enable heartbeat automation in OMNIbus. For instructions on enabling OMNIbus heartbeat automation, see the *Agent Autonomy* chapter in the *IBM Tivoli Monitoring: Administrator's Guide*.	Unidirectional	Netcool/OMNIbus
4.	Configure the Netcool/OMNIbus Probe for SNMP to process IBM Tivoli Monitoring agent events. For instructions on configuring OMNIbus to receive SNMP alerts, see the *Agent Autonomy* chapter in the *IBM Tivoli Monitoring: Administrator's Guide*.	Unidirectional	Netcool/OMNIbus
5.	For each IBM Tivoli Monitoring Agent: configure the IBM Tivoli Monitoring Agent to forward events to the SNMP Probe. See the *Agent Autonomy* chapter in the *IBM Tivoli Monitoring: Administrator's Guide* for information on configuring agents to send events to OMNIbus.	Unidirectional	IBM Tivoli Monitoring
6.	"Verifying installation and configuration" on page 745.	Unidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Netcool/OMNIbus Multitiered and High-availability Architecture

The IBM Tivoli Netcool/OMNIbus product can be deployed in a multitiered configuration to increase performance and event handling capacity. High-availability is also supported by adding primary and backup servers to the tiers. In the standard multitiered environment, three sets of Netcool/OMNIbus ObjectServers are included:

1. The collection tier includes the ObjectServers to which probes connect. In a high-availability architecture, primary and backup pairs of ObjectServers are included in this tier. Each ObjectServer in the collection tier is connected to an ObjectServer in the aggregation tier by using a unidirectional gateway.

2. The aggregation tier includes up to two ObjectServers if you are using a high-availability architecture. The ObjectServers are connected by a bidirectional gateway. The bulk of the event processing occurs in the aggregation tier.
3. The display tier includes one or more ObjectServers. They are connected to an ObjectServer in the aggregation tier by using a unidirectional gateway. The Netcool/OMNIbus desktop event list users and web GUI users connect to ObjectServers in this tier. The operator connects their Netcool/OMNIbus Desktop UI to the display tier in dual-server desktop mode. Events are retrieved from the display ObjectServers, but updates made to events go to both the display and the aggregation tiers.

High-availability can also be achieved in a single-tier environment by having a primary and backup ObjectServer and a bidirectional gateway between them. In a single-tier architecture, the probes and user interfaces connect directly to these ObjectServers and the ObjectServers perform the collection, aggregation, and display functions.

The IBM Tivoli Monitoring triggers are assigned to the primary_only trigger group. In a high-availability Netcool/OMNIbus architecture, the standard Netcool/OMNIbus automations enable the triggers in the primary_only trigger group on the acting primary ObjectServer and disable them on the backup ObjectServer.

For more information about the Netcool/OMNIbus ObjectServer multitiered and high-availability architecture and setup instructions, see the Netcool/OMNIbus information center: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm.

You can also set up peer-to-peer failover mode for Netcool/OMNIbus probes to reduce event loss. Two instances of a probe can run simultaneously in a peer-to-peer failover relationship. One instance is designated as the master; the other instance acts as a slave and is on hot standby. If the master instance fails, the slave instance is activated. For more information about configuring and using master and slave probes for high availability, see the Netcool/OMNIbus information center: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm.

In Netcool/OMNIbus multitier and high-availability architectures, the Tivoli Enterprise Monitoring Servers are configured to forward events to the ObjectServers via the probes. If probe failover mode is used, you must configure situation events in IBM Tivoli Monitoring with multiple event destinations – one destination to the master probe and a second destination to the slave probe.

In a multitiered architecture, IBM Tivoli Monitoring provides triggers to add to the collection tier, database schema updates and triggers for the aggregation tier, and database schema updates for the display tier. The gateways also must be updated with mapping entries. In this architecture, the IBM Tivoli Monitoring Situation Update Forwarder is installed on each ObjectServer in the aggregation tier.

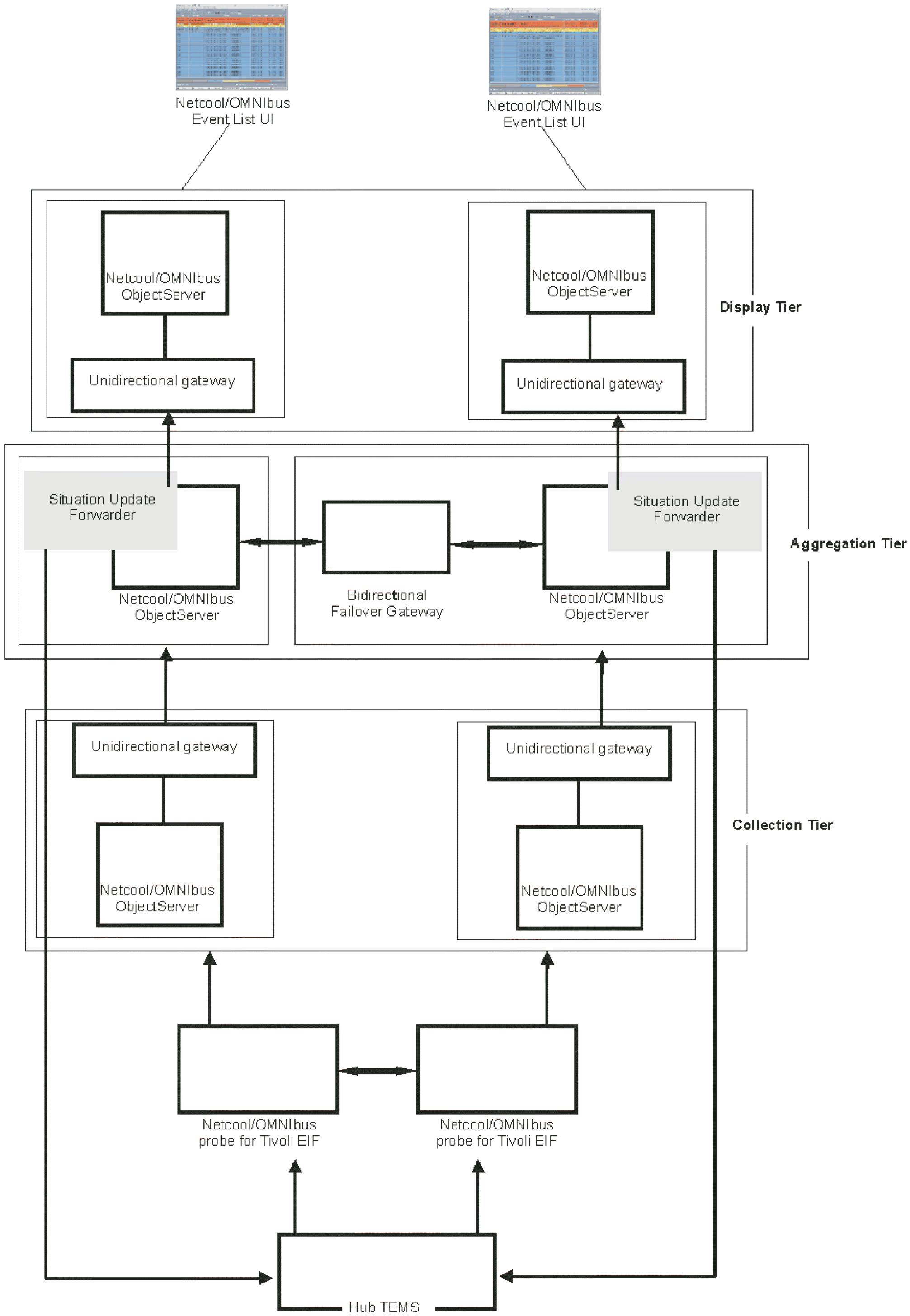

Netcool/OMNIbus Event List UI
Netcool/OMNIbus Event List UI
Netcool/OMNIbus ObjectServer
Unidirectional gateway
Netcool/OMNIbus ObjectServer
Unidirectional gateway
Display Tier
Situation Update Forwarder
Netcool/OMNIbus ObjectServer
Bidirectional Failover Gateway
Situation Update Forwarder
Netcool/OMNIbus ObjectServer
Aggregation Tier
Unidirectional gateway
Netcool/OMNIbus ObjectServer
Unidirectional gateway
Netcool/OMNIbus ObjectServer
Collection Tier
Netcool/OMNIbus probe for Tivoli EIF
Netcool/OMNIbus probe for Tivoli EIF
Hub TEMS

Figure 167. Standard multitiered architecture with high availability

In a single-tier high-availability architecture, the IBM Tivoli monitoring triggers and database schema updates are applied to the primary and backup ObjectServers and mapping entries are configured for the bidirectional failover gateway between the ObjectServers. The IBM Tivoli Monitoring Situation Update Forwarder is installed on each ObjectServer.

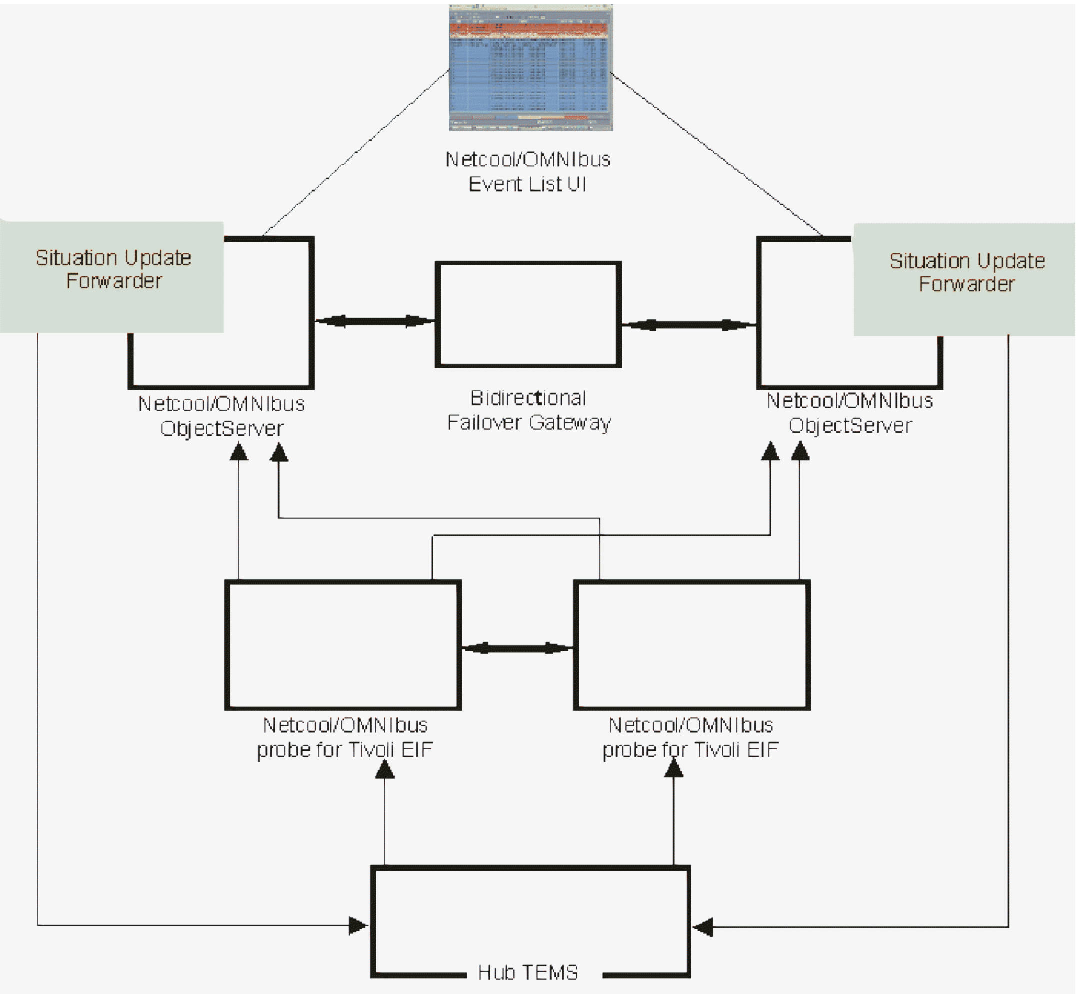


Figure 168. single-tier high-availability architecture

Uses

You can use the Netcool/OMNIbus multitiered architecture if you need to add scalability to your Netcool/OMNIbus environment to increase performance and event handling. High availability can be added for the ObjectServers in your single-tier and multitier architectures so that your environment can continue to operate at full capacity (and with minimal event loss) in the event of ObjectServer failure.

Adding peer-to-peer failover for the probes can be used to reduce event loss in the event of a probe failure.

Installation and configuration

If you are installing IBM Tivoli Monitoring and Netcool/OMNIbus event integration for the first time, perform the tasks in Table 136 on page 713. However, if you are upgrading an existing event integration environment, see “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746.

Table 136. Installation and configuration: Netcool/OMNIbus Multitiered and High-Availability Architecture ObjectServers

	Task	Architecture Type	Administrator
1.	Install Netcool/OMNIbus ObjectServers in a multitiered architecture or single-tier high availability environment if they are not already installed, or upgrade them to the fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
2.	Install the Netcool/OMNIbus Probe for Tivoli EIF if it is not already installed or upgrade it to the release required by IBM Tivoli Monitoring. If you are using peer-to-peer probe failover, install and configure master and slave EIF probes. See the Netcool/OMNIbus Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
3.	For each Netcool/OMNIbus ObjectServer in the aggregation tier or for each ObjectServer in a single-tier architecture, see "Installing the IBM Tivoli Monitoring Event Synchronization Component" on page 720. **Note:** If you are installing Netcool/OMNIbus ObjectServer in a high availability environment, you must install the event synchronization component on the primary and backup ObjectServer in the single-tier architecture and in the aggregation tier.	Unidirectional and bidirectional	Netcool/OMNIbus

Table 136. Installation and configuration: Netcool/OMNIbus Multitiered and High-Availability Architecture ObjectServers (continued)

	Task	Architecture Type	Administrator
4.	For each gateway between ObjectServers in a multitier or single-tier architecture, see "Updating gateways to map attributes" on page 730.	Unidirectional and bidirectional	Netcool/OMNIbus
5.	For each bidirectional failover gateway in an aggregation tier or single-tier with high availability, see "Updating the bidirectional failover gateway to replicate tables" on page 731.	Unidirectional and bidirectional	Netcool/OMNIbus
6.	*For each Netcool/OMNIbus ObjectServer in the collection tier:* "Updating the OMNIbus database schema in the collection tier" on page 732.	Unidirectional and bidirectional	Netcool/OMNIbus
7.	For each Netcool/OMNIbus ObjectServer in the aggregation tier or for each ObjectServer in the single-tier architecture, see "Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers" on page 733.	Unidirectional and bidirectional	Netcool/OMNIbus
8.	For each ObjectServer in the aggregation tier or in a single-tier architecture, see "Changing the default deduplication trigger" on page 735	Unidirectional and bidirectional	Netcool/OMNIbus
9.	For each Netcool/OMNIbus ObjectServer in the aggregation tier or in the single-tier architecture, see "Configuring the OMNIbus server for program execution from scripts" on page 742.	Bidirectional	Netcool/OMNIbus
10.	For each Netcool/OMNIbus ObjectServer in the aggregation tier or in the single-tier architecture, Start the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see "Starting and stopping the IBM Tivoli Monitoring Situation Update Forwarder" on page 739.	Bidirectional	Netcool/OMNIbus
11.	For each Netcool/OMNIbus ObjectServer in the display tier, see "Updating the OMNIbus database schema in the display tier" on page 738.	Unidirectional and bidirectional	Netcool/OMNIbus

Table 136. Installation and configuration: Netcool/OMNIbus Multitiered and High-Availability Architecture ObjectServers (continued)

	Task	Architecture Type	Administrator
12.	Restart each gateway that was updated in steps 4 and 5 based on the following suggested restart order for a multitier architecture: 1. Restart gateways between the collection and aggregation tiers. 2. Restart bidirectional failover gateways in the aggregation tier. 3. Restart gateways between the aggregation and display tiers. See the Netcool/OMNIbus Information Center for more details on restarting the gateways: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm.	Unidirectional and bidirectional	Netcool/OMNIbus
13.	For each EIF probe, see "Configuring the Netcool/OMNIbus EIF probe" on page 740. **Note:** If you are using a multitier architecture you must also ensure that the EIF probes are configured to connect to an ObjectServer in the collection tier.	Unidirectional and bidirectional	Netcool/OMNIbus
14.	For each hub monitoring server, see "Configuring the hub monitoring server to forward events" on page 743. If you are using peer-to-peer probe failover, configure the Hub to forward events to the master probe.	Unidirectional and bidirectional	IBM Tivoli Monitoring
15.	If you are using peer-to-peer probe failover, use the `tacmd createEventDest` command to create an event destination for the EIF slave probe. For more information, see the *IBM Tivoli Monitoring: Command Reference*.	Unidirectional and bidirectional	IBM Tivoli Monitoring
16.	If you are using peer-to-peer probe failover, configure each situation to be forwarded to the event destination for the master probe and to the event destination for the slave probe using the Tivoli Enterprise Portal Situation Editor or the `tacmd createsit` or `editsit` commands.	Unidirectional and bidirectional	IBM Tivoli Monitoring
17.	"Verifying installation and configuration" on page 745.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Table 136. Installation and configuration: Netcool/OMNIbus Multitiered and High-Availability Architecture ObjectServers (continued)

	Task	Architecture Type	Administrator
18.	Determine if additional configuration tasks should be performed. See "Customizing Event Integration" on page 750.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Integration with Tivoli Business Service Manager

In this scenario, IBM Tivoli Monitoring is configured to forward events to Netcool/OMNIbus. The Tivoli Business Service Manager Data Server analyzes the events in the Netcool/OMNIbus ObjectServer for matches against the incoming-status rules configured for service models. If the event's data matches a status rule, the status of the service model is changed accordingly.

Tivoli Business Service Manager can be used to install Netcool/OMNIbus ObjectServer and the Netcool/OMNIbus Probe for Tivoli EIF if you do not have an existing installation of these components. See the Tivoli Business Service Manager information center for details.

You can use any of the architecture scenarios described in this chapter for configuring forwarding of events from IBM Tivoli Monitoring to Netcool/OMNIbus when integrating with Tivoli Business Service Manager.

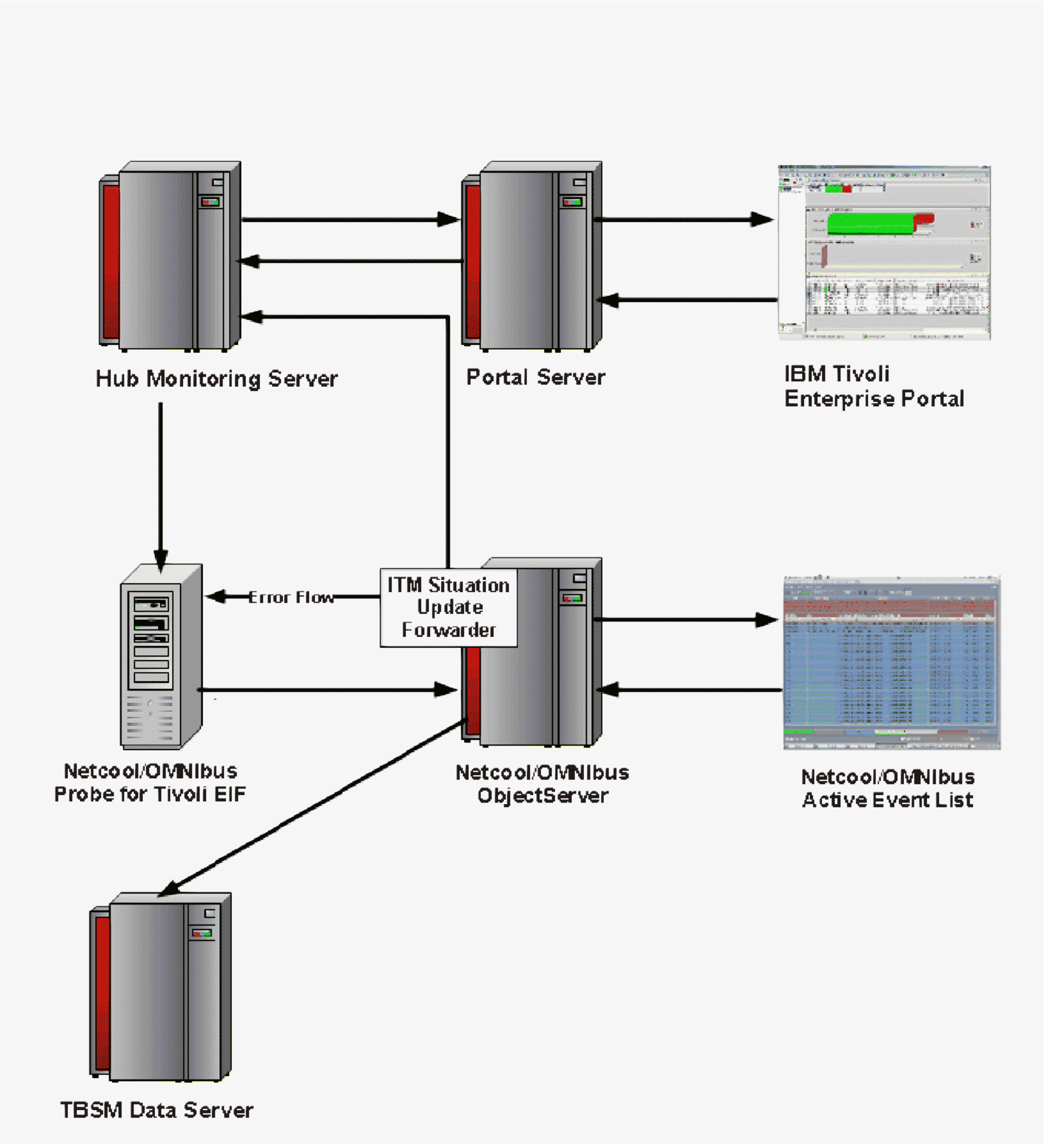

Hub Monitoring Server
Portal Server
IBM Tivoli Enterprise Portal
Error Flow
ITM Situation Update Forwarder
Netcool/OMNIbus Probe for Tivoli EIF
Netcool/OMNIbus ObjectServer
Netcool/OMNIbus Active Event List
TBSM Data Server

Figure 169. Integrating IBM Tivoli Monitoring, Netcool/OMNIbus, and Tivoli Business Service Manager

Uses

If you are using Tivoli Business Service Manager to manage your business services, and you want to use monitoring events to update business service status, then you must use the IBM Tivoli Monitoring probe rules that set the OMNIbus attributes required by Tivoli Business Service Manager.

Installation and configuration

If you are installing IBM Tivoli Monitoring and Netcool/OMNIbus event integration for the first time, perform the tasks in the following table. However, if you are upgrading an existing event integration environment, see “Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration” on page 746.

You should use the IBM Tivoli Monitoring probe rules file and `.sql` files provided with IBM Tivoli Monitoring instead of the related Tivoli Monitoring rules file and `.sql` files provided with Tivoli Business Service Manager Version 4.2.1.

Table 137. Installation and configuration

	Task	Architecture Type	Administrator
1.	Install Netcool/OMNIbus ObjectServer using the Tivoli Business Service Manager installation program or follow the procedures in the Tivoli Business Service Manager documentation for updating an existing ObjectServer with database schema updates required by Tivoli Business Service Manager. See the Tivoli Business Service Manager Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.tivoli.itbsm.doc/welcome.htm.	Unidirectional and bidirectional	Netcool/OMNIbus
2.	Install the Netcool/OMNIbus Probe for Tivoli EIF using the Tivoli Business Service Manager installation program or follow the procedures in the Tivoli Business Service Manager documentation for updating an existing EIF probe. See the Tivoli Business Service Manager Information Center for detailed instructions on this task: http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.tivoli.itbsm.doc/welcome.htm.	Unidirectional and bidirectional	Netcool/OMNIbus
3.	"Installing the IBM Tivoli Monitoring Event Synchronization Component" on page 720.	Unidirectional and bidirectional	Netcool/OMNIbus
4.	"Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers" on page 733.	Unidirectional and bidirectional	Netcool/OMNIbus
5.	"Changing the default deduplication trigger" on page 735.	Unidirectional and bidirectional	Netcool/OMNIbus
6.	"Configuring the OMNIbus server for program execution from scripts" on page 742.	Bidirectional	Netcool/OMNIbus
7.	Start the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see "Starting and stopping the IBM Tivoli Monitoring Situation Update Forwarder" on page 739.	Bidirectional	Netcool/OMNIbus

Table 137. Installation and configuration (continued)

	Task	Architecture Type	Administrator
8.	If you have more than one monitoring server forwarding events to the Netcool/OMNIbus ObjectServer, define additional monitoring servers to the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see “Updating the IBM Tivoli Monitoring Situation Forwarder to forward event status updates to additional monitoring servers” on page 752.	Bidirectional	Netcool/OMNIbus
9.	To use the IBM Tivoli Monitoring probe rules include files for Tivoli Business Service Manager integration, see “Configuring the Netcool/OMNIbus EIF probe” on page 740. **Note:** Some monitoring agents also have their own probe rules include file. See the user guide for each monitoring agent that you plan to install to determine if it has a rules include file that should be copied to the computer system where the probe is installed and uncommented in the probe's main rules file, `tivoli_eif.rules`.	Unidirectional and bidirectional	Netcool/OMNIbus
10.	*For each hub monitoring server:* “Configuring the hub monitoring server to forward events” on page 743.	Unidirectional and bidirectional	IBM Tivoli Monitoring
11.	“Verifying installation and configuration” on page 745.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus
12.	Determine if additional configuration tasks should be performed. See “Customizing Event Integration” on page 750.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Installation and configuration

The following products must be installed and configured before you install the event synchronization component and configure event forwarding for Netcool/OMNIbus:

- IBM Tivoli Netcool/OMNIbus V7.2 or later
- IBM Tivoli Netcool/OMNIbus probe for Tivoli EIF version 10 or later and the non-native probe version 12 or later
- IBM Tivoli Monitoring

Note: fix packs might be required for each of the supported Netcool/OMNIbus releases, for more information see “Required software for event integration with Netcool/OMNIbus” on page 158.

The steps for installing and configuring event integration depend on the type of architecture you are installing. The section on "Architecture scenarios" on page 696 describes various architecture types and steps to implement the solution. First select the type of architecture you want to implement, and then follow the steps outlined for that architecture type.

Setting up event forwarding between IBM Tivoli Monitoring and Netcool/OMNIbus involves both IBM Tivoli Monitoring and Netcool/OMNIbus configuration tasks, therefore setup should be coordinated between both the IBM Tivoli Monitoring and Netcool/OMNIbus administrators.

Installing the IBM Tivoli Monitoring Event Synchronization Component

Installing the IBM Tivoli Monitoring Event Synchronization component also installs the IBM Tivoli Monitoring Situation Update Forwarder, IBM Tivoli Monitoring rules file for the Netcool/OMNIbus Probe for Tivoli EIF, and SQL files to update the Netcool/OMNIbus ObjectServer database for EIF event handling. On Windows, a Situation Update Forwarder service is also created. The tasks in this section should be completed by the Netcool/OMNIbus administrator.

The IBM Tivoli Monitoring Event Synchronization Component installer is located on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.

- Bidirectional architecture:

  Install the IBM Tivoli Monitoring Event Synchronization Component on the host of the Netcool/OMNIbus ObjectServer. The bidirectional architecture requires the IBM Tivoli Monitoring Situation Update Forwarder to be located on the Netcool/OMNIbus ObjectServer.
- Unidirectional architecture:

  A unidirectional architecture does not need the IBM Tivoli Monitoring Situation Update Forwarder, therefore the IBM Tivoli Monitoring Event Synchronization Component can be installed on any system. The files required for later setup steps, such as probe rules and database SQL files, can be copied manually to the Netcool/OMNIbus Probe for Tivoli EIF and Netcool/OMNIbus ObjectServer. During installation you are asked questions that only apply to the bidirectional architecture. You must specify a valid value for these parameters even though they will not be used.

If you are using a Netcool/OMNIbus multitier architecture, install the Tivoli Monitoring Event Synchronization component with each ObjectServer in the aggregation tier. If you are using a single-tier architecture, install the event synchronization component with each of your ObjectServers.

Use one of the following three methods to install the Tivoli Monitoring Event Synchronization component:
- Installing from a wizard
- Installing from the command-line
- Installing from the command-line using a silent installation

Note:

- You cannot install event synchronization for Netcool/OMNIbus on the same system as an IBM Tivoli Enterprise Console event server.
- If the ObjectServer is running on Windows 2003 and you are planning to install the event synchronization remotely (using a program such as Terminal Services to connect to that Windows 2003 computer), you must run the **`change user /install`** command before you run the installation, which puts the computer into the required "install" mode. After the installation, run the **change user /execute** command to return the computer to its previous mode.
- If you have a monitoring server on an operating system like UNIX or Linux, you must configure your TCP/IP network services in the `/etc/hosts` file to return the fully qualified host name if your Netcool/OMNIbus ObjectServer must use the fully qualified hostname to send event status updates to the monitoring server.
- Linux or UNIX users can run the event synchronization installer under a root or a non-root user ID. If you are installing as a non-root user the `/etc/TME` directory must already exist.

- The results of the event synchronization installation are written to the %TEMP%\ itm_tec_event_sync_install.log file on Windows and to the /tmp/ itm_tec_event_sync_install.log file on UNIX.

Installing from a wizard

Take the following steps to install event synchronization by using the installation wizard:

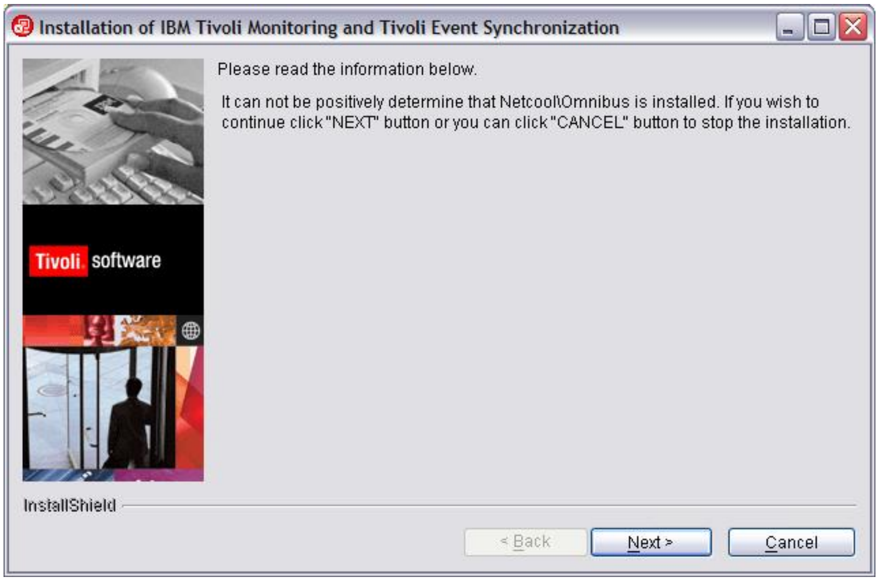
1. On the computer where the ObjectServer is installed, launch the event synchronization installation:
   - On Windows systems, double-click the ESync2300Win32.exe file in the tec subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
   - On Linux or UNIX operating systems, change to the tec subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image and run the following command:

     ```
     ESync2300operating_system.bin
     ```

     where *operating_system* is the operating system you are installing on (Aix, HP11, Linux, linux390, or Solaris). For example, run the following command on an AIX computer:

     ```
     ESync2300Aix.bin
     ```

     If the installer cannot locate OMNIbus in its usual place, the following window is displayed. Click **Next** to continue installing the event synchronization:


Installation of IBM Tivoli Monitoring and Tivoli Event Synchronization
Tivoli software
InstallShield
< Back
Next >
Cancel

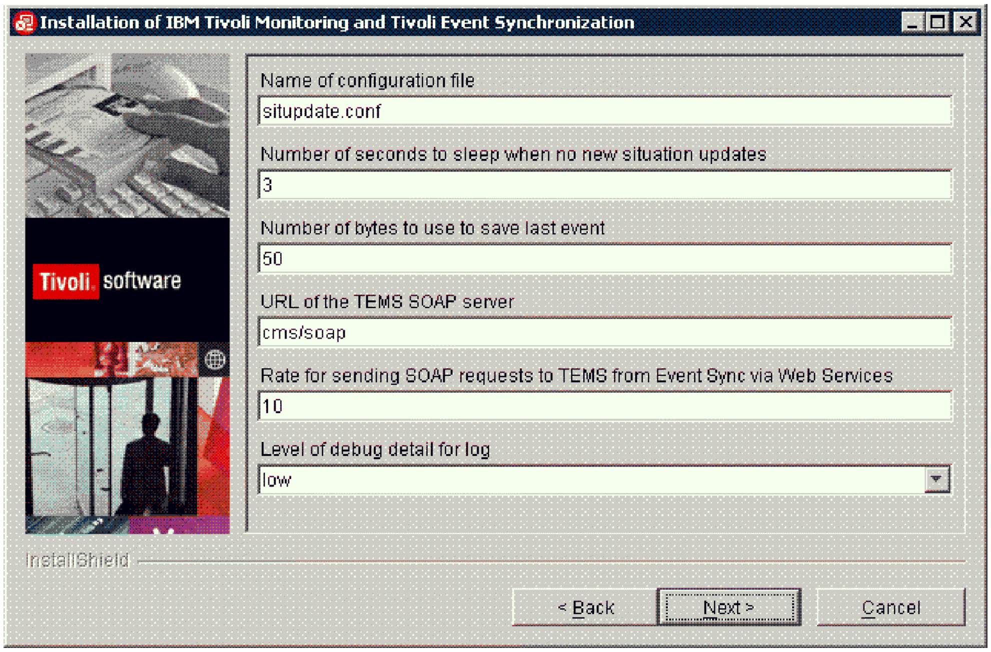
Please read the information below.

It can not be positively determine that Netcool\Omnibus is installed. If you wish to continue click "NEXT" button or you can click "CANCEL" button to stop the installation.

Figure 170. Installation of IBM Tivoli Monitoring and Tivoli Event Synchronization

2. Click **Next** on the Welcome window.
3. Review the license agreement, select **I accept the terms in the license agreement** and click **Next**.
4. Click **Next** to install the synchronization component in the default location, or use the **Browse** button to select another location.
   - On Windows systems, the default installation directory is C:\Program Files\IBM\SitForwarder.
   - On Linux/UNIX systems, the default installation directory is /opt/IBM/SitForwarder.

   Click **Next** to continue.
5. Complete the fields in the installation windows by using the configuration values described in Table 138 on page 722 and click **Next**.


Installation of IBM Tivoli Monitoring and Tivoli Event Synchronization
Tivoli. software
InstallShield
< Back
Next >
Cancel

Name of configuration file
situpdate.conf

Number of seconds to sleep when no new situation updates
3

Number of bytes to use to save last event
50

URL of the TEMS SOAP server
cms/soap

Rate for sending SOAP requests to TEMS from Event Sync via Web Services
10

Level of debug detail for log
low

Figure 171. Netcool/OMNIbus event synchronization configuration fields

Table 138. Netcool/OMNIbus event synchronization configuration fields

Field	Description
Name of configuration file	*For bidirectional architecture only:* The name of the file where the IBM Tivoli Monitoring Situation Update Forwarder configuration information is stored. The default name is `situpdate.conf`.
Number of seconds to sleep when no new situation updates	*For bidirectional architecture only:* The polling interval, in seconds used by the IBM Tivoli Monitoring Situation Update Forwarder to determine if there are new event updates to send from Netcool/OMNIbus to the hub Tivoli Enterprise Monitoring Server. The minimum value is 1, while the default value is 3. If no situation events are found, the IBM Tivoli Monitoring Situation Update Forwarder rests for 3 seconds.
Number of bytes to use to save last event	*For bidirectional architecture only:* Number of bytes that the IBM Tivoli Monitoring Situation Update Forwarder uses when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) value is 50.
URL of the Tivoli Enterprise Monitoring Server SOAP Server	*For bidirectional architecture only:* The URL for the SOAP Server configured on the computer where the hub monitoring server is running. The default value is `cms/soap`. Do not change this value. The actual URL for the SOAP Server is dynamically constructed from the incoming event.
Rate for sending SOAP requests to the monitoring server from the OMNIbus probe via web services	*For bidirectional architecture only:* The maximum number of event updates sent by the IBM Tivoli Monitoring Situation Update Forwarder to the hub monitoring server at one time. The minimum (and default) value is 10 events.

Table 138. Netcool/OMNIbus event synchronization configuration fields (continued)

Field	Description
Level of debug detail for log	*For bidirectional architecture only:* The level of information for event synchronization that is logged. The following choices are available: • Low (default) • Medium • Verbose

6. Complete the fields on the installation window by using the values described in Table 139 and click **Next**.

Table 139. Netcool/OMNIbus event synchronization configuration fields, continued

Field	Description
Maximum size of any single cache file	*For bidirectional architecture only:* The maximum permitted size, in bytes, for any one event cache file used by the IBM Tivoli Monitoring Situation Update Forwarder. The minimum (and default) value is 50000. Do not use commas when specifying this value (specify 50000 instead of 50,000).
Maximum number of caches files	*For bidirectional architecture only:* The maximum number of event caches files at any given time used by the IBM Tivoli Monitoring Situation Update Forwarder. The minimum value is 2, while the default value is 10. When this value is reached, the oldest file is deleted to make room for a new file.
Directory for cache files to be located	*For bidirectional architecture only:* The location where IBM Tivoli Monitoring Situation Update Forwarder event cache files are located. The default locations are as follows: • On Windows: `C:\Program Files\IBM\SitForwarder\persistence`. • On UNIX: `/opt/IBM/TEC/SitForwarder/persistence`

7. Type the following information for each hub monitoring server for which you want to use bidirectional synchronization and click **Add**. You must specify information for at least one hub monitoring server even if you are only using unidirectional synchronization.

   Host name
   The fully qualified host name for the computer where the hub monitoring server is running. The name must match the information that will be in events coming from this hub monitoring server.

   User ID
   The user ID to access the computer where the hub monitoring server is running.

   Password
   The password to access the computer.

   Confirmation
   The password, again.

   Repeat this step to add a short host name for the same hub monitoring server by specifying the short host name value for the `Host name` parameter. You can add information for up to 10 hub monitoring servers host names in this wizard. If you want to add additional monitoring servers, do so after installation by using the steps outlined in "Defining additional monitoring servers to the event server" on page 664 on page 664. If you have configured the Hot Standby feature, you must configure the host name information for both the primary and secondary hub monitoring servers.

8. When you have provided information about all of the monitoring servers, click **Next**.

   A summary window is displayed.

9. Click **Next** to proceed.

   The installation begins and a progress indicator is displayed. When the installation is completed, a message is displayed telling you that the installation has been successful.

10. Click **Finish** to exit the wizard.

    Note: If any configuration errors occurred during installation and configuration, you are directed to a log file that contains additional troubleshooting information.

Installing from the command-line

Use the following steps to install the event synchronization from the command-line on your event server:

1. Change to the `tec` directory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
2. Run the following command to launch the installation:

   On Windows:

   ```
   ESync2300Win32.exe -console
   ```

   On UNIX or Linux:

   ```
   ESync2300operating_system.bin -console
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, or `Solaris`). For example, run the following command on an AIX computer:

   ```
   ESync2300Aix.bin -console
   ```

   The following prompt is displayed:

   ```
   Press 1 for Next, 3 to Cancel or 4 to Redisplay [1]
   ```

3. Type `1` to start the installation and press Enter.

   The following prompt is displayed:

   ```
   Software Licensing Agreement:
   Press Enter to display the license agreement on your screen. Please
   read the agreement carefully before installing the Program. After
   reading the agreement, you will be given the opportunity to accept it
   or decline it. If you choose to decline the agreement, installation
   will not be completed and you will not be able to use the Program.
   ```

4. Press Enter to display the software license agreement.
5. Type `1` and press Enter to accept the license.

   The following prompt is displayed:

   ```
   Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
   ```

6. Type `1` and press Enter to continue.

   The following prompt is displayed:

   ```
   Name of configuration file [situpdate.conf]
   ```

7. Press Enter to use the default configuration file `situpdate.conf` for the IBM Tivoli Monitoring Situation Update Forwarder. If you want to use a different configuration file, type the name and press Enter.

   The following prompt is displayed:

   ```
   Number of seconds to sleep when no new situation updates [3]
   ```

8. Type the number of seconds that you want to use for the polling interval used by the IBM Tivoli Monitoring Situation Update Forwarder to determine if there are new event updates to send from Netcool/OMNIbus to the hub Tivoli Enterprise Monitoring Server and press Enter. The default value is `3`, while the minimum value is `1`. This configuration value is used for bidirectional architecture only.

   The following prompt is displayed:

```
Number of bytes to use to save last event [50]
```

9. Type the number of bytes the IBM Tivoli Monitoring Situation Update Forwarder uses to save the last event and press Enter. The default and minimum value is `50`. This configuration value is used for bidirectional architecture only.

   The following prompt is displayed:

   ```
   URL of the CMS SOAP server [cms/soap]
   ```

10. Type the URL for the hub monitoring server SOAP Server and press Enter. The default value is `cms/soap` (which you can use if you set up your monitoring server using the defaults for SOAP server configuration). This configuration value is used for bidirectional architecture only.

    The following prompt is displayed:

    ```
    Rate for sending SOAP requests to CMS from TEC via Web Services [10]
    ```

11. Type maximum number of event updates to be sent by the IBM Tivoli Monitoring Situation Update Forwarder to the hub monitoring server at one time and press Enter. The default and minimum value is `10`. This configuration value is used for bidirectional architecture only.

    The following prompt is displayed:

    ```
    Level of debug for log

    [x] 1 low
    [ ] 2 med
    [ ] 3 verbose

    To select an item enter its number, or enter 0 when you are finished: [0]
    ```

12. Type the level of debugging that you want to use and press Enter. The default is Low, indicated by an `x` next to `Low`.

13. Type `0` when you have finished and press Enter.

    The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```

14. Type `1` and press Enter to continue.

    The following prompt is displayed:

    ```
    Maximum size of any single cache file, in bytes [50000]
    ```

15. Type the maximum size, in bytes, for the cache file used by the IBM Tivoli Monitoring Situation Update Forwarder and press Enter. The default is `50000`. Do not use commas (,) when specifying this value. This configuration value is used for bidirectional architecture only.

    The following prompt is displayed:

    ```
    Maximum number of cache files [10]
    ```

16. Type the maximum number of cache files used by the IBM Tivoli Monitoring Situation Update Forwarder to have at one time and press Enter. The default is 10, while the minimum is 2. This configuration value is used for bidirectional architecture only.

    On Windows, the following prompt is displayed:

    ```
    Directory for cache files to reside [C:/Program Files/IBM/SitForwarder/persistence]
    ```

    On UNIX, the following prompt is displayed:

    ```
    Directory for cache files to reside [/opt/IBM/SitForwarder/persistence]
    ```

17. Type the directory for the IBM Tivoli Monitoring Situation Update Forwarder cache files and press Enter. The default directory on Windows is `C:\Program Files\IBM\SitForwarder\persistence`; on UNIX, `/opt/IBM/SitForwarder/persistence`. This configuration value is used for bidirectional architecture only.

    The following prompt is displayed:

    ```
    Press 1 for Next, 2 for Previous, 3 to Cancel, or 4 to Redisplay [1]
    ```

18. Type `1` and press Enter to continue.

19. The following prompt is displayed:

```
--- Tivoli Enterprise Monitoring Server 1 ---

Host name []
```

Type the fully qualified host name for the computer where the hub monitoring server is running. This should match the information that will be in events coming from this monitoring server. This configuration value is used for bidirectional architecture only. Press Enter.

The following prompt is displayed:

```
User ID []
```

20. Type the user ID to use to access the computer where the hub monitoring server is running. This configuration value is used for bidirectional architecture only. Press Enter.

    The following prompt is displayed:

    ```
    Password:
    ```

21. Type the password to access the hub monitoring server. This configuration value is used for bidirectional architecture only. Press Enter.

    The following prompt is displayed:

    ```
    Confirmation:
    ```

22. Type the password again to confirm it and press Enter.

    The following prompt is displayed:

    ```
    --- Tivoli Enterprise Monitoring Server 2 ---

    Host name []
    ```

23. Repeat steps 19 - 22 to add the short host name for the same hub monitoring server. Then repeat the same steps for each additional hub monitoring server that the IBM Tivoli Monitoring Situation Update Forwarder should send event status updates to. If you have configured the Hot Standby feature, you must configure the host name information for both the primary and secondary hub monitoring servers.

    When you have provided information for all of the hub monitoring servers *and* you specified information for less than 10 monitoring server host names, press Enter to move through the remaining fields defining additional monitoring servers. Do not specify any additional monitoring server information.

24. When you see the following prompt, type 1 and press Enter to continue:

    ```
    Press 1 for Next, 2 for Previous, 3 to cancel or 4 to Redisplay [1]
    ```

    The event synchronization is installed. The following prompt is displayed:

    ```
    IBM Tivoli Monitoring and Tivoli Enterprise Console Event
    Synchronization will
    be installed in the following location:

    /opt/IBM/SitForwarder

    for a total size:

    101.3 MB

    Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1]
    ```

25. Type 1 and press Enter to continue.

    The following prompt is displayed:

    ```
    The InstallShield Wizard has successfully installed IBM Tivoli
    Monitoring and Tivoli Event Synchronization.
    Choose Finish to exit the wizard.
    Press 3 to Finish, or 4 to Redisplay [1]
    ```

26. Type 3 to finish and press Enter.

Installing from the command-line using a silent installation

Use the following steps to install the event synchronization using a silent installation from the command-line on your event server. This installation method runs silently, so you do not see status messages during the actual installation.

1. Change to the `tec` directory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
2. Run the following command to generate the configuration file:

   On Windows:

   ```
   ESync2300Win32.exe -options-template filename
   ```

   where *filename* is the name of the configuration file to create, for example, es_silentinstall.conf.

   On UNIX:

   ```
   ESync2300operating_system.bin -options-template filename
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, `or Solaris`). For example, run the following command on an AIX computer:

   ```
   ESync2300Aix.bin -options-template filename
   ```

3. Edit the output file to specify the values described in Table 140.

   Notes:

   a. Remove the number signs (###) from the beginning of any value that you want to specify.

   b. You must specify the following values:

      ```
      ### -P installLocation="value"
      ### -W configInfoPanel3.fileLocn="value"
      ```

      and the following information for at least one monitoring server:

      ```
      cmdSvrsPnlNotGuiMode.hostname1,
      cmdSvrsPnlNotGuiMode.userID1
      cmdSvrsPnlNotGuiMode.pswd1
      cmdSvrsPnlNotGuiMode.retypePswd1
      ```

      If you do not specify any of the other values, the default values are used. If you specify values, ensure that the values you specify meets the minimum required values. Otherwise, the installation stops and an error is written to the log file.

Table 140. Netcool/OMNIbus event synchronization configuration values

Value	Description

installLocation	The directory where you want to install the product. If the directory path contains spaces, enclose it in double quotation marks (" "). For example, to install the product to C:\Program Files\My Product, use
	`-P installLocation="C:\Program Files \My Product"`
	On Windows systems, the default installation directory is:
	`C:\Program Files\IBM\SitForwarder`
	On Linux/UNIX systems, the default installation directory is:
	`/opt/IBM/SitForwarder`

configInfoPanel.filename	*For bidirectional architecture only:* The name of the file where IBM Tivoli Monitoring Situation Update Forwarder configuration information is stored. The default file name is `situpdate.conf`.

Table 140. Netcool/OMNIbus event synchronization configuration values *(continued)*

Value	Description
configInfoPanel.pollingInt	*For bidirectional architecture only:* The polling interval, in seconds used by the IBM Tivoli Monitoring Situation Update Forwarder to determine if there are new event updates to send from Netcool/OMNIbus to the hub Tivoli Enterprise Monitoring Server. The minimum value is 1, while the default value is 3. If there are no situation events, the IBM Tivoli Monitoring Situation Update Forwarder rests for 3 seconds.
configInfoPanel.crcByteCnt	*For bidirectional architecture only:* Number of bytes that the IBM Tivoli Monitoring Situation Update Forwarder uses when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) value is 50.
configInfoPanel.cmsSoapURL	*For bidirectional architecture only:* The URL for the SOAP Server configured on the computer where the hub monitoring server is running. The default value is `cms/soap`. Do not change this value. The actual URL for the SOAP Server is dynamically constructed from the incoming event.
configInfoPanel.bufFlushRate	*For bidirectional architecture only:* The maximum number of event updates to be sent by the IBM Tivoli Monitoring Situation Update Forwarder to the hub monitoring server at one time. The minimum (and default) value is 10 events.
configInfoPanel.logLevel	*For bidirectional architecture only:* The level of debug information for the IBM Tivoli Monitoring Situation Update Forwarder that is logged. You have the following choices: • Low (default) • Medium • Verbose
configInfoPanel3.filesize	*For bidirectional architecture only:* The maximum permitted size, in bytes, for any one event cache file used by the IBM Tivoli Monitoring Situation Update Forwarder. The minimum (and default) value is 50000. Do not use commas when specifying this value (50,000 instead of 50000).
configInfoPanel3.fileNumber	*For bidirectional architecture only:* The maximum number of event caches files at any given time used by the IBM Tivoli Monitoring Situation Update Forwarder. The minimum value is 2, while the default value is 10. When this value is reached, the oldest file is deleted to make room for a new file.
configInfoPanel3.fileLocn	*For bidirectional architecture only:* The location where IBM Tivoli Monitoring Situation Update Forwarder event cache files are located. The default locations are as follows: • On Windows: `C:\Program Files\IBM\SitForwarder\persistence`. • On UNIX: `/opt/IBM/SitForwarder/persistence`

Table 140. Netcool/OMNIbus event synchronization configuration values (continued)

Value	Description
cmsSvrsPnlNotGuiMode.hostname*#* **Note:** The pound sign (#) stands for a number 1 - 10. For example, "hostname1".	*For bidirectional architecture only:* The host name of each hub monitoring server that sends events to the event server. For each hub monitoring server, specify one property with the monitoring server's fully qualified hostname and specify another property with the monitoring server's short hostname. Specify up to 10 monitoring server host names. If you have configured the Hot Standby feature, you must configure the host name information for both the primary and secondary hub monitoring servers.
cmsSvrsPnlNotGuiMode.userID*#*	*For bidirectional architecture only:* The user ID for the hub monitoring server, identified in host name*#*, to use to access the computer where the monitoring server is running.
cmsSvrsPnlNotGuiMode.pswd*#*	*For bidirectional architecture only:* The password for the user ID used to access the computer where the hub monitoring server is running.
cmsSvrsPnlNotGuiMode.retypePswd*#*	*For bidirectional architecture only:* The password confirmation for the user ID.

4. Save the file.
5. Run the following command:

   On Windows:

   ```
   ESync2300Win32.exe -options filename -silent
   ```

   where *filename* is the name of your configuration file.

   On UNIX:

   ```
   ESync2300operating_system.bin -options filename -silent
   ```

   where *operating_system* is the operating system you are installing on (`Aix`, `HP11`, `Linux`, `linux390`, `Solaris`). For example, on AIX, run the following command:

   ```
   ESync2300Aix.bin -options filename -silent
   ```

When installation is complete, the results are written to the **itm_tec_event_sync_install.log** file. On UNIX systems, this log file is always created in the `/tmp` directory. For Windows systems, this file is created in the directory defined by the %TEMP% environment variable. To determine where this directory is defined for the current command-line window, run the following command:

```
echo %TEMP%
```

If you specified the monitoring servers in the silent installation configuration file, you might consider deleting that file after installation, for security reasons. The passwords specified in the files are not encrypted.

If you want to define additional monitoring servers (in addition to the one required monitoring server), run the sitconfuser.sh command. Repeat this command for each monitoring server.

If you specified your monitoring servers after the installation, you must stop and restart the Situation Update Forwarder process manually.

Updating the Netcool/OMNIbus ObjectServer with IBM Tivoli Monitoring attributes, tables, and triggers

IBM Tivoli Monitoring provides updates to the Netcool/OMNIbus ObjectServer attributes, tables, and triggers. The type of updates required depend on whether you are using a single-tier architecture, a single-tier high-availability architecture, or a multitier architecture. A single-tier architecture uses a single ObjectServer. A single-tier high-availability architecture uses a primary and backup ObjectServer with a bi-directional gateway between them. A multitier architecture uses multiple sets of ObjectServers. For more information on the last two architecture types, see "Netcool/OMNIbus Multitiered and High-availability Architecture" on page 709.

Table 141 indicates which ObjectServer updates are required for each architecture type.

Table 141. ObjectServer updates required for each architecture type

Task	single-tier architecture (single ObjectServer)	single-tier, high-availability architecture	Multitier architecture
"Updating gateways to map attributes."		X Perform this task for the gateway between the primary and backup ObjectServers.	X Perform this task for each unidirectional and bidirectional gateway between the ObjectServers.
"Updating the bidirectional failover gateway to replicate tables" on page 731.		X Perform this task for the bidirectional gateway between the ObjectServers.	X Perform this task for each bidirectional gateway used between ObjectServers in the aggregation tier.
"Updating the OMNIbus database schema in the collection tier" on page 732.			X Perform this task on each ObjectServer in the collection tier.
"Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers" on page 733.	X	X Perform this task on each ObjectServer.	X Perform this task for each ObjectServer in the aggregation tier.
"Changing the default deduplication trigger" on page 735.	X	X Perform this task on each ObjectServer.	X Perform this task for each ObjectServer in the aggregation tier.
"Updating the OMNIbus database schema in the display tier" on page 738.			X Perform this task on each ObjectServer in the display tier.

Updating gateways to map attributes

If a Netcool/OMNIbus multitier architecture or single-tier high-availability architecture is used, then the gateways between ObjectServers need to be modified to add the relevant attributes for IBM Tivoli Monitoring integration with OMNIbus.

This task should be performed by a Netcool/OMNIbus administrator.

In a multitier architecture, update the map definition file for each of these gateways that exist in your environment:

- Collection tier to aggregation tier unidirectional gateways.
- Aggregation tier bidirectional failover gateways.
- Aggregation tier to display tier unidirectional gateways.

In a single-tier high-availability architecture, update the map definition file for the bidirectional failover gateway between the primary and backup ObjectServers.

Perform the following steps for each affected gateway:

1. Copy the `GATE_itm.map` file from an ObjectServer where the IBM Tivoli Event Synchronization component was installed. The `GATE_itm.map` file can found be in the `event_sync_install_dir/omnibus/multitier directory` where `event_sync_install_dir` is the location where the IBM Tivoli Monitoring Event Synchronization component is installed.
2. Copy the mappings in the `GATE_itm.map` file into the gateway map definition file that also uses the `.map` extension. The following two types of mappings can be found in the `GATE_itm.map` file:
   a. **Custom IBM Tivoli Monitoring attributes for the ObjectServer alerts.status table**: these attributes should be added to the STATUSMAP mapping entry in the gateway map definition file. Your gateway map file might contain this comment block to identify where to add these custom attributes:
      ```
      ##############################################################################
      #
      #       CUSTOM alerts.status FIELD MAPPINGS GO HERE
      #
      ##############################################################################
      ```
   b. **Custom IBM Tivoli Monitoring table mappings**: Custom Tivoli Monitoring tables can be added to the ObjectServer database. The mapping definitions for custom tables only need to be copied to the gateway map definition file for bidirectional failover gateways:
      - in the aggregation tier of a multitier high-availability architecture, or
      - in a single-tier high-availability architecture

      Copy the mapping definitions for the custom tables from the `GATE_itm.map` file to the end of the bidrectional failover gateway map definition file.

Note: You must restart the gateways for the changes to take effect. However, restart the gateways only after all other updates to the OMNIbus database schema are complete. See Table 136 on page 713 to determine when to restart the gateways.

Updating the bidirectional failover gateway to replicate tables

If a Netcool/OMNIbus multitier architecture or single-tier high-availability architecture is used, then the bidirectional gateways between ObjectServers must be modified to replicate IBM Tivoli Monitoring custom tables in the ObjectServer database.

In a multitier high-availability architecture, update the bidirectional failover gateways used between the ObjectServers in the aggregation tier. In a single-tier high-availability architecture, update the bidirectional failover gateway between the primary and backup ObjectServers.

This task should be performed by a Netcool/OMNIbus administrator. Perform the following steps for each affected bidirectional failover gateway:

1. Copy the `GATE_itm.tblrep.def` file from an ObjectServer where the IBM Tivoli Event Synchronization component was installed.

   You can find the `GATE_itm.tblrep.def` file in the `event_sync_install_dir/omnibus/multitier directory` where `event_sync_install_dir` is the location where the IBM Tivoli Monitoring Event Synchronization component is installed.

2. Copy the table replication definitions in the GATE_itm.tblrep.def file to the bottom of the gateway table replication definition file that also uses the .tblrep.def file extension. Use the following criteria to determine which table definitions to copy:
   a. Always copy the replication definition for the alerts.itm_loopback_events table.
   b. Copy the replication definition for the alerts.itm_heartbeat_events table if monitoring agents will be sending events directly to Netcool/OMNIbus and the agents will be configured to send heartbeat events. If you plan to use agent heartbeat settings, you must also run the itm_heartbeat.sql file when updating the OMNIbus database schema in a single-tier architecture or in the aggregation tier of a multitier architecture.
   c. Copy the replication definition for the alerts.itm_cleared_event_cache table if you are planning to use bidirectional event synchronization between the monitoring servers and ObjectServer. If you copy this definition, you must also execute the itm_event_cache.sql file when updating the OMNIbus database schema in a single-tier architecture or in the aggregation tier of a multitier architecture.

Note: You must restart the gateways for the changes to take effect. However, the restart should occur after all other updates to the OMNIbus database schema are complete. See Table 136 on page 713 to determine when to restart the gateways.

Updating the OMNIbus database schema in the collection tier

In a Netcool/OMNIbus multitier architecture, add IBM Tivoli Monitoring triggers to each ObjectServer in the collection tier.

This task should be performed by a Netcool/OMNIbus administrator.

For each Netcool/OMNIbus Object Server defined in the collection tier:

1. Copy the collection_itm.sql file from an ObjectServer in the aggregation tier to each ObjectServer in the collection tier.

   The collection_itm.sql file can found be in the event_sync_install_dir/omnibus/multitier directory where event_sync_install_dir is the location where the IBM Tivoli Monitoring Event Synchronization component is installed on an ObjectServer in the aggregation tier.
2. Update the Object Server database with the following command, which pipes the SQL command set into the SQL command-line tool and performs the updates to the ObjectServer database:
   - On Windows:

```
type path_to_file\collection_itm.sql | %OMNIHOME%\..\bin\redist\isql
-U username
-P password
-S server_name
```

where:

$OMNIHOME
Is the system-defined variable defining the installation location of OMNIbus.

username
Is the OMNIbus Object Server user name.

password
Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

- On UNIX:

  ```
  $OMNIHOME/bin/nco_sql -user username
    -password password
    -server server_name
    < path_to_file/collection_itm.sql
  ```

  where:

  $OMNIHOME
  Is the system-defined variable defining the installation location of OMNIbus

  username
  Is the OMNIbus ObjectServer user name

  password
  Is the OMNIbus ObjectServer password

  server_name
  Is the OMNIbus ObjectServer name defined for process control.

  path_to_file
  Is the fully qualified path to the specified SQL file.

Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers

Update the Netcool/OMNIbus ObjectServer with the attributes and triggers that allow the ObjectServer to correctly process IBM Tivoli Monitoring events and forward events to the IBM Tivoli Monitoring Situation Update Forwarder when using bidirectional communication. To verify if specific customization settings are needed, see "Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754.

Perform the steps in this section for each ObjectServer in a single-tier architecture or for each ObjectServer in the aggregation tier of a multitier architecture.

The tasks in this section should be completed by the Netcool/OMNIbus Administrator.

If you are setting up a unidirectional architecture and have installed the IBM Tivoli Monitoring Event Synchronization Component on a system other than your Netcool/OMNIbus ObjectServer, you must manually copy the SQL files to your ObjectServer. The files are located in the event_sync_install_dir/omnibus directory. Also, for a unidirectional architecture, you must modify the contents of the itm_proc.sql file, changing every instance of the word *REJECT* to the word *ACCEPT* and ensure that the path specified for the itmfile variable exists on the system with Netcool/OMNIbus ObjectServer. You only need to load the itm_proc.sql and itm_db_update.sql files. If you later decide to convert your environment to a bidirectional architecture, you must also load the itm_sync.sql file.

If you are setting up a bidirectional architecture, you might want to customize the itm_proc.sql file before loading it. For more information, see "Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754. Also review the information at the end of the topic about ensuring that the eventcmd procedure is correctly configured to send events back to the hub monitoring server.

Note: If you are integrating IBM Tivoli Monitoring and Tivoli Business Service Manager, you must also complete the following steps:

1. If you are using Tivoli Business Service Manager V4.2.1, add the Tivoli Business Service Manager schema updates before you add the Tivoli Monitoring schema updates. If you add the Tivoli Monitoring schema updates before the Tivoli Business Service Manager schema updates, rerun the procedure in this section to add the IBM Tivoli Monitoring schema updates again to ensure that the latest definitions are used.

2. After updating the OMNIbus schema with the Tivoli Monitoring updates, run the Tivoli Business Service Manager discover schema utility (rad_discover_schema). See the Tivoli Business Service Manager Information Center for detailed instructions on using this utility: http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.tivoli.itbsm.doc/welcome.htm.
3. After running the discover schema utility, remember to restart the Tivoli Business Service Manager Dataserver. Failure to do so can cause connection problems.

If the OMNIbus ObjectServer is running on a UNIX system as a non-root user and the event synchronization component is installed and run as either root or another user, verify that the user under which the ObjectServer is running has write permission to the `event_sync_install_dir/log` directory before updating the OMNIbus database schema. If the user does not have write permission to this directory there will be error messages related to '`itmfile`' when the `itm_proc.sql` file is loaded into Netcool/OMNIbus and ITM-OMNIbus event synchronization will not work.

The command to configure OMNIbus pipes the SQL command set into the SQL command-line tool and performs the updates to the ObjectServer.

1. Update the Object Server database with the following commands:
   - On Windows:

```
type path_to_file\itm_proc.sql | %OMNIHOME%\..\bin\redist\isql -U username
-P password
-S server_name

type path_to_file\itm_db_update.sql | %OMNIHOME%\..\bin\redist\isql -U username
-P password
-S server_name
```

Only run this command if you are using the bidirectional architecture:

```
type path_to_file\itm_sync.sql | %OMNIHOME%\..\bin\redist\isql -U username
-P password
-S server_name
```

Only run this command if you are using the bidirectional architecture:

```
type path_to_file\itm_event_cache.sql  | %OMNIHOME%\..\bin\redist\isql -U username
-P password
-S server_name
```

where:

$OMNIHOME
Is the system-defined variable defining the installation location of OMNIbus.

username
Is the OMNIbus Object Server user name.

password
Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

- On UNIX:

  ```
  $OMNIHOME/bin/nco_sql -user username
    -password password
    -server server_name
    < path_to_file/itm_proc.sql
  ```

  ```
  $OMNIHOME/bin/nco_sql -user username
    -password password
    -server server_name
    < path_to_file/itm_db_update.sql
  ```

  Only run this command if you are using the bidirectional architecture:

  ```
  $OMNIHOME/bin/nco_sql -user username
    -password password
    -server server_name
    < path_to_file/itm_sync.sql
  ```

  Only run this command if you are using the bidirectional architecture:

  ```
  $OMNIHOME/bin/nco_sql -user username
    -password password
    -server server_name
    < path_to_file/itm_event_cache.sql
  ```

  where:

  $OMNIHOME
  Is the system-defined variable defining the installation location of OMNIbus.

  username
  Is the OMNIbus ObjectServer user name.

  password
  Is the OMNIbus ObjectServer password.

  server_name
  Is the OMNIbus ObjectServer name defined for process control.

  path_to_file
  Is the fully qualified path to the specified SQL file. The SQL files are located in the `event_sync_install_dir/omnibus directory` where `event_sync_install_dir` is the location where the IBM Tivoli Monitoring Event Synchronization component is installed.

Notes:

1. “Object exists” and "Attempt to insert duplicate row" errors occur if the scripts were previously run. These errors are harmless.
2. If you are running the SQL database schema for the first time, the following harmless error messages are displayed:
   - ERROR=Object not found on line 4 of statement '-- A workspace table for the ITM event clear automation...', at or near 'itm_event_clear'.
   - ERROR=Object not found on line 1 of statement 'delete from alerts.itm_problem_events;...', at or near 'itm_problem_events'.
   - ERROR=Object not found on line 1 of statement 'drop table alerts.itm_problem_events;...', at or near 'itm_problem_events'.

Changing the default deduplication trigger

The deduplication triggers provided by the IBM Tivoli Monitoring event synchronization component and the standard deduplication triggers provided with Netcool/OMNIbus are in conflict with each other. The standard deduplication triggers modify the Severity and Summary attributes that results in those attributes having incorrect values for IBM Tivoli Monitoring events for both unidirectional and bidirectional event synchronization.

The standard deduplication triggers must be modified by the Netcool/OMNIbus administrator to ignore events from IBM Tivoli Monitoring. The deduplication trigger to modify depends on the Netcool/OMNIbus architecture type:

single-tier architecture:

- In a single-tier architecture without high availability, modify the deduplication trigger.
- In a single-tier architecture with a bidirectional failover gateway for high availability, modify the `agg_deduplication` trigger in each ObjectServer.

Multitier architecture:

- In a multitier architecture without high availability, modify the deduplication trigger on the Objectserver in the aggregation tier.
- In a multitier architecture with a bidirectional failover gateway for high availability, modify the `agg_deduplication` trigger on each ObjectServer in the aggregation tier.

You can modify the appropriate trigger from either the Netcool/OMNIbus Administrator `nco_config` interface or by using the `nco_sql` CLI on the Netcool/OMNIbus ObjectServer.

Using Netcool/OMNIbus Administrator:

1. Run the `nco_config` utility to start Netcool/OMNIbus Administrator.
2. Connect to the Netcool/OMNIbus ObjectServer where the change will be made.
3. Select **AutomationTriggers** from the menu.
4. Go to the editor for the trigger to be modified. See the preceding paragraphs to determine if you should modify the deduplication or agg_deduplication trigger.
5. On the **When** tab, enter: **`new.Type != 20`** and **`new.Type != 21`**.
6. Save the trigger and exit from `nco_config` interface.

Using the nco_sql command to modify the deduplication trigger in environments without a bidirectional failover gateway:

1. Open the following file:
   - On Windows systems: `%OMNIHOME%\etc\automation.sql`.
   - On Linux/UNIX systems: `$OMNIHOME/etc/automation.sql`.
2. Locate the command that creates the deduplication trigger. See the following example of the deduplication trigger command:

   Note: Your deduplication trigger might be different than what is shown in the following example depending on the Netcool/OMNIbus release that you are using or if you have customized the trigger.

```
create or replace trigger deduplication
group default_triggers
priority 1
comment 'Deduplication processing for ALERTS.STATUS'
before reinsert on alerts.status
for each row
begin
 set old.Tally = old.Tally + 1;
 set old.LastOccurrence = new.LastOccurrence;
 set old.StateChange = getdate();
 set old.InternalLast = getdate();
 set old.Summary = new.Summary;
 set old.AlertKey = new.AlertKey;
 if (( old.Severity = 0) and (new.Severity > 0))
 then
 set old.Severity = new.Severity;
 end if;
end;
go
```

3. Copy this command to a temporary file. In this example, the file name is `/tmp/dedup.sql`.
4. Edit `/tmp/dedup.sql` and add the line that is highlighted in bold in the following example command so that the trigger ignores IBM Tivoli Monitoring events:

```
create or replace trigger deduplication
group default_triggers
priority 1
comment 'Deduplication processing for ALERTS.STATUS'
before reinsert on alerts.status
for each row
when (new.Type != 20) and (new.Type != 21)
begin
set old.Tally = old.Tally + 1;
set old.LastOccurrence = new.LastOccurrence;
set old.StateChange = getdate();
set old.InternalLast = getdate();
set old.Summary = new.Summary;
set old.AlertKey = new.AlertKey;
if (( old.Severity = 0) and (new.Severity > 0))
then
set old.Severity = new.Severity;
end if;
end;
go
```

5. Save the file, and run the following command to replace the deduplication trigger:
   - On Windows systems: `%OMNIHOME%\..\bin\redist\isql -U username -P password -S server_name < C:\tmp\dedup.sql`.
   - On Linux and UNIX systems: `$OMNIHOME/bin/nco_sql -user username -password password -server server_name < /tmp/dedup.sql`.

   where:

   OMNIHOME
   Is the system-defined variable that defines the installation location of Netcool/OMNIbus ObjectServer.

   username
   Is the Netcool/OMNIbus ObjectServer user name.

   password
   Is the Netcool/OMNIbus ObjectServer password.

   server_name
   Is the Netcool/OMNIbus ObjectServer name defined for process control.

Using nco_sql to modify the agg_deduplication trigger in environments with a bidirectional failover gateway:

1. Open the following file:
   - On Windows systems: `%OMNIHOME%\extensions\multitier\objectserver\aggregation.sql`.
   - On Linux/UNIX systems: `$OMNIHOME/extensions/multitier/objectserver/aggregation.sql`.
2. Copy the agg_deduplication trigger logic to a temporary file. In this example, the temporary file name is `/tmp/agg_dedup.sql`.
3. Edit `/tmp/agg_dedup.sql` and add the line highlighted in bold in the following example so that the trigger ignores IBM Tivoli Monitoring events.

   Note: Only the beginning portion of the agg_deduplication trigger is shown in the example that follows. The remaining logic is not shown because it does not need any modifications. Your agg_deduplication trigger might be different than what is shown in the following example depending on the Netcool/OMNIbus release that you are using or if you have customized the trigger.

```
CREATE OR REPLACE TRIGGER agg_deduplication
GROUP default_triggers
PRIORITY 2
COMMENT 'Replacement reinsert trigger (alerts.status) for multi-ObjectServer environments.'
BEFORE REINSERT ON alerts.status
FOR EACH ROW
when (new.Type != 20) and (new.Type != 21)
declare

        now utc;
begin
```

4. Save the file, and run the following command to replace the agg_deduplication trigger:
   - On Windows systems: `%OMNIHOME%\..\bin\redist\isql -U username -P password -S server_name < C:\tmp\agg_ dedup.sql.`
   - On Linux/UNIX `$OMNIHOME/bin/nco_sql -user username -password password -server server_name < /tmp/agg_dedup.sql.`

   where:

   OMNIHOME
   Is the system-defined variable that defines the installation location of Netcool/OMNIbus ObjectServer.

   username
   Is the Netcool/OMNIbus ObjectServer user name.

   password
   Is the Netcool/OMNIbus ObjectServer password.

   server_name
   Is the Netcool/OMNIbus ObjectServer name defined for process control.

Updating the OMNIbus database schema in the display tier

In a Netcool/OMNIbus multitier architecture, add IBM Tivoli Monitoring attributes to the ObjectServer alerts.status table in the display tier.

This task should be performed by a Netcool/OMNIbus administrator.

For each Netcool/OMNIbus Object Server defined in the display tier, complete the following steps:

1. Copy the `display_itm.sql` file from an ObjectServer in the aggregation tier to each ObjectServer in the display tier.

   The `display_itm.sql` file can found be in the `event_sync_install_dir/omnibus/multitier directory` where `event_sync_install_dir` is the location where the IBM Tivoli Monitoring Event Synchronization component is installed on an ObjectServer in the aggregation tier.
2. Update the Object Server database with the following command, which pipes the SQL command set into the SQL command-line tool and performs the updates to the ObjectServer database:
   - On Windows:

     ```
     type path_to_file\display_itm.sql | %OMNIHOME%\..\bin\redist\isql
     -U username
     -P password
     -S server_name
     ```

     where:

     $OMNIHOME
     Is the system-defined variable defining the installation location of OMNIbus.

username
Is the OMNIbus Object Server user name.

password
Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

- On UNIX:

```
$OMNIHOME/bin/nco_sql -user username
  -password password
  -server server_name
  < path_to_file/collection_itm.sql
```

where:

$OMNIHOME
Is the system-defined variable defining the installation location of OMNIbus

username
Is the OMNIbus ObjectServer user name

password
Is the OMNIbus ObjectServer password

server_name
Is the OMNIbus ObjectServer name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

Starting and stopping the IBM Tivoli Monitoring Situation Update Forwarder

If you are using bidirectional architecture to forward event status updates from Netcool/OMNIbus back to the hub monitoring server, you must start the IBM Tivoli Monitoring Situation Update Forwarder.

The tasks in this section should be completed by the Netcool/OMNIbus Administrator.

This process is started automatically when the Netcool/OMNIbus ObjectServer starts. To start the process manually, change to the `event_sync_install_dir/bin` directory where *event_sync_install_dir* is the directory where the IBM Tivoli Monitoring Event Synchronization Component is installed. Then run the following command:

On Windows:

```
startSUF.cmd
```

On UNIX:

```
startSUF.sh
```

To stop the process, run the following command:

On Windows:

```
stopSUF.cmd
```

On UNIX:

```
stopSUF.sh
```

On Windows, you can also start and stop the IBM Tivoli Monitoring Situation Update Forwarder service to start or stop the forwarding of event updates. You can start and stop this service either from the Windows Service Manager utility or with the following commands:

```
net start situpdate
net stop situpdate
```

Configuring the Netcool/OMNIbus EIF probe

Use this step to configure the Netcool/OMNIbus EIF Probe with the rules for mapping situation EIF events to OMNIbus events. The default mapping of IBM Tivoli Monitoring EIF attribute slots to OMNIbus attributes is shown in “Default mapping of situation events to OMNIbus events” on page 766. To configure the mapping, you must uncomment the include statement for the `itm_event.rules` in the probe's `tivoli_eif.rules` file and copy the IBM Tivoli Monitoring rules files to the probe's operating system directory.

Three Tivoli Monitoring rules files are available:

- The `itm_event.rules` file is included by the `tivoli_eif.rules` probe rules file. This is located in the omnibus directory under the event synchronization component's installation directory.
- The `tbsm_eif_event.rules` file is included by the `itm_event.rules` file and sets attributes specific to Tivoli Business Service Manager integration. This is located in the omnibus/tbsm directory under the event synchronization component's installation directory.
- The `itm_custom_overrides.rules` is included by the `itm_event.rules` file and can be used to customize the IBM Tivoli Monitoring rules behavior. See “Customizing the rules file” on page 742 for more details. This is located in the omnibus directory under the event synchronization component's installation directory.

If IBM Tivoli Monitoring is sending virtualization events or predictive analytics events to Netcool/OMNIbus, there are rules and triggers available with Netcool/OMNIbus ObjectServer for additional event enrichment and correlation handling. These rules also set some of the OMNIbus attributes (including Node and Summary) differently from the values that are normally used for monitoring events. See the Netcool/OMNIbus Information Center for details on extending the functionality of Netcool/OMNIbus for virtualization and predictive events. The virtualization rules and predictive rules files have include statements in the `tivoli_eif.rules` file. Verify that you are using a version of the EIF Probe's `tivoli_eif.rules` file that includes the virtualization rules file (`tivoli_eif_virtualization_pt2.rules`) and predictive rules file (`predictive_event.rules`) after `itm_event.rules`. If either of these rules files are included before `itm_event.rules` then move the include statements. You should also verify that you are using Netcool/OMNIbus Version 7.3.0 FP 6 or later, or Netcool/OMNIbus Version 7.3.1 FP2 or later, as those fix packs have the required updates to the rules files for integration with IBM Tivoli Monitoring V6.2.3.

Some monitoring agents might also provide their own probe rules include files that should also be included after `itm_event.rules` in the probe's `tivoli_eif.rules` file. See the documentation for the specific agents you are using in your environment to determine if the agent has its own rules file.

Note: You must recycle the probe after you update any of the rules files.

If you are using a Netcool/OMNIbus multitier architecture, you should also ensure the EIF Probe is configured to connect to an ObjectServer in the collection tier.

The tasks in this section should be completed by the Netcool/OMNIbus Administrator.

Take the following steps to update the rules files of the EIF probe:

1. Uncomment the include statement for `itm_event.rules` in the `tivoli_eif.rules` file in the probe's operating system directory:

   %OMNIHOME%\probes\arch (Windows)

or

`$OMNIHOME/probes/`*`arch`* `(UNIX)`

where:

OMNIHOME
: Is a system-defined variable defining the installation location of OMNIbus.

arch
: Represents the operating system directory on which the probe is installed; for example, solaris2 when running on a Solaris system, and win32 for a Windows system.

Note: If the probe's version of the `tivoli_eif.rules` does not contain a statement to include the `itm_event.rules` file, you must upgrade to version 10 or later of the EIF Probe.

2. Copy the `itm_event.rules` file in the omnibus directory where the event synchronization component is installed, to this directory on the machine where the EIF probe is installed:

   `%OMNIHOME%\probes\`*`arch`* `(Windows)`

   or

   `$OMNIHOME/probes/`*`arch`* `(UNIX)`

   where:

   OMNIHOME
   Is a system-defined variable defining the installation location of OMNIbus.

   arch
   Represents the operating system directory on which the probe is installed; for example, solaris2 when running on a Solaris system, and win32 for a Windows system.
3. If you are integrating Tivoli Business Service Manager as well, uncomment the line in `itm_event.rules` with the include statement for `tbsm_eif_event.rules`, and copy the `tbsm_eif_events.rules` from the omnibus/tbsm directory to this directory on the machine where the EIF probe is installed:

   `%OMNIHOME%\probes\`*`arch`* `(Windows)`

   or

   `$OMNIHOME/probes/`*`arch`* `(UNIX)`

   where:

   OMNIHOME
   Is a system-defined variable defining the installation location of OMNIbus.

   arch
   Represents the operating system directory on which the probe is installed; for example, solaris2 when running on a Solaris system, and win32 for a Windows system.
4. If rules customizations are done in `itm_custom_override.rules`, uncomment the line in `itm_event.rules` with the include statement for `itm_custom_override.rules`. Copy the `itm_custom_override.rules` file with the customizations to this directory on the machine where the EIF probe is installed:

   `%OMNIHOME%\probes\`*`arch`* `(Windows)`

   or

   `$OMNIHOME/probes/`*`arch`* `(UNIX)`

   where:

   OMNIHOME
   Is a system-defined variable defining the installation location of OMNIbus.

arch
Represents the operating system directory on which the probe is installed; for example, solaris2 when running on a Solaris system, and win32 for a Windows system.

5. Stop the EIF probe.
   - On Windows systems: In the Control Panel, open **Administrative Tools**, then **Services**. In the list of services, double-click the EIF probe and then click **Stop**.
   - On UNIX systems: Issue the following command:

     ```
     $OMNIHOME/bin/nco_pa_stop -process probe_name
     ```

     where:

     $OMNIHOME
     Is the system-defined variable defining the installation location of OMNIbus.

     probe_name
     Is the OMNIbus EIF probe name defined for Process Control.
6. Restart the OMNIbus probe.
   - On Windows systems: In the list of services, double-click **OMNIbus EIF Probe** and then click **Start**.
   - On UNIX systems: Issue the following command:

     ```
     $OMNIHOME/bin/nco_pa_start -process probe_name
     ```

Customizing the rules file

If you need to customize the IBM Tivoli Monitoring probe rules, add your customizations to the `itm_custom_override.rules` file, that can be included by `itm_event.rules` by uncommenting the line with the include statement for it. You should not modify any of the OMNIbus attributes that are used by the IBM Tivoli Monitoring triggers to perform event synchronization. See "Default mapping of situation events to OMNIbus events" on page 766 for the list of attributes you should not change.

If you want to map agent specific slots to Netcool/OMNIbus attributes, see "Generic mapping for agent-specific slots" on page 770 for information on slot naming and other considerations.

The `itm_custom_overrides.rules` file is copied into the following directory:

```
%OMNIHOME%\probes\arch (Windows)
```

or

```
$OMNIHOME/probes/arch (UNIX)
```

where:

OMNIHOME
Is a system-defined variable defining the installation location of OMNIbus.

arch
Represents the operating system directory on which the probe is installed; for example, solaris2 when running on a Solaris system, and win32 for a Windows system.

After updating the rules file, restart the probe using the instructions in "Configuring the Netcool/OMNIbus EIF probe" on page 740.

Configuring the OMNIbus server for program execution from scripts

To run the IBM Tivoli Monitoring Situation Update Forwarder program from SQL automation scripts for sending synchronization events to the hub monitoring server, the Netcool/OMNIbus ObjectServer must be running under process control and the **PA.Username** and **PA.Password** properties must be set in *OMNIHOME*/etc/NCOMS.props file, where *OMNIHOME* is the system-defined variable defining the installation location of Netcool/OMNIbus.

The tasks in this section should be completed by the Netcool/OMNIbus Administrator if you are using bidirectional event synchronization. If you are using a Netcool/OMNIbus multitier architecture, these tasks should only be performed on the ObjectServers in the aggregation tier and are not necessary for the ObjectServers in the collection and display tiers.

For Linux and UNIX: The **PA.Username** property must be set to the user name for connecting to the process control agent. The **PA.Password** property must be set to the password of the user connecting to the process control agent. By default, process control grants access to the members of ncoadmin group so configure **PA.Username** to specify a user in this group.

For Windows: The **PA.Username** property must be set to a Windows account name, and the **PA.Password** property must be set to the password for that account.

See OMNIbus documentation for more information about configuring OMNIbus server under process control and for information about the nco_pa_crypt utility that encrypts the PA.Password property value.

After you change the **PA.Username** and **PA.Password** properties in the `OMNIHOME/etc/NCOMS.props` file, complete the following procedure to restart the OMNIbus ObjectServer:

1. Stop the OMNIbus server:
   - On Windows: In the Control Panel, open Administrative Tools, then Services. In the list of services, double-click OMNIbus server, then click Stop.
   - On UNIX: Issue the following command from command-line

     ```
     $OMNIHOME/bin/nco_pa_stop -process server_name
     ```

     where:

     $OMNIHOME
         Is the system-defined variable defining the installation location of OMNIbus.

     server_name
         Is the OMNIbus ObjectServer name defined for process control.
2. Restart the OMNIbus server.
   - On Windows: In the list of services, double-click OMNIbus server, then click Start.
   - On UNIX: Issue the following command from command-line:

     ```
     $OMNIHOME/bin/nco_pa_start -process server_name
     ```

Configuring the hub monitoring server to forward events

Before the hub monitoring server forwards any situation events to Netcool/OMNIbus, you must enable event forwarding and define a default event destination.

The tasks in this section should be completed by the IBM Tivoli Monitoring Administrator.

Take the following steps to enable event forwarding for each hub monitoring server from which you want to forward events to an OMNIbus ObjectServer. If the Hot Standby feature is configured, you should perform these steps for both the primary and secondary hub monitoring servers:

For Windows monitoring servers *only*, perform the following steps:

1. Open Manage Tivoli Enterprise Monitoring Services.
2. Right-click the monitoring server and click **Reconfigure**.
3. On the configuration options window, select **Tivoli Event Integration Facility**.
4. Click **OK** and **OK**.
5. Complete the following fields on the Event Server: Location and Port Number window and click **OK**:

Server or EIF Probe Location
Type the host name or IP address for the computer where the EIF probe is installed.

Port Number
Type the port number on which the probe is listening. Check with the Netcool/OMNIbus administrator that installed the EIF probe to determine what port number the EIF probe is using, or see the *IBM Tivoli Monitoring: Troubleshooting Guide* for instructions on how to determine the EIF probe port number. If the default probe port number was used, the number depends on the version of the probe and how the probe was installed. If you are using Netcool/OMNIbus Probe for Tivoli EIF Version 8 or later, the default port number is 9998. If Tivoli Business Service Manager V4.2.1 was used to install the probe, the default probe port number is 5530.

For Linux and UNIX monitoring servers: You configured the EIF probe and port information for the Linux/UNIX monitoring server during installation, if you installed the monitoring server using the configuration instructions in this installation guide. However, if you did not configure this information, see “Configuring the hub monitoring server” on page 215 for the procedure.

You can use the `tacmd createEventDest` command to configure additional event destinations if the hub monitoring server should forward events to more than one Netcool/OMNIbus Probe for EIF. For more details on this command, see the *IBM Tivoli Monitoring: Command Reference*. For details on how to specify which event destinations to use for each situation, see “Specifying which situations forward events to Netcool/OMNIbus” on page 758.

Controlling event forwarding rate

The `om_tec.config` file controls the forwarding of events from IBM Tivoli Monitoring to IBM Tivoli Netcool/OMNIbus. This file contains the following parameter:

`BufferFlushRate=`*`events_per_minute`*
Specifies the number of events that are sent per minute when the adapter has reestablished its connection to the Netcool/OMNIbus Probe for Tivoli EIF. Once the adapter has recovered the lost connection, if there are events in the buffer, the events are sent at this rate per minute. The default value is 0, meaning all events are sent in one burst.

If your environment can have many open situation events, you might want to adjust this parameter to control the rate at which events are sent to the probe. To edit this file and change this parameter:

- On Windows:
  1. Open Manage Tivoli Enterprise Monitoring Services.
  2. Right-click Tivoli Enterprise Monitoring Server, and click Advanced → Edit EIF Configuration.
- On Linux or UNIX, edit file *`install_dir`*`/tables/`*`hostname`*`/TECLIB/ om_tec.config`, where *install_dir* is your Tivoli Monitoring installation directory and *hostname* is the name of the host running this monitoring server.

Restart the hub monitoring server after completing these steps:

- On Windows computers:
  1. Start the Manage Tivoli Enterprise Monitoring Services utility (**Start → (All) Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**).
  2. Right-click the hub monitoring server and select **Recycle** from the popup menu.
- On UNIX and Linux computers, enter the following commands:

  ```
  $ITM_INSTALL_PATH/bin/itmcmd stop tems_server
  $ITM_INSTALL_PATH/bin/itmcmd start tems_server
  ```

  where:

  `$ITM_INSTALL_PATH`
  Is the system variable defining the installation location of the monitoring server.

`tems_server`
Is the host name of the computer where the monitoring server is installed (the host where you are executing this command).

Verifying installation and configuration

After following the steps in the previous section, your Netcool/OMNIbus Probe for Tivoli EIF and ObjectServer should be receiving IBM Tivoli Monitoring situation events. Before additional customization, you should verify that the event synchronization component is installed and event integration is configured successfully. The verification requires coordination between the Netcool/OMNIbus and IBM Tivoli Monitoring administrators.

To verify unidirectional event flow:

1. Start the Tivoli Enterprise Portal.
2. Ensure you have a situation that has been configured to forward EIF events, and that the situation condition can be easily created.
3. After you have created a situation condition, use the Tivoli Enterprise Portal to verify that one or more situation events are open.

   The Tivoli Enterprise Portal Physical Navigator displays an event severity indicator at the top level of the Navigator and on all affected nested nodes in the view. As a result, events generated by a situation are always visible. You can also see the complete list of open situation events in the Situation Event Console workspace.
4. Start the Netcool/OMNIbus Event List UI.

   For example, on the OMNIbus server machine, run the following command to start the Netcool/OMNIbus Native Event List: `$OMNIHOME/bin/nco_event` where `$OMNIHOME` is the directory location of the Netcool/OMNIbus installation.
5. Refresh the event view in the Netcool/OMNIbus Event List UI and check for situation events from IBM Tivoli Monitoring. You should see the same set of events that are in the Tivoli Enterprise Portal Situation Event Console.

If you see events from IBM Tivoli Monitoring in the OMNIbus Event List UI, you have successfully set up unidirectional event flow in your environment.

If you do not see events from IBM Tivoli Monitoring in the OMNIbus Event List UI, see the *IBM Tivoli Monitoring: Troubleshooting Guide* for information about how to resolve any issues.

To verify bidirectional event flow:

1. Follow the steps in the previous section to first verify unidirectional event flow in your environment.
2. In the event view in the Netcool/OMNIbus Event List UI, perform an action (acknowledge, deacknowledge, delete, or clear) on one of the forwarded situation events from IBM Tivoli Monitoring.
3. In the Tivoli Enterprise Portal, verify that the state of the situation event changes to the expected state. See Table 129 on page 678 for details on the expected behavior of events.

If you see the status of the corresponding situation change within the Tivoli Enterprise Portal, you have successfully set up bidirectional event flows in your environment.

If you do not see the status of the corresponding situation event change within the Tivoli Enterprise Portal, see the *IBM Tivoli Monitoring: Troubleshooting Guide* for information about resolving issues.

Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration

If you have already installed and configured event integration between IBM Tivoli Monitoring and Netcool/OMNIbus and you are upgrading IBM Tivoli Monitoring, perform the tasks listed in Table 142 to upgrade each of the environments that will be receiving events from the latest version of IBM Tivoli Monitoring.

Table 142. Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration.

	Task	Architecture Type	Administrator
1.	Upgrade Netcool/OMNIbus ObjectServer, if necessary, to a release or fix pack version required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for instructions on performing the upgrade: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
2.	Back up the rules files in the Netcool/OMNIbus Probe for Tivoli EIF's operating system directory and then upgrade the Netcool/OMNIbus Probe for Tivoli EIF, if necessary, to the release required by IBM Tivoli Monitoring. See the Netcool/OMNIbus Information Center for instructions on performing the upgrade: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm. See also "Required software for event integration with Netcool/OMNIbus" on page 158.	Unidirectional and bidirectional	Netcool/OMNIbus
3.	"Upgrading the IBM Tivoli Monitoring Event Synchronization Component" on page 747.	Unidirectional and bidirectional	Netcool/OMNIbus or IBM Tivoli Monitoring
4.	"Upgrading existing IBM Tivoli Monitoring events to use new OMNIbus attributes" on page 748.	Unidirectional and bidirectional	Netcool/OMNIbus

Table 142. Upgrading from a previous installation of IBM Tivoli Monitoring and Netcool/OMNIbus integration. (continued)

	Task	Architecture Type	Administrator
5.	"Updating the Netcool/OMNIbus ObjectServer with IBM Tivoli Monitoring attributes, tables, and triggers" on page 730. **Note:** If you have made any customizations to the IBM Tivoli Monitoring triggers in the Netcool/OMNIbus ObjectServer, you should make the same updates to the new version of the triggers and procedures. You can edit the .sql files provided with the IBM Tivoli Monitoring Event Synchronization before loading them into Netcool/OMNIbus ObjectServer. For a list of customizations that might have been made to the existing triggers, see "Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754.	Unidirectional and bidirectional	Netcool/OMNIbus
6.	Ensure that you have a default deduplication trigger. For more information, see "Replacing the default deduplication trigger" on page 749.	Unidirectional and bidirectional	Netcool/OMNIbus
7.	If you did not replace the default deduplication trigger in the previous task, ensure that you have changed the existing default duplication trigger to ignore IBM Tivoli Monitoring events. See "Changing the default deduplication trigger" on page 735 for details.	Unidirectional and bidirectional	Netcool/OMNIbus
8.	"Updating the EIF Probe Rules" on page 750.	Unidirectional and bidirectional	Netcool/OMNIbus

Upgrading the IBM Tivoli Monitoring Event Synchronization Component

When you Upgrade the IBM Tivoli Monitoring event synchronization component for IBM Tivoli Netcool/OMNIbus you install the newest version of the Situation Update Forwarder, and replace any scripts that have been updated since the base version. The latest version of the IBM Tivoli Monitoring probe rules files and ObjectServer triggers are also copied to the system by the installation program and will replace the previous versions. The `tivoli_eif.rules` file will also be deleted in the omnibus directory under the event synchronization installation directory because it is no longer used.

If you are using a Netcool/OMNIbus multitier architecture, upgrade the Tivoli Monitoring Event Synchronization component with each ObjectServer in the aggregation tier. If you are using a single-tier architecture, upgrade the event synchronization component with each of your ObjectServers.

Upgrading from a wizard

Follow these steps to upgrade the event synchronization component via the installation wizard. The tasks in this section can be completed by either the Netcool/OMNIbus Administrator or the IBM Tivoli Monitoring Administrator.

1. On the host where you installed the previous version of the event synchronization component (normally the host of the Netcool/OMNIbus ObjectServer), start the event synchronization upgrade installation.
   - On Windows, double-click the `ESUpgrade23win32.exe` file in the `tec` subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image.
   - On Linux or UNIX, change to the `tec` subdirectory on the IBM Tivoli Monitoring V6.2.3 Tools DVD or DVD image, and run the following command:

     ```
     ESUpgrade23operating_system.bin
     ```

     where *operating_system*is the operating system you are installing on (`aix`, `HP11`, `Linux`, `linux390`, or `Solaris`).

   For example, run the following command on an AIX computer:

   ```
   ESUpgrade23Aix.bin
   ```
2. Click **Next** on the Welcome window.
3. Select **I accept the terms in the license agreement**, and click **Next**.

   A window is displayed indicating the installation is about to proceed.
4. Click **Next** to proceed with the installation.

   A progress indicator shows the progress of the installation.
5. Click **Finish** to exit the installer.

Upgrading existing IBM Tivoli Monitoring events to use new OMNIbus attributes

If you are upgrading from IBM Tivoli Monitoring Version 6.2.2 (or earlier) to the latest version of IBM Tivoli Monitoring, you must upgrade the existing monitoring events in the Netcool/OMNIbus ObjectServer alerts.status table to use a new OMNIbus attribute. To upgrade you apply the `itm_migrate.sql` file provided with the IBM Tivoli Monitoring Event Synchronization component to the ObjectServer.

The task in this section should be completed by the Netcool/OMNIbus Administrator.

The command to configure OMNIbus pipes the SQL command set into the SQL command-line tool and performs the updates to the ObjectServer.

1. Update the Object Server database with the following commands:
   - On Windows systems:

     ```
     %OMNIHOME%\..\bin\redist\isql -U username
       -P password
       -S server_name
       < path_to_file\itm_migrate.sql
     ```

     Where:

     `$OMNIHOME`
       Is the system-defined variable defining the installation location of OMNIbus.

     `username`
       Is the OMNIbus Object Server user name.

     `password`
       Is the OMNIbus Object Server password.

     `server_name`
       Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

- On UNIX systems:

```
$OMNIHOME/bin/nco_sql -user username
  -password password
  -server server_name
  < path_to_file/itm_migrate.sql
```

Where:

$OMNIHOME
Is the system-defined variable defining the installation location of OMNIbus.

username
Is the OMNIbus Object Server user name.

password
Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file. The SQL files are located in the `event_sync_install_dir/omnibus directory` where `event_sync_install_dir` is the location where the IBM Tivoli Monitoring Event Synchronization component is installed.

Replacing the default deduplication trigger

With previous versions of the triggers provided by the IBM Tivoli Monitoring event synchronization component, the default deduplication trigger was overwritten in the Netcool/OMNIbus ObjectServer. The overwriting causes you to lose any customizations you made to the trigger. The latest version of the event synchronization component provides a deduplication trigger specific to Tivoli Monitoring called `itm_deduplication`. If your default deduplication trigger has IBM Tivoli Monitoring logic, follow these steps to apply the default deduplication trigger to the Netcool/OMNIbus ObjectServer. The tasks in this section should be completed by the Netcool/OMNIbus Administrator.

1. Open the `automation.sql` file located at `%OMNIHOME%\etc\automation.sql` (Windows) or `$OMNIHOME/etc/automation.sql` (UNIX).
2. Locate the command that creates the deduplication trigger. For example:

```
create or replace trigger deduplication
group default_triggers
priority 1
comment 'Deduplication processing for ALERTS.STATUS'
before reinsert on alerts.status
for each row
begin
   set old.Tally = old.Tally + 1;
   set old.LastOccurrence = new.LastOccurrence;
   set old.StateChange = getdate();
   set old.InternalLast = getdate();
   set old.Summary = new.Summary;
   set old.AlertKey = new.AlertKey;
   if (( old.Severity = 0) and (new.Severity > 0))
   then
     set old.Severity = new.Severity;
   end if;
end;
go
```

3. Copy this command to a temporary file. For this example, the temporary file is `/tmp/dedup.sql`.
4. Edit the command in the temporary file to add the line in bold type so that the deduplication trigger ignores IBM Tivoli Monitoring events and does not change the summary and severity of those events:

```
for each row
when (new.Type != 20) and (new.Type != 21)
begin
```

5. Run the command to replace the deduplication trigger.
   - On Windows:

     ```
     %OMNIHOME%\..\bin\redist\isql -U username -P password -S server_name < C:\tmp\dedup.sql
     ```

   - On UNIX:

     ```
     $OMNIHOME/bin/nco_sql -user username -password password -server server_name < /tmp/dedup.sql
     ```

   where:

   OMNIHOME
   Is the system-defined variable defining the installation location of OMNIbus

   username
   Is the OMNIbus Object Server user name

   password
   Is the OMNIbus Object Server password

   server_name
   Is the OMNIbus Object Server name defined for process control.

Updating the EIF Probe Rules

The Netcool/OMNIbus Probe for Tivoli EIF should be updated with the latest versions of the IBM Tivoli Monitoring probe rules files. This task should be performed by the Netcool/OMNIbus administrator.

IBM Tivoli Monitoring requires the Netcool/OMNIbus Probe for Tivoli EIF version 10 or later. This release of the EIF probe provides a `tivoli_eif.rules` file that includes rules files for other products. You should use the probe's version of `tivoli_eif.rules` instead of the `tivoli_eif.rules` file that was provided by IBM Tivoli Monitoring releases prior to V6.2.3. The latest version of the IBM Tivoli Monitoring probe rules are in the `itm_event.rules` include file that is included by the probe's version of `tivoli_eif.rules`.

If you are using a `tivoli_eif.rules` file from a previous release of IBM Tivoli Monitoring and you customized the rules, determine what customizations were made to the `tivoli_eif.rules` file that you backed up before upgrading the EIF Probe. Then perform one of the following actions:

- Merge your customized rules for IBM Tivoli Monitoring events into the `itm_event.rules` provided with the latest version of IBM Tivoli Monitoring. Or:
- Determine if you can use the `itm_event.rules` as it is, and put your IBM Tivoli Monitoring event customization in the new `itm_custom_override.rules` file that is included by `itm_event.rules` to make upgrades easier in the future.

To update the EIF probe to use the latest IBM Tivoli Monitoring rules, see "Configuring the Netcool/OMNIbus EIF probe" on page 740.

Customizing Event Integration

After verifying that event integration is working as expected, additional configuration tasks are available to customize the event integration between IBM Tivoli Monitoring and Netcool/OMNIbus.

Table 143. Additional configuration tasks after event integration installation.

Task	Architecture type	Administrator
"Configuring IBM Tivoli Monitoring Situation Update Forwarder event flow to OMNIbus" on page 751.	Bidirectional	Netcool/OMNIbus

Table 143. Additional configuration tasks after event integration installation. (continued)

Task	Architecture type	Administrator
"Changing the configuration values of the IBM Tivoli Monitoring Situation Update Forwarder" on page 752.	Bidirectional	Netcool/OMNIbus
"Updating the IBM Tivoli Monitoring Situation Forwarder to forward event status updates to additional monitoring servers" on page 752.	Bidirectional	Netcool/OMNIbus
"Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers" on page 754.	Bidirectional	Netcool/OMNIbus
"Changing the default acknowledgment timeout used when sampled events are deleted or cleared in Netcool/OMNIbus" on page 756.	Bidirectional	Netcool/OMNIbus
"Editing the configuration parameters that the hub monitoring server uses to forward events to the Netcool/OMNIbus Probe for Tivoli EIF and configure up to five failover EIF probes" on page 757.	Unidirectional and bidirectional	IBM Tivoli Monitoring
"Specifying which situations forward events to Netcool/OMNIbus" on page 758.	Unidirectional and bidirectional	IBM Tivoli Monitoring Operator
"Understanding and Customizing the Event Contents" on page 765.	Unidirectional and bidirectional	IBM Tivoli Monitoring
"Converting from unidirectional architecture to bidirectional architecture" on page 759.	Bidirectional	Netcool/OMNIbus
"Customizing event status processing behavior when agent switching is used or the agent goes offline" on page 760.	Unidirectional and bidirectional	IBM Tivoli Monitoring
"Creating a Netcool/OMNIbus WebGUI tool to launch from WebGUI to the Tivoli Enterprise Portal" on page 763.	Unidirectional and bidirectional	Netcool/OMNIbus

Configuring IBM Tivoli Monitoring Situation Update Forwarder event flow to OMNIbus

If you are using the bidirectional architecture, the IBM Tivoli Monitoring Situation Update Forwarder can send events to Netcool/OMNIbus in the following cases:

- The Situation Update Forwarder sends events to Netcool/OMNIbus if it detect errors when it is processing event status updates initiated from the OMNIbus Object Server.
- The Situation Update Forwarder sends an event to Netcool/OMNIbus to update the default acknowledgement timeout when the `sitconf.sh` or `sitconf.cmd` commands are executed.

To send events from the Situation Update Forwarder to Netcool/OMNIbus, update the values for the following parameters in the *event_sync_install_dir*/omnibus/errorevent.conf file:

```
ServerName
ServerPort
```

where:

`event_sync_install_dir`
Is the location where the IBM Tivoli Monitoring Event Synchronization Component is installed on your Netcool/OMNIbus ObjectServer system.

`ServerName`
Is the name of the computer where the EIF probe is running.

`ServerPort`
Is the listening port for the EIF probe. The default value is 9998.

Changing the configuration values of the IBM Tivoli Monitoring Situation Update Forwarder

After installation of IBM Tivoli Event Synchronization, you can change the configuration values of the Situation Update Forwarder that were specified during installation. See Table 138 on page 722 and Table 139 on page 723 for the list of configuration values that can be modified. Use the procedure in this section to change the configuration parameters of the IBM Tivoli Monitoring Situation Update Forwarder when you are using the bidirectional architecture.

If you want to change any of the settings for the IBM Tivoli Monitoring Situation Update Forwarder on the Netcool/OMNIbus ObjectServer system, use the `sitconfig.sh` command on Linux/UNIX or the `sitconf.cmd` command on Windows. You have two options for running this command:

- Manually modify the IBM Tivoli Monitoring Situation Update Forwarder configuration file for event synchronization (named `situpdate.conf` by default and located in `<`*`event_sync_install_dir`*`>/etc`) and then run: `sitconfig.sh update <config_filename>`.
- Run the `sitconfig.sh` command directly, specifying only those settings that you want to change.

See the *IBM Tivoli Monitoring: Command Reference* for Tivoli Netcool/OMNIbus commands for the full syntax of this command.

After you change the configuration of the event synchronization, you must manually stop and restart the IBM Tivoli Monitoring Situation Update Forwarder process from the `<event_sync_installdir>/bin` directory. On a Windows operating system, use the **`stopSUF.cmd`** and **`startSUF.cmd`** commands. On operating systems such as UNIX, use the **`stopSUF.sh`** and **`startSUF.sh`** commands.

Updating the IBM Tivoli Monitoring Situation Forwarder to forward event status updates to additional monitoring servers

If you are using the bidirectional architecture and have multiple hub monitoring servers forwarding situation events to the same Netcool/OMNIbus ObjectServer, the IBM Tivoli Monitoring Situation Update Forwarder must be defined with the list of monitoring servers that it should send event status updates to. When you install the IBM Tivoli Monitoring Event Synchronization Component and specify the IBM Tivoli Monitoring Situation Forwarder configuration parameters, you specify the SOAP URL of one hub monitoring server.

To add additional hub monitoring servers to the list that can receive event status updates from the Netcool/OMNIbus ObjectServer, issue the `sitconfuser` command as described in the following steps. Ensure the IBM Tivoli Monitoring Situation Update Forwarder is configured for both the fully qualified host name and the short host name of the hub monitoring server.

If you have configured the Hot Standby feature, you must configure the host name information for both the primary and secondary hub monitoring servers.

This task is performed on the system where the Netcool/OMNIbus ObjectServer and IBM Tivoli Monitoring Situation Update Forwarder are installed.

- On Windows operating systems, change to the `<event_sync_installdir>\bin` directory and enter the following command:

  ```
  sitconfuser.cmd add serverid=server userid=user password=password
  pathc=path_to_conf_file type=OMNIBUS
  ```

  where:

  `<event_sync_installdir>`
  Is the directory where the IBM Tivoli Monitoring event synchronization component is installed on your Netcool/OMNIbus ObjectServer system.

  `server`
  Is the fully qualified host name of the hub monitoring server.

  `user`
  Is the user ID used to access the computer where the hub monitoring server is running.

  `password`
  Is the password of the user ID specified by the user parameter.

  `path_to_conf_file`
  Is the directory containing the `situser.conf` file. By default, the `situser.conf` file is in the `<event_sync_installdir>\etc` directory.

  Repeat this command to add short host name information for the same hub monitoring server by specifying the short host name value for the `serverid` parameter.

- On UNIX operating systems, change to the `<event_sync_installdir>/bin` directory and enter the following command:

  ```
  sitconfuser.sh add serverid=server userid=user password=password
  pathc=path_to_conf_file type=OMNIBUS
  ```

  where:

  `<event_sync_installdir>`
  Is the directory where the IBM Tivoli Monitoring event synchronization component is installed on your Netcool/OMNIbus ObjectServer system.

  `server`
  Is the fully qualified host name of the monitoring server.

  `user`
  Is the user ID used to access the computer where the monitoring server is running.

  `password`
  Is the password of the user ID specified by the user parameter.

  `path_to_conf_file`
  Is the directory containing the `situser.conf` file. By default, the `situser.conf` file is in the `<event_sync_installdir>/etc` directory.

  Repeat this command to add short host name information for the same monitoring server by specifying the short host name value for the `serverid` parameter.

Repeat this step for each monitoring server that you want to add.

You can also delete monitoring servers, see the *IBM Tivoli Monitoring: Command Reference* for the full syntax of this command.

After you change the configuration of the event synchronization, you must manually stop and restart the IBM Tivoli Monitoring Situation Update Forwarder process from the <event_sync_installdir>/bin directory. On Windows operating systems use stopSUF.cmd and startSUF.cmd. On UNIX operating systems use stopSUF.sh and startSUF.sh.

Customizing how the IBM Tivoli Monitoring OMNIbus triggers handle event status updates from the monitoring servers

You can configure how the IBM Tivoli Monitoring OMNIbus triggers handle acknowledgment expiration and resurface status update events from the hub monitoring servers when the bidirectional architecture is used. You can also configure how stop situation event status updates are handled for pure situation events for both the unidirectional and bidirectional architecture. For more information about handling event status updates, see “Event behavior” on page 678.

It might also be necessary to edit the procedure that sends event status updates from Netcool/OMNIbus to IBM Tivoli Monitoring if Netcool/OMNIbus is installed into a directory path containing spaces or the Situation Update Forwarder will be run by a user other than root.

Configuring acknowledgement expiration and resurface status update event behavior

The **get_config_parms** procedure in the <event_sync_install_dir>/omnibus/itm_proc.sql file defines three configuration parameters:

```
set sit_ack_expired_def_action = 'REJECT'
set sit_resurface_def_action = 'ACCEPT'
set situpdate_conf_file = 'situpdate.conf'
```

Note: When unidirectional architecture is used, the itm_proc.sql file must be configured to accept the acknowledgment expiration and resurface status update events from the hub monitoring servers.

The sit_ack_expired_def_action variable defines the action to be taken for an event by the OMNIbus server when an acknowledgment expiration status update is received for an event from a hub monitoring server. An acknowledgment expiration status update occurs if the event was acknowledged in the Tivoli Enterprise Portal with a timeout and the event is still true when the timeout expires. The default action is to reject the request if the event is still in the acknowledged state in the Netcool/OMNIbus ObjectServer. In this case, OMNIbus sends information back to the hub monitoring server to change the state of the event from Acknowledgement Expired to Acknowledged. If you want to change the action taken by the OMNIbus server to Accept the acknowledgment expiration, modify the statement in itm_proc.sql to set sit_ack_expired_def_action = 'ACCEPT'. In this case, the event is deacknowledged in Netcool/OMNIbus but still has the acknowledged with expiration status in the hub monitoring server.

The variable sit_resurface_def_action defines the action to be taken by the OMNIbus server when a situation event has resurfaced in IBM Tivoli Monitoring. An event is resurfaced when it has been deacknowledged in the Tivoli Enterprise Portal Situation Event Console. The default action of the OMNIbus server is to Accept this request and Deacknowledge the event. If you would like to change the action taken by OMNIbus server to Reject the resurface of the event, modify the statement to set sit_resurface_def_action = 'REJECT' in itm_proc.sql. If the action is REJECT, OMNIbus then sends information back to the hub monitoring server to change the state of the event back to Acknowledged and the event remains in the acknowledged state in the Netcool/OMNIbus ObjectServer.

The variable situpdate_conf_file specifies the name of the configuration file to be used by the IBM Tivoli Monitoring Situation Update Forwarder. If you would like to change the name of the configuration file, modify the statement to set situpdate_conf_file = 'newname.conf'.

After modifying itm_proc.sql, issue the following command on your Netcool/OMNIbus ObjectServer:

- On Windows operating systems:

```
%OMNIHOME%\..\bin\redist\isql -U <username>
-P <password>
-S <server_name>
< <path_to_file>\itm_proc.sql
```

- On Linux/UNIX operating systems:

  ```
  $OMNIHOME/bin/nco_sql -user <username>
  -password <password>
  -server <server_name>
  < <path_to_file>/itm_proc.sql
  ```

  Where:

  OMNIHOME
  Is the system-defined variable defining the installation location of OMNIbus.

  username
  Is the OMNIbus ObjectServer user name.

  password
  Is the OMNIbus ObjectServer password.

  server_name
  Is the OMNIbus ObjectServer name defined for process control.

  path_to_file
  Is the fully qualified path to the specified SQL file.

Configuring stop situation event behavior for pure situations

When the IBM Tivoli Monitoring triggers in Netcool/OMNIbus process a situation stop event, the triggers clear all events for the situation that are detected by the remote monitoring server specified by the `ITMThruNode` OMNIbus attribute. However, you can configure the IBM Tivoli Monitoring triggers to ignore situation stop events for pure events. To configure this behavior, you must update the following statement in the `itm_event_clear` trigger from:

```
set puresitstop = 'CLOSE';
```

to:

```
set puresitstop = 'OPEN';
```

You can either use the Netcool/OMNIbus Administrator to edit the `itm_event_clear` trigger or modify the trigger in the `itm_db_update.sql` file provided with the IBM Tivoli Monitoring event synchronization component and then re-load the `itm_db_update.sql` file into Netcool/OMNIbus ObjectServer. After modifying `itm_db_update.sql`, issue the following command on your Netcool/OMNIbus ObjectServer:

- On Windows operating systems:

  ```
  %OMNIHOME%\..\bin\redist\isql -U <username>
  -P <password>
  -S <server_name>
  < <path_to_file>\itm_db_update.sql
  ```

- On Linux/UNIX operating systems:

  ```
  $OMNIHOME/bin/nco_sql -user <username>
  -password <password>
  -server <server_name>
  < <path_to_file>/itm_db_update.sql
  ```

  Where:

  OMNIHOME
  Is the system-defined variable defining the installation location of OMNIbus.

  username
  Is the OMNIbus ObjectServer user name.

password
Is the OMNIbus ObjectServer password.

server_name
Is the OMNIbus ObjectServer name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

Editing the eventcmd procedure

The `eventcmd` procedure in the `<event_sync_install_dir>/omnibus/itm_proc.sql` file might also need the following modifications:

- The value of the executable variable must not contain any spaces. If the Situation Update Forwarder was installed in a directory with spaces, change the executable variable setting from $EVENTCMD to a path for the `eventcmd.bat` file that does not contain spaces, for example, on Windows: `C:\Progra~1\IBM\SitForwarder\omnibus\eventcmd.bat`.
- Change the host variable to reflect the actual host name on which the Object Server is running.
- You might have to change the user and group variables from the default settings if the executable cannot be run as root.

Editing the writeitmcmd procedure

You might also need to make the following modifications to the `writeitmcmd` procedure in the `<event_sync_install_dir>/omnibus/itm_proc.sql` file:

- Change the host variable to reflect the actual host name on which the Object Server is running.
- You might have to change the user and group variables from the default settings if the executable cannot be run as root.

Changing the default acknowledgment timeout used when sampled events are deleted or cleared in Netcool/OMNIbus

When a situation event from a sampled situation is forwarded to the Tivoli Netcool/OMNIbus ObjectServer and that event is later deleted or cleared in the ObjectServer, the behavior of the bidirectional event synchronization architecture is to send a request to the hub Tivoli Enterprise Monitoring Server to acknowledge the situation with a specified timeout. The reason for this behavior is that you cannot close sampled situation events unless the monitoring agent determines the situation condition is no longer true.

If the acknowledgment timeout of the situation expires and the situation is still true, IBM Tivoli Monitoring sends an acknowledgement expiration status update event to Netcool/OMNIbus. The status update event causes a new event to be opened in the Netcool/OMNIbus ObjectServer and the Netcool/OMNIbus operator is notified that the event condition has not been resolved. If the situation condition becomes false, then the event is closed in IBM Tivoli Monitoring, and the event remains cleared in the Netcool/OMNIbus ObjectServer.

The default acknowledgment expire time for sampled situations is 59 minutes. This default time can be changed in the situation timeouts configuration file on the ObjectServer (`sit_timeouts.conf`). Also, expiration times for individual situations can be configured in this file. After editing this file, you can have the expire times dynamically loaded into the ObjectServer by using the `sitconf.sh` refresh (UNIX) or `sitconf.cmd` refresh (Windows) command in `<event_sync_installdir>/bin`.

Note: You must perform the task in the section "Configuring IBM Tivoli Monitoring Situation Update Forwarder event flow to OMNIbus" on page 751 before running the `sitconf`command.

Editing the configuration parameters that the hub monitoring server uses to forward events to the Netcool/OMNIbus Probe for Tivoli EIF and configure up to five failover EIF probes

Edit the hub monitoring server Tivoli Event Integration Facility EIF configuration file to customize the configuration parameters that specify up to five failover Netcool/OMNIbus EIF probes or to adjust size of the event cache.

When the Tivoli Event Integration Facility (EIF) has been enabled on the hub monitoring server and the default EIF server (Tivoli Enterprise Console Event Server or Netcool/OMNIbus EIF probe) and port number have been specified, the EIF configuration file is updated with the information. This is the default EIF receiver of forwarded situation events.

See "Configuring the hub monitoring server to forward events" on page 743 for instructions on configuring the monitoring server to enable the Tivoli Event Integration Facility.

Take these steps to edit the EIF configuration file:

1. Open the `om_tec.config` file:
   a. In the Manage Tivoli Monitoring Services window, right-click Tivoli Enterprise Monitoring Server and click **Advanced → Edit EIF Configuration**.
   b. Open *`Install_dir`*`/tables/`*`tems_name`*`/TECLIB/om_tec.config` in a text editor where `Install_dir` is the directory where IBM Tivoli Monitoring is installed and *`tems_name`* is the name of the hub monitoring server supplied during monitoring server installation.
2. Edit any of the event server configuration parameters for the event integration facility.
3. When you are finished editing `om_tec.config`, save the file.
4. You must restart the hub monitoring server or, alternatively, you can use the `tacmd refreshTECinfo` command to complete the updates without having to restart the monitoring server. To use this command, log in to the command-line interface with `tacmd` login, then run `tacmd refreshTECinfo -t eif` to complete the EIF configuration. For more information on this command, see the *IBM Tivoli Monitoring: Command Reference*.

Table 144. Supported Netcool/OMNIbus Probe for Tivoli EIF configuration parameters for the Event Integration Facility (EIF).

Parameters	Value	Remarks
ServerLocation=	`tec_server_addr`	Host name or IP address of the OMNIbus EIF probe. To provide event failover, you can indicate up to five default EIF probes, separating each with a comma. When the default EIF probe is unavailable, the situation event goes to the next server in the list.
ServerPort=	[port:0]	OMNIbus EIF probe listening port, which is 9998 by default. Specify 0 if the OMNIbus EIF probe uses the port mapper. If you specified multiple server locations, add the corresponding port numbers here, separating each with a comma.
EventMaxSize=	4096	Maximum number of characters allowed in the event. This parameter is disabled by default. To enable it, remove the pound symbol (#) at the beginning of the line.

Table 144. Supported Netcool/OMNIbus Probe for Tivoli EIF configuration parameters for the Event Integration Facility (EIF). (continued)

Parameters	Value	Remarks
RetryInterval=	5	The number of times to retry connection with the EIF probe before returning an error.
getport_total_timeout_usec=	50500	The number of seconds to continue attempting to connect to the EIF probe port before timing out. The default is 14 hours.
NO_UTF8_CONVERSION=	YES	Events are already in UTF8, no conversion is needed. Must be set to YES.
ConnectionMode=	co	The connection mode.
BufferEvents=	YES	Whether the EIF buffers the event. This must be set to YES.
BufEvtMaxSize=	4096	Maximum size of the event cache. The default is initially 4096 KB and you can change it here.
BufEvtPath=	`./TECLIB/om_tec.cache`	Path of the event cache file. The default is `./TECLIB/om_tec.cache`.
FilterMode=	OUT	Enable event filtering. This is set to OUT by default.

Filter:	Class=ITM_Generic;
	master_reset_flag=”;
		To filter out specific classes, use this keyword. By default, situation events of the class `ITM_Generic` and those that send no master reset flag are not forwarded.

Specifying which situations forward events to Netcool/OMNIbus

By default, when the hub Tivoli Enterprise Monitoring Server has been configured for the Tivoli Event Integration Facility, events for situations enabled for event forwarding are sent to the default EIF receiver. See the “Configuring the hub monitoring server to forward events” on page 743 section for details on how to specify the default EIF receiver. If you want to forward situation events to multiple Netcool/OMNIbus Probes for Tivoli EIF, you must use the `tacmd createEventDest` command to create additional event destinations and then use the Tivoli Enterprise Portal Situation Editor to specify which event destinations to use for the situation’s events. For details on the `tacmd createEventDest` command, see the *IBM Tivoli Monitoring: Command Reference*.

Also by default, event forwarding is not enabled for a new situation unless you base the new situation definition on an existing situation that already has event forwarding enabled, or you explicitly enable event forwarding for the situation. You can also use the Tivoli Enterprise Portal Situation Editor to ensure that Event Integration Facility forwarding is enabled for each situation whose events should be forwarded to Netcool/OMNIbus.

The tasks in this section can be completed by the IBM Tivoli Monitoring Tivoli Enterprise Portal Operator.

Complete these steps to specify the destination EIF receiver and severity for a forwarded event:

1. In the Tivoli Enterprise Portal Navigator view, either right-click the Navigator item that the situation is associated with and click **Situations** or click **Situation Editor** in the main toolbar.
2. Select the situation whose events should be forwarded to Netcool/OMNIbus.
3. Click the **EIF** tab.

4. Select **Forward Events to an EIF Receiver** to specify that an EIF event is sent for each event that opens for this situation.
5. Select the **EIF Severity** to apply to forwarded events for this situation. <Default EIF Severity> uses the same severity as is used for the situation at this Navigator item.
6. Assign any other EIF receivers:
   - To add a destination, select it from the **Available EIF Receivers** list and move to the Assigned list. (After selecting the first destination, you can use Ctrl+click to select other destinations or Shift+click to select all destinations between the first selection and this one.)
   - To remove a destination, select it from the **Assigned EIF Receivers** list and move to the Available list.

   The **Available EIF Receivers** list shows all of the defined EIF destinations that were created with the `tacmd createEventDest` command.
7. Save the situation definition with your changes by clicking **Apply**, to keep the Situation editor open, or **OK**, to close the Situation editor.

Converting from unidirectional architecture to bidirectional architecture

If you are using the unidirectional architecture for event integration between your hub monitoring servers and Netcool/OMNIbus and you decide to switch to the bidirectional architecture, you must perform the following tasks for each of your Netcool/OMNIbus ObjectServers that will be sending event status updates back to a hub monitoring server:

1. Ensure you have the IBM Tivoli Monitoring Event Synchronization Component installed on the machine with your Netcool/OMNIbus ObjectServer.
2. Configure the OMNIbus server to support execution from scripts for running the Situation Update Forwarder. For more information, see “Configuring the OMNIbus server for program execution from scripts” on page 742.
3. Define each monitoring server that is forwarding events to Netcool/OMNIbus ObjectServer to the IBM Tivoli Monitoring Situation Update Forwarder and start the IBM Tivoli Monitoring Situation Update Forwarder. For more information, see “Updating the IBM Tivoli Monitoring Situation Forwarder to forward event status updates to additional monitoring servers” on page 752.
4. Update the Netcool/OMNIbus ObjectServer database schema to add the IBM Tivoli Monitoring triggers that send bi-directional event status updates to your hub monitoring servers. Follow the procedure in “Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers” on page 733 to load the `itm_sync.sql` file. You do not need to load the `itm_db_update.sql` or `itm_proc.sql` files because they were loaded into the ObjectServer when you enabled unidirectional event integration.
5. Edit the get_config_parms procedure in Netcool/OMNIbus Object to choose the default bidirectional behavior when an Acknowledgement Expired event status update is sent from the hub monitoring server to Netcool/OMNIbus:
   a. Use the nco_config utility to start Netcool/OMNIbus Administrator.
   b. Connect to the Netcool/OMNIbus ObjectServer where you are making the change.
   c. Select **Automation** → **Procedures** from the menu.
   d. Edit the get_config_parms procedure and change:

      ```
      set sit_ack_expired_def_action = 'ACCEPT';
      ```

      to

      ```
      set sit_ack_expired_def_action = 'REJECT';
      ```

   e. Save the trigger and exit from Netcool/OMNIbus Administrator.

Customizing event status processing behavior when agent switching is used or the agent goes offline

The variables in Table 145 on page 761 can be added to the monitoring server's environment file to customize the behavior of event status processing when agent switching is used or when the agent goes offline. The first two variables help ensure that events are not closed by the agent's primary monitoring server after the agent has switched to its secondary monitoring server. The monitoring server's environment file can be found in these locations:

- On Windows systems:

  ```
  ITM_HOME\cms\KBBENV
  For example: C:\IBM\ITM\cms\KBBENV
  ```

- On Linux/UNIX systems:

  ```
  ITM_HOME/config/tems_hostname_ms_tems_name.config
  For example: /opt/IBM/ITM/config/edinburg_ms_labtems.config
  ```

  For Linux/UNIX systems, you must add the variables to the `.config` and the `.ini` files. The name and location of the `.ini` file is `ITM_HOME/config/ms.ini`.

- On z/OS systems:

  ```
  &shilev.&rtename.RKANPARU(KDSENV)
  For example: ITM.SYP1.RKANPARU(KDSENV)
  ```

  Note: The `&shilev` and `&rtename` are variables that correspond to high level qualifiers of the RKANPARU(KDSENV) partitioned dataset. These variables can take 1 to 8 characters.

You must recycle the Tivoli Enterprise Monitoring Server after modifying the environment file for your changes to be picked up.

Table 145. Variables to customize the behavior of event status processing when agent switching is used.

Variable	Architecture Type	Details	Administrator
IRA_MIN_NO_DATA_WAIT_TIME	Unidirectional and bidirectional	The minimum time to wait before the monitoring server closes a situation event. This parameter is defined in number of seconds. The default value is zero. By default, after an agent is disconnected from a Tivoli Enterprise Monitoring Server, situations already open will remain open for three situation polling intervals. For example, take two sampled situations, S1 and S2, with intervals of 30 seconds and 15 minutes respectively. Both situations are open when the agent loses connection. Situation S1 closes after at least one minute and 30 seconds. Situation S2 closes after at least 45 minutes. With agent switching, if a situation closes too soon it might generate duplicate events because the agent did not have sufficient time to connect to the backup monitoring server before the primary server closes the original event. This is particularly true for situations with very short polling intervals. In such a scenario you can use the **IRA_MIN_NO_DATA_WAIT_TIME** variable to set the minimum wait time before a situation is closed. Using the example above, if IRA_MIN_NO_DATA_WAIT_TIME is set to 600 (5 minutes), S1 will close after 5 minutes not 90 seconds. S2 is unaffected and will close after 45 minutes as before. **Note:** You should set this variable in the environment file for all of your monitoring servers.	IBM Tivoli Monitoring

Table 145. Variables to customize the behavior of event status processing when agent switching is used. (continued)

Variable	Architecture Type	Details	Administrator
CMS_SIT_TIME_VALIDATION	Unidirectional and bidirectional	Valid entries are Y or N. The default is N. By default, the monitoring server handles situation events on a first-come-first-serve basis. In a scenario where agent switching is enabled an agent might send events through two different monitoring servers. The events that arrive first might not necessarily be the earlier events if one of the monitoring servers encountered connection issues. This generally has little impact on situation event processing, except when a monitoring server is shutdown and some situations might be closed prematurely even though the agent is already connected to a different monitoring server. You must perform two actions to avoid this scenario: 1. All monitoring server hosts should synchronize time, preferable through Internet Time Protocol (ITP) clients. 2. You should add the **CMS_SIT_TIME_VALIDATION=Y** variable to all monitoring server environment files. This switches the LCLTMSTMP column in situation events to use UTC time instead of local time, which is then used to determine event order.	IBM Tivoli Monitoring

Table 145. Variables to customize the behavior of event status processing when agent switching is used. (continued)

Variable	Architecture Type	Details	Administrator
CMS_SIT_CHECK_NODESTS	Unidirectional and bidirectional	Valid entries are Y or N. The default is N. This variable is only applicable for users of event integration with Tivoli Enterprise Console and Omnibus. When the **CMS_SIT_CHECK_NODESTS** variable is set to Y in the environment file, the hub monitoring servers check the agent status whenever a close status update event is forwarded to Netcool/Omnibus. The CMS_SIT_CHECK_NODESTS variable should only be added to the hub monitoring server environment file. If the agent is offline the close status update event is tagged with a special *OFFLINE* indicator in the `situation_eventdata` EIF slot. If you do not want events to be closed in Netcool/OMNIbus when an agent goes offline, you can customize the EIF probe rules to ignore close events where the situation_status slot is set to **N** and the situation_eventdata EIF slot is set to **OFFLINE**. See "Customizing the rules file" on page 742 for details on how to add customizations to the EIF probe rules. You should also consider setting the IRA_MIN_NO_DATA_WAIT_TIME environment variables described in this table so that close status events are not sent to Netcool/OMNIbus until after the agent offline condition has been detected. (If the default agent heartbeat interval is used, it can take between 10 to 20 minutes before the monitoring server detects that the agent is no longer online.) **Note:** You should only set the CMS_SIT_CHECK_NODESTS variable in the environment file of your hub monitoring server.	IBM Tivoli Monitoring

Creating a Netcool/OMNIbus WebGUI tool to launch from WebGUI to the Tivoli Enterprise Portal

If you want your Netcool/OMNIbus WebGUI users to launch the Tivoli Enterprise Portal to see additional details about events shown in the WebGUI, you can create a WebGUI tool that launches the Tivoli Enterprise Portal from the Netcool/OMNIbus Event List UI. The Tivoli Enterprise Portal will be launched in the context of the managed system associated with the event. This support allows WebGUI users to quickly determine and resolve issues in their IT environment.

Follow the instructions in the IBM Tivoli Netcool/OMNIbus Information Center to create event management tools for the WebGUI and choose Script as the tool type. For more details on the tool creation procedure, see the Netcool/OMNIbus information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm.

For the script command, use the format example below that supports your environment.

Note: The examples below use http and port number 1920. If you use https or a different port number to access the Tivoli Enterprise Portal Server, you must customize the script commands below to reflect that environment.

Scenario 1: Single hub monitoring server is sending events to the Netcool/OMNIbus ObjectServer. The Tivoli Enterprise Portal Server and hub monitoring server are on the same host. Use the following script command format for this scenario:

```
var  str_appllabel = '{@ITMApplLabel}';
if  (!(str_appllabel == "A:P:S")&& !(str_appllabel == "A:P:L") )
{ window.open("http://{@ITMHostname}:1920///cnp/kdh/lib/cnp.html?managed_system_name={@ITMSitOrigin}"); }
```

Scenario 2: Single hub monitoring server is sending events to the Netcool/OMNIbus ObjectServer. Tivoli Enterprise Portal Server and the hub monitoring server are on different hosts. Use the following script command format for this scenario:

```
var  str_appllabel = '{@ITMApplLabel}';
if  (!(str_appllabel == "A:P:S")&& !(str_appllabel == "A:P:L") )
{ window.open("http://<teps-hostname>:
1920///cnp/kdh/lib/cnp.html?managed_system_name={@ITMSitOrigin}"); }
```

In this example, replace `<teps-hostname>` with the IP address or hostname of your Tivoli Enterprise Portal Server.

Scenario 3: Multiple hub monitoring servers are sending events to the same Netcool/OMNIbus ObjectServer. Use the following script command format for this scenario:

```
var  str_appllabel = '{@ITMApplLabel}';
if ( ! (str_appllabel == "A:P:S") &&  !(str_appllabel == "A:P:L") )
{

var strtems={@ITMHostname};
if( strtems == "<tems_host1>" )
{ window.open("http://<teps_host1>:1920///cnp/kdh/lib/cnp.html?managed_system_name={@ITMSitOrigin}"); }
else if( strtems == "<tems_host2>")
{ window.open("http://<teps_host2>:1920///cnp/kdh/lib/cnp.html?managed_system_name={@ITMSitOrigin}"); }
}
```

Change the following in this script command example:

- *<tems_host1>* to the IP address or the host name of the server where the first hub monitoring server exists.
- *<tep_host1>* to the IP address or the host name of the server where the first portal server exists.
- *<tems_host2>* to the IP address or the host name of the server where the second hub monitoring server exists.
- *<tep_host2>* to the IP address or the host name of the server where the second portal server exists.

If you have more than two hub monitoring servers sending events to the Netcool/OMNIbus ObjectServer, add additional *else if* checks for each hub monitoring server.

You can configure the Netcool/OMNIbus WebGUI and Tivoli Enterprise Portal to authenticate users from an external LDAP repository. If you do this configuration and add the Single Sign-On (SSO) capability between both servers (WebGUI and Tivoli Enterprise Portal server), you can open and work on both consoles with the same user credentials (authenticated by LDAP server). You do not need to login separately to each console.

In order to configure Tivoli Enterprise Portal console for Single Sign-On, see the *IBM Tivoli Monitoring: Administrator's Guide*. To configure the WebGUI console for Single Sign-On, see the Netcool/OMNIbus Information Center http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.tivoli.namomnibus.doc/welcome_ob.htm.

Understanding and Customizing the Event Contents

From the **Situation editor EIF** tab of the Tivoli Enterprise Portal, you can create map definitions for situation events sent to the EIF Probe. The EIF Slot Customization window, which is opened from the EIF tab, is used to customize how situation events are mapped to forwarded EIF events, thus overriding the default mapping between situation event attributes and EIF attribute slots.

With IBM Tivoli Monitoring you can also use the `itm_custom_override.rules` file to customize the mapping of EIF event attributes to OMNIbus attributes. If you want to map an EIF event attribute to a new OMNIbus attribute, you must add the attribute to the ObjectServer alerts.status table before updating the `itm_custom_override.rules` file.

Note: Neither EIF Slot Customization or custom rules should change the values of the attributes that are used by the IBM Tivoli Monitoring triggers in the Netcool/OMNIbus ObjectServer. For details on attributes that should not be modified, see “Default mapping of situation events to OMNIbus events” on page 766.

Table 146. Common tasks for understanding and customizing an the contents of an event

Task	Architecture Type	Administrator
See “Default mapping of situation events to OMNIbus events” on page 766 and “Generic mapping for agent-specific slots” on page 770 for details on the default mappings of event attributes to EIF slots and OMNIbus attributes.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus
See the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* or the Tivoli Enterprise Portal online help for details on using the EIF slot customization function to customize how situation event attributes are mapped to EIF attribute slots.	Unidirectional and bidirectional	IBM Tivoli Monitoring
“Adding event classes for new and updated agents to the MCS Attribute service used by the EIF Slot Customization function” on page 772.	Unidirectional and bidirectional	IBM Tivoli Monitoring
“About customizing the format of the event message attribute” on page 774.	Unidirectional and bidirectional	IBM Tivoli Monitoring
“Localizing the event message attribute” on page 775.	Unidirectional and bidirectional	IBM Tivoli Monitoring
See “Customizing the rules file” on page 742 for details on how to customize the mapping of IBM Tivoli Monitoring EIF event attributes to Netcool/OMNIbus ObjectServer attributes.	Unidirectional and bidirectional	IBM Tivoli Monitoring and Netcool/OMNIbus

Default mapping of situation events to OMNIbus events

You can use the mapping information in this topic when you forward a situation event to the Tivoli Netcool/OMNIbus ObjectServer and want to write probe rules or SQL procedures and triggers in the ObjectServer.

The situation event forwarder generates an event integration facility (EIF) event with an event class based on the attribute group used in the situation. When the situation event is forwarded to the EIF Probe, the Tivoli Netcool/OMNIbus EIF probe translates the EIF event into an OMNIbus alert format. The EIF event contains all of the base attributes described by the *Omegamon_Base* class and its parent class EVENT. The EIF event also contains the agent specific attributes from the attribute group referenced by the situation definition when the event is opened. The agent specific attributes are also called extended slots in the Tivoli Enterprise Portal EIF Slot Customization function. For more information on agent-specific slots included in events, see "Generic mapping for agent-specific slots" on page 770.

Note: Agent-specific attributes are not included in any event status updates (including the event status update message that is sent when the event is closed in IBM Tivoli Monitoring).

Omegamon_Base is described as follows:

```
Omegamon_Base ISA EVENT
DEFINES {
cms_hostname: STRING;
cms_port: STRING;
integration_type: STRING;
master_reset_flag: STRING;
appl_label:STRING;
situation_name: STRING;
situation_origin: STRING;
situation_displayitem: STRING;
situation_time: STRING;
situation_status: STRING;
situation_eventdata: STRING;
situation_type: STRING;
situation_thrunode: STRING;
situation_group: STRING;
situation_fullname: STRING; }; END;
```

As part of the generic mapping for these situations, the IBM Tivoli Monitoring event forwarder assigns associated values for each of the *Omegamon_Base* event class attributes when forwarding an event to the EIF Probe. In addition to these event class attributes, values are assigned to the host name, origin, severity, and message attributes that are inherited from the EVENT class as shown in Table 147.

Table 147. Base IBM Tivoli Monitoring EIF attribute slots.

EIF slot attributes	Values and meaning
appl_label	Indicates if the event was sent from a monitoring agent. For events sent from a monitoring agent, this attribute is set to A:P:S if the event is a private situation event or is set to A:P:L if the event is a lifecycle event. For events sent from a hub monitoring server, a value is not provided for this slot.
cms_hostname	TCP/IP host name of the hub Tivoli Enterprise Monitoring Server that forwards the event.
cms_port	The hub monitoring server port on which the SOAP web service is listening.
ClassName	ITM_ + the name of the attribute group that the situation definition is based on, for example: ITM_NT_Process.
fqhostname	The fully qualified host name, if available. of the managed system where the event originated. [1]

Table 147. Base IBM Tivoli Monitoring EIF attribute slots. (continued)

EIF slot attributes	Values and meaning
hostname	The TCP/IP host name of the managed system where the event originated, if available. [1]
integration_type	Indicator to help OMNIbus performance. • N for a new event, the first time the event is raised • U for update event, subsequent event status changes The integration_type value is used solely by the OMNIbus synchronization rule to improve its performance. It has no other meaning related with the event.
master_reset_flag	Master reset indicator set for master reset events that indicate a hub monitoring server has been started. Value is NULL for all other events: • R for monitoring server recycle master_reset • S for hotstandby master_reset
msg	The situation name and formula when a situation event is opened and the situation name for all other event messages, for example acknowledgement, close, and so on.
origin	The TCP/IP address of the managed system where the event originates, if available. The address is in dotted decimal format. [1]
severity	The EIF severity of the event.
situation_displayitem	Display item of associated situation, if available.
situation_eventdata	Event data attributes in key-value pair format if multiple rows of the attribute group match the situation condition, if the situation does not have a display item, or if multiple rows have the same display item value. The data from the first row that matches the situation condition is not included in this EIF slot. The event data can be truncated because the event integration facility imposes a 2 KB size limit.
situation_group	One or more situation group names (up to 5) that the situation is a member of.
situation_fullname	Display name of the associated situation if it is different from the situation identifier in the situation_name slot.
situation_name	Unique identifier given to the situation.
situation_origin	Managed system name where the situation event originated. It has the same value as sub_source.

Table 147. Base IBM Tivoli Monitoring EIF attribute slots. (continued)

EIF slot attributes	Values and meaning
situation_status	Status of the situation event. The following values are valid: • Y if the situation condition is true and event has been opened in the hub monitoring server. • N if the situation condition has become false and the event is closed in the hub monitoring server. • A if the event has been acknowledged. • E if the event has been resurfaced (deacknowledged). • F if the acknowledgment timeout for the event has expired. • P if the situation has been stopped. • D if the situation has been deleted. • X the situation includes an error.
situation_time	Timestamp of the situation event.
situation_type	Indicator of whether the IBM Tivoli Monitoring situation that caused the event is a sampled or pure situation.
situation_thrunode	The hub or remote Tivoli Enterprise Monitoring Server that the managed system is connected to.
source	Contains IBM Tivoli Monitoring.
sub_source	The origin managed system name for the associated situation.
Notes: 1. If there is an IBM Tivoli Monitoring firewall gateway between the managed system and its monitoring server, these slots contain the hostname and IP address of the firewall gateway instead of the managed system.	

The IBM Tivoli Monitoring rules for the Tivoli Netcool/OMNIbus EIF probe map the EIF attribute slots of the situation event into ObjectServer attributes, which are defined in the alerts.status table of the Netcool/OMNIbus ObjectServer. See Table 148 for details. Although the EIF event also contains the agent specific attribute group slots when the event is opened, these slots are not mapped to ObjectServer attributes by the probe rules.

Table 148. Mapping of EIF attribute slots to OMNIbus attributes

EIF slot attribute	OMNIbus attribute
situation_name + situation_origin + situation_displayitem + event_class	Identifier (for ITMProblem)
situation_name + situation_origin + situation_displayitem + event_class + ITMResolution	Identifier (for ITMResolution)
situation_name	AlertKey
situation_origin	ITMSitOrigin
situation_origin	Node
situation_origin	NodeAlias
source	Agent
default	Type (20) (for ITMProblem)
situation_status = “P” and integration_type = “U”	Type (21) (for ITMResolution)
situation_status = “D” and integration_type = “U”	Type (21) (for ITMResolution)

Table 148. Mapping of EIF attribute slots to OMNIbus attributes (continued)

EIF slot attribute	OMNIbus attribute
situation_status = “N” and integration_type = “U”	Type (21) (for ITMResolution)
situation_displayitem	ITMDisplayItem
situation_status	ITMStatus
situation_thrunode	ITMThruNode
situation_time	ITMTime
situation_type	ITMSitType
situation_eventdata	ITMEventData
cms_hostname	ITMHostname
master_reset_flag	ITMResetFlag
integration_type	ITMIntType
ClassName	AlertGroup
msg	Summary
“tivoli_eif probe on ”+hostname()	Manager
6601 if integrating with Tivoli Business Service Manager 87722 if the event was sent by the hub monitoring server 87723 if the event was sent from an IBM Tivoli Monitoring agent	Class

severity
`FATAL / 60 = Critical`
`CRITICAL / 50 = Critical`
`MINOR / 40 = Minor`
`WARNING / 30 = Warning`
`UNKNOWN / 10 = Indeterminate`
	Severity

getdate	LastOccurrence/FirstOccurrence
date	TECDate
repeat_count	TECRepeatCount
fqhostname	TECFQHostname
hostname	TECHostname
cms_port	ITMPort
situation_fullname	ITMSitFullName
situation_group	ITMSitGroup
appl_label	ITMApplLabel

Note: If you use the virtualization or predictive rules files provided with Netcool/OMNIbus, those rules might set some of the OMNIbus attributes differently than the settings in the table above, for example, Node, NodeAlias, Class, and Summary.

You should not customize the EIF probe rules or use the EIF Slot Customization window of the Situation Editor to change any of the situation EIF attribute slots that are used by the IBM Tivoli Monitoring triggers in the Netcool/OMNIbus ObjectServer to process IBM Tivoli Monitoring events. The EIF slots and OMNIbus attributes that should not be customized are shown in table Table 149 on page 770.

Table 149. EIF attribute slots to OMNIbus attributes, not to be customized

EIF Slot	OMNIbus attribute
appl_label	ITMApplLabel
cms_hostname	ITMHostname
cms_port	ITMPort
integration_type	ITMIntType
master_reset_flag	ITMResetFlag
situation_displayitem	ITMDisplayItem
situation_eventdata	ITMEventData
situation_name	AlertKey
situation_origin	ITMSitOrigin
situation_status	ITMStatus
situation_thrunode	ITMThruNode

Note: The situation_origin EIF slot is also mapped to the Node and NodeAlias OMNIbus attributes. However, since the IBM Tivoli Monitoring triggers do not use the Node and NodeAlias attributes, you can customize their values in the `itm_custom_override.rules` file. However, if you are using the virtualization or predictive rules files provided with Netcool/OMNIbus, verify that your customizations of Node and NodeAlias are consistent with the values set by these rules.

Generic mapping for agent-specific slots

Generic mapping is used for events for which there is no event mapping file and is based on the target event class that is identified by the situation definition.

For situation events that do not have a custom event mapping specified for the forwarded event, an event is generated with a unique ClassName value based on the attribute group used in the situation. The ClassName attribute of the EIF event is set to a combination of **ITM_** plus the attribute group name associated with the situation. For example, a situation using the NT_Process attribute group generates an EIF event with ClassName set to *ITM_NT_Process*. The EIF event contains the base attribute slots described in Table 147 on page 766.

Additional agent-specific event slot values are populated with attribute values from the situation event data. The EIF slot names are the attribute names from the attribute group. For example, a situation using the Process_CPU attribute causes generation of a slot called process_cpu in the EIF event forwarded to the OMNIbus EIF probe. If an agent-specific attribute name conflicts with a slot name in the Tivoli Enterprise Console EVENT class or Omegamon_Base class, then the agent product code associated with the attribute group, for example: knt_, is prepended to the attribute name to form a unique slot name.

By default, the agent specific EIF event slots are not mapped to OMNIbus attributes. However, you can perform custom mapping of these EIF slots to OMNIbus attributes in the `itm_custom_override.rules` file. You can either map the agent specific EIF slots to name-value pairs in the ExtendedAttr OMNIbus attribute, or map them to a new OMNIbus attribute. Also some agents, for example, ITCAM for Transactions agents, might provide their own rules and `.sql` file for mapping agent specific attributes to new or existing OMNIbus attributes. (If you create a new OMNIbus attribute, you must add the attribute to the ObjectServer alerts.status table before updating the rules file.) However, it is important to note that the agent specific slot values are only included in the EIF event that is sent when the event is opened and not in any subsequent status update events, for example when the event is acknowledged or closed. If you are using bidirectional event synchronization between the monitoring server and Netcool/OMNIbus and you map agent specific slots to a new OMNIbus attribute, you must also update the IBM Tivoli Monitoring table and triggers that are used to cache sampled events that are cleared or deleted in the OMNIbus UI. The event data is cached so that it can be used to fully populate a new event if the event is re-opened when the acknowledge timeout expires in the monitoring server. See Table 129 on page 678 for more information about the behavior of events originating from a hub monitoring server.

Perform the following procedure to customize the list of OMNIbus attributes that are cached when a sampled event is cleared or deleted after the new attribute has been added to the ObjectServer alerts.status table:

1. Open the `itm_event_cache.sql` file using a text editor. This file is available in the `event_sync_install_dir/omnibus` where `event_sync_install_dir` is the directory where the IBM Tivoli Monitoring event synchronization component is installed.
2. Find "`--NewAttributeGoesHere type(size)`". Copy the line, insert a new line immediately below, and paste the copied line. In the newly added line delete the characters "`--`", replace "`NewAttributeGoesHere type(size)`" with the new OMNIbus attribute name, type, and size.
3. Find "`--statusr.NewAttributeGoesHere,`". Copy the line, insert a new line immediately below, and paste the copied line. In the newly added line delete the characters "`--`", replace "`NewAttributeGoesHere`" with the appropriate OMNIbus attribute name. Do not delete the comma (,) at the end of the line.
4. Find "`--set new.NewAttributeGoesHere = cachedevent.NewAttributeGoesHere ;`". Copy the line, insert a new line immediately below, and paste the copied line. In the newly added line delete the characters "`--`", replace "`NewAttributeGoesHere`" with the appropriate OMNIbus attribute name. Do not delete the semicolon (;) at the end of the line.
5. Execute the `itm_event_cache.sql` file to update the OMNIbus database schema by using the instructions in “Updating the OMNIbus database schema on single-tier or aggregation tier ObjectServers” on page 733.

For complex situations, the situation definition can involve more than one attribute group. In this case, the EIF event class used is derived from the first attribute group encountered in the situation event data of the triggering situation. For example, if a situation is written for the NT_Process and Local Time attribute groups, NT_Process being the first attribute group, the EIF event class ITM_NT_Process is used. Additional event slots are generated based on the attributes of the first attribute group only.

Table 150. Special characters for attribute groups and names in EIF events generated from forwarded situation events.

Character:	**Converts to:**
<uppercase> (applies only to attribute name)	<lowercase> (applies only to attribute name)
% percent sign	pct_
I/O	io
/ forward slash	_per_
\ backward slash	_ (underscore)
<space>	_ (underscore)
(open parenthesis) close parenthesis	_ (underscore)
< open pointed bracket > close pointed bracket	_ (underscore)

All strings and time stamp types are mapped to STRING types, and all integer types are mapped to INTEGER in the event class definition. No default values are assigned to the attribute slots. Attributes that have a specified non-zero scale/precision value are mapped to the string type of REAL.

Note: If you are mapping from an attribute to a slot and the resulting slot name has a trailing underscore, the trailing underscore is removed in the final slot name. Final slot names never have a trailing underscore.

Adding event classes for new and updated agents to the MCS Attribute service used by the EIF Slot Customization function

The EIF Slot Customization facility uses the MCS Attribute Service to present a list of predefined event classes in the Event class name list of the EIF Slot Customization window, which is available through the EIF tab of the Tivoli Enterprise Portal Situation editor. Only the event classes belonging to the OS agents are predefined and they are in an MCS Attribute Service jar file. When a new type of agent is added into the Tivoli Management Services infrastructure or a new event class is added to an updated version of an agent, you must generate a new MCS XML file and point the Tivoli Enterprise Portal Server to the new XML file before the new event classes will appear in the Event class name list.

See the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* or the Tivoli Enterprise Portal online help for details on using the EIF slot customization function.

To generate a new MCS XML file, install the Tivoli Enterprise Console Event Definition Generator (TEDGEN) tool supplied on the IBM Tivoli Monitoring Tools DVD. The TEDGEN tool should be installed on a distributed computer where the Hub monitoring server or portal server is installed. The TECLIB directory of the Hub monitoring server or portal server contains the agent BAROC files that are processed by the TEDGEN tool.

Note: The definitions in MCS XML file supersede those defined in the shipped MCS Attribute Services jar file (they are not merged). To obtain a MCS XML file that contains both the event classes definitions of the OS agents as well as new or updated agents, be sure all the BAROC definitions for the OS agents and other agents used in your environment are all in the TECLIB directory of your portal server or monitoring server before running the TEDGEN utility to generate the MCS XML file.

Installation and Configuration

1. Install the TEDGEN tool from the IBM Tivoli Monitoring Tools DVD on either the Hub monitoring server or portal server. The tool is located in the `tec/tedgen` directory on the Tools DVD and the installation and configuration instructions are in the `README.txt` file in the same directory.
2. If you installed the TEDGEN tool on a portal server on Linux or UNIX, you must also perform the following configuration steps:
   a. Create the `Install_dir/tables/cicatrsq/TECLIB` directory if it does not already exist, where `Install_dir` is the directory where IBM Tivoli Monitoring is installed.
   b. Copy the `om_tec.baroc` and `kib.baroc` files from the `Install_dir/arch/cq/TECLIB` directory to the `Install_dir/tables/cicatrsq/TECLIB` directory where `arch` is the architecture directory for the portal server.

Procedure

These steps assume that you have installed the TEDGEN utility on the computer where the Hub Tivoli Enterprise Monitoring Server or the Tivoli Enterprise Portal Server are installed and you have installed the latest application support for the agents whose event slots will be customized.

Notes:

1. If you are generating the MCS XML file on the portal server on Linux or UNIX, the BAROC files are not present by default and you must install them by running the `install.sh` script on the portal server computer and selecting the **Install TEMS support for remote seeding** option. See “Installing application support files on a computer with no monitoring server” on page 289 for more details. This action places the BAROC files on the portal server under the `Install_dir/tables/cicatrsq/TECLIB` directory.

2. If you installed the TEDGEN utility with a portal server on Windows, verify that the portal server's TECLIB directory contains the `.baroc` files for the agents whose events you want to customize. Not all agents include their `.baroc` files in their Tivoli Enterprise Portal Server application support. If an agent's `.baroc` file is not present, you can copy it from the Hub monitoring server's TECLIB directory.

Perform these steps to run the TEDGEN utility:

1. If you are running the TEDGEN utility on Windows, issue the following command:

   ```
   set CANDLE_HOME=Install_dir
   ```

   where `Install_dir` is the directory where IBM Tivoli Monitoring is installed.
2. If you are running the TEDGEN utility on UNIX, issue the following command:

   ```
   export CANDLEHOME=Install_dir
   ```

   where `Install_dir` is the directory where IBM Tivoli Monitoring is installed.
3. Change to the `TEDGEN_Install_dir/scripts` directory where `TEDGEN_Install_dir` is the directory where the TEDGEN utility is installed.
4. Run the TEDGEN tool to create a new MCS XML file:
   - On Windows systems:

     ```
     tedgen -itmDir Install_dir\{CMS|CNPS}
     \TECLIB -id server_id -xmlPath output_xml_file_path
     ```
   - On Linux/UNIX systems:

     ```
     tedgen -itmDir
     Install_dir/tables/{tems_name|cicatrsq}/TECLIB
     -id server_id -xmlPath output_xml_file_path
     ```

   Where:

   Install_dir

   Is the directory where the monitoring server or portal server is installed.

   server_id

   Is any string. The EIF slot customization function does not use the value of this string.

   output_xml_file_path

   Is the name of the MCS XML file that should be created by the tool. Absolute and relative paths are supported.

   Example: In the following example, the hub monitoring server named `mytems` has the BAROC files in the `TECLIB` directory. The output file goes to the same directory and is named `tems.xml`.

   ```
   tedgen -itmDir C:\IBM\ITM\CMS\TECLIB -id mytems -xmlPath tems.xml
   ```
5. If the TEDGEN tool was run on the Hub monitoring server, copy the newly generated MCS XML file to the computer where the Tivoli Enterprise Portal Server is installed.
6. Edit the portal server environment file to specify the path to the XML file:

   a. On Windows systems: In the Manage Tivoli Monitoring Services window, right-click **Tivoli Enterprise Portal Server** and click **Advanced → Edit ENV File** to open the `kfwenv` file in the text editor.

   b. On Linux/UNIX systems: Open `Install_dir/config/cq.ini` in a text editor.

   c. Locate the `KFW_MCS_XML_FILES` environment variable and type = (equal sign) followed by the path to the MCS XML file.

   d. Save and close the environment file.

   e. On Windows systems, restart the portal server. On Linux/UNIX systems, reconfigure the portal server and then start it.

About customizing the format of the event message attribute

The OMNIbus EIF probe maps the msg attribute slot in the EIF event sent from the Tivoli Enterprise Monitoring Server into the Summary attribute of the ObjectServer. The Summary attribute gives you a descriptive way of looking at an alert in OMNIbus.

The situation name alone does not provide detailed event identification where you have large numbers of like-events from various sources. Therefore, the msg EIF slot is expanded to include the following event attributes when an event is opened:

```
Situation-Identifier [(formula) ON Managed-System-Name ON DISPLAY-ITEM
(threshold Name-Value pairs)]
```

where:

Situation-Name
The unique identifier of the situation.

formula
The formula tells how the situation is evaluated.

Managed-System-Name
The agent or the managed system.

DISPLAY-ITEM
The identifier that triggered the situation if there is more than one instance. This is optional and is used only if a display item is specified in the situation definition.

threshold Name-Value pairs
The raw data that the situation uses to evaluate whether it is triggered.

Examples:

```
NT_Criticial_Process [(Process_CPU > 4 AND Thread_Count > 50)
ON IBM-AGX02:NT
(Process_CPU = 8 AND Thread_Count = 56)]

NT_Disk_Full [(Free_Megabytes < 1000000)
ON "IBM-AGX02:NT"
ON D: (Free_Megabytes = 100)]
```

You can use the EIF Slot Customization window of the Tivoli Enterprise Portal Situation Editor to override the default msg slot setting.

When customizing the msg attribute slot, the Literal value column can be used to define a custom message template. The message template can consist of fixed message text and variable substitution references, or symbols. The symbol can be base or extended slot data, or a special reference to the situation formula. Base slots are those that are included in all forwarded events, such as situation_name. Extended slots are those specific to the attribute group used by the situation definition. See the following syntax:

- For an extended slot, use the fully qualified attribute name **($Attribute_Table.Attribute_Name$)**.
- For a base slot, use the variable name that is not fully qualified (no . periods) unless it is the situation formula symbol.
- For a situation formula, use **$formula$**.

These characters are not supported: less than, greater than, quotation mark, single quotation mark, and ampersand. The Literal value column cannot be used to define a message template if a value is selected in the Mapped attribute column.

See the Tivoli Enterprise Portal online help or the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* for more information about using the EIF slot customization function.

Localizing the event message attribute

Edit the KMS_OMTEC_GLOBALIZATION_LOC variable on the hub monitoring server to enable globalization of the EIF event msg attribute slot that is mapped to the OMNIbus Summary attribute by the OMNIbus EIF probe rules.

By default, this variable is set to United States English and the msg attribute slot contains United States English messages. Take these steps to edit the variable to enable any language packs that are installed in your hub monitoring server environment:

1. On the computer where the hub Tivoli Enterprise Monitoring Server is installed, open the KBBENV file:
   - Start Manage Tivoli Monitoring Services, right-click **Tivoli Enterprise Monitoring Server** and click **Advanced → Edit ENV** file.
   - In a text editor, open the `<install_dir>/config/ <tems_name>_ms_<address>.cfg` file, where `<install_dir>` is the directory where IBM Tivoli Monitoring is installed, `<tems_name>` is the value of the hub monitoring server supplied during the monitoring server configuration, and `<address>` is the IP address or fully qualified name of the hub monitoring server computer.
2. Locate (or add) the KMS_OMTEC_GLOBALIZATION_LOC environment variable and enter the desired language and country code in the format KMS_OMTEC_GLOBALIZATION_LOC=xx_XX, where
   - xx is the language.
   - XX is the country code: de_DE, en_US, en_GB, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, zh_CN, or zh_TW.

   For example, KMS_OMTEC_GLOBALIZATION_LOC=pt_BR (for Brazilian Portuguese) or KMS_OMTEC_GLOBALIZATION_LOC=zh_CN (for Simplified Chinese).
3. Save and close the monitoring server environment file.
4. Restart the hub monitoring server.

Appendix A. Installation worksheets

Use the following worksheets to gather information you need during the installation of the IBM Tivoli Monitoring components:

Windows hub monitoring server worksheet

Use the following worksheet to gather information for your installation of the hub monitoring server on a Windows computer.

Table 151. Windows hub monitoring server installation worksheet

Host name of computer	
IP address	
IBM Tivoli Monitoring installation directory	
Encryption key	Must use the same encryption key on all IBM Tivoli Monitoring components. Do not specify the following characters: **&** ampersand **|** pipe **'** single quote **=** equal sign **$** dollar sign In addition, do not specify double-byte (DBCS) characters.
Agents to install on this computer	
Agents to add to the deployment depot (plus non-agent bundles, if your site uses them)	
Program folder name	
Monitoring server name	
Communications protocol details	See Monitoring server communications protocol details worksheet.
Agents for which to add application support data	

Linux or UNIX hub monitoring server installation worksheet

Use the following worksheet to gather information for the installation of the hub monitoring server on a Linux or UNIX computer.

Table 152. Linux or UNIX hub monitoring server installation worksheet

Host name of computer	
IP address	
IBM Tivoli Monitoring installation directory	
Encryption key	Must use the same encryption key on all IBM Tivoli Monitoring components. Do not specify the following characters: **&** ampersand **|** pipe **'** single quote **=** equal sign **$** dollar sign In addition, do not specify double-byte (DBCS) characters.
Monitoring server name	
Communications protocol details	See Monitoring server communications protocol details worksheet.
KDC_PARTITION	
NIC interface name ("Optional Primary Network Name")	
Agents to install on this computer	
Agents for which to add application support data	
Agents to add to the deployment depot (plus non-agent bundles, if your site uses them)	

Windows remote monitoring server worksheet

Use the following worksheet to gather information for the installation of the remote monitoring server on a Windows computer.

Table 153. Windows remote monitoring server installation worksheet

Host name of computer	
IP address	
IBM Tivoli Monitoring installation directory	
Encryption key	Must use the same encryption key on all IBM Tivoli Monitoring components. Do not specify the following characters: **&** ampersand **|** pipe **'** single quote **=** equal sign **$** dollar sign In addition, do not specify double-byte (DBCS) characters.
Agents to install on this computer	
Agents to add to the deployment depot (plus non-agent bundles, if your site uses them)	
Program folder name	
Monitoring server name	
Agents for which to add application support data	
Hub monitoring server name	
Hub monitoring server host name	
Hub monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.

Linux or UNIX remote monitoring server installation worksheet

Use the following worksheet to gather information for the installation of the remote monitoring server on a Linux or UNIX computer.

Table 154. Linux or UNIX remote monitoring server installation worksheet

Host name of computer	
IP address	
IBM Tivoli Monitoring installation directory	
Encryption key	Must use the same encryption key on all IBM Tivoli Monitoring components. Do not specify the following characters: **&** ampersand **|** pipe ' single quote **=** equal sign **$** dollar sign In addition, do not specify double-byte (DBCS) characters.
Monitoring server name	
KDC_PARTITION	
NIC interface name ("Optional Primary Network Name")	
Agents for which to add application support data	
Hub monitoring server host name	
Hub monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.
Agents to add to the deployment depot (plus non-agent bundles, if your site uses them)	

Windows portal server worksheet

Use the following worksheet to gather information for the installation of the portal server on a Windows computer.

Table 155. Windows portal server worksheet

Installation location	
Encryption key used on the hub monitoring server	Must specify the same encryption key on all IBM Tivoli Monitoring components.
Program folder	
Host name of the computer where you are installing the portal server	
Portal server database administrator ID	
Portal server database administrator password	
Portal server database user ID (default = TEPS)	
Portal server database user password	
Warehouse database administrator ID	
Warehouse database administrator password	
Warehouse database user ID (default = ITMUser)	
Warehouse database user password	
Warehouse data source name (default = ITM Warehouse)	
Warehouse database name	
Hub monitoring server host name	
Hub monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.

Linux portal server worksheet

Use the following worksheet to gather information for the installation of the portal server on a Linux computer.

Table 156. Linux portal server worksheet

Installation location	
Encryption key for the hub monitoring server	Must specify the same encryption key on all IBM Tivoli Monitoring components.
Host name for the hub monitoring server	
Hub monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.
NIC interface name (Primary Optional Network Name)	
DB2 for Linux, UNIX, and Windows instance name (default = db2inst1)	
DB2 for Linux, UNIX, and Windows administrator ID (default = db2inst1)	
DB2 for Linux, UNIX, and Windows administrator password	
Portal server database name (default = TEPS)	
Portal server database user (default = itmuser)	
Portal server database user password	
Warehouse database name (default = WAREHOUS)	
Warehouse database user (default = itmuser)	
Warehouse database user password	

Generic Windows monitoring agent worksheet

Use the following worksheet to gather information for the installation of a monitoring agent on a Windows computer. Depending on the agent you are installing, you might need additional information to configure the agent. See the agent user's guide for more information.

Table 157. Generic Windows monitoring agent worksheet

Installation directory	
Encryption key for the hub monitoring server	Must specify the same encryption key on all IBM Tivoli Monitoring components.
Agents to install	
Program folder name	
Monitoring server host name	
Monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.

Generic Linux or UNIX monitoring agent worksheet

Use the following worksheet to gather information for the installation of a monitoring agent on a Linux or UNIX computer. Depending on the agent you are installing, you might need additional information to configure the agent. See the agent user's guide for more information.

Table 158. Generic monitoring agent for a Linux or UNIX computer worksheet

Installation directory	
Encryption key for the hub monitoring server	Must specify the same encryption key on all IBM Tivoli Monitoring components.
Agents to install	
Agent product code or codes	
Monitoring server host name	
Monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.
KDC_PARTITION	
NIC interface name (Optional Primary Network Name)	
root user password	
User group name	
Optional user name	

Windows portal desktop client worksheet

Use the following worksheet to gather information for the installation of a the portal desktop client on a Windows computer.

Table 159. Windows portal desktop client worksheet

Installation directory	
Encryption key for the monitoring server	Must specify the same encryption key on all IBM Tivoli Monitoring components.
Program folder name	
Portal server host name	
Monitoring server host name	
Monitoring server communications protocol details	See Monitoring server communications protocol details worksheet.

Linux portal desktop client worksheet

Use the following worksheet to gather information for the installation of a the portal desktop client on a Windows computer.

Table 160. Linux portal desktop client worksheet

Installation directory	
Encryption key for the monitoring server	Must specify the same encryption key on all IBM Tivoli Monitoring components.
Instance name for the portal server	
Portal server host name	

Monitoring server communications protocol details worksheet

Use the following worksheet to gather the communications protocol details for your hub and remote monitoring servers.

Table 161. Monitoring server communications protocol details worksheet

IP.UDP Settings	
Host name or IP address	
Port number or port pools	
IP.PIPE Settings	
Host name or IP address	
Port number	
IP.SPIPE Settings	
Host name or IP address	
Port number	
SNA Settings	
Network name	
LU name	
LU 6.2 LOGMODE	
TP name	
Local LU alias	

Appendix B. Performing a silent installation of IBM Tivoli Monitoring

This appendix provides information about installing IBM Tivoli Monitoring using the silent installation method. This method of installation is useful for advanced users who supply installation information once through a response file, instead of repeatedly through an installation wizard.

You might run through the installation wizard one time to determine the values that you need to set for your monitoring needs and then use silent installation to install the rest of your environment. For more information about installing through the installation wizard, see Chapter 9, “Installing IBM Tivoli Monitoring,” on page 207.

The following table outlines the steps for performing a silent installation of IBM Tivoli Monitoring.

Table 162. Installation and configuration steps

Step	Where to find detailed information
Assess your monitoring needs to determine the best deployment of IBM Tivoli Monitoring components.	Chapter 6, “Preparing for installation,” on page 125
Ensure you have the required hardware and software.	“Hardware and software requirements” on page 138
Gather any information required for successful installation (such as DB2 user information and security specifications).	“Specific information to have ready” on page 125
Run the silent installation.	“Creating and using a Windows response file” “Performing a silent installation on a Linux or UNIX computer” on page 793
Install application support files on the monitoring server, portal server, and portal desktop client.	“Installing and enabling application support” on page 266
Start the portal client to verify that you can view the monitoring data.	“Starting the Tivoli Enterprise Portal client” on page 320

For information on performing a silent uninstallation, see “Uninstalling components and agents silently” on page 859.

Creating and using a Windows response file

A sample Windows silent installation response file is provided on the product installation media. Use the following steps to edit that response file as appropriate for your environment:

1. Locate on the product installation media the appropriate silent_*.txt file, as dictated by the Tivoli Management Services component for which you're building a response file.

   silent_server.txt
   for a server image

   silent_agent.txt
   for an agent image

   silent_WIA64.txt
   for an agent image for 64-bit Windows Itanium

   For example, when installing an agent silently you must use the appropriate response file located in `ITM_build\Windows\Deploy`.
2. Copy this file to a temporary directory on your system.
3. Open your copy of the silent_*.txt file in a text editor.

4. Change the parameters as appropriate for your environment. The silent_*.txt file contains descriptions of all the parameters, including directions on how to use them.

   Complete all of the steps listed in the file. Each line of the file must be either a comment (containing a semi-colon in column one) or a meaningful statement that starts in column one.

   Note: If you want to use the TCP/IP protocol, make sure to specify "IP.UDP." If you specify "TCP/IP," IP.PIPE is used by default.

   Attention: Do not modify any other files supplied with the installation (for example, the SETUP.ISS file).
5. Save the file and close the editor.
6. Run the silent installation using one of the following methods:
   - "Running the silent installation from the command-line with parameters" on page 792
   - "Running the silent installation using SMS" on page 792

Notes:

1. A **silentInstall.cmd** script has been added to the Agents DVD. To install this agent you need to run this script:

   ```
   silentInstall.cmd
   ```

   To install the agent in a different directory than the default one (CANDLE_HOME), use the -h option:

   ```
   silentInstall.cmd -h directory
   ```

   If this directory name contains spaces, make sure you enclose the name in quotation marks:

   ```
   silentInstall.cmd -h "directory_with_spaces"
   ```

   To review the usage of the **silentInstall.cmd** file, enter `silentInstall.cmd -?`.
2. To silently install the tacmd command interface (component KUE) so you can continue running commands based on the previous CLI, uncomment this line in the FEATURES sections of your response file:

   ```
    KUEWICMA= Tivoli Enterprise Services User Interface
   ```
3. If the installation fails for any reason, a log file, "Abort IBM Tivoli Monitoring *date time*.log," is created to document the problem. If the installation fails before reading in the installation location, the log file is written to the Windows boot drive, typically the C:\ drive. If the installation fails after reading the installation location, the log file is written to an `\install` subdirectory in the installation directory. For example, if you use the default installation directory, the log file is written to the `C:\ibm\itm\installITM` directory.

Automatically creating agent response files on Windows

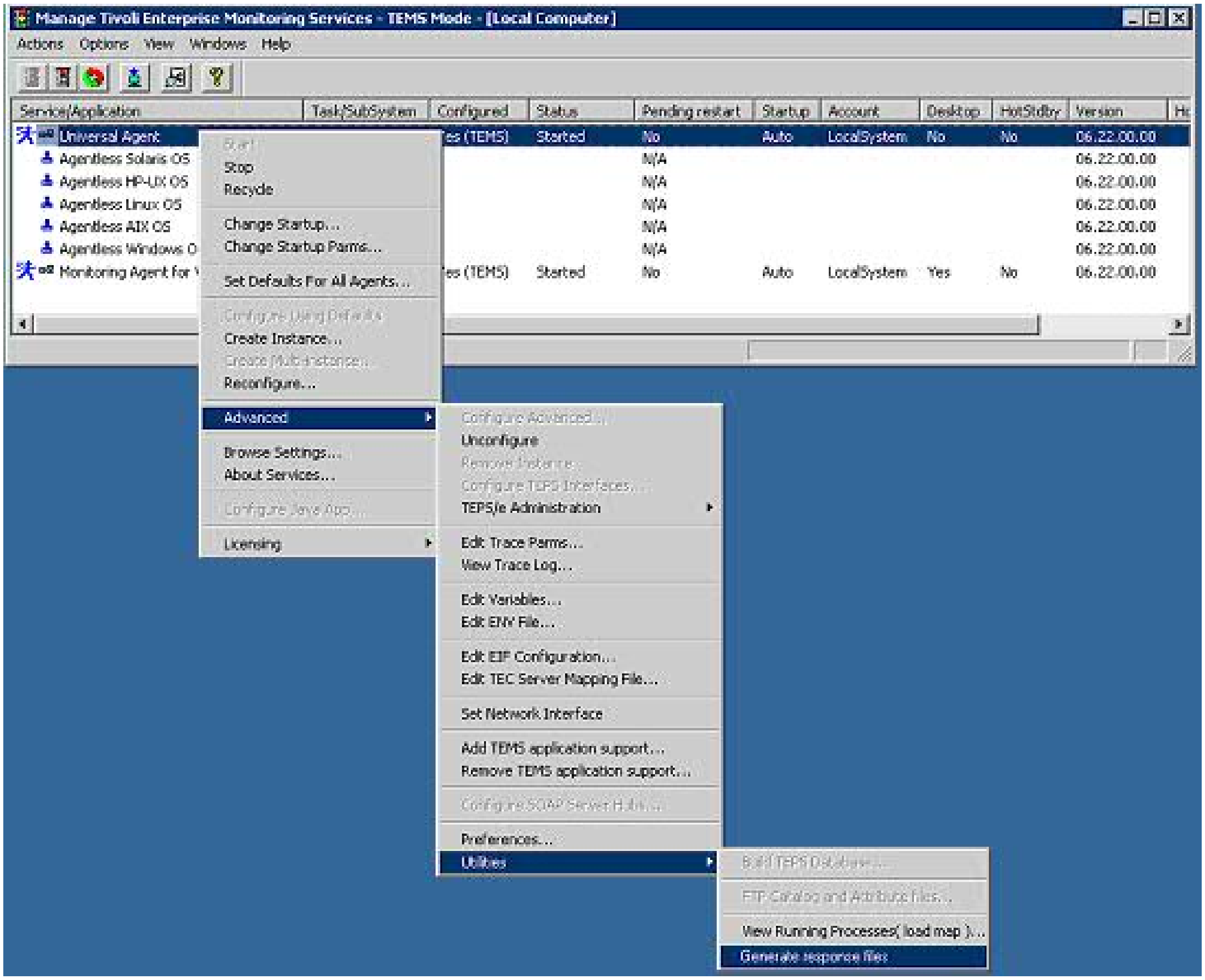
As of IBM Tivoli Monitoring V6.2.2, you can have Tivoli Monitoring create the response files for you after configuring an agent. This new feature vastly simplifies and speeds up the creation and customization of response files for agent installation and deployment. It also reduces the likelihood of user error when specifying their contents. The resulting response files can be used to either install or deploy similar agents across your environment.

Note: The automatic generation of response files does not apply to multi-instance agents or to server components.

The agent must be successfully installed and configured prior to generating the response file. To have Tivoli Monitoring generate a response file, follow these steps:

1. From the Manage Tivoli Enterprise Monitoring Services screen, right-click the agent whose configuration information you want saved, and select the **Advanced → Utilities → Generate Response**

Files option from the pop-up menu. See Figure 172.



Service/Application	Task/SubSystem	Configured	Status	Pending restart	Startup	Account	Desktop	HotStdby	Version	Hc
Universal Agent		es (TEMS)	Started	No	Auto	LocalSystem	No	No	06.22.00.00	
Agentless Solaris OS				N/A					06.22.00.00	
Agentless HP-UX OS				N/A					06.22.00.00	
Agentless Linux OS				N/A					06.22.00.00	
Agentless AIX OS				N/A					06.22.00.00	
Agentless Windows O				N/A					06.22.00.00	
Monitoring Agent for \		es (TEMS)	Started	No	Auto	LocalSystem	Yes	No	06.22.00.00	

Figure 172. The Generate Response Files option

Note: This option does not appear if the agent has not yet been configured.

2. A window opens where you can specify the path into which you want the generated response files written; the default directory is *ITM_Install_Home*\response.

   The newly generated response files are named `silent_install_`*pc*`.txt` and `silent_deploy_`*pc*`.txt`, where *pc* is the product code for the agent whose configuration parameters you want saved. See Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815 for the list of product codes. If the directory you specify already contains files with these names, you are prompted to either replace them or specify new names.

The installation and remote-deployment response files are created. When they are available, you can use them to either install or remotely deploy identical agents across your IBM Tivoli Monitoring network:

1. Edit the response file, and supply missing security parameters such as passwords and encryption keys.

   Note: For security reasons, the IBM Tivoli Monitoring encryption key entered during installation is not saved in the generated response file, and password parameters are blanked out. You must supply these values yourself.

2. Install the agent.

- If you install it locally on the target computer, perform a silent installation using the generated response file for installation: `silent_install_`*`pc`*`.txt`.
- If instead you remotely deploy the new agent, use the generated response file for remote deployment: `silent_deploy_`*`pc`*`.txt`.

When the silent installation completes successfully, the new agent is installed and configured with identical settings as the first one.

Note: Menu option **Generate Response Files** creates silent response files for the selected agent; such files contain all the installation parameters required to install the same agent with identical settings on other machines. However, the generated response files apply only to the Common Installer. They do not work with Solution Installer-based agent images.

Running the silent installation from the command-line with parameters

Use the following steps to run the installation from the command-line:

1. Start a DOS Command Shell.
2. From the shell, change to the directory containing this installation (where setup.exe and setup.ins are located).
3. Run the setup as follows. You must specify the parameters in the same order as listed.

   ```
   start /wait setup /z"/sfC:\temp\SILENT_*.TXT" /s /f2"C:\temp\silent_setup.log"
   ```

   where:

   `/z"/sf"`
   Specifies the name of the installation driver you customized for your site. This is a required parameter. This file must exist.

   `SILENT_*.TXT`
   is the name of the input silent_server.txt, silent_agent.txt, or silent_WIA64.txt file, as described above.

   `/s` Specifies that this is a silent installation. This causes nothing to be displayed during installation.

   `/f2`
   Specifies the name of the InstallShield log file. If you do not specify this parameter, the default action is to create Setup.log in the same location as the setup.iss file. In either case, the Setup program must be able to create and write to this file. If the specified directory does not exist, the Setup program cannot create and write to the file. Therefore, the specified directory path must exist.

Running the silent installation using SMS

Use the following steps:

1. Copy all the installation files to a LAN-based disk that SMS will mount on the desired computers. (Copy all files in the directory that contains the setup.exe and setup.ins files.)
2. Replace the original silent_server.txt, silent_agent.txt, or silent_WIA64.txt file on the LAN disk with your modified version.
3. Edit the PDF file located with the setup.exe file, and change the Setup invocation as follows:

   ```
    Setup /z"/sfC:\temp\SILENT_*.TXT" /s /f2"C:\temp\silent_setup.log"
   ```

   where:

   `SILENT_*.TXT`
   is the name of the input silent_server.txt, silent_agent.txt, or silent_WIA64.txt file, as described above.

Performing a silent installation on a Linux or UNIX computer

Just as the interactive installation on Linux and UNIX requires both an installation of code and then a separate configuration, so does the silent installation method. Both the installation and configuration use parameter files to define what you are installing or configuring. Sample installation and configuration parameter files are included with IBM Tivoli Monitoring and with monitoring agents. The files are located in the following locations:

- Silent installation files:
  - On the product installation media (both base IBM Tivoli Monitoring and agent installation media)
  - After installation, a sample file is located in the `*install_dir*/samples` directory
- Silent configuration files: After you install the product, a configuration file for each component that requires configuration is located in the `*install_dir*/samples` directory. There is also a sample configuration file that you can use to configure any component.

Before editing any of the response files, note the following syntax rules:

- Comment lines begin with a pound sign (#).
- Blank lines are ignored.
- Parameter lines are PARAMETER=value. Do not use a space before the parameter; you can use a space before or after an equal sign (=).
- Do not use any of the following characters in any parameter value:

  $ dollar sign

  = equal sign

  | pipe

Use the following procedures to perform silent installations:

- “Installing components with a response file”
- “Configuring components with a response file” on page 795

Installing components with a response file

The silent_install.txt response file specifies the installation parameters for IBM Tivoli Monitoring components. To use this file to perform a silent installation, edit the file to identify what you want to install and then run the following command:

```
./install.sh -q -h install_dir -p response_file [-k]
```

where:

install_dir

identifies the installation location for the IBM Tivoli Monitoring component. The default installation location is `/opt/IBM/ITM`.

Note: You must not specify the path of the directory containing `./install.sh` as your IBM Tivoli Monitoring home directory. On certain platforms, this can cause the plugin JAR files to overwrite themselves and become zero length files. The installation will fail as a result.

response_file

identifies the response file that you edited to specify installation parameters. Specify the absolute path to this file.

[-k]

is an optional parameter to secure your IBM Tivoli Monitoring installation. If you do not secure the installation at this point you will be asked at the end of this procedure if you want to do so.

[-c]

is an optional parameter to print diagnostic messages to the console.

`[-d]`
is an optional parameter to indicate the location of the product CD.

`[-h]`
is an optional parameter to indicate the location of the CandleHome.

`[-j]`
is an optional parameter to indicate the JRE installed location. This is available on Tandem system only.

`[-p]`
is an optional parameter for silent mode execution using the specified **PARAMETER_FILE**.

The parameters that you can configure in the silent_install.txt file vary depending on the component that you are installing. Each of the files contains comments that explain the options. The following procedure is an example of installing all available components on one computer. Use this to determine the type of information you need to gather when you are setting up your own silent installation.

Use the following steps to perform a silent installation on a UNIX computer:

1. Edit the `silent_install.txt` file located in `ITM_build\Windows\Deploy`.
2. Set the following parameters as appropriate for your environment:

Table 163. Silent installation parameters for UNIX

Parameter	Definition
INSTALL_ENCRYPTION_KEY	REQUIRED. The data encryption key used to encrypt data sent between systems. This key must be the same for all components in your IBM Tivoli Monitoring environment. Do not use the following characters in the key: **&** ampersand **|** pipe **'** single quote **=** equal sign **$** dollar sign In addition, do not specify double-byte (DBCS) characters.
INSTALL_FOR_PLATFORM	The operating system for which to install the components. You can specify an architecture code. If you do not specify an architecture code, the operating system for the current computer is used. You can find a list of the architecture codes for the supported architectures in **archdsc.tbl** in the registry directory; they are also listed in Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815.

Table 163. Silent installation parameters for UNIX (continued)

Parameter	Definition

INSTALL_PRODUCT	The product code for the components (or "products") that you want to install. See Appendix D, "IBM Tivoli product, platform, and component codes," on page 815 for a list of the codes for the base components. You can use the **./cinfo** command to view the product codes for the applications installed on this computer. You can also find a list of the product codes in the registry directory in `proddsc.tbl`.
	You can specify "all" to install all available components.
	To install multiple components (but not all), repeat this parameter for each component that you want to install. For example:
	`INSTALL_PRODUCT=ms`
	`INSTALL_PRODUCT=cj`
	`INSTALL_PRODUCT=cq`
	This example installs the monitoring server, portal server, and portal desktop client on a Linux computer.
	In the same pass, you can install support with the components. After you have specified the components that you want to install, specify the support that you want to install using the following parameters:
	`INSTALL_PRODUCT_TMS (for TEMS support)`
	`INSTALL_PRODUCT_TPS (for TEPS support)`
	`INSTALL_PRODUCT_TPW (for TEP Browser Client support)`
	`INSTALL_PRODUCT_TPD (for TEP Desktop Client support)`
	`INSTALL_PRODUCT_TPA (for ITPA Domain support)`

MS_CMS_NAME	If you are installing a monitoring server, use this parameter to specify the name for the monitoring server, such as HUB_*hostname*. Do not specify an IP address or fully qualified host name.
SKIP_SDA_CHECK	You can use this parameter to overwrite product support that was seeded in self-describing mode. By default, this parameter is set to **NO** and a self-describing mode seeding status check is performed. If this parameter is set to **YES**, the seeding process at the end of the installation does not check the self-describing mode seeding status.

3. Save and close the file.
4. Run the following command to install IBM Tivoli Monitoring in the /opt/IBM/ITM directory:

   ```
   ./install.sh -q -h /opt/ibm/itm -p /tmp/silent_install.txt
   ```

Configuring components with a response file

You can use the **itmcmd config** command with the **-p** *filename* parameter to configure IBM Tivoli Monitoring components silently.

The following sample configuration response files are provided with IBM Tivoli Monitoring:

- **ms_silent_config.txt**: Used to configure the monitoring server
- **cq_silent_config.txt**: Used to configure the portal server
- **cj_silent_config.txt**: Used to configure the portal desktop client

- **silent_config.txt**: A generic configuration file used to configure agents that do not require unique configuration parameters

Note that these sample configuration files are available only after the monitoring server has been installed in a UNIX or Linux system. The files are created in the *itm_installdir*/samples directory.

If an agent requires unique configuration parameters, configure the agent by using the **itmcmd config** command or the Manage Tivoli Enterprise Monitoring Services utility.

Use the following steps to configure a component using the silent method:

1. Edit the configuration file for the component that you want to configure.
2. Complete the parameters identified in the file. Each file contains comments that define the available parameters and the values to specify.
3. Save the file and exit.
4. Run one of the following commands.

   To configure the monitoring server:

   ```
   ./itmcmd config -S -p response_file -t ms_name
   ```

   To configure the portal server, desktop client, or an agent:

   ```
   ./itmcmd config -A -p response_file pc
   ```

   where:

   response_file
   Is the name of the configuration response file. Specify an absolute path to this file.

   ms_name
   Is the name of the monitoring server that you want to configure.

   pc Is the product code for the component or agent that you want to configure. See Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815 for the list of product codes.

Automatically creating agent response files on Linux or UNIX

As of IBM Tivoli Monitoring V6.2.2, you can have Tivoli Monitoring create the response files for you after configuring an agent. This new feature vastly simplifies and speeds up the creation and customization of response files for agent installation and deployment. It also reduces the likelihood of user error when specifying their contents. The resulting response files can be used to either install or deploy similar agents across your environment.

Note: The automatic generation of response files does not apply to multi-instance agents or to server components.

The agent must be successfully installed and configured prior to generating the response file. To have Tivoli Monitoring generate a response file, invoke the itmcmd CLI command with the resp option:

```
itmcmd resp [-d directory] pc
```

where:

directory
is the name of the directory where you want the generated files stored. The default directory is *itm_installdir*/response.

pc is the product code for the agent whose configuration parameters you want saved. See Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815 for the list of product codes.

Possible errors are that either the *directory*path or the product code, *pc*, is invalid. In either case, an error message is displayed, and the response files are not generated.

When it completes, the itmcmd resp command creates these installation and remote-deployment response files:

```
silent_install_pc.txt
silent_config_pc.txt
```

When response files are available, you can use them to either install or remotely deploy, and then configure, identical agents across your IBM Tivoli Monitoring network:

1. Edit the response files, and supply missing security parameters such as passwords and encryption keys.

   Note: For security reasons, the IBM Tivoli Monitoring encryption key entered during installation is not saved in the generated response file, and password parameters are blanked out. You must supply these values yourself.
2. Install the agent.
   - If you install it locally on the target computer, perform a silent installation using the generated response file for installation: `silent_install_pc.txt`.
   - If instead you remotely deploy the new agent, use the generated response file for remote deployment: `silent_deploy_pc.txt`.
3. Using the configuration response file (`silent_config_pc.txt`), configure the newly installed agent:

Appendix C. Firewalls

There are four options for achieving interoperability between Tivoli Management Services components across network firewalls:

1. Automatic (do nothing)
2. Configure IP.PIPE with the `ephemeral` keyword
3. Use a broker partition file
4. Implement a firewall gateway

Use the following topics to select and implement the appropriate option for your environment:

- “Determining which option to use”
- “Basic (automatic) implementation” on page 800
- “Implementation with ephemeral pipe” on page 800
- “Implementation with partition files” on page 802
- “Implementation with firewall gateway” on page 806

Determining which option to use

To determine which option is needed in a particular the firewall environment, four factors must be considered:

- “Flow of connection establishment”
- “Permission at the firewall”
- “Server address continuity” on page 800
- “Number of internet zones” on page 800

Flow of connection establishment

Products based on Tivoli Management Services typically establish connections in a traditional client-server flow: connect requests flow from a client in the public network to a server in the trusted network. If the network configuration allows this connection flow, permissions at the firewall are the next consideration in determining how interoperability is achieved.

If the configuration requires that the physical connect requests flow from the secure, trusted server network to the public, untrusted client network, then option 4, a firewall gatewall, is required for interoperability. See “Implementation with firewall gateway” on page 806.

Permission at the firewall

If the network configuration allows traditional connection flow, the next consideration is what firewall permissions, if any, are required of the firewall that separates the private, trusted server network from the public, untrusted client network. For simplicity the firewall between these disjoint networks is referred to as the *barrier firewall*.

If all ports are permitted across the barrier firewall, then server address continuity becomes a consideration (see “Server address continuity” on page 800).

If no ports are permitted at the barrier firewall, then to achieve interoperability among components in this firewall environment, full-duplex traffic must be permitted by the firewall administrator for as many ports as there are servers being concurrently accessed. For example, if Tivoli Enterprise Monitoring Agents are accessing only the Tivoli Enterprise Monitoring Server, then only one port must be permitted (for full-duplex traffic) or opened at the barrier firewall. This is the well-known monitoring server port (the default is 1918 for IP.PIPE, 3660 for IP.SPIPE). If agents are accessing two servers concurrently, a monitoring server and a Warehouse Proxy server, then two ports must be opened at the firewall, one for the monitoring server (typically 1918) and one for the Warehouse Proxy (typically 63358) for interoperability in this firewall environment.

Server address continuity

If the barrier firewall has no restrictions and all ports are permitted, the next factor to consider is server address continuity. Address continuity refers to the validity and reachability of published IP addresses. Address continuity exists when a published server address is universally reachable by all network clients requesting that service. An example of a server address with address continuity is update.microsoft.com (207.46.211.124).

Tivoli Management Services server components register their services and the location of these services (IP address) with a *location broker*. Clients send queries to the location broker to request address information for a service, and receive responses that a list of protocols (address families) and IP addresses at which these services are available. The client then sends a specific server request to one of the addresses in the list received from the location broker. Service registration with the location broker assumes address continuity.

If the published address of the Tivoli service (a remote monitoring server, for example) is identical and reachable for either side of the barrier firewall, then nothing further needs to be done to achieve interoperability in this firewall environment. If the same address cannot be reached from either side of the barrier firewall, then option 2 (ephemeral pipe configuration) or option 3 (broker partition files) is required for interoperability.

Both options are used when traversing a firewall with Network Address Translation (NAT) in effect. While option 2 (ephemeral pipe) is easier to implement, it restricts the endpoint: ephemeral endpoints cannot warehouse data. If warehousing data at the endpoint is required, then partition files must be used for interoperability in this firewall environment (see “Implementation with partition files” on page 802); otherwise ephemeral pipes are sufficient to traverse the translating firewall (see “Implementation with ephemeral pipe”).

Number of internet zones

The final factor to be taken into consideration is the number of internet zones, that is, how many barrier firewalls must be crossed. If there are two or more translating firewalls, then option 4, a firewall gateway, must be used for interoperability in this firewall environment (see “Implementation with firewall gateway” on page 806).

Basic (automatic) implementation

IBM Tivoli Monitoring supports most common firewall configurations. To enable this support, IBM Tivoli Monitoring uses the IP.PIPE socket address family, a TCP-based protocol that opens a single port on the firewall for communication by IBM Tivoli Monitoring components. If your target environment includes a firewall between any IBM Tivoli Monitoring components, you must specify IP.PIPE or IP.SPIPE as your communication protocol during configuration. Typically, no other special configuration is needed unless one of the factors discussed in “Determining which option to use” on page 799 requires it.

Implementation with ephemeral pipe

Configure your communications to use ephemeral pipe when addresses on either side of a barrier firewall are not identical or are not reachable from either side. (Note that use of ephemeral pipe may be dictated by the per-machine agent count, as well as any form of discontiguous addressing such as a NAT firewall.)

You configure an IP.PIPE or IP.SPIPE connection as an ephemeral pipe by adding the ephemeral keyword `ephemeral:y` the KDE_TRANSPORT environment variable immediately following the protocol keyword (IP.PIPE or IP.SPIPE) in the associated K*PP*ENV file for that process. You must then restart the process for the change to be effective.

Some K*PP*ENV files use KDC_FAMILIES instead of KDE_TRANSPORT. The process is exactly the same for the KDC_FAMILIES environment variable: adding the ephemeral keyword `ephemeral:y` immediately following the protocol keyword (IP.PIPE or IP.SPIPE) that is to be designated ephemeral.

For example, to configure the KNTAGENT to make an ephemeral connection to the monitoring server, change KDE_TRANSPORT (or KDC_FAMILIES) in the file KNTENV from

```
KDE_TRANSPORT=IP.PIPE PORT:1918 IP SNA
```

to

```
KDE_TRANSPORT=IP.PIPE ephemeral:y PORT:1918 IP SNA
```

or from

```
KDC_FAMILIES=IP.PIPE PORT:1918 IP SNA
```

to

```
KDC_FAMILIES=IP.PIPE ephemeral:y PORT:1918 IP SNA
```

To configure a remote monitoring server to make an ephemeral connection to the hub, change KDE_TRANSPORT (or KDC_FAMILIES) in the file KDSENV from

```
KDE_TRANSPORT=IP.PIPE PORT:1918 IP SNA
```

to

```
KDE_TRANSPORT=IP.PIPE ephemeral:y PORT:1918 IP SNA
```

or from

```
KDC_FAMILIES=IP.PIPE PORT:1918 IP SNA
```

to

```
KDC_FAMILIES=IP.PIPE ephemeral:y PORT:1918 IP SNA
```

Monitoring agents that configure their connections as ephemeral cannot warehouse data unless KPX_WAREHOUSE_LOCATION is also configured at the remote monitoring server to which the monitoring agent reports. The variable KPX_WAREHOUSE_LOCATION is an optional list of fully qualified, semicolon-delimited network names that must be added to environment file of the monitoring server to which the agents are connected. This file is located in different places, depending on the platform:

- On Windows systems: *install_dir*\CMS\KBBENV
- On UNIX systems:*install_dir*/config/kbbenv.ini

The syntax is:

```
KPX_WAREHOUSE_LOCATION=family_protocol:#network_address[port_number];
```

For example:

```
KPX_WAREHOUSE_LOCATION=ip.pipe:#192.168.0.14[18303];ip:#192.168.0.14[34543];
```

The KPX_WAREHOUSE_LOCATION variable does not work if you are using EPHEMERAL ports and the Warehouse Proxy Agent is on a server other than the remote monitoring server for that agent. When this is the case, and when "EPHEMERAL:Y" is used in the KDC_FAMILIES specification of the agent, the agent will route all communications, including data exports, to its monitoring server. In the agent log, the initial lookup for the Warehouse Proxy does retrieve the correct Warehouse Proxy Agent IP and port from the monitoring server's Local Location Broker. However, the agent switches to using the monitoring server's IP address for the export instead of the Warehouse Proxy Agent IP address due to the EPHEMERAL:Y setting.

When the Warehouse Proxy Agent is collocated with the agent's monitoring server, the export works normally because the virtual port that the request gets routed to is the correct listening port for the Warehouse Proxy Agent. The IP address for the Warehouse Proxy Agent and monitoring server are also the same. However, when the Warehouse Proxy Agent is on a different machine to the agent's monitoring server, there is nothing listening on the Warehouse Proxy Agent's port on the monitoring server machine. As a result a `Status = 8` error message is received when trying to route the export. There are a few possible solutions:

- Enable and use the KDE Gateway component between the affected agents and their monitoring server.
- Place the Warehouse Proxy Agent on the same monitoring server as the agent.
- If EPHEMERAL:Y is only there as a security blanket to ensure monitoring server ⇔ agent communications, remove it from the KDC_FAMILIES variables of the agents and recycle the agents.

Note: Another possible solution is to change the collection location for all attribute groups from the Tivoli Enterprise Monitoring Agent to the Tivoli Enterprise Monitoring Server. Note that this will considerably increase the monitoring server's processing load and is not usually a preferred method for this reason. However, this method is an option if the previous solutions cannot be implemented.

See also “Setting a permanent socket address for a proxy agent” on page 608.

Implementation with partition files

Address translation is an enhanced security feature of some firewall configurations. With this feature, components that must be reached across the firewall have two unique, but corresponding addresses: the external address (valid for components outside the firewall) and the internal address (valid for components inside the firewall).

In IBM Tivoli Monitoring, the component that typically must be reached for connection is the monitoring server; however, the Warehouse Proxy, which runs on Windows as a server-type application, must also be accessible to clients and also requires an external and internal address. A component on either side of the firewall knows only about the address that is valid for its side, its *partition*.

To accommodate sites with address translation, IBM Tivoli Monitoring uses a partition-naming strategy. This strategy requires two steps:

- The creation of a text file, called a *partition file*, as part of the configuration of a hub or remote monitoring server (or Warehouse Proxy). The partition file contains an entry that defines the address of that component in the other partition.
- The specification of a *partition name* (any alphanumeric string up to 32 characters) as part of the configuration of any agent, a hub or remote monitoring server, or Warehouse Proxy. A partition name must be specified for each component regardless of which side of the firewall it is located on.
- “Sample scenarios”
- “Creating or modifying the partition file in Manage Tivoli Enterprise Monitoring Services” on page 804
- “Creating the partition file manually” on page 805

Sample scenarios

The following scenarios illustrate how to implement partition files in various configurations. In these scenarios, your site has one firewall that contains two partitions, which are named OUTSIDE and INSIDE:

- “Scenario 1: Hub monitoring server INSIDE and monitoring agents OUTSIDE” on page 803
- “Scenario 2: Hub and remote monitoring servers INSIDE and monitoring agents OUTSIDE” on page 803

- “Scenario 3: Hub monitoring server INSIDE, remote monitoring server and agents OUTSIDE”

Scenario 1: Hub monitoring server INSIDE and monitoring agents OUTSIDE

The hub monitoring server is contained within the firewall in a partition named INSIDE. A partition file named `parthub.txt` is included in this partition and contains the following entry:

```
OUTSIDE ip.pipe: hub's_external_address
```

OUTSIDE is the partition name outside the firewall and *hub's_external_address* is the address of the hub monitoring server that is valid for the agents.

As part of the configuration of each agent, specify the name of the partition that each is located in OUTSIDE.

When an agent starts, parthub.txt is searched for an entry that matches the partition name *OUTSIDE* and sees the monitoring server address that is valid for the agents (the external address).

Scenario 2: Hub and remote monitoring servers INSIDE and monitoring agents OUTSIDE

Note: In Scenarios 2 and 3, all agents report to the remote monitoring server.

As part of the configuration of the hub monitoring server, specify the name of the partition that it is located in INSIDE. No partition file is needed because the only component that reports to it (the remote monitoring server) is also inside the firewall.

As part of the configuration of the remote monitoring server, specify the name of the partition that it is located in INSIDE. A partition file, `partremote.txt`, must also be created at the remote monitoring server. It contains the following entries:

```
OUTSIDE ip.pipe: remote's_external_address
```

When configuring the agents (all of which are outside the firewall, reporting to the remote monitoring server), specify the name of the partition that they are located in OUTSIDE. When the agents start, partremote.txt is searched for an entry that matches the partition name OUTSIDE and sees the remote monitoring server address that is valid for them (the external address).

Scenario 3: Hub monitoring server INSIDE, remote monitoring server and agents OUTSIDE

As part of the configuration of the hub monitoring server, specify the name of the partition that it is located in INSIDE. Create a partition file, `parthub.txt`, containing the following entry:

```
OUTSIDE ip.pipe: hub's_external_address
```

OUTSIDE is the partition name outside the firewall and *hub's_external_address* is the address of the hub monitoring server that is valid for the remote monitoring server.

As part of the configuration of both the agents and the remote monitoring server, specify the name of the partition they are located in OUTSIDE.

A partition file `partremote.txt` also must be created at the remote monitoring server. It contains the following entry:

```
INSIDE ip.pipe: remote's_internal_address
```

If the hub monitoring server needs to communicate with the remote monitoring server (for example, to issue a report request from an agent that is connected to the remote monitoring server), the `partremote.txt` file is searched for an entry that matches the partition name INSIDE and sees the remote monitoring server address that is valid for it (the internal address).

Creating or modifying the partition file in Manage Tivoli Enterprise Monitoring Services

The following instructions provide the steps for creating or editing the partition file using Manage Tivoli Enterprise Monitoring Services:

- "Windows: Editing the partition file"
- "UNIX and Linux: Editing the partition file"

Windows: Editing the partition file

You can create or modify the partition file using the Tivoli Enterprise Monitoring Server Configuration option in Manage Tivoli Enterprise Monitoring Services. Use the following steps:

1. Right-click the monitoring server you want to configure and select **Reconfigure**.

   The Tivoli Enterprise Monitoring Server Configuration window is displayed.
2. Select **IP.PIPE** as the communications protocol.
3. Select **Address Translation**.
4. Click **OK**.

   The Hub (or Remote) TEMS Configuration window is displayed.
5. Click **NAT Settings**.

   The NAT Settings window is displayed.
6. To create the file:
   a. For **Partition File**, type the fully qualified name of the partition file, for example `C:\IBM\ITM\CMS\KDCPARTITION.TXT`.

   b. For **Partition Name**, type the name of the partition to which the file applies.

   c. Click **Edit File**.

      A message appears saying that the partition file cannot be found and asking you if you want to create it.

   d. Click **Yes** to create the file.

      The file is created and opened in Notepad.

   e. Create the entry for the partition.

      The format for the entries is `PARTITION-ID IP.PIPE:`*`nn.nn.nn.nn`* `IP.PIPE:`*`nn.nn.nn.nn`*. For example, to create a monitoring server partition for a typical scenario with a monitoring agent outside of a NAT firewall connecting to a monitoring server behind a firewall, use the partition ID of your monitoring agent, two spaces, and then the IP address of the host of the monitoring server. Add additional IP.PIPE:*nn.nn.nn.nn* addresses on a single line for multiple network interface cards. See "Sample partition file" on page 805 for more information on creating entries in the partition file.

   f. Save and close the file.
7. Click **OK** to save the changes and close the configuration window.

UNIX and Linux: Editing the partition file

You can create or modify the partition file using the Tivoli Enterprise Monitoring Server Configuration option in Manage Tivoli Enterprise Monitoring Services. Use the following steps:

1. Select the monitoring server you want to configure.
2. Click **Action** → **Configure** → **Basic Settings**.
3. Select **IP.PIPE** as the communications protocol.
4. Select **Use Address Translation**.
5. Enter the full path and file name for the partition file.
6. Click **Create** to create the file (if it does not exist) or **Modify** to edit the file.
7. Enter the partition ID in the first column.

8. Enter the IP address in the second column. If you require a second IP address, enter it in the third column. (If more than two IP addresses are required for a partition ID, use a text editor to add the additional addresses. See "Sample partition file.")
9. Click **Save** to save the file and exit or **Cancel** to return to the previous screen without modifying the file.

Creating the partition file manually

If your site is using address translation, you must create a partition file. The partition file is a text file containing the name of a partition and its constituent interface address. You must create or modify this file before implementing firewall support with monitoring servers and agents, the portal server and the hub monitoring server, clients and the portal server, and monitoring agents and Warehouse Proxy Agents.

When Tivoli Management Services components need to communicate across a firewall that performs NAT, those components must be able to retrieve an IP address of the other component that is valid on its side of the firewall. To support this capability, the location broker namespace is logically divided into partitions with unique partition IDs. Partition IDs are specified using the KDC_PARTITION environment variable. The partition file is the means to insert appropriate IP addresses into the location broker namespaces.

When an IBM Tivoli Monitoring component performs a location broker lookup operation, the partition ID of its partition is automatically supplied. The location broker returns only addresses that have been defined for that partition namespace and no other. In effect, the IBM Tivoli Monitoring component sees only addresses that are valid for its partition.

A partition file is a standard text file defined to the system using the KDC_PARTITIONFILE environment variable. Within this file, each line describes a partition name with its constituent IP addresses using space delimited tokens. The format is as follows:

```
PARTITION-ID IP.PIPE:nn.nn.nn.nn IP.PIPE:nn.nn.nn.nn
```

The first token on each line is used as a case-insensitive partition ID. The partition ID can be any alphanumeric string with a maximum length of 32 characters. Subsequent tokens specified are treated as interface addresses in standard NCS format (`address-family:address`). For communication across firewalls, use only IP.PIPE for address-family.

The expected default location of the file is /*install_dir*/tables/*tems_name*.

Sample partition file

The following sample partition file illustrates the format and content expected.

```
# SAMPLE PARTITION FILE
#
# IMPORTANT: Do not overwrite this file. Copy to another directory
# before making changes.
#
# Lines beginning with a '#' are treated as comments and are ignored.
# Note: Do not specify a line that starts with an '*' as it might prevent
# proper functioning.
#
# Basic Format
# PARTITION-ID IP.PIPE:nn.nn.nn.nn IP.PIPE:nn.nn.nn.nn
#
# Procedure to edit this sample partition file.
# To create a monitoring server partition file for a typical scenario
# (monitoring agent outside of a NAT firewall connecting to a
# monitoring server behind the firewall),do the following:
# 1) Replace the "$OUTSIDE-PID$" with the partition id of your monitoring agent
# 2) Replace the "$OUTSIDE-TEMS-HOST-ADDRESS$" with the ip address of the monitoring
# server host outside of the firewall.
# 3) Add additional IP.PIPE:nn.nn.nn.nn addresses on a single line for multiple
# Network Interface Cards (NICs) as in the format above. Separate entries with
# two spaces.Lines can be continued by placing a backslash ('\') char at
# the end of the line.
#
################################################################################
$OUTSIDE-PID$ IP.PIPE:$OUTSIDE-CMS-HOST-ADDRESS$
```

Implementation with firewall gateway

The firewall gateway feature enables additional end-to-end connectivity options for use in environments with specific TCP/IP connection management policies. The firewall gateway is capable of negotiating multiple firewall hops and network address translation (NAT). It also allows you to configure network traffic so that it is initiated from the more secure network zone.

The Firewall Gateway provides the following functionality:

- Gateway instances interoperate over a single physical relay connection. Logical connections are multiplexed over the relay. The origination direction of the relay connection is configurable to match enterprise firewall transit requirements.
- Relay support enables a logical connection to span multiple firewall zones. Each relay instance can optionally provide access to the upstream management network. Multiple relays can be chained to provide seamless hops across multiple zones.
- Proxy support provides a transparent interface to IBM Tivoli Monitoring V6.2 components. Server proxy components reside downstream and listen for inbound connections. Client proxy components reside upstream and make connections to services on behalf of downstream endpoints.
- All ports used by gateway instances are configurable. Port pooling is available to constrain client proxy connections to designated port values.
- Multiple failover addresses can be configured for all gateway connections.

NAT alone is not a reason to use the firewall gateway, which is content-neutral and can proxy any TCP connection. In most cases, NAT processing is handled by the PIPE protocol (IP.PIPE or IP.SPIPE), which can be used without the firewall gateway. Use the gateway when you have any of the following scenarios:

- A single TCP connection cannot be made to span between IBM Tivoli Monitoring components An example would be that there are multiple firewalls between these components and a policy that does not allow a single connection to traverse multiple firewalls.
- Connection requirements do not allow the IBM Tivoli Monitoring default pattern of connections to the hub monitoring server. An example here would be agents residing in a less-secure zone connecting to a monitoring server residing in a more-secure zone. Security policy would only allow a connection to be established from a more-secure zone to a less-secure zone, but not the other way round.
- You must reduce open firewall ports to a single port or connection. For example, rather than opening the port for every system being monitored, you would like to consolidate the ports to a single “concentrator”. Connection requirements do not allow the IBM Tivoli Monitoring default pattern of connections to the hub monitoring server.
- You must reduce open firewall ports to a single port or connection. You must manage agent failover and monitoring server assignment symbolically at the hub monitoring server end of the gateway. Because gateway connections are made between matching service names, an administrator can change the failover and monitoring server assignment of downstream gateway agents by changing the client proxy bindings next to the hub monitoring server.

In the context of firewalls, the server and client relationship can best be described in terms of *upstream* and *downstream*. Those entities that open a socket to listen for requests are at the upstream or server end. Those entities connecting to the server are at the downstream or client end. Using one or more relay configurations, logical connection requests flow from a listening downstream server proxy interface, and terminate in an outbound connection from an upstream client proxy interface to a listening server. Intermediate relay configurations consist of an upstream relay interface containing at least one downstream relay interface.

Configuration

The gateway component is configured through an XML document that specifies a set of zones, each of which contain at least one upstream interface with one or more imbedded downstream interfaces. The following sections provide information on the activation, structure, and content of the XML document:

- "Activation"
- "IPv4 Address Data"
- "IPv6 Address Data"
- "XML Document Structure"

Activation

The gateway feature can be activated within any IBM Tivoli Monitoring process. However, use must be limited to the host computer operating system agent to prevent potential resource consumption conflicts with Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server processes.

The configuration variable `KDE_GATEWAY` is set to the XML configuration file name. A line of the form `KDE_GATEWAY=`*`filename`* must be added to the following configuration files, depending on your environment:

- On Windows computers, configuration variables for the Windows operating system agent are located in the *`ITMinstall_dir`*`\tmaitm6\KNTENV` file.
- On UNIX computers, configuration variables for the UNIX operating system agent are located in the *`ITMinstall`*`_dire/config/ux.ini` and `ITMinstall_dir/config/ux.config` files. Add the entry to both files for reliable results.
- On Linux computers, configuration variables for the Linux operating system agent are located in the *`ITMinstall_dir`*`/config/lz.ini` and *`ITMinstall_dir`*`/config/lz.config` files. Add the entry to both files for reliable results.

After you make these changes, stop and restart the monitoring agents.

For information on configuring firewall support on z/OS, see the *OMEGAMON XE shared publications: Common Planning and Configuration Guide* and *IBM Tivoli Monitoring: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

IPv4 Address Data

Internet Protocol Version 4 (IPv4) addresses supplied as data to **<bind>** and **<connection>** tags can be in absolute dotted decimal or symbolic form. An address-specific port number override can be specified following a trailing colon (":") character.

IPv6 Address Data

Internet Protocol Version 6 (IPv6) addresses supplied as data to **<bind>** and **<connection>** tags can be in absolute uncompressed hexadecimal, absolute compressed hexadecimal, or symbolic form. Absolute hexadecimal expressions must be enclosed within parentheses ("(" and ")") with one-to four-digit groups separated by a colon (":"). Compression of a run of 0 digits can occur at most once and is indicated by double colons ("::"). An address-specific port number override can be specified following a trailing colon; this specification is outside the parentheses that wrap an absolute address.

XML Document Structure

The relationships between XML configuration elements are illustrated in Figure 173 on page 808. Attributes are described on affected elements; default values for most attributes can be supplied on outer elements with noted exceptions.

```
<tepgwml:gateway xmlns:tepgwml="http://xml.schemas.ibm.com/tivoli/tep/kde/">
 <zone>
  <interface>       upstream interface
   <bind>
    <connection>
    </connection
   </bind
   <interface>    downstream interface
    <bind>
     <connection>
     </connection>
    </bind>
   </interface>
  </interface>
 </zone>
 <portpool>
 </portpool>
</tepgwml:gateway>
```

Figure 173. Structure of firewall gateway XML configuration document

<gateway>

A **<gateway>** element in the assigned namespace `http://xml.schemas.ibm.com/tivoli/tep/kde/` contains configuration elements described within this document. The gateway XML processor semantically ignores valid XML until the container is opened, allowing for configuration documents to be imbedded in other documents. This element cannot contain data.

`name`

The **name** attribute is required, cannot contain imbedded delimiters, and must begin with a nonnumeric. This attribute is used to identify a specific gateway instance.This attribute cannot be inherited from an outer element.

`threads`

The **threads** attribute specifies the number of worker threads in a general purpose thread pool. The specification must satisfy `1 <= value <= 256`, and defaults to `32`. Threads in this pool are shared by all defined zones, and are used only by interface startup logic, and to recover from outbound buffer exhaustion conditions. The default value is generally more than adequate.

<zone>

A zone is a container of interfaces sharing communication resources. This element cannot contain data.

`name`

The **name** attribute is required, cannot contain imbedded delimiters, and must begin with a nonnumeric. This attribute is used to identify a specific zone instance. This attribute cannot be inherited from an outer element.

`maxconn`

The **maxconn** attribute imposes an upper limit on the number of concurrent gateway connections within the zone. Each proxy physical connection and each logical connection crossing a relay interface consume this value. The specification must satisfy `8 <= value <= 4096`, and defaults to `256`.

`bufsize`

The **bufsize** attribute sets the data buffer size within the zone. The specification must satisfy `256 <= value <= 16384`, and defaults to `2048`.

`minbufs`

The **minbufs** attribute sets the minimum number of buffers in the zone buffer pool that are reserved for inbound traffic. The specification must satisfy `4 <= value <= 1024`, and defaults to `64`.

maxbufs

The **maxbufs** attribute sets the maximum number of buffers in the zone buffer pool that are reserved for inbound traffic. The specification must satisfy `minbufs <= value <- 2048`, and defaults to `128`.

<interface>

An interface describes a set of network bindings that exhibit a fixed behavior according to a specified role, and based on whether it is defined as upstream, which means that the enclosing element is `<zone>`, or downstream, where the enclosing element is `<interface>`. In all roles, logical connections arrive through one or more downstream interfaces and are forwarded through the upstream interface. After a logical connection has been established end to end, data flow is full duplex. A valid configuration requires an upstream interface to contain at least one downstream interface. This element cannot contain data.

name

The **name** attribute is required, cannot contain imbedded delimiters, and must begin with a nonnumeric. This attribute is used to identify a specific interface instance. This attribute cannot be inherited from an outer element.

role

The **role** attribute is required, and describes the behavior of network bindings contained within. The role attribute must be specified as “proxy”, “listen”, or “connect”. Downstream proxy interfaces represent local listening endpoints, and function as a server proxy. Upstream proxy interfaces represent local connecting endpoints, and function as a client proxy. Relay interfaces are assigned either “listen” or “connect”. No configuration restriction is made on the relay connection role other than peer relay connections must specify the opposite role. Relay connections are considered persistent, are initiated at gateway startup, and automatically restarted in the event of a network disruption.

<bind>

A **<bind>** element represents connection resources on one or more local interfaces. When specified within interfaces that “listen” (downstream proxy, relay listen), bind elements represent listening ports on local interfaces. For “connect” interfaces (upstream proxy, relay connect), they represent the local binding to be used for the outbound connection. Specific local interface addresses can be supplied as data; the default interface is `any`.

localport

The **localport** attribute is required within “listen” interfaces, and is optional within “connect” interfaces. The value supplied can be either a number that satisfies `1 <= value <= 65535`, or for “connect” based roles, can only contain the name of a portpool element defined within the gateway.

ipversion

The**ipversion** attribute declares the address family to be used for activity within the tag scope. Valid values are `4` or `6`, with a default of `4`.

ssl

The **ssl** attribute controls SSL negotiation for connections within the scope of this binding. When specified as `yes`, a successful negotiation is required before a connection is allowed on the gateway. The default value is `no`, meaning no SSL negotiation occurs on behalf of the gateway connection. Note that this does not restrict the conveyance of SSL streams across a gateway, only whether or not the gateway acts as one end of the SSL negotiation. When this operand is specified on a relay binding, it can be used to secure relay traffic, and must be specified on both ends of the relay connection.

service

The **service** attribute is a character string used to represent a logical connection between client and server proxy interfaces. Each connection accepted by a server proxy must find an upstream client proxy connection with a matching service string. No value restrictions are imposed.

<connection>

The **<connection>** tag is used to supply remote network interfaces as data. When applied to a “listen” mode binding, the connection tag represents the list of remote interface addresses that are allowed to make a connection, and is optional. This tag is required for “connect” mode bindings, and describes the remote end of the connection. Multiple addresses can be supplied for failover purposes.

remoteport

The **remoteport** attribute supplies the default port number of remote interfaces described within this tag. The value supplied must satisfy `1 <= value <= 65535`.

<portpool>

The **<portpool>** tag is used to create a list of local port numbers to be used for outbound connections. Port numbers are supplied as data, and can be specified discretely or as a range expression separated by hyphen ("-"). Range expressions are limited to 1024 bytes to prevent syntax errors from resulting in larger ranges than expected. Multiple specifications of either form are allowed.

name

The **name** attribute is required, cannot contain imbedded delimiters, and must begin with a nonnumeric. This attribute is used to identify a specific portpool instance. This attribute cannot be inherited from an outer element, and is referenced by a *localport* attribute on a bind element.

Warehouse Proxy Configuration

To ensure that the Warehouse Proxy Agent listens at a fixed port number across the monitoring enterprise, append the following configuration text to the `KDC_FAMILIES` configuration variable for the Warehouse Proxy:

```
IP.PIPE SKIP:15 COUNT:1
```

The effect of this configuration change is to force the Warehouse Proxy to listen at the Tivoli Enterprise Monitoring Server well-known port number (default 1918) plus the quantity 4096 multiplied by 15. For example purposes, if the monitoring server port is defaulted to 1918, this causes the Warehouse Proxy to listen at 63358. The following examples assume this recommendation has been implemented.

Example gateway configuration scenario

This example uses a three-hop firewall scenario, shown in Figure 174 on page 811. This scenario makes the following assumptions:

- Connections can only cross a firewall from the more trusted side to the less trusted side.
- Relay data crossing a zone enters and leaves on separate ports.

The effects of NAT on cross-zone addresses are not shown for clarity. NAT connections are fully supported. Dynamic NAT connections may require that inbound connection verification be removed. This is accomplished by removing the **<connection>** tag under the "listening" **<bind>**.

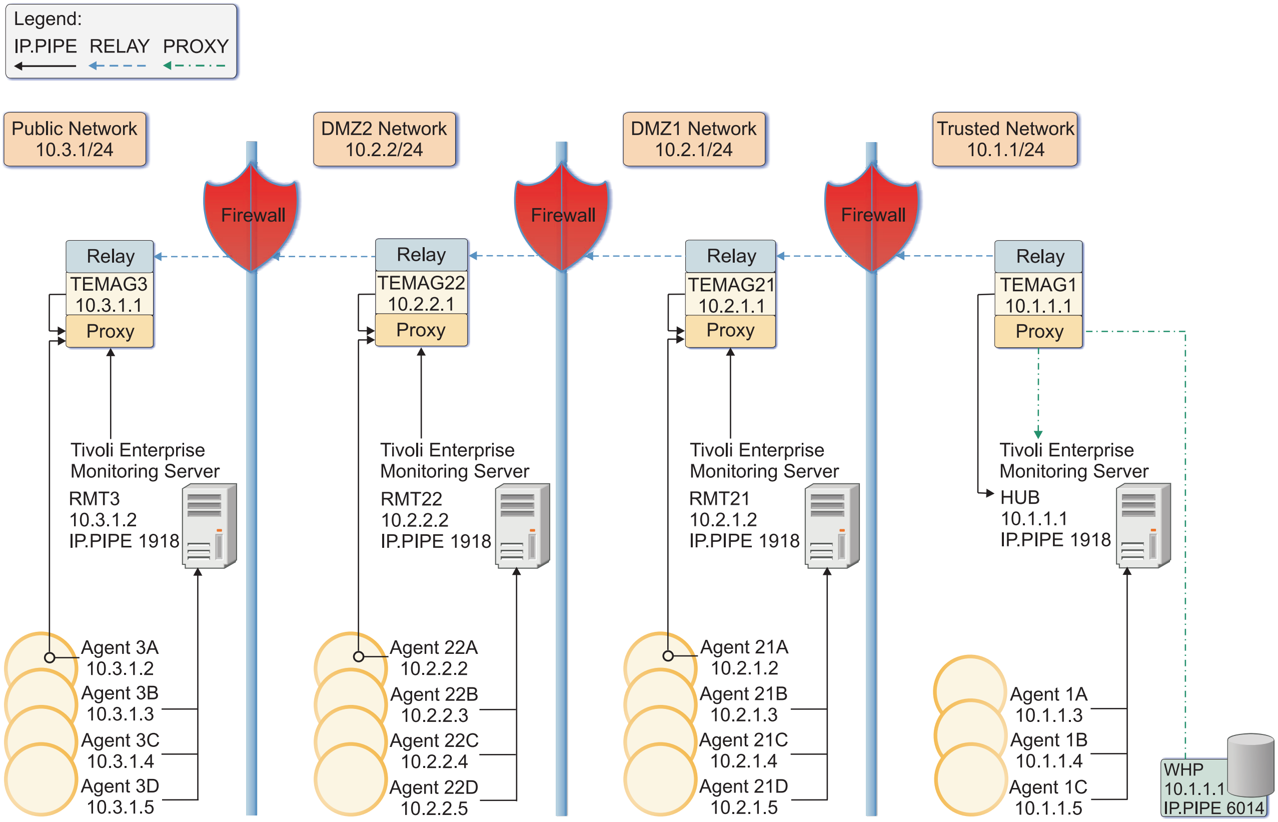


Figure 174. Three-hop firewall scenario

Public Network (TEMAG3):

The public network has the following characteristics:

- Gateway service is configured as part of operating system agent TEMAG3 on 10.3.1.1.
- TEMAG3 accepts a relay connection on port 10030 only from TEMAG22, port 10030.
- The Tivoli Monitoring components within this zone will contact the hub monitoring server and Warehouse Proxy server proxy ports 1918 and 63358 through the TEMAG3 interface address.

  The agents and the monitoring server at these zones needs to be configured as follows: Agents 3B, 3C, and 3D pointed directly to RMT3. RMT3 needs to be configured to point to TEMAG3, even if the configuration dialog asks for the hostname or address of the primary hub monitoring server. Agent3A as well as TEMAG3 itself should both point to TEMAG3.

  In general, a gateway agent should point to itself, except for the final gateway, which should point to a monitoring server as usual. In this example, TEMAG1 should point to HUB.

  In terms of node topology, all the agents and monitoring servers in this example that are pointed to the gateway agents will appear as if they are directly connected to the hub monitoring server.
- A remote monitoring server resides on a computer other than TEMAG3 to prevent conflict on port 1918.

The TEMAG3 gateway has the following configuration:

```
<tep:gateway name="temag3"
 xmlns:tep="http://xml.schemas.ibm.com/tivoli/tep/kde/" >
 <zone name="least_trusted">
  <!--
  upstream interface, listens for incoming relay
  connections, accepts traffic from downstream interfaces.
  -->
  <interface name="uprelay" ipversion="4" role="listen">
```

```
   <bind localport="10030">10.3.1.1
    <connection remoteport="10030">10.2.2.1
    </connection>
   </bind>
   <!--
   downstream interface, listens for incoming proxy
   connections, routes traffic over upstream relay.
   -->
   <interface name="serverproxy" ipversion="4" role="proxy">
    <bind localport="1918" service="tems"/>
    <bind localport="63358" service="whp"/>
   </interface>
  </interface>
 </zone>
</tep:gateway>
```

DMZ2 Network (TEMAG22):

The DMZ2 network has the following characteristics:

- Gateway service configured as part of OS agent TEMAG22 on 10.2.2.1.
- TEMAG22 originates a relay connection to TEMAG3 port 10030 using local port 10030.
- TEMAG22 accepts a relay connection on port 10022 only from TEMAG21, port 10022.
- Tivoli Monitoring components within this zone will contact the hub monitoring server and Warehouse Proxy server proxy ports 1918 and 63358 through the TEMAG22 interface address.
- A remote monitoring server resides on a computer other than TEMAG22 to prevent conflicts on port 1918.

The TEMAG22 gateway has the following configuration:

```
<tep:gateway name="temag22"
 xmlns:tep="http://xml.schemas.ibm.com/tivoli/tep/kde/" >
 <zone name="dmz2">
  <!--
  upstream interface, listens for incoming relay
  connections, accepts traffic from downstream
  interfaces
  -->
  <interface name="uprelay" ipversion="4" role="listen">
   <bind localport="10022">10.2.2.1
    <connection remoteport="10022">10.2.1.1</connection>
   </bind>
   <!--
   downstream interface, originates relay connection to
   downstream relay, routes traffic over upstream relay.
   -->
   <interface name="downrelay" ipversion="4" role="connect">
    <bind localport="10030">10.2.2.1
     <connection remoteport="10030">10.3.1.1</connection>
    </bind>
        </interface>
   <!--
   downstream interface, listens for incoming proxy
   connections, routes traffic over upstream relay.
   -->
   <interface name="serverproxy" ipversion="4" role="proxy">
    <bind localport="1918" service="tems"/>
    <bind localport="63358" service="whp"/>
   </interface>
  </interface>
 </zone>
</tep:gateway>
```

DMZ1 Network (TEMAG21):

The DMZ1 network has the following characteristics:

- Gateway service is configured as part of OS agent TEMAG21 on 10.2.1.1.
- TEMAG21 originates a relay connection to TEMAG22 port 10022 using local port 10022.
- TEMAG21 accepts a relay connection on port 10021 only from TEMAG1 port 10021.
- Tivoli Monitoring components within this zone will contact the hub monitoring server and Warehouse Proxy server proxy ports 1918 and 63358 via the TEMAG21 interface address.
- A remote monitoring server resides on a computer other than TEMAG21 to prevent conflicts on port 1918.

The TEMAG21 gateway has the following configuration:

```
<tep:gateway name="temag21"
 xmlns:tep="http://xml.schemas.ibm.com/tivoli/tep/kde/" >
 <zone name="dmz1">
  <interface name="uprelay" ipversion="4" role="listen">
   <bind localport="10021">10.2.1.1
    <connection remoteport="10021">10.1.1.1</connection>
   </bind>
   <interface name="downrelay" ipversion="4" role="connect">
    <bind localport="10022">10.2.1.1
     <connection remoteport="10022">10.2.2.1</connection>
    </bind>
   </interface>
   <interface name="serverproxy" ipversion="4" role="proxy">
    <bind localport="1918" service="tems"/>
    <bind localport="63358" service="whp"/>
   </interface>
  </interface>
 </zone>
</tep:gateway>
```

Trusted Network (TEMAG1):

The Trusted Network has the following characteristics:

- Gateway service is configured as part of operating system agent TEMAG1 on 10.1.1.1.
- TEMAG1 originates a relay connection to TEMAG21 port 10021 using local port 10021.
- TEMAG1 makes client proxy connections to the hub monitoring server using source ports in the range 20000-20099.
- TEMAG1 makes client proxy connections to the Warehouse Proxy Agent at destination port 63358 using source ports in the range 20100-20199.

The TEMAG1 gateway has the following configuration:

```
<tep:gateway name="temag1"
 xmlns:tep="http://xml.schemas.ibm.com/tivoli/tep/kde/" >
 <zone name="most_trusted">
  <!--
  upstream interface, traffic from downstream
  interfaces and originates proxy connections on behalf
  of downstream server proxy clients
  -->
  <interface name="clientproxy" ipversion="4" role="proxy">
   <bind localport="poolhub" service="tems">
    <connection remoteport="1918">10.1.1.1</connection>
   </bind>
   <bind localport="poolwhp" services="whp">
    <connection remoteport="63358">10.1.1.1</connection>
   </bind>
   <!--
   downstream interface, originates connection to
   downstream relay, routes traffic to upstream proxy
```

```
    -->
   <interface name="downrelay" ipversion="4" role="connect">
    <bind localport="10021">10.1.1.1
     <connection remoteport="10021">10.2.1.1</connection>
    </bind>
   </interface>
  </interface>
 </zone>
 <portpool name="poolhub">20000-20099</portpool>
 <portpool name="poolwhp">20100-20199</portpool>
</tep:gateway>
```

Appendix D. IBM Tivoli product, platform, and component codes

Table 164 lists the product codes that identify the different IBM Tivoli Monitoring components and base agents. Use these codes when running commands. For information on the product codes for Tivoli Monitoring distributed agents and other Tivoli Management Services based agents, see the product documentation.

Table 164. Component product codes for infrastructure components and base monitoring agents

Component	Product code
Endpoint monitoring agent	tm
i5/OS monitoring agent	a4
Linux OS monitoring agent	lz
Summarization and Pruning Agent	sy
Tivoli Enterprise Monitoring Server	ms
Tivoli Enterprise Portal browser client	cw
Tivoli Enterprise Portal desktop client	cj
Tivoli Enterprise Portal Server	cq
IBM Tivoli Universal Agent	um
IBM Message Service Client Library	w0
UNIX Log Alert monitoring agent	ul
UNIX OS monitoring agent	ux
Warehouse Proxy	hd
Windows OS monitoring agent	nt
Eclipse Help Server	kf
Agentless monitoring for Windows operating systems	r2
Agentless monitoring for AIX operating systems	r3
Agentless monitoring for Linux operating systems	r4
Agentless monitoring for HP-UX operating systems	r5
Agentless monitoring for Solaris operating systems	r6
Business System Manager common agent	r9
Tivoli Performance Analyzer	pa
Tivoli Performance Analyzer Domain for DB2	p0
Tivoli Performance Analyzer Domain for OS agent	p3
Tivoli Performance Analyzer Domain for Oracle	p4
Tivoli Performance Analyzer Domain for System P	p6
Tivoli Performance Analyzer Domain for ITCAM RT	pi
Tivoli Performance Analyzer Domain for VMware	pu
IBM Tivoli Composite Application Manager Agent for DB2	ud
IBM Tivoli Composite Application Manager Extended Agent for Oracle	rz

A complete, alphabetical list of product codes for IBM Tivoli Monitoring can be found at this Web site: http://www-01.ibm.com/support/docview.wss?uid=swg21265222&myns=swgtiv&mynp=OCSSZ8F3 &mync=E.

Table 165 lists the platform codes required by the commands related to remote agent deployment.

Table 165. Platform codes required for remote agent deployment

Platform	Code
AIX R4.1	aix4
AIX R4.2	aix42
AIX R4.2.0	aix420
AIX R4.2.1	aix421
AIX R4.3	aix43
AIX R4.3.3	aix433
AIX R5.1 (32 bit)	aix513
AIX R5.1 (64 bit)	aix516
AIX R5.2 (32 bit)	aix523
AIX R5.2 (64 bit)	aix526
AIX R5.3 (32 bit)	aix533
AIX R5.3 (64 bit)	aix536
HP-UX R10.01/10.10	hp10
HP-UX R10.20	hp102
HP-UX R11 (32 bit)	hp11
HP-UX R11 (64 bit)	hp116
HP-UX R11 Integrity (32 bit)	hpi113
HP-UX R11 Integrity (64 bit)	hpi116
Linux Intel R2.2	li622
Linux Intel R2.2 (32 bit)	li6223
Linux Intel R2.4	li624
Linux Intel R2.4 GCC 2.9.5 (32 bit)	li6242
Linux Intel R2.4 (32 bit)	li6243
Linux Intel R2.4 GCC 2.9.5 (64 bit)	li6245
Linux Intel R2.6 GCC 2.9.5 (32 bit)	li6262
Linux Intel R2.6 (32 bit)	li6263
Linux Intel R2.6 GCC 2.9.5 (64 bit)	li6265
Linux S390 R2.2 (31 bit)	ls322
Linux S390 R2.2 (31 bit)	ls3223
Linux S390 R2.2 (64 bit)	ls3226
Linux S390 R2.4 (31 bit)	ls324
Linux S390 R2.4 GCC 2.9.5 (31 bit)	ls3242
Linux S390 R2.4 (31 bit)	ls3243
Linux S390 R2.4 GCC 2.9.5 (64 bit)	ls3245
Linux S390 R2.4 (64 bit)	ls3246
Linux S390 R2.6 GCC 2.9.5 (31 bit)	ls3262
Linux S390 R2.6 (31 bit)	ls3263
Linux S390 R2.6 GCC 2.9.5 (64 bit)	ls3265
Linux S390 R2.6 (64 bit)	ls3266

Table 165. Platform codes required for remote agent deployment (continued)

Platform	Code
Linux AMD64 R2.4 (32 bit)	lx8243
Linux x86_64 R2.4 (64 bit)	lx8246
Linux AMD64 R2.6 (32 bit)	lx8263
Linux x86_64 R2.6 (64 bit)	lx8266
Linux ia64 R2.4 (64 bit)	lia246
Linux ia64 R2.6 (64 bit)	lia266
Linux ppc R2.4 (64 bit)	lpp246
Linux ppc R2.6 (32 bit)	lpp263
Linux ppc R2.6 (64 bit)	lpp266
MVS™	mvs
Digital UNIX	osf1
OS/2	os2
OS/400®	os400
Solaris R2.4	sol24
Solaris R2.5	sol25
Solaris R2.6	sol26
Solaris R7 (32 bit)	sol273
Solaris R7 (64 bit)	sol276
Solaris R8 (32 bit)	sol283
Solaris R8 (64 bit)	sol286
Solaris R9 (32 bit)	sol293
Solaris R9 (64 bit)	sol296
Solaris R10 (32 bit)	sol503
Solaris R10 (64 bit)	sol506
Solaris R10 Opteron (32 bit)	sol603
Solaris R10 Opteron (64 bit)	sol606
Tandem Itanium (64 Bit)	ta6046
Tandem MIPS (64 Bit)	tv6256
Tru64 V5.0	tsf50
Tivoli Enterprise Monitoring Server support	tms
Tivoli Enterprise Portal Server support	tps
Tivoli Enterprise Portal Desktop client support	tpd
Tivoli Enterprise Portal Browser client support	tpw
Common code, sample libraries and documentation	unix
Windows Itanium 64 bit	WIA64
Windows x86-64 64 bit	WIX64
Windows (all other environments)	winnt

Table 166 lists the various IBM Tivoli Monitoring components and the codes that represent them; these show up when invoking the **cinfo -t** command.

Table 166. Application support codes

Component	Code
Tivoli Enterprise Monitoring Server	TEMS
Tivoli Enterprise Portal Server	TEPS
Tivoli Enterprise Portal	TEP
Windows	WI
Tivoli Enterprise Monitoring Server application support (Windows systems)	WICMS
Tivoli Enterprise Portal Server application support (Windows systems)	WICNS
Tivoli Enterprise Portal browser client application support (Windows systems)	WIXEB
Tivoli Enterprise Portal desktop client application support (Windows systems)	WIXEW
Tivoli Enterprise Monitoring Server application support (Linux or UNIX systems)	tms
Tivoli Enterprise Portal Server application support (Linux or UNIX systems)	tps
Tivoli Enterprise Portal browser client application support (Linux or UNIX systems)	tpw
Tivoli Enterprise Portal desktop client application support (Linux or UNIX systems)	tpd

Appendix E. Agent configuration and environment variables

This appendix contains information about the IBM Tivoli Monitoring configuration files that are preserved during an upgrade, behavior of customized configuration settings, information about how to create persistent configuration changes, as well as a list of environment variables related to the components and monitoring agents of IBM Tivoli Monitoring.

The following table lists the information topics related to IBM Tivoli Monitoring agent configuration and environment variables:

Table 167. Information topics related to IBM Tivoli Monitoring agent configuration and environment variables

Topic	**Where to find information**
Configuration files that are preserved during an upgrade.	To understand which configuration files are preserved during an upgrade, and which configuration files are not preserved, see "Configuration files preserved during an upgrade."
Product behavior with customized configuration settings.	For an overview of product behavior regarding customized configuration settings, see "Product behavior with custom configuration settings" on page 823.
Persistent configuration changes.	For information about how to create persistent configuration changes for configuration environment variables, see "Persistent configuration changes" on page 823.
IBM Tivoli Monitoring environment variables.	For a list of environment variables related to the components and monitoring agents of IBM Tivoli Monitoring, see "Environment variables" on page 826.

Configuration files preserved during an upgrade

Custom settings and variables located in configuration files might be lost when upgrading IBM Tivoli Monitoring. These files include `K??CMA.ini` and `K??ENV` files for Windows systems, and `*.ini`, `*.config`, `k*env` files for Linux or UNIX, and more. This section explains which configuration files are preserved during an upgrade, which configuration files are not preserved, and the behavior of custom settings in configuration files.

Preserving user custom settings

During an upgrade many product files are replaced with newer versions. Other files are merged with existing files to produce the updated version. Still other files are generated by the installation process using values you provide. For more information see, "Product behavior with custom configuration settings" on page 823.

The following general rules apply to how user custom settings are preserved:

- User-defined constructs are kept. For example, situations, policies, queries, and workspaces are always preserved automatically on upgrade.
- Values you can change through a supported product interface are preserved.
- Values that you changed manually (for example, because of a technote or as directed by IBM software support) are probably preserved. Any value that you changed manually that was restored to a default value during an upgrade is recoverable from the backups made during the upgrade process.

The configuration process works with two basic types of files:

- **K??CMA.ini, k*env** (initialization) files are used to collect the inputs from the installation process. This input information is the set of responses to installation questions that are captured as keyword-value pairs. This information is laid down with default values and basic information about the installed components.

- **K??ENV, *.config** (configuration) files are generated from the values in the initialization files and the values entered during configuration.

The initialization files are source files and the configuration files are output files. Although the source files are modified by the configuration tools and sometimes by hand, the configuration output files are rarely modified. Configuration files are generated files, therefore anything changed manually in configuration files is lost during the reconfiguration of that component.

To recover a manual customization after upgrading, perform the following steps:

1.
   - On Windows systems: Compare the new version of the K??CMA.ini or K??ENV file with the version saved in the `itm_home\backup\backups\date_and_time_of_upgrade directory`.
   - On Linux or UNIX systems: Compare the new version of the *.ini file with the version saved in the same directory as *.ini.bak.
2. Change the installer-supplied defaults to the hand-edited values found in the backup file. Make any changes required to carry your hand-edited custom settings forward, and save the new file.

Files that are preserved on upgrade

The following files are preserved when you upgrade your IBM Tivoli Monitoring environment. If you modified these files or settings, you can expect the changes to be preserved on upgrade:

`-D flags in Applet.html`
Any changes made to the `-D` flags are preserved.

`Bannerimage.html`
Bannerimage.html in the CNB directory. If you add a customer image for your own banner, it is preserved.

`-D flags in "cnp" batch or script files`
Any changes made to the -D flags on the Java calls of these files are preserved. The specific file names are as follows:

- On Linux or UNIX systems: `cnp.sh and cnp_`*`inst`*`.sh`
- On Windows systems: `cnp.bat and cnp_`*`inst`*`.bat`

where *inst* is the name of an instance of the Tivoli Enterprise Portal to connect to. The -D flags in cnp.sh are not preserved.

`K??ENV, *.ini`
The current settings from ENV files are preserved by checking "key = value" and adding keys that did not exist in the new file from the old file, and replacing the value from the old file in the new file. Keys with default values are preserved.

`K??CMA.ini, k*env`
If a change was made using a provided configuration tool, the value is always preserved. If you were instructed by a technote or by an IBM support engineer to make a manual change the value is most likely, but not always, preserved (depending on what was changed and why).

`OM_TEC.config`
In one of the following directories:

- On Linux or UNIX systems: `itm_home/tables/temsname/TECLIB`
- On Windows systems: `itm_home\cms\TECLIB`

`tecserver.txt`
In one of the following directories:

- On Linux or UNIX systems: `itm_home/tables/temsname/TECLIB`
- On Windows systems: `itm_home\cms\TECLIB`

Files that are not preserved on upgrade

The following files are not preserved when you upgrade your IBM Tivoli Monitoring environment. If you modified these files or settings, you must make a backup before performing the upgrade. After the upgrade, you must reconstitute your modifications from your backups.

For the Manage Scripts feature in Tivoli Enterprise Portal

The Manage Scripts feature in the Tivoli Enterprise Portal has a set of built-in scripts that manage 3270 terminal session navigation. These built-in functions persist after you install IBM Tivoli Monitoring. However, you must save the custom scripts that you define for the navigation of 3270 terminal sessions. For more information, see the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.

For components running on Windows systems OR Linux or UNIX systems

The following IBM Tivoli Enterprise Console event synchronization files are not preserved on upgrade:

- In the TEC_CLASSES directory of the rulebase created during IBM Tivoli Enterprise Console event synchronization installation: omegamon.baroc, Sentry.baroc.
- In the TEC_RULES directory of the rulebase created during IBM Tivoli Enterprise Console event synchronization installation: omegamon.rls.

Before these files are replaced, backup copies are made automatically and placed in the same directories as the original files and have the **.bac** suffix added to their names. You can open these backup files and manually migrate modifications.

Note: The **.baroc** and **.rls** files are backed up only if you choose to automatically upgrade the specified rulebase.

For components running on Windows systems

The following files are not preserved on upgrade:

- On the Tivoli Enterprise Portal Server, the **CNP.bat** and **applet.html** files are built based on the content of the CNPS, CNP, and CNB directories. Except for the -D flags, everything in these files is regenerated.
- The **buildpresentation.bat** file is generated during installation. Any updates to this file are lost.
- The **CNB/jrelevel.js** file is regenerated during installation. Any updates to this file are lost.

For components running on Linux or UNIX systems

The following files are not preserved on upgrade:

- *ARCH*/**cq|cj|cw/***, where ARCH is a specific architecture, such as **li6263**) and one of the following two-letter codes: cq, cj, or cw). These files are overwritten during upgrade. For example, changes to **ARCH/cw/jrelevel.js** are not preserved.
- The autostart scripts. These files are regenerated during the installation and configuration and have different names based on the platform:
  - Linux, Solaris: /etc/init.d/ITMAgents*
  - HP-UX: /sbin/init.d/ITMAgents*
  - AIX: /etc/rc.itm*

Special exceptions

On Linux or UNIX systems, several IBM Tivoli Monitoring configuration files exist with values that do not typically persist during an upgrade. However, the values in the following files do persist:

- itm_home/tables/temsname/*.txt, where ***.txt** refers to all file names that have the text (***.txt**) tag, including **partition.txt** (previously including **glb_site.txt**).

  Note: The **glb_site.txt** file is not supposed to be preserved. It is a configuration output file and its values are gathered and stored during normal configuration.
- The following files are generated from **kbbenv.ini** and **ms.ini**:
  - *itm_home*/tables/*temsname*/KBBENV
  - *itm_home*/config/hostname_ms_temsname.config

Changes made to these files are saved in the same way that changes made to the source .ini files are saved.

Product behavior with custom configuration settings

This section provides an overview of product behavior regarding custom configuration settings.

Before you upgrade, make sure you back up your IBM Tivoli Monitoring environment, especially if you have a more complicated environment. Most companies have a standard process for preserving a backup image of all computer systems, including the IBM Tivoli Monitoring environment.

General operations

Several types of variables control the operation of an IBM Tivoli Monitoring component. See the following basic types of variables:

`User-modified variables`
You generate these settings when doing a configuration, changing a value in the CLI or GUI configuration GUIs, and then saving the changes. These variables are user-modifiable variables. They are typically stored in initialization K??CMA.ini, k*env files on the disk. The output (K??ENV, *.config) files use these variables, such as **key=$VAR1$**, which are then substituted with the value that you specify.

`Static variables`
The second type of setting is a variable for an internal component that you cannot configure through the common configuration tools. These variables are stored in K??CMA.ini, K??ENV, k*env, *.ini files, in entries such as **NUM_TIMES_TO_TRY=4**. No dynamic substitution takes place. This type of setting is static.

The following list describes persistence in several scenarios:

- During an upgrade from one version of a component to another, the user-modified variables are always persisted with the installation values that are set through the configuration tools.
- If you manually modify these values, for example, adding a new value to the end of an existing **$VR$** value, that new value is not persisted.
- The static values for the system are typically not persisted because they are internal component variables, and not typically exposed to the user for configuration. These static variables are not documented. In most cases, do not change them. Changing these values without instructions from a technote or IBM Software Support can lead to unpredictable results. To ensure stability of the system, the default values are restored when you upgrade to a new component version level.

  Although these variables normally are not persisted, a variable that you add that does not already have a key is persisted. This type of value is normally created under the guidance of IBM Software Support.

Keep notes about any changed static variables that you modify in response to instructions from technotes, IBM Software Support, and so on. After performing an upgrade, examine if the values need to be reapplied.

Persistent configuration changes

Advanced users can apply override values to component customization. Applying override values ensures that values are retained during an upgrade. You should test this in your environment first before you apply it globally.

Windows persistent configuration changes

The Windows agent process preserves configuration changes by design. Updated variables in the `kxxcma.ini` file, where `xx` denotes the product code, are kept in the Override Local Settings section. These variables are used during each configuration to update the Windows registry entries that the agents use at run time.

Example:

1. Select the monitoring agent that you want to update in the Manage Tivoli Enterprise Monitoring Services window.
2. Right-click the monitoring agent and select **Advanced** > **Edit Variables**.
3. Click **Add**.
4. Select **KDC_FAMILIES** from the drop-down list of variables. The `@Protocol@` value is displayed in the **Value** field.
5. Append **ephemeral:y** to the current value. The resulting string is `@Protocol@ ephemeral:y`.
6. Click **OK** to save the change.
7. Click **OK** again.

Linux or UNIX persistent configuration changes

The files that contain configuration information for IBM Tivoli Monitoring Linux or UNIX processes are located in the *itm_home*/config directory. One such file is the `xx.ini` file where `xx` denotes the product code. For example, `ux` is the product code for the UNIX OS agent. Additional information is stored in locations such as *itm_home*/config/.ConfigData/kxxenv where the user defaults and configuration dialog variables are stored. Some information is also stored in other files such as `env.config`. All these files are processed to create a `.config` file.

Each time an agent is started (or stopped) the agent `.config` files are created in one of the following formats:

- xx.config
- xx_instance.config
- hostname_xx_instance.config

Tivoli Enterprise Monitoring Server `.config` files are created during the Tivoli Enterprise Monitoring Server configuration process:

```
./itmcmd config -S -t temsname
```

Tivoli Enterprise Monitoring Server `.config` files have the format: `hostname_ms_temsname.config`. The agent uses the `.config` file to define the runtime configuration environment variables.

Making persistent configuration changes

To make persistent configuration changes, create the file `xx.environment` in the config directory. In this file define variables in the **key=value** format. Variables defined in the `xx.environment` file have override variables from the `.config` file during agent startup.

Example 1: Overriding the CTIRA_HOSTNAME to change the host name section of the Managed System Name: Create a file in the `config` directory called `xx.environment` with the following contents:

```
CTIRA_HOSTNAME=myname
```

The contents of `xx.environment` is included in the environment.

Result: When the component starts, the lines in the `xx.environment` file are included and processed after other lines in the `.config` file.

Example 2: Altering an existing variable KDC_FAMILIES: The KDC_FAMILIES environment variable is particularly challenging because much of the data is created from the `.ConfigData/kxxenv` dialog file. For example, suppose that in the existing `.config` file the KDC_FAMILIES variable looks similar to the following variable:

```
export KDC_FAMILIES='ip.pipe port:1918 ip use:n ip.spipe use:n sna use:n HTTP:1920'
```

But you must have `ephemeral:y` added. In this case, the contents of the `xx.environment` file might be similar to the following variable:

```
KDC_FAMILIES=ip.pipe port:1918 ephemeral:y ip use:n ip.spipe use:n sna use:n HTTP:1920
```

A more robust content line might be written as follows:

```
KDC_FAMILIES=ephemeral:y ${KDC_FAMILIES}
```

Create an `xx.environment` file for each product that you need to modify. That way your overrides persists for upgrades in current releases, and in future releases.

Notes:

1. If you add a variable to the `xx.environment` file and during a later configuration, change a parameter that is also in the `xx.environment` file, the `xx.environment` file will override your configuration changes. If you use the more robust way of overriding existing variables as described in Example 2, those configuration changes will be included.
2. Future IBM Tivoli Monitoring release levels might change the meaning and defaults of configuration variables. You must be aware of changes made to configuration variables and make appropriate alterations to achieve your intended result.
3. In the worst case scenario, an override might prevent the agent from starting or running successfully. This scenario requires a manual login and correction. Therefore you must always thoroughly test the setup.
4. Some agents use other installation configuration files. The method outlined in this section works only on Linux or UNIX `.ini` file configuration.
5. You can find values passed to the component process in the *itm_home*`/logs/xx.env` file.

Environment variables

This section contains environment variables related to the components and monitoring agents that comprise the base IBM Tivoli Monitoring product.

Common environment variables

The following table lists the environment variables that are common to all components.

Table 168. Common environment variables

Variable	Value type	Purpose
KDC_DEBUG	Y or N	Default is N. Diagnosing RPC communications or connectivity problems between the following components: • The portal server and the monitoring server. • An agent and the monitoring server. • A remote monitoring server and the hub monitoring server.
KDE_DEBUG	Y or N	Default is N. Diagnosing general communication problems, including TCP and SSL setup and connections, local interface discovery, host name resolution, and so on.
KBS_DEBUG	Y or N	Default is N. Consolidates KDC_DEBUG, KDE_DEBUG, and KDH_DEBUG, into one environment variable to simplify the process of collecting diagnostic information.
KDE_TRANSPORT		Change the default port settings for a monitoring agent. For more information, see “Tivoli Monitoring protocol usage and protocol modifiers” on page 379.
KDC_FAMILIES		Change the default port settings for a monitoring agent. For more information, see “Tivoli Monitoring protocol usage and protocol modifiers” on page 379.
KDEB_INTERFACELIST		Contains interface addresses in the order for which these interface addresses should be discovered and used.
KDH_DEBUG	Y or N	Default is N. Diagnosing connectivity problems with the integrated web server.
KBB_RAS1	Trace specification string	Default tracing level: KBB_RAS1=ERROR Diagnosing client request problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) Diagnosing client or Tivoli Enterprise Monitoring Server interaction problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) (UNIT:ctcmw IN ER) (UNIT:kv4 IN ER) Diagnosing SQL generation problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) (UNIT:ctreport ALL) Diagnosing login problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) (UNIT:ctauth ALL)
KBB_RAS1_LOG		Determines the count limit and the maximum files settings.
KBB_SIG1	To generate tracing, core dumps or stacktraces, on error.	• Possible values: -trace, -dumpoff, -asyncoff • Default value: blank

Tivoli Enterprise Portal Server environment variables

The following table lists Tivoli Enterprise Portal Server environment variables.

Table 169. Tivoli Enterprise Portal Server environment variables

Variable	Value type	Purpose
DSUSER1	Y or N	Part of a set of 9 variables DSUSER1...DSUSER9 that you can set by using the **tacmd configureportalserver** command. For more information, see the *IBM Tivoli Monitoring Administrator's Guide*.
KBB_RAS1	Trace specification string	Default tracing level: KBB_RAS1=ERROR Diagnosing client request problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) Diagnosing client or Tivoli Enterprise Monitoring Server interaction problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) (UNIT:ctcmw IN ER) (UNIT:kv4 IN ER) Diagnosing SQL generation problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) (UNIT:ctreport ALL) Diagnosing login problems: KBB_RAS1=ERROR (UNIT:ctsql IN ER) (UNIT:ctdata IN ER) (UNIT:ctauth ALL)
KDH_SERVICEPOINT	Service point string	Specifies an alternate service point string, which labels an ITM process in the IBM Tivoli Monitoring Service Index web page. If not specified, the service point string defaults to `<process-owner>.<system-name>_<product-code>`.
KFW_CMW_SPECIAL_HUB_ ENTERPRISE=N	Y or N	Default is N. When set to Y, associates situations to the Tivoli Enterprise Portal Server.
KFW_DATABUS_INPUT_ TRACE_IGNORE_HEARTBEAT	Y or N	Default is N. Reduces trace volume by skipping client heartbeat requests when (UNIT:ctdata IN ER) request tracing is used.
KFW_DATABUS_QUERY_ VERBOSE	Y or N	Default is N. Tivoli Enterprise Portal Server client side response time and request life cycle tracing.
KFW_MIGRATE_CMS	Y or N	Default is N. When set to Y, causes initial migration of managed objects and user IDs from the monitoring server the first time the Tivoli Enterprise Portal Server starts after installation.
KFW_MIGRATE_FORCE	Y or N	Default is N. When set to Y, SQL seed files are processed even if the date, time, or size of the files have not changed since prior seeding. This occurs when running the `buildpresentation.bat` file (Windows) and when running the `InstallPresentation.sh` file or reconfiguring the Tivoli Enterprise Portal Server (UNIX/Linux). To be effective, this value must be set in `kfwalone` (Windows) or in `cq.ini` or `lnxenvnocms` (UNIX/Linux).
KFW_MIGRATE_VERBOSE	Y or N	Default is N. When set to Y, provides greater detail about operations performed in `migrate.log` when seeding the Tivoli Enterprise Portal Server database. This occurs when running the `buildpresentation.bat` file (Windows) and when running the `InstallPresentation.sh` file or reconfiguring the Tivoli Enterprise Portal Server (UNIX/Linux). To be effective, this value must be set in `kfwalone` (Windows) or in `cq.ini` or `lnxenvnocms` (UNIX/Linux).

Table 169. Tivoli Enterprise Portal Server environment variables (continued)

Variable	Value type	Purpose
KFW_REPORT_FIND_ WAREHOUSE_AT_ STARTUP	Y or N	Default is Y. When set to Y, the portal server queries the data warehouse during startup. If set to N, the warehouse will be initialized at the first deliberate query instead of pre-initialized.
KFW_REPORT_ROW_LIMIT	positive integer or 0	Default is 100000. Set this variable to limit the number of rows returned for a report request. A KFWITM574W warning message displays in the view indicating the result set was truncated. Set to 0 to disable. This variable affects only report requests when the user selects the "return all rows" option on the **View Properties** window.
KFW_REPORT_TERM_ BREAK_POINT	Y or N	Default is N. When set to Y, specifies the point where a historical request selects from short-term or long-term history data.
KFW_REPORT_WITH_UR	Y or N	Default is N. When set to Y, it appends "with UR" to the SQL statement against the DB2 Tivoli Data Warehouse or Summarization and Pruning table issued by the Tivoli Enterprise Portal Server. It does not lock out the row and tables in the database; and allows other commands access to the Warehouse database at the same time.
KFW_SQL_VERBOSE	Y or N	Default is N. When set to Y, provides RAS1 trace of each SQL statement issued by the Tivoli Enterprise Portal Server to the Tivoli Enterprise Monitoring Server or to the DB2/ODBC data source. The use of this variable is deprecated because the same capability is available using the standard RAS1 trace setting "KBB_RAS1=ERROR (UNIT:ctsql IN ER)"
MSG_MODE	kms or MSG2	Specify kms if you want to use IBM Tivoli Monitoring Operations Logging. Specify MSG2 to use MSG2 logging.
TEPS_SDA	Y or N	`N` disables the self-describing agent capability at the portal server, whereas `Y` enables it. The self-describing agent capability is enabled by default at the portal server. **Note:** Do not specify `YES` or `NO`; instead, always specify `Y` or `N`.

Table 169. Tivoli Enterprise Portal Server environment variables (continued)

Variable	Value type	Purpose
TEPS_MANIFEST_PATH	Any valid path	The location where you want the portal server to store the manifest and JAR files it collects from the self-describing agents. This parameter is normally set during component installation. Use the following steps: • On Windows: 1. On the computer where the monitoring server is installed, click **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services.** 2. Right-click **Tivoli Enterprise Portal Server** and click **Advanced→ Edit Variables**. 3. When you are asked if you want to stop the service, answer **Yes**. 4. The default file location that is set for **TEPS_MANIFEST_PATH** is `*ITMinstall_dir*\cnps\support` where *ITMinstall_dir* is the directory where you installed the product. You can change this location if you want to do so. • On Linux or UNIX: 1. On the computer where the monitoring server is installed, change to the `<install_dir>/config/directory`. 2. Open the `cq.config` file. 3. The default file location that is set for **TEPS_MANIFEST_PATH** is *`ITMinstall_dir/arch/`*`cq/support` where *ITMinstall_dir* is the directory where you installed the product. You can change this location if you want to do so.

Tivoli Enterprise Monitoring Server environment variables

The following table lists Tivoli Enterprise Monitoring Server environment variables.

Table 170. Tivoli Enterprise Monitoring Server environment variables

Variable	Purpose
ATTRLIB	Specifies the Tivoli Enterprise Monitoring Server attribute (ATR) files directory.
CANDLE_HOME	The directory where the product was installed.
CMS_BINPATH	Specifies the Tivoli Enterprise Monitoring Server binary files directory.
CMS_EXTERNALBROKERS	Specifies whether there are internal brokers.
CMS_FTO	Specifies whether to use Tivoli Enterprise Monitoring Server Hot Standby.
CMS_MSGBASE	Applies to is/5 platform Tivoli Enterprise Monitoring Agent only. Specify the MSG2 message file for Tivoli Enterprise Monitoring Agent framework messages.
CMS_NODEID	The Tivoli Enterprise Monitoring Server IBM Tivoli Monitoring Node ID.
CMS_NODE_VALIDATION	Enables the rejection of incorrect managed system names.
DEPLOYQUEUESIZE	Specifies the maximum number of requests that the request queue on the agent deployment controller.
DEPLOYTHREADPOOLSIZE	Specifies the number of threads that are available to the deployment controller.

Table 170. Tivoli Enterprise Monitoring Server environment variables (continued)

Variable	Purpose
DEPOTHOME	Specifies the root directory of the agent depot on the file system.
KDC_GLBSITES	Specifies the Tivoli Enterprise Monitoring Server global sites network file with Tivoli Enterprise Monitoring Server Hub host names.
KDH_SERVICEPOINT	Specifies an alternate service point string, which labels an ITM process in the IBM Tivoli Monitoring Service Index web page. If not specified, the service point string defaults to `<process-owner>.<system-name>_<product-code>`.
KDS_CATLGLIB	Specifies the catalog library.
KDS_HUB	Specifies that this Tivoli Enterprise Monitoring Server is a Hub Tivoli Enterprise Monitoring Server (*LOCAL) or Remote Tivoli Enterprise Monitoring Server (*REMOTE).
KDS_NCS	Whether to use IBM Tivoli Monitoring network lookup services.
KDS_RULELIB	Specifies the rule library.
KDS_RUN	Specifies Tivoli Enterprise Monitoring Server components or probes to run at Tivoli Enterprise Monitoring Server startup.
KDS_START	Specifies Tivoli Enterprise Monitoring Server KDS component startup.
KDS_VALIDATE	Whether to use Tivoli Enterprise Monitoring Server authentication.
KGLCB_FSYNC_ENABLED	Whether to enable UNIX or Linux fsync calls.
KGL_CBTBUFCNT	Specifies Tivoli Enterprise Monitoring Server internal table buffer count.
KGL_CBTBUFSZ	Specifies Tivoli Enterprise Monitoring Server internal table buffer size.
KGL_KEYRING_FILE	Specifies LDAP authentication SSL GSKit keyring file.
KGL_KEYRING_LABEL	Specifies LDAP authentication SSL GSKit keyring label.
KGL_KEYRING_PASSWORD	Specifies LDAP authentication SSL GSKit keyring password.
KGL_KEYRING_STASH	Specifies LDAP authentication SSL GSKit keyring password stash file.
KGL_KGLMSGBASE	Specifies the Tivoli Enterprise Monitoring Server KGL message file directory.
KGL_LDAP_BASE	LDAP authentication LDAP search base.
KGL_LDAP_BIND_ID	LDAP authentication LDAP server bind Distinguished Name (DN).
KGL_LDAP_BIND_PASSWORD	LDAP authentication LDAP server bind password.
KGL_LDAP_HOST_NAME	LDAP authentication LDAP server host name.
KGL_LDAP_PORT	LDAP authentication LDAP server port.
KGL_LDAP_SSL_ENABLED	Whether to use LDAP SSL communications.
KGL_LDAP_USER_FILTER	LDAP authentication user filter.
KGL_LDAP_VALIDATE	Whether to use LDAP authentication.
KGL_MSG2_EVENTLOG	Whether to format Event Log.
KGL_MSG2_UNIVERSAL	Whether to enable Universal Messages.
KGL_TRC1	Whether to enable the error log.
KHD_HISTRETENTION	Specifies the default retention period in hours for the short-term history files (default is 24 hours). This can be used to reduce the amount of data kept on disk after a successful upload to the warehouse is performed.
KHD_HISTRETENTION	Specifies the default retention period in hours for the short-term history files (default is 24 hours). This can be used to reduce the amount of data kept on disk after a successful upload to the warehouse is performed.
KIB_MAXCOLS	Tivoli Enterprise Monitoring Server internal dictionary column maximum.

Table 170. Tivoli Enterprise Monitoring Server environment variables (continued)

Variable	Purpose
KMS_DISABLE_TEC_EMITTER	TEC Emitter to be disabled.
KMS_OMTEC_ GLOBALIZATION_LOC	TEC Integration Globalization locale.
KMS_OMTEC_INTEGRATION	TEC Integration enabled.
KPX_WAREHOUSE_LOCATION	Allows a fixed warehouse route for the agents connected to that Tivoli Enterprise Monitoring Server when the usage of the Global Location Broker default algorithm is not supported. It is a list of fully qualified, semicolon delimited network names such as: `KPX_WAREHOUSE_LOCATION= family_protocol: #network_address[port_number]; ...`
KPX_WAREHOUSE_REGCHK	Number of minutes to wait between rechecking the Global Location Broker for any warehouse proxy agent registration changes. The default value is set to 60 minutes.
KSH_DIRECTORY	Specifies the Tivoli Enterprise Monitoring Server SOAP Server HTML files directory.
KT1_TEMS_SECURE	YES is required for the tacmd commands, **putfile**, **getfile**, and **executecommand**. Also required when **executeaction** is issued with advanced options. Not required when issuing **executeaction** to agent without advanced options.
MSG_MODE	Specify kms if you want to use IBM Tivoli Monitoring Operations Logging. Specify MSG2 to use MSG2 logging.
NLS1_LOCALEDIR	The directory of the locale file.
RKDSCATL	Specifies the Tivoli Enterprise Monitoring Server catalogue (CAT) files directory.
SQLLIB	Specifies the Tivoli Enterprise Monitoring Server seeding and query (SQL) files directory.
TIMEOUT	Specifies the time in seconds that Agent Deployment tool has to complete a task. If the tool does not complete in the task in the time specified by the TIMEOUT value, the task is terminated. The default value is 600 seconds.
KMS_SDA	N disables the self-describing agent capability and Y enables it. By default, the self-describing agent capability is disabled at the hub monitoring server. Self-describing agent environment variables are enabled by default for remote monitoring servers, portal servers, and self-describing agent enabled agents. However, these components disable self-describing agent capability if connected to a hub monitoring server that has the self-describing agent capability disabled. Enabling or disabling the self-describing agent capability at the hub monitoring server controls the capability across all components, but you can disable the capability individually at a remote monitoring server, portal server, or agent. You must restart the hub monitoring server for a change in the KMS_SDA variable to take effect. **Note:** Do not specify `YES` or `NO`; instead, always specify `Y` or `N`.

Table 170. Tivoli Enterprise Monitoring Server environment variables (continued)

Variable	Purpose
KMS_SDM_HUB_WAIT_TIMEOUT	The number of seconds that the Remote Tivoli Enterprise Monitoring Server waits for the hub monitoring server to complete a self-describing agent installation request initiated by this Remote Tivoli Enterprise Monitoring Server. If the hub monitoring server does not complete the self-describing agent installation request within this time interval, the Remote Tivoli Enterprise Monitoring Server gives up waiting for completion and returns a status of KFASDM_ST_Server_TimedOut to the agent. One or more self-describing agents will continue to retry the self-describing agent installation for this product until the hub monitoring server product installation is confirmed. A valid value is a number in seconds greater than or equal to 180. The default value is 180 seconds. This parameter is an internal parameter that you might use, in a rare situation, to fine-tune the self-describing agent capability. In general, accept the default value of this variable.
KMS_TAPPLOG_MAX_QUEUED_EVENTS	Do not change this value unless directed to do so by IBM Software Support. This variable determines the maximum number of application support installation event notifications that are stored in the TAPPLOGT notification queue. Modifying this value might result in a loss of self-describing agent installation change notices and unpredictable self-describing agent results. A valid value is a number greater than or equal to 20. The default value is 100. This parameter is an internal parameter that you might use, in a rare situation, to fine-tune the self-describing agent capability. In general, accept the default value of this variable.
KMS_CONFIG_PROP_USE_CACHE	Do not change this value unless directed to do so by IBM Software Support. When set to Y, this variable directs the Tivoli Enterprise Monitoring Server self-describing agent installation manager to use the local Tivoli Enterprise Monitoring Server internal cache mechanism of the application support installation records table to speed up self-describing agent operations. When set to N, operations against the application support installation records table use only the TEMS KEY1 database I/O routines and bypass the table internal cache. This might slow down self-describing agent operations. This parameter is an internal parameter that you might use, in a rare situation, to fine-tune the self-describing agent capability. In general, accept the default value of this variable.
TEMS_JAVA_BINPATH	This variable locates the JAVA installation path within the z/OS USS environment. For more information, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

Table 170. Tivoli Enterprise Monitoring Server environment variables (continued)

Variable	Purpose
TEMS_MANIFEST_PATH=*file_loc*	The location where you want the monitoring server to store the manifest and JAR files it collects from the self-describing agents. You must set this parameter to enable the self-describing capability. This parameter is normally set during component installation. Use the following steps to set the location where you want the monitoring server to store the manifest and JAR files: • On Windows: 1. On the computer where the monitoring server is installed, click **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services.** 2. Right-click **Tivoli Enterprise Monitoring Server** and click **Advanced→ Edit Variables**. 3. When you are asked if you want to stop the service, answer **Yes**. 4. The default file location that is set for **TEMS_MANIFEST_PATH** is `ITMinstall_dir\CMS\support` where *ITMinstall_dir* is the directory where you installed the product. You can change this location if you want to do so. • On Linux or UNIX: 1. On the computer where the monitoring server is installed, change to the `ITMinstall_dir/config/ directory`. 2. Open the `tems_hostname_ms_tems_name.config` file. 3. The default file location that is set for **TEMS_MANIFEST_PATH** is `ITMinstall_dir/tables/<TEMS_NAME>/support` where *ITMinstall_dir* is the directory where you installed the product. You can change this location if you want to do so.

Universal Agent environment variables

The following table lists environment variables you can set to customize tracing for the Universal Agent.

Table 171. Universal Agent environment variables

Variable	Purpose
KBB_RAS1=ERROR (UNIT:kumamain ALL)	Problems involving managed system online/offline processing.
KBB_RAS1=ERROR (UNIT:kumpdpda Error Output) (UNIT:kumpmd2a Error Detail)	Incorrect report data.
KBB_RAS1=ERROR (UNIT:kumpfile Error State Detail Flow Metrics) (UNIT:kumpdcmf ALL)	Detailed File Data Provider tracing.
KBB_RAS1=ERROR (UNIT:kumpsosr ALL) (UNIT:kumpspst ALL) (UNIT:kumpscku ALL) (UNIT:kumpstcp ALL) (UNIT:kumplpba ALL)	Detailed API or Socket Data Provider tracing.
KBB_RAS1=ERROR METRICS	Problems involving Universal Agent memory usage.
KUMA_DCH_TRAPEMIT	SNMP Emitter tracing. Use to display emitted traps in the UAGENT Action report.

Table 171. Universal Agent environment variables (continued)

Variable	Purpose
KUMA_VERBOSE	Tracing a Universal Agent API client program. This variable must be set on the system where the API client program is executing, not where Universal Agent is running) Example: KUMP_API_VERBOSE=Y dpapi.log. The KUMP_API_VERBOSE option is valuable when debugging an API program that communicates with the Universal Agent API data provider.
KUMP_HTTP_DEBUG	HTTP Data Provider tracing.
KUMP_ODBC_DEBUG	ODBC Data Provider tracing.
KUMP_SCRIPT_DEBUG	Script Data Provider tracing.

KUMP_SNMP_DEBUG_TRAP
KUMP_SNMP_DEBUG_DISCOVERY_ROUTE
KUMP_SNMP_DEBUG_DISCOVERY_NETWORK
KUMP_SNMP_DEBUG_MIB_MANAGER
KUMP_SNMP_DEBUG_MIB_IO
	SNMP Data Provider tracing. All of the debug environment variables listed default to No. As an example, if you use the SNMP Data Provider and have problems collecting MIB data, you set these two environment variables:
	KUMP_SNMP_DEBUG_MIB_MANAGER=Y
	KUMP_SNMP_DEBUG_MIB_IO=Y

For a complete list of the Universal Agent environment variables, see the *IBM Tivoli Universal Agent User's Guide*.

Tivoli Data Warehouse environment variables

The following Warehouse Proxy agent parameters can be changed in the `KHDENV` file on Windows systems and the `hd.ini` file in systems other than Windows systems to customize tracing for the Warehouse Proxy agent.

Table 172. Warehouse proxy environment variables

Variable	Purpose
CTIRA_NCSLISTEN	The number of RPC threads.
KHD_BATCH_USE	Allows inserts to be batched together, reducing the number of transactions against the database, improving performance and reducing network traffic. Default is Y and can be edited by GUI or CLI.
KHD_CNX_POOL_SIZE	The total number of preinitialized ODBC connection objects available to the work queue export threads. The default value should match the KHD_EXPORT_THREADS value.
KHD_CNX_WAIT	The time in minutes to wait before trying to reconnect. Default is 10 minutes.
KHD_CNX_WAIT_ENABLE	A time to wait before a retry. Default is Y. Changing this variable to N does not wait before retries. Setting this variable to N can generate a large log file if the tests to the database fail at each retry. It is advisable not to change this variable.
KHD_EXPORT_THREADS	The number of worker threads exporting data to the database. The default value should match the KHD_CNX_POOL_SIZE value.
KHD_QUEUE_LENGTH	The length of the KHD work queue. This integer identifies the maximum number of export work requests that can be placed on the work queue before the queue starts rejecting requests. The default value of KHD_QUEUE_LENGTH is 1000. Setting this value to 0 means the queue length has no limit.

Table 172. Warehouse proxy environment variables (continued)

Variable	Purpose
KHD_SRV_STATUSTIMEOUT	The time in seconds set by default to 600s = 10 minutes. Set KHD_SRV_STATUSTIMEOUT less than KHD_STATUSTIMEOUT by at least 60 seconds. It is advisable not to change this variable.
KHD_WAREHOUSE_TEMS_LIST	A space or coma-delimited separated list of monitoring servers. A warehouse proxy agent serves all the agents reporting to those monitoring servers listed. The same monitoring server name must not appear more than once in all the warehouse proxy Tivoli Enterprise Monitoring Server lists in the entire enterprise environment.
KHD_DB_COMPRESSION	Used to change the configuration of the Warehouse Proxy Agent to enable or disable database compression.
KHD_WHLOG_ENABLE	Used to change the configuration of the Warehouse Proxy Agent to enable or disable data warehouse log tables. This variable is set to N by default.
KHD_SERVER_DIST_ COMPRESSION_ENABLE	Used to send distributed agent compressed data. This variable is set to Y by default.
KHD_SERVER_Z_ COMPRESSION_ ENABLE	Used to allow z/OS clients to send compressed data. This variable is set to N by default.
KHD_CLIENT_ COMPRESSION_ ENABLE	Specifies that historical data is not compressed if set to N, which is true even if the Warehouse Proxy Agent server has allowed the compression. This variable does not exist by default.

Table 173 lists the Summarization and Pruning agent parameters that you can edit in the `KSYENV` file on Windows systems and the `SY.ini` file on non-Windows systems to customize tracing for the Summarization and Pruning Agent.

Table 173. Summarization and Pruning agent environment variables

Variable	Purpose
KSY_BLACKOUT	Specifies a comma-separated list of exception times where the Summarization and Pruning agent should not start when using the flexible scheduling. The values in the list should be in the format HH:MM-HH:MM where HH must be between 00 and 23 and MM must be between 00 and 59. The starting time must be smaller than the end time of the exception period. For example, to block the Summarization and Pruning agent from starting between 1 and 2 AM and 5 and 6 PM, use the following time statement: 01:00-01:59,17:00-17:59.
KSY_CNP_SERVER_HOST	Tivoli Enterprise Portal Server connection default host.
KSY_CNP_SERVER_PORT	Tivoli Enterprise Portal Server connection default port.
KSY_DAY_AGE_UNITS	The minimum age of data for daily data in days before aggregation is done.
KSY_EVERY_N_DAYS	How often to run the schedule in days (number).
KSY_EVERY_N_MINS	Indicates the frequency between Summarization and Pruning agent runs when using the flexible scheduling. Must be a multiple of 15 minutes, with a minimum of 15 and a maximum of 1440 (one day).
KSY_FIXED_SCHEDULE	Indicates whether the Summarization and Pruning agent is configured for fixed schedule if set to Y or flexible schedule when set to N.
KSY_HOUR_AGE_UNITS	The minimum age of data for hourly data in hours before aggregation is done.
KSY_HOUR_AM_PM	Whether to run in the AM or PM (AM/PM).
KSY_HOUR_TO_RUN	The hour of the day to run. Valid values are 1-12.

Table 173. Summarization and Pruning agent environment variables (continued)

Variable	Purpose
KSY_MAX_ROWS_PER_ TRANSACTION	Maximum rows per database transaction.
KSY_MAX_WORKER_THREADS	Maximum number of simultaneous worker threads. Default is 1. Recommended value is the number of processors on your server minus 1.
KSY_MINUTE_TO_RUN	The minute of the day to run (number).
KSY_SHIFT1_HOURS	A comma-separated list of off-peak hour numbers for the shift.
KSY_SHIFT2_HOURS	A comma-separated list of peak hour numbers for the shift.
KSY_SHIFTS_ENABLED	Shift periods. Only two shifts are allowed. If shifts are enabled, each hour (0-23) must be specified once.
KSY_START_OF_WEEK_DAY	Start of the week day, for example: 0 = Sunday 1 = Monday.
KSY_TIMEZONE_IND	Time zone indicator. AGENT specifies use the time zone offset of the agent. WAREHOUSE specifies use the time zone offset of the warehouse .
KSY_VACATION_DAYS	Vacation days in a comma-separated list of days in YYYYMMDD format.
KSY_VACATIONS_ENABLED	Whether vacation is enabled with Y or N.
KSY_WAREHOUSEAGGREGLOG_PRUNE	Use to specify the pruning for the WAREHOUSEAGGREGLOG table. The format of the value should be *number.unit*, where *number* is the number of units to retain and *unit* specifies the retention unit, which can be one of day, month or year.
KSY_WAREHOUSELOG_PRUNE	Use to specify the pruning for the WAREHOUSELOG table. The format of the value should be *number.unit*, where *number* is the number of units to retain and *unit* specifies the retention unit, which can be one of day, month or year.
KSY_WAREHOUSE_DRIVER	The Warehouse Database Connection JDBC Driver.
KSY_WAREHOUSE_PASSWORD	The Warehouse Database Connection Password (encrypted).
KSY_WAREHOUSE_URL	The Warehouse Database Connection JDBC URL.
KSY_WAREHOUSE_USER	The Warehouse Database Connection User (encrypted).
KSY_WEEKENDS_AS_VACATIONS	Whether weekends are vacation with Y or N.
KSZ_JAVA_ARGS	Java arguments.
KSY_WHLOG_ENABLE	Used to change the configuration of the Summarization and Pruning Agent to enable or disable data warehouse log tables. The default value is N.
KSY_DB_COMPRESSION	Used to change the configuration of the Summarization and Pruning Agent to enable (Y) or disable (N) database compression.

Common agent environment variables

The following environment variables are used by all IBM Tivoli Monitoring agents.

Table 174. Common agent environment variables

Variable	Purpose
CMS_MSGBASE	Applies only to the i5/OS platform. Specifies the MSG2 message file for agent framework messages.
CT_CMSDIRECT	*Obsolete.* Replaced by IBM Tivoli Monitoring V6 firewall communications services. Specify the full NAT firewall address of the monitoring server to connect to, including *protocol:address[port#]*.

Table 174. Common agent environment variables (continued)

Variable	Purpose
CT_CMSLIST	Required variable that specifies the primary or secondary Tivoli Enterprise Monitoring Server the agent must connect with. Takes a list of monitoring server names in the form *network_protocol:hostname* or *network_protocol:address* and delimited by semicolons.
CTIRA_CELL_NAME	*Obsolete.* Replaced by the **CT_CMSLIST** agent configuration variable.
CTIRA_HEARTBEAT	The interval, in minutes, of the agent-to-monitoring server heartbeat data sample. The default value is 10 minutes. Shorter heartbeat intervals increase network traffic between the agent and the Tivoli Enterprise Monitoring Server.
CTIRA_HIST_DIR	Required variable that specifies the directory where agent-based short-term history data files are stored. Does not apply to the Tivoli Enterprise Monitoring Server's short-term history data files. The directory must already exist.
CTIRA_HOSTNAME	Used by many, but not all, agents to provide an alternate host name qualifier (*subsystem:hostname:nodetype*) for the published agent managed system name. Used to remove a long network domain name (such as **acme.us.com**) from the default agent hostname. Not honored by all agents. For some agents, **CTIRA_HOSTNAME** might cause unpredictable results in the Tivoli Enterprise Portal navigation tree.
CTIRA_IP_PORT	*Applies to z/OS agents only. Do not modify.* Set this variable to 0 so the operating system can provide the agent RPC listen port, which prevents a port conflict for some z/OS configurations.
CTIRA_LOG_PATH	Required variable that specifies the directory where the agent's Operations Log file is stored. The file names themselves use the suffixes `.LG0` and `.LG1`. This variable does not apply to agents running on a z/OS system.
CTIRA_MAX_RECONNECT_ TRIES	*Obsolete.* Number of consecutive unsuccessful attempts the agent makes to connect to a Tivoli Enterprise Monitoring Server before giving up and exiting. The default value of 0 means that the agent remains started regardless of its connection status with the monitoring server. (Before IBM Tivoli Monitoring V6.2.2, the default value was 720.)
CTIRA_NCSLISTEN	Number of RPC listen server threads to create for the agent. The default value is 10.
CTIRA_NODETYPE	Supplies the agent node type qualifier (*subsystem:hostname:nodetype*) of the agent managed system name (*msn*). Provide the agent product indicator in this name. This value is set by the application and should not be set by the user.
CTIRA_OS_INFO	Overrides the default value for agent entries in the Tivoli Enterprise Monitoring Server's "ManagedSystem.Host_Info" column. This variable is used to build the navigation tree for the Tivoli Enterprise Portal Server. The value must match an existing entry in the **CNPS/osnames** file. This variable is not applicable to subnode type records in the monitoring server's ManagedSystem table.
CTIRA_PRIMARY_FALLBACK_ INTERVAL	Forces the agent to switch from the primary Tivoli Enterprise Monitoring Server to one of the defined secondary monitoring servers, because the primary monitoring server is offline or due to network connectivity issues. You want the agent to switch back to the primary monitoring server as soon as possible when it becomes available. This parameter controls the frequency with which the agent performs a lookup of the primary monitoring server. If the primary monitoring server is found, the agent disconnects from the secondary monitoring server and reconnects to the primary monitoring server. A value of zero disables this feature. The minimum value must be 2.5 times `CTIRA_RECONNECT_WAIT` value. The default value is 4500 seconds, or 75 minutes.
CTIRA_PRODUCT_SEP	Supplies an alternate qualifier for the agent's managed system name (*msn*). The default value is a colon (:).
CTIRA_RECONNECT_WAIT	Time interval, in seconds, that the agent waits between attempts to register with a Tivoli Enterprise Monitoring Server. The default value is 600 seconds.

Table 174. Common agent environment variables (continued)

Variable	Purpose
CTIRA_REFLEX_ATOMIC	For subnode targets only. Evaluates the situation state by any existing specified display item column name when deciding which reflex situation automation command the agent should execute. Not applicable to reflex situation commands executed or evaluated by the Tivoli Enterprise Monitoring Server. Disable by setting to **N**. The default value is **Y**.
CTIRA_REFLEX_TARGET	For subnode targets only. Evaluates situation state by subnode name value in the ORIGINNODE column when deciding which reflex situation automation command the agent should execute. Not applicable to reflex situation commands executed or evaluated by the Tivoli Enterprise Monitoring Server. Disable by setting to **N**. The default value is **Y**.
CTIRA_SIT_CLEAN	Number of seconds between garbage cleanup of stale entries in the agent persistent situation file. The default value is 900 seconds.
CTIRA_SIT_FILE	Specifies an alternate name for the default agent-based persistent situation file. This variable should be used only in exceptional conditions because the file name reflects the agent's managed system name. Unsupported for z/OS-based agents.
CTIRA_SIT_MGR	Specifies whether or not to use the agent's persistent situation file when registering with the Tivoli Enterprise Monitoring Server. Using this file improves performance of the monitoring server. Set this variable to **N** to disable usage. For a z/OS agent, the value must be **N**, because this feature is not implemented for a z/OS-based monitoring server. For all other platforms, the default value is **Y**.
CTIRA_SIT_PATH	Required variable that specifies the directory where the agent-based persistent situation file is stored. This agent-only file contains a copy of the Tivoli Enterprise Monitoring Server monitoring situations for the agent's use while registering with the monitoring server. The file is named `psit_`*`msn`*`.str`, where *msn* is the managed system name of the agent process. Unsupported for z/OS-based agents.
CTIRA_SUBSYSTEM_ID	Optional variable that overrides the subsystem ID qualifier (*subsystem:hostname:nodetype*) of the agent managed system name (*msn*). Describes a monitored resource instance to help make this name unique. Value may also be set by the agent itself.
CTIRA_SYSTEM_NAME	Sets an alternate system name for agent entries in the Tivoli Enterprise Monitoring Server's ManagedSystem.Host_Address column within the **<NM>mysystem</NM>** tags. Used to build the Tivoli Enterprise Portal Server's navigation tree. Not applicable to subnode type records in the monitoring server's ManagedSystem table. The maximum allowable length of the value used for this variable is 31 characters.
CTIRA_THRESHOLDS	Specifies the fully qualified name of the XML-based adaptive (dynamic) threshold override file. The default file is located in `$CANDLE_HOME/localconfig/`*`pc`*`/`*`pc`*`_thresholds.xml`, where *pc* is the agent product code. On z/OS systems, the default file name is *pc* THRES.

Table 174. Common agent environment variables (continued)

Variable	Purpose
IRA_ADAPTIVE_THRESHOLD_ MODE	Specifies the adaptive (dynamic) threshold operation mode, either **CENTRAL** or **LOCAL**. In **CENTRAL** mode, threshold overrides are centrally created and distributed to the agent. Do not modify the override XML file, which is the default mode. In **LOCAL** mode, central distribution to the agent is inhibited, and threshold overrides are locally created and managed. Use **LOCAL** mode to specify that the agent should ignore enterprise distribution; its affinity will not be registered, so the Tivoli Enterprise Portal cannot override the agent's managed system node. Use this mode cautiously because it causes the Tivoli Enterprise Monitoring Server's thresholds and the agent's thresholds to be out of sync. You must create and manually write override definitions in the same file that is created in **CENTRAL** mode: `*managed-system-name_product-code*_thresholds.xml`. For instance, on Windows, this file is named `Primary_*myagent*_NT_thresholds.xml`; on Linux, `*myagent*_LZ_thresholds.xml`; on UNIX, `*myagent*_UX_thresholds.xml`. On Windows, the file is stored in the `%CANDLEHOME%\TMAITM6` directory; on Linux and UNIX, the file is stored in `$CANDLEHOME/*interp*/*product-code*/bin`. The names of the columns to be used when specifying overrides is taken from the attributes file. The override name and **objectId** must be unique in the XML file. **Timestamp** is not required. If later you switch back from **LOCAL** mode to **CENTRAL** mode, centrally located overrides will again override the local definitions.
IRA_AUTONOMOUS_LIMIT	Sets the saved event limit for autonomous operation. If the specified value is a number (for example, 500), then it is the maximum number of situation event records to be saved by the agent. If the specification is in common disk space units such as KB, MB, or GB (for example, 5 MB), then it is the total amount of disk space to be used by the agent for saving situation event data. The default value is 2 MB.
IRA_AUTONOMOUS_MODE	Turns on (**YES**) or off (**NO**) agent autonomous mode. While in autonomous mode, the agent continues to run Enterprise situations. The situation event data persists on disk even after agent restart. On reconnection to the Tivoli Enterprise Monitoring Server, the agent uploads saved situation event data to the monitoring server. The default value is **Y**.
IRA_DEBUG_AUTONOMOUS	Turns on (**Y**) or off (**N**) debug trace for agent autonomous operation. This variable is used for diagnosing problems in this area. The default value is **N**.
IRA_DEBUG_EVENTEXPORT	Turns on (**Y**) or off (**N**) agent event export operations, such as SNMP trap and debug trace. The default value is **N**.
IRA_DEBUG_PRIVATE_ SITUATION	Turns on (**Y**) or off (**N**) debug trace when processing the private situations of an agent. The default value is **N**.
IRA_DEBUG_SERVICEAPI	Turns on (**Y**) or off (**N**) debug trace for agent service interface processing. The default value is **N**.
IRA_DEBUG_TRANSCON	Turns on (**Y**) or off (**N**) debug trace for agent transport conduit processing. This variable is used for diagnosing problems in this area. The default value is **N**.
IRA_DEBUG_EIF	Turns on (**Y**) or off (**N**) debug trace for the EIF event emitting feature in this agent.
IRA_DUMP_DATA	Used by both agents and the Tivoli Enterprise Monitoring Server for debugging. Set to **Y** to do a hexadecimal dump of RPC transaction content data contents into the RAS1 log. The default value is **N**. Can produce voluminous RAS1 message output if enabled.

Table 174. Common agent environment variables (continued)

Variable	Purpose
IRA_EVENT_EXPORT_ CHECKUSAGE_INTERVAL	Specifies the frequency with which the agent checks and calculates the autonomous operation saved event usage limit that is defined by the **IRA_AUTONOMOUS_LIMIT** parameter. The default value is 90 seconds; the agent enforces the minimum setting of 30 seconds.
IRA_EIF_ENABLE_LOG	Enables logging of EIF events sent for private situations directly from the Tivoli Enterprise Monitoring Agent to the agent's *.LG0 operational log if set to "Y".
IRA_EVENT_EXPORT_EIF	Enables (**Y**) or disables (**N**) the EIF event emitting feature for any private situations configured for this agent.
IRA_EVENT_EXPORT_ LISTSTAT_INTERVAL	Defines the frequency with which the agent outputs collected situation statistics to the debug trace log. The default value is 900 seconds, or 15 minutes.
IRA_EVENT_EXPORT_ LISTSTAT_OUTPUT	Enables (**Y**) or disables (**N**) periodic output of situation operation statistics data to the trace log. The default is **N**.
IRA_EVENT_EXPORT_SIT_ STATS	Enables (**Y**) or disables (**N**) basic situation operation statistics data collection. The basic situation data includes situation first start time, situation first event time, situation last start time, situation last stop time, situation last true event time, situation last false event time, number of times situation recycles, number of times situation enters autonomous operation. The default value is **Y**.
IRA_EVENT_EXPORT_SIT_ STATS_DETAIL	Enables (**Y**) or disables (**N**) details situation operation statistics data collection. The detail data collected includes 8 days' situation operation profile such as hourly true event count, hourly false event count, hourly data row count, hourly true event ratio, and hourly false event ratio. The default value is **N**.
IRA_EVENT_EXPORT_SNMP_ TRAP	Enables (**Y**) or disables (**N**) the SNMP trap emitter capability. When enabled, the SNMP trap configuration file is required and must exist for the agent to emit SNMP V1, V2, or V3 trap to configured SNMP trap managers. The default value is **Y**.
IRA_EVENT_EXPORT_SNMP_ TRAP_CONFIG	Specifies the fully qualified SNMP trap configuration file name. The default file is located in `$CANDLE_HOME/localconfig/`*`pc`*`/`*`pc`*`_trapcnfg.xml` (member *`pc`*`TRAPS` on z/OS systems), where *pc* is the agent product code.
IRA_LOCALCONFIG_DIR	Specifies the local configuration directory path that contains locally customized configuration files such as threshold overrides, private situations, and SNMP trap configuration file. The default directory is the `localconfig` subdirectory of the directory specified by the **CANDLE_HOME** environment variable, which is the RKANDATV DD name on z/OS systems.
IRA_PRIVATE_SITUATION_ CONFIG	Specifies the fully qualified autonomous Private Situation configuration file name. The default file on distributed systems is located in `$CANDLE_HOME/localconfig/`*`pc`*`/`*`pc`*`_situations.xml`, where *pc* is the agent product code. The default file on z/OS systems is the SICNFG member in the RKANDATV data set.
IRA_SERVICE_INTERFACE_ DEFAULT_PAGE	Instructs the agent to open the named product-specific HTML page instead of the default `navigator.htm` page when logging on to the agent service interface. By default, the agent looks for the product-specific file in `CANDLE_HOME/localconfig` on distributed systems and the `RKANDATV` dataset on z/OS systems. However, if the **IRA_SERVICE_INTERFACE_DIR** environment variable has been set, the agent looks in the directory specified by that environment variable. If you set **IRA_SERVICE_INTERFACE_DEFAULT_PAGE** (but not **IRA_SERVICE_INTERFACE_DIR**), you should put any product-specific HTML pages in the `CANDLE_HOME/localconfig/html` directory on distributed systems. Therefore, if you create `myPage.htm` and put it in `CANDLE_HOME/localconfig/html` set **IRA_SERVICE_INTERFACE_DEFAULT_PAGE=/html/myPage.htm**.

Table 174. Common agent environment variables (continued)

Variable	Purpose
IRA_SERVICE_INTERFACE_DIR	Defines the path specification of the agent service interface HTML directory. In conjunction with the **IRA_SERVICE_INTERFACE_DEFAULT_PAGE** parameter, the agent constructs the file path to a specific, requested HTTP GET object. The default filepath is `CANDLE_HOME/localconfig` on distributed systems and the RKANDATV dataset on z/OS systems. The parameter is equivalent to the **IRA_HTML_PATH** parameter. Example: If **IRA_SERVICE_INTERFACE_DIR="\mypath\private"** and you enter `http://localhost:1920///kuxagent/kuxagent/html/myPage.htm` in your browser, `myPage.htm` is retrieved from `\mypath\private\html\` instead of `%CANDLE_HOME%\localconfig\html\`.
IRA_SERVICE_INTERFACE_NAME	Specifies a unique agent interface name to represent this agent. The default agent service interface name is *pc* `agent`, where *pc* is the application product code. In the scenario where multiple instances of the same agent are running in the system, this parameter enables customization of a unique service interface name to correspond to this agent.

ITM_MANIFEST_PATH=*file_loc*	The location of the self-describing agent's manifest and JAR files. Normally this parameter is set during agent installation and needs to be changed only if the files are moved to a network file system.
	File locations:
	• On Windows:
	`<inst_dir>\TMAITM6\support\<pc>`
	`<inst_dir>\TMAITM6_x64\support\<pc>`
	• On Linux or UNIX:
	`<inst_dir>/<binarch>/<pc>/support`
	where pc is the two-character product code.
	This parameter is an internal parameter that is set and used by the agent installers only. You should not change this variable.

KBB_RAS1	Sets the level of agent tracing: **`ERR (UNIT:KRA ST)`** View the state of main agent functions such as situation and report processing. **`ERR (UNIT:KRA ALL)`** View detailed debug messages for agent functions. **`ERR (UNIT:KHDX ST)`** View the state of the agent's short-term history data uploads to the Tivoli Data Warehouse. **`ERR (UNIT:KHD ALL)`** View detailed debugging messages about short-term history data uploads to the Tivoli Data Warehouse.
KCA_CACHE_LIMIT	Maximum time (in hours) that an alert is cached and seen in the **Alerts** view of the Agent Managent Service (AMS) workspace. **Possible values:** Greater than 0. **Default value:** 24.

Table 174. Common agent environment variables (continued)

Variable	Purpose
KCA_CAP_DIR	Directories looking for CAP files. **Possible values:** Any valid PATH. **Default value:** • On Windows systems: %CANDLE_HOME%\TMAITM6_x64\CAP;@BinPath@\CAP;%ALLUSERSPROFILE%\Application Data\IBM\CAP • On Linux or UNIX: $CANDLEHOME$/config/CAP:/opt/IBM/CAP
KCA_CAP_DISCOVERY_INTERVAL	Time interval (in seconds) for discovering new or changed CAP files. **Possible values:** Greater than 0. **Default value:** 30.
KCA_CMD_TIMEOUT	Timeout (in seconds) for external commands created by PAS. **Possible values:** Greater than or equal to 90. **Default value:** 90.
KCA_DISCOVERY_INTERVAL	How frequently the watchdog polls for new, running instances of agents managed by Agent Management Services. **Possible values:** Greater than 0. **Default value:** • On Windows systems: 30 • On Linux or UNIX: 120
KCA_IP_DIR	Only for Linux and UNIX; directory for creating sockets to accept external commands. **Possible values:** Any valid existing PATH. **Default value:** $CANDLEHOME$/$BINARCH$/$PRODUCTCODE$/bin/pasipc **Note:** Some systems can have a problem managing sockets with long path names. When installing the agent on a CANDLEHOME path that is longer than 50 characters you should set KCA_IP_DIR to a value shorter than the default value. For example KCA_IP_DIR=/tmp/pasic.
KCA_ITM_DISCOVERY_INTERVAL	Time interval (in seconds) for discovering new unmanaged agent instances of IBM Tivoli Monitoring agents (cinfo command). **Possible values:** Greater than KCA_DISCOVERY_INTERVAL. **Default value:** 600.
KCA_MAX_RETRIES_ON_PIPE	Maximum number of consecutive failed attempts to check the availability of the OS agent through the PAS internal socket (pipe). When KCA_MAX_RETRIES_ON_PIPE is reached, the OS agent is restarted. An undefined variable, or less than 0, means an infinite number of retries (no restart). **Possible values:** Greater than 5. **Default value:** Not defined.

Table 174. Common agent environment variables (continued)

Variable	Purpose
KCAWD_WRITE_DIR	Writable directory for kcawd (WatchDog agent). **Possible values:** Any valid directory. **Default value:** The installation directory of the OS agent binary.
KHD_HISTSIZE_EVAL_ INTERVAL	Controls how often (in seconds) the Warehouse Proxy Agent client code checks the size of the historical files. For more information, see the *IBM Tivoli Monitoring: Administrator's Guide*.
KHD_HISTRETENTION	Specifies the default retention period in hours for the short-term history files (default is 24 hours). This value can be used to reduce the amount of data kept on disk after a successful upload to the warehouse is performed.
KHD_STATUSTIMEOUT	The time in seconds set by default to 900s = 15 minutes. An export request on the application agent is resent if a status is not received from the Warehouse Proxy agent before the timeout expires. It is advisable not to change this variable.
KHD_TOTAL_HIST_MAXSIZE	Controls the threshold that the Warehouse Proxy Agent client code uses when checking the size of the historical files. When their combined size is over the specified value in MB, writing to them is suspended until the threshold is no longer exceeded. A value of 0 disables the periodic checking. For more information, see the *IBM Tivoli Monitoring: Administrator's Guide*.
TIMEOUT	Specifies the time in seconds that Agent Deployment tool has to complete a task. If the tool does not complete in the task in the time specified by the TIMEOUT value, the task is terminated. The default value is 600 seconds.
TEMA_SDA	N disables the self-describing agent capability at the agent, whereas Y enables it. A value of N blocks the Tivoli Enterprise Monitoring Server from retrieving any product support files from this agent and provides you with control on a per agent basis without stopping the self-describing agent feature on the Tivoli Enterprise Monitoring Server for other products. **Note:** Do not specify `YES` or `NO`; instead, always specify `Y` or `N`.
TEMA_SDA_ACK_WAIT	Time interval in seconds for the agent to wait on a response from the Tivoli Enterprise Monitoring Server for the completion status of an actual self-describing agent installation. A valid value is a number in seconds greater than or equal to 60. The default value is 300 seconds. This parameter is an internal parameter that you might use, in a rare situation, to fine-tune the self-describing agent capability. In general, accept the default value of this variable.
TEMA_SDA_RETRY_WAIT	Time interval in seconds for the agent to wait before attempting another self-describing agent installation request. Not all failed self-describing agent registration or installation responses from the Tivoli Enterprise Monitoring Server are retried. A valid value is a number in seconds greater than or equal to 60. The default value is 600 seconds. This parameter is an internal parameter that you might use, in a rare situation, to fine-tune the self-describing agent capability. In general, accept the default value of this variable.

Table 174. Common agent environment variables (continued)

Variable	Purpose
TEMA_SDA_MAX_ATTEMPTS	The maximum number of times the agent tries to register a request to initiate a self-describing agent product installation with the Tivoli Enterprise Monitoring Server. The agent will continue to try its self-describing agent registration until the Tivoli Enterprise Monitoring Server self-describing agent manager has completed the installation of this agent's product support. The minimum value is 1. The agent must always try its self-describing agent registration once (unless the TEMA_SDA value is N). This maximum attempt count should only be modified with direction from IBM Software Support. If the value is set too low, your self-describing agent product installation might not complete as expected. The agent must be able to retry its self-describing agent registration until the product installation has completed. In the event of an agent attempting excessive self-describing agent product install retry attempts, it might generate excessive or unwanted audit or error messages that you want to stop. Decreasing this number can help in such a case. This parameter is an internal parameter that you might use, in a rare situation, to fine-tune the self-describing agent capability. In general, accept the default value of this variable.

Operating system agent environment variables

This section lists IBM Tivoli Monitoring OS agent environment variables that you can customize.

Windows OS monitoring agent

The following table lists the Windows OS monitoring agent environment variables.

Table 175. Windows OS monitoring agent environment variables

Variable	Description	Possible values	Default value
NT_EXCLUDE_PERF_OBJS	To explicitly exclude collection of metrics from the specified Performance Objects.	List of comma-separated strings.	Blank
NT_EXCLUDE_UNMAPPED_DISKS	To avoid reporting of unmapped hard disks among Logical Disks (having instance name not mapped to a letter yet).	0 or 1.	0
NT_LOG_DUPLICATE	To enable reporting of the Duplicate Record Count for duplicated events in the event log.	0 or 1.	0
NT_LOG_MAX_EVTS	To define the max number of past events to be searched in the Event Log. 0 means that the missed events feature is disabled.	0, or greater than or equal to 1.	0
NT_LOG_MAX_TIME	To define the maximum interval, in seconds, of past events to be searched in the Event Log. 0 means that the missed events feature is disabled.	0, or greater than or equal to 1.	0
NT_LOG_THROTTLE	To drop the specified number of duplicate events every read cycle of the event log. 0 means throttling is disabled.	0, or greater than or equal to 1.	0
NT_PERFMON_MEMORY_CHECK	To disable the exclusion of Performance Objects that are leaking memory, as a result of the check performed by the agent at initialization time.	0 or 1.	1

Table 175. Windows OS monitoring agent environment variables (continued)

Variable	Description	Possible values	Default value
REVERSE_LOOKUP_ ACCEPTED_FAILURES	To avoid long Network Port attribute group response times due to reverse lookup failures when retrieving Local and Remote Names from corresponding IP addresses. 0 means that reverse lookup is disabled.	0, or greater than or equal to 1.	30

Linux OS monitoring agent

The following table lists the Linux OS monitoring agent environment variables.

Table 176. Linux OS monitoring agent environment variables

Variable	Description	Possible values	Default value
KBB_NFS_TIMEOUT	To adjust the timeout on NFS file systems monitoring (in seconds).	1 - 30.	2
KBB_SHOW_NFS	Specifies whether NFS monitoring is enabled. The default value is false on Linux and UNIX systems.	[true|false], not case sensitive.	false, not case-sensitive
KDEBE_FIPS_MODE_ ENABLED	To request GSKit compliance to FIPS 140-2 standards when computing checksums for files in the File Information attribute group.	[yes|no], not case-sensitive.	no, not case-sensitive
KLZ_CPUSTAT_SAMPLE_ SECS	To tune the sampling interval for the CPU attribute group (in seconds).	Greater than or equal to 5.	30
KLZ_DISK_SAMPLE_HRS	To tune the sampling interval for the Disk Usage Trends attribute group (in seconds).	Greater than or equal to 1.	3600
KLZ_IOSTAT_SAMPLE_ SECS	To tune the sampling interval for the Disk IO attribute group (in seconds).	Greater than or equal to 5.	30
KLZ_NETSTAT_SAMPLE_ SECS	To tune the sampling interval for the Network attribute group (in seconds).	Greater than or equal to 5.	30
KLZ_PINGHOSTLIST	To set the full qualified path of the file containing the list of servers for the Host Availability attribute group.	A valid UNIX pathname.	Blank
KLZ_PROCESS_SAMPLE_ SECS	To tune the sampling interval for the Process Instant Busy CPU (Percent) attribute in the Process group (in seconds).	0, or greater than or equal to 5.	60
KLZ_PROCNORM_TEST	To enable Process short-term CPU average attributes also for Queries.	[y|n], not case-sensitive.	n, not case-sensitive
KLZ_SETLPARVMID	To set the CTIRA_HOSTNAME parameter for the agent as Lparname.VMname:LZ (Linux 390 systems only).	[y|n], not case-sensitive.	n, not case-sensitive
KLZ_SKIP_DISABLED_ DISKS	To avoid the OS agent reporting on SAN Physical Disks that are not available to the system.	[TRUE|true|True] or false, not case-sensitive.	false, not case-sensitive
KLZ_SWAPTREND_ SAMPLE_HRS	To tune the sampling interval for the Swap Rate attribute group (in seconds).	Greater than or equal to 1.	3600
KLZ_SYSSTAT_SAMPLE_ SECS	To tune the sampling interval for the System Statistics attribute group (in seconds).	Greater than or equal to 5.	30
KLZ_TCPSTAT_SAMPLE_ SECS	To tune the sampling interval for the TCP Statistics attribute group.	Greater than or equal to 5.	30

Table 176. Linux OS monitoring agent environment variables (continued)

Variable	Description	Possible values	Default value
UID_USERNAME_ REFRESH_TIME	To tune the refresh interval of uid<->username and gid<->groupname mapping tables (in minutes).	Greater than or equal to 1.	60

UNIX OS monitoring agent

The following table lists the UNIX OS monitoring agent environment variables.

Table 177. UNIX OS monitoring agent environment variables

Variable	Description	Possible values	Default value
AGENT_TIMING	To enable agent internal timers.	Y	Y
EXCLUDE_LEGACY_ DISKS	To exclude monitoring of disks that follow the HPUX legacy naming convention.	[TRUE|true|True] or false, not case-sensitive.	false, not case-sensitive
KBB_EXCLUDE_SLOW_ ISATTY	To check the response time of the isatty() system call and eventually skip it to avoid agent hangs.	0 or 1.	0
KBB_HPUX_SAR	To force usage of the sar command for CPU statistics measurements on HP-UX systems.	[TRUE|true] or false, not case-sensitive.	false, not case-sensitive
KBB_HPUX_VMSTAT	To force usage of the vmstat command for CPU statistics measurements on HP-UX systems.	[TRUE|true] or false, not case-sensitive.	false, not case-sensitive
KBB_SHOW_CUSTOMFS	To enable monitoring of a specific filesystem type.	Between 0 and 30.	blank
KBB_SHOW_NFS	Specifies whether NFS monitoring is enabled. The default value is false on Linux and false on UNIX.	[true|false], not case-sensitive.	false, not case-sensitive
KUX_AIXDP	To disable the AIX data provider metrics collection (new attributes inherited from the AIX Premium Agent). **Note:** The minimum AIX requirements to collect metrics from UNIX OS monitoring agent and Premium Monitoring Agent for AIX convergence are: • AIX53S = AIX 5.3 TL12 • AIX61F = AIX 6.1 TL5	[true|false], not case-sensitive.	true, not case-sensitive
KUX_DEFINED_ USERS	To enable the Defined Users group on AIX systems.	True or false, not case-sensitive.	false, not case-sensitive
KUX_GETARGS_DELAY	To adjust the interval between subsequent attempts to get Processes command line arguments (milliseconds - AIX systems only).	Between 0 and 1000.	10
KUX_GETPROCS_DELAY	To adjust the sampling interval for Process CPU percent utilization (milliseconds - AIX only).	0, or greater than or equal to 1.	2000
KUX_IGNORE_MPSTAT	To avoid usage of the mpstat command for CPU statistics measurements on AIX.	[TRUE|true] or false, not case-sensitive.	false, not case-sensitive

Table 177. UNIX OS monitoring agent environment variables (continued)

Variable	Description	Possible values	Default value
KUX_PINGHOSTLIST	To set the full qualified path of the file containing the list of servers for the UNIX Ping attribute group.	A valid UNIX pathname.	blank
KUX_PRCTL_OFF	To disable collecting CPUSHARES and SHAREPCT to avoid crashes in the prctl command due to a Solaris Zones bug.	[TRUE|true|True] or false, not case-sensitive.	false, not case-sensitive
KUX_SKIP_DISABLED_ DISKS	To avoid the OS agent reporting on SAN Physical Disks that are not actually available to the system.	[TRUE|true|True] or false, not case-sensitive.	false, not case-sensitive
KUX_SKIP_HP_ UTMPX	To skip counting the number of user sessions for the All Users group on HPUX.	True or false, not case-sensitive.	false, not case-sensitive
KUX_SKIP_TTY	To improve process monitoring performances by skipping TTY names resolution.	[TRUE|true|True] or false, not case-sensitive.	false, not case-sensitive
KUX_SKIP_UTF8CONV_ PROCESS	To reduce OS agent high CPU consumption when converting strings to UTF8 for large number of Processes.	[TRUE|true|True] or false, not case-sensitive.	false, not case-sensitive
KUX_TCPSTAT_SAMPLE_ SECS	To tune the sampling interval for the TCP Statistics attribute group (in seconds).	Greater than or equal to 5.	30
KBB_NFS_TIMEOUT	To tune the timeout on NFS filesystems monitoring (in seconds).	Between 1 and 30.	2
KDEBE_FIPS_MODE_ ENABLED	To request GSKit compliance to FIPS 140-2 standards when computing checksums for files in the File Information attribute group.	[yes|no], not case-sensitive.	no, not case-sensitive
MAX_NUMBER_OF_ DISKS_HPUX	To define an upper limit on the number of disks that the agent can monitor (HP-UX only).	Between 0 and 30000.	3500
NETWORK_INTERFACE _REFRESH_TIME_HPUX	To tune the frequency of full refresh of network interfaces data (in hours - HPUX only).	Between 1 and 168.	0
TTY_REFRESH_TIME	To tune the refresh interval of tty<->ttyname mapping table (in minutes).	Greater than or equal to 1.	60
UID_USERNAME_ REFRESH_TIME	To tune the refresh interval of uid<->username and gid<->groupname mapping tables (in minutes).	Greater than or equal to 1.	60

Appendix F. Maintaining the EIB on Linux or UNIX

To ensure the effective operation of your monitoring server, back up your Enterprise Information Base (EIB) tables as part of your routine maintenance. The EIB contains the attributes and other data that define the agents to the server. The following files, which are stored in the *install_dir*/tables/eib directory, compose the EIB.

Table 178. EIB Files

*.db Files		*.idx Files	
qa1cacts.db	qa1daggr.db	qa1cacts.idx	qa1daggr.idx
qa1cckpt.db	qa1dcct.db	qa1cckpt.idx	qa1dcct.idx
qa1ccobj.db	qa1dcct2.db	qa1ccobj.idx	qa1dcct2.idx
qa1ccomm.db	qa1dmobj.db	qa1ccomm.idx	qa1dmobj.idx
qa1ceibl.db	qa1dmtmp.db	qa1ceibl.idx	qa1dmtmp.idx
qa1chost.db	qa1dobja.db	qa1chost.idx	qa1dobja.idx
qa1ciobj.db	qa1dpcyf.db	qa1ciobj.idx	qa1dpcyf.idx
qa1cmcfg.db	qa1drnke.db	qa1cmcfg.idx	qa1drnke.idx
qa1cnodl.db	qa1drnkg.db	qa1cnodl.idx	qa1drnkg.idx
qa1cplat.db	qa1dsnos.db	qa1cplat.idx	qa1dsnos.idx
qa1cpset.db	qa1dspst.db	qa1cpset.idx	qa1dspst.idx
qa1cruld.db	qa1dstms.db	qa1cruld.idx	qa1dstms.idx
qa1csitf.db	qa1dstsa.db	qa1csitf.idx	qa1dstsa.idx
qa1csmni.db	qa1dstua.db	qa1csmni.idx	qa1dstua.idx
qa1cstsc.db	qa1dswrs.db	qa1cstsc.idx	qa1dswrs.idx
qa1cstsh.db	qa1dswus.db	qa1cstsh.idx	qa1dswus.idx
qa1cthre.db	qa1dwgrp.db	qa1cthre.idx	qa1dwgrp.idx
qa1dactp.db	qa1dwork.db	qa1dactp.idx	qa1dwork.idx

Appendix G. Securing your IBM Tivoli Monitoring installation on Linux or UNIX

On UNIX and Linux operating systems, the product installation process creates the majority of directories and files with **world write** permissions. This configuration creates a security situation that is not acceptable in many enterprises. The **secureMain** utility helps you bring the monitoring environment into compliance with the security standards of your company. Run the **secureMain** utility on all installations, especially those installations that include the UNIX OS Agent, to prevent privilege escalation.

Note: You do not need to be logged in as a root user to run this utility, but you are prompted for the root password when it is required.

Usage

The **secureMain** commands use the following syntax:

```
secureMain [-h install_dir] [-g common_group] [-t type_code] lock
secureMain [-h install_dir] [-g common_group] unlock
```

where variables are defined as follows:

- *install_dir* is the directory path for the IBM Tivoli Monitoring installation. If this parameter is not supplied, the script attempts to determine the installation directory.
- *common_group* is a group ID common to all of the user IDs that are used to run components in this installation. The user ID that is used to perform the installation must also be a member of the group ID specified. The only exception is that the **root** ID is not required to be a member of the group ID specified.
- **type_code** is a component code belonging to an installed component. You can specify multiple **-t** options to create a list of component codes to be processed.

If **secureMain** is invoked with no parameters, the usage text is displayed.

secureMain lock is used to tighten permissions in an IBM Tivoli Monitoring 6.1 installation. It should be run after installing or configuring components.

When **secureMain lock** is invoked with no other parameters, the permissions are tightened generally to 755. However, a number of directories and some files are still left with **world write** permissions. When certain components which are commonly run using multiple user IDs are present in the installation, many more files have **world write** permissions.

When **secureMain lock** is invoked with the **-g common_group** parameter, the permissions are tightened generally to 775 and the directories and files have their group owner changed to **common_group** specified. There are no directories or files left with **world write** permissions. Even when certain components which are commonly run using multiple user IDs are present in the installation, no files will have **world write** permissions. Additionally, the **common_group** value specified is written to a file and is used for all future **secureMain lock** invocations in this installation, unless the **-g** option is specified and the **common_group** is different from the previous value.

When **secureMain lock** is invoked with the **-t** *type_code* parameter, sections of the installation might be skipped when tightening permissions. Common directories, like `bin`, `config`, `registry`, and `logs`, and the files in them are always processed. Only directories and files specific to the specified **type_code** components are processed. The other component directory trees are skipped.

secureMain unlock is used to loosen permissions in an IBM Tivoli Monitoring 6.2 installation. **secureMain unlock** is normally not necessary, but can be run if desired. It should be run before installing or configuring components.

secureMain unlock does not return the installation to the permission state that it was in before running **secureMain lock**. It only processes the common directories, like `bin`, `config`, `registry`, and `logs`, and the files in them.

Examples

The following example locks the installation using the common group **itmgroup**:

```
secureMain -g itmgroup lock
```

The following example locks the base and `mq` component directories using the common group **itmgroup**:

```
secureMain -g itmgroup -t mq lock
```

Scenario with secureMain

The following scenario illustrates the use of **secureMain**:

1. Complete the following operations using **root** authorization:
   a. Install OS Agent.
   b. Configure OS Agent.
   c. List files with **world write** permissions, using the following command: `find . -perm -o+w -ls`
   d. Run the following command: `secureMain -g itmgroup -t ux lock`
   e. Install the 32-bit Enterprise Svcs UI to get the 32-bit framework.
   f. Install the MQ agent.
   g. Run the following command: `secureMain -g itmgroup -t mq lock`
   h. List files with **world write** permissions, using the following command: `find . -perm -o+w -ls`
   i. Start the OS agent.
2. Complete the following operations using **mquser** authorization:
   a. Start the MQ agent for a queue manager.
   b. Start the MQ agent for a second queue manager.
   c. Stop the MQ agent for the first queue manager.
   d. Stop the MQ agent for the second queue manager.
3. Complete the following operations using **root** authorization:
   a. Stop the OS Agent.
   b. List files with **world write** permissions, using the following command: `find . -perm -o+w -ls`

Securing your IBM Tivoli Monitoring environment at installation

Starting with IBM Tivoli Monitoring V6.2.3, the command `./install.sh` contains the optional `–k group` parameter. The `-k` parameter runs the **secureMain** utility at the end of the installation process.

Example:

```
./install.sh –h /opt/IBM/ITM –k itmgroup
```

If you do not specify the `-k` parameter when you run `./install.sh`, you will be asked at the end of the installation if you want to secure your IBM Tivoli Monitoring environment. If your IBM Tivoli Monitoring environment is already secured this question is skipped. When you install a monitoring agent into an already secured environment, **secureMain** will automatically execute at the end of the installation. For more information, see “Installing into an existing installation” on page 136.

Appendix H. Uninstalling IBM Tivoli Monitoring

Use the following steps to uninstall IBM Tivoli Monitoring:

- “Uninstalling the entire IBM Tivoli Monitoring environment”
- “Uninstalling an individual IBM Tivoli Monitoring agent or component” on page 856
- “Uninstalling components and agents silently” on page 859
- “Uninstalling the event synchronization component” on page 862

Uninstalling the entire IBM Tivoli Monitoring environment

Use the following procedures to remove the entire IBM Tivoli Monitoring environment from your computer.

- “Uninstalling the environment on Windows”
- “Uninstalling the environment on Linux or UNIX” on page 855

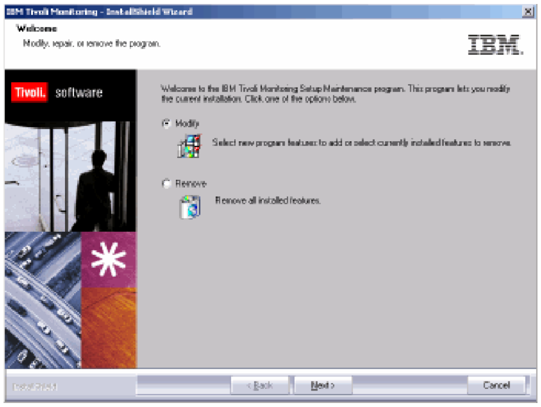
If you want to remove just one component such as an agent, see “Uninstalling an individual IBM Tivoli Monitoring agent or component” on page 856.

Note: If you plan to reinstall IBM Tivoli Monitoring into a different directory than the one used for this installation, you must stop and restart this computer before reinstalling IBM Tivoli Monitoring.

Uninstalling the environment on Windows

Use the following steps to uninstall IBM Tivoli Monitoring from a Windows computer:

1. From the desktop, click **Start** → **Settings** → **Control Panel** (for Windows 2000) or **Start** → **Control Panel** (for Windows 2003).
2. Click **Add/Remove Programs**.
3. Select **IBM Tivoli Monitoring** and click **Change/Remove**. The following window is displayed.



Welcome

Modify, repair, or remove the program.

Welcome to the IBM Tivoli Monitoring Setup Maintenance program. This program lets you modify the current installation. Click one of the options below.

● Modify

Select new program features to add or select currently installed features to remove.

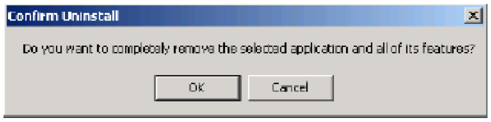
○ Remove

Remove all installed features.

Figure 175. Uninstalling IBM Tivoli Monitoring

4. Select **Remove** and click **Next**.

   The following window is displayed.

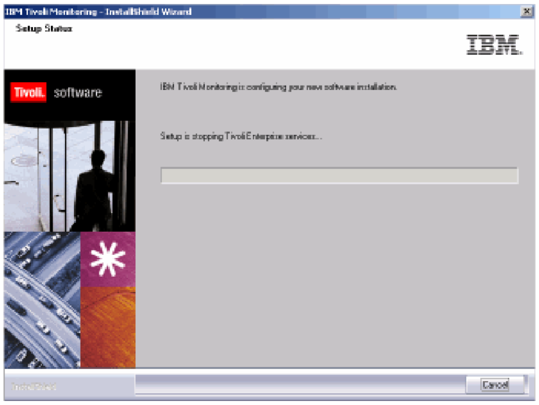


Do you want to completely remove the selected application and all of its features?

Figure 176. Confirming the uninstallation

5. Click **OK**.

   The following progress window is displayed.

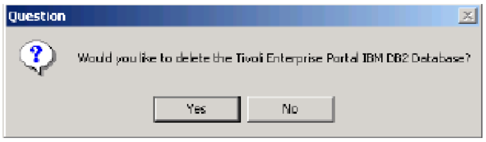


IBM Tivoli Monitoring is configuring your new software installation.

Setup is stopping Tivoli Enterprise services...

Figure 177. Stopping Tivoli components before uninstallation

After Tivoli Enterprise services have stopped, you are asked if you want to remove the Tivoli Enterprise Portal database.

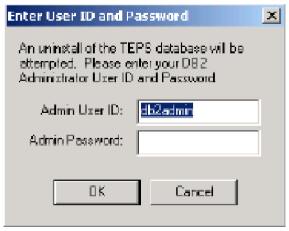


Would you like to delete the Tivoli Enterprise Portal IBM DB2 Database?

Figure 178. Removing the portal database

6. Click **Yes**.

   The following window is displayed, requesting information required to remove the database:



An uninstall of the TEPS database will be attempted. Please enter your DB2 Administrator User ID and Password

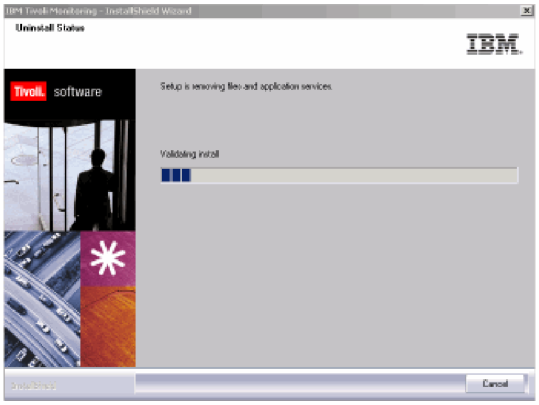
Admin User ID: db2admin

Admin Password:

Figure 179. Database information

7. Type the password for the database administrator in the **Admin Password** field and click **OK**.

   The following progress window is displayed.


IBM Tivoli Monitoring - InstallShield Wizard

Uninstall Status

Setup is removing files and application services.

Validating install

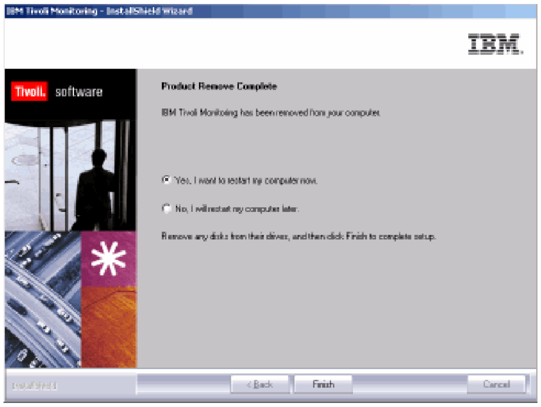
Figure 180. Uninstallation progress window

A pop-up window, indicating that GSKit is being uninstalled, is displayed.

Uninstalling GSKit, please wait.

Figure 181. GSKit uninstallation

After GSKit is uninstalled, the following window is displayed:


IBM Tivoli Monitoring - InstallShield Wizard

Product Remove Complete

IBM Tivoli Monitoring has been removed from your computer.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

Figure 182. Successful uninstallation

8. Click **Finish**.

Uninstalling the environment on Linux or UNIX

Use the following steps to uninstall IBM Tivoli Monitoring from a UNIX computer:

1. From a command prompt, run the following command to change to the appropriate /bin directory:

   ```
   cd install_dir/bin
   ```

   where *install_dir* is the path for the home directory for IBM Tivoli Monitoring.
2. Run the following command:

   ```
   ./uninstall.sh
   ```

   A numbered list of product codes, architecture codes, version and release numbers, and product titles is displayed for all installed products.

3. Type the number for the installed product that you want to uninstall. Repeat this step for each additional installed product you want to uninstall.
4. After you have removed all installed components, you are asked if you want to remove the installation directory. Type `y` and press Enter.

You can also run the following command to remove all installed components from the command-line:

```
./uninstall.sh REMOVE EVERYTHING
```

After the command completes, you can manually remove the IBM Tivoli Monitoring installation directory.

Note: If for any reason, the UNIX uninstallation is not successful, run the following command to remove all IBM Tivoli Monitoring directories:

```
rm -r install_dir
```

This uninstallation program does not delete the database created for Tivoli Enterprise Portal on a Linux portal server. If you want to delete that database, you must remove it manually. See the documentation for your database software for information about deleting a database.

Uninstalling an individual IBM Tivoli Monitoring agent or component

Use the following procedures to remove an agent or other individual IBM Tivoli Monitoring component from your computer:

- "Uninstalling a component on Windows"
- "Uninstalling a component on Linux or UNIX" on page 857
- "Uninstalling OMEGAMON Monitoring Agents" on page 857
- "Removing an agent through the Tivoli Enterprise Portal" on page 859

Note: If you plan to reinstall a IBM Tivoli Monitoring component into a different directory than the one used for this installation, you must stop and restart this computer before reinstalling the IBM Tivoli Monitoring component.

Uninstalling a component on Windows

Use the following steps to remove a component on a Windows computer. You can uninstall a single agent or the entire agent bundle (such as IBM Tivoli Monitoring for Databases). You cannot uninstall the application support files laid down for a Windows Tivoli Enterprise Monitoring Server without uninstalling the monitoring server.

1. From the desktop, click **Start → Settings → Control Panel** (for Windows 2000 or 2003) or **Start → Control Panel** (for Windows XP).
2. Click **Add/Remove Programs**.
3. Complete one of the following:
   - To uninstall a single IBM Tivoli Monitoring component, such as the portal server or portal client (but not all components), select **IBM Tivoli Monitoring**.
   - To uninstall an agent bundle or a specific agent, select the agent bundle.
4. Click **Change/Remove**.
5. Complete one of the following steps:
   - To uninstall a specific agent or component, select **Modify**.
   - To uninstall the entire agent bundle, select **Remove**.
6. Click **Next**.
7. If you are uninstalling an agent bundle, click **OK** to confirm the uninstallation.
8. If you are uninstalling an agent or component, complete the following steps:

a. For an agent, expand **Tivoli Enterprise Monitoring Agents** and select the agent you want to uninstall.

b. For a component, select the component (such as **Tivoli Enterprise Portal Desktop Client**).

c. For Tivoli Performance Analyzer, clear the Tivoli Performance Analyzer check box in all sections (such as Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, Tivoli Enterprise Portal Desktop Client, and so on).

d. Click **Next**.

e. Click **Next** on the confirmation screen.

f. Depending on the remaining components on your computer, there might be a series of configuration panels. Click **Next** on each of these panels.

Note: When you are uninstalling the Tivoli Enterprise Portal Server, the installer gives you the option to remove the portal server database. If there are other databases created by Tivoli Monitoring in this or previous version on the computer, the installer gives you the option to remove them as well.

9. Click **Finish** to complete the uninstallation.

Uninstalling a component on Linux or UNIX

Use the following steps to remove a component on a UNIX computer. You can uninstall a single agent or the entire agent bundle (such as IBM Tivoli Monitoring for Databases).

1. From a command prompt, run the following command to change to the appropriate /bin directory:

```
cd install_dir/bin
```

where *install_dir* is the path for the home directory for IBM Tivoli Monitoring.

2. Run the following command:

```
./uninstall.sh
```

A numbered list of product codes, architecture codes, version and release numbers, and product titles is displayed for all installed products.

3. Type the number for the agent or component that you want to uninstall. Repeat this step for each additional installed product you want to uninstall.

Note: When you are uninstalling the Tivoli Enterprise Portal Server, the installer gives you the option to remove the portal server database. If there are other databases created by Tivoli Monitoring in this or previous version on the computer, the installer gives you the option to remove them as well.

Uninstalling OMEGAMON Monitoring Agents

Use the following steps to remove OMEGAMON agents from a computer. Table 179 on page 858 and Table 180 on page 858 list the agents by internal code, release, and descriptive name.

1. Launch Manage Candle Services (350 or 360) or Manage Tivoli Enterprise Monitoring Services.
2. Use the **Description** and **Release** columns to locate the agent service name.
3. Stop the service by right-clicking the name and clicking **Stop**.
4. Take note of any task or subsystem names that are listed in the column for your agent. Usually this column lists `Primary` unless your agent supports instances. If your agent supports instances, record these names for later use.
5. Unconfigure the agent by right-clicking the name and clicking **Advanced → Unconfigure**. The **Configured** column changes from `Yes` to `No`. Continue to unconfigure all instances found in step 4.
6. Open the Windows Explorer and navigate to the installation directory for OMEGAMON 350 or 360 products and IBM Tivoli Monitoring. The default directories are C:\Candle Candle OMEGAMON and C:\IBM\ITM for IBM Tivoli Monitoring. Then navigate to the CMA directory.

7. Delete files K??ENV (Task/SubSystem name Primary) and any instances shown as K??ENV_INSTANCENAME (Task/SubSystem name from step 4 on page 857).
8. Delete any *PC**.EXE or *PC**.DLL files for the product. *PC* is the product internal identifier three-character code from the tables.
9. Exit Manage Candle Services or Manage Tivoli Enterprise Monitoring Services, and launch it again. The agent and all instances should not be shown under the **Service/Application** column.

Note: You can also use this procedure to remove IBM Tivoli Monitoring agents if you use the TMAITM6 directory instead of the CMA directory in step 6 on page 857. All of the other steps do not change.

Table 179. Candle OMEGAMON Release 04R1

Internal identifier	Release	Description
K3Z	400	Windows Server Active Directories Monitoring Agent
KA2	120	Alert Adapter for AF/Remote
KA4	300	Monitoring Agent for OS/400
KBL	320	CASP Directory Server Monitoring Agent
KBR	320	CASP Exchange Connector Monitoring Agent
KEZ	251	eBA Solutions Monitoring Agent
KIC	100	WebSphere InterChange Server Monitoring Agent
KIE	100	WebSphere InterChange Server Data Source
KMA	201	Alert Adapter for Remedy ARS
KMC	360	WebSphere MQ Configuration Agent
KMQ	360	WebSphere MQ Monitoring Agent
KNW	300	NetWare Monitoring Agent
KOQ	301	Microsoft SQL Server Monitoring Agent
KOR	301	Oracle Monitoring Agent
KOY	300	Sybase Monitoring Agent
KPT	201	Alert Adapter for Peregrine Service Center
KQI	120	WebSphere Integration Brokers Monitoring Agent
KSA	301	R/3 Monitoring Agent
KTX	300	Tuxedo Monitoring Agent
KUD	400	DB2 Universal Database Monitoring Agent
KWE	130	WebSphere Application Server Monitoring Agent
KWL	100	BEA WebLogic Server Monitoring Agent
KWN	100	Windows Management Web Service

Table 180. Candle OMEGAMON Release BIV110

Internal identifier	Release	Description
KIC	110	WebSphere InterChange Server Monitoring Agent
KIE	110	WebSphere InterChange Server Data Source
KMC	370	WebSphere MQ Configuration Agent
KMQ	370	WebSphere MQ Agent
KQI	130	WebSphere Integration Brokers Monitoring Agent

Removing an agent through the Tivoli Enterprise Portal

You can also uninstall non-OS monitoring agents from the Tivoli Enterprise Portal by stopping the agent and removing its configuration settings. After you have removed the agent from the enterprise, you can completely uninstall the agent from the managed system. When you remove an agent, it is removed from any managed system groups to which it is assigned, any situation or policy distribution lists it was on, and any custom Navigator view items to which it was assigned.

Note: You cannot use the Tivoli Enterprise Portal to remove or uninstall OS agents.

Note: If the Manage Tivoli Enterprise Monitoring Services utility is running when you uninstall the agent, it is shut down automatically by the uninstallation process.

Use the following steps to remove and optionally uninstall an agent:

1. In the Tivoli Enterprise Portal, right-click the agent Navigator item and click **Remove**.
2. Click **Yes** when you are asked to confirm the removal of the agent.
3. When you are asked to confirm that you want to permanently uninstall the agent, click **Yes** to uninstall or **No** to leave the agent installed on the computer.

Uninstalling the Warehouse Proxy

When you uninstall the Warehouse Proxy, the warehouse database is not removed and historical situations on the agent are not stopped.

Before you uninstall, complete the following steps to uninstall the warehouse database and historical situations on the agent:

1. Stop the historical situations.
2. Drop the warehouse database.
3. Remove the ODBC data source.
4. Remove the Windows user, ITMUser, that was created to connect to a DB2 for Linux, UNIX, and Windows database.

Removing the ODBC data source connection

When you uninstall IBM Tivoli Monitoring, the ODBC data source created for the Warehouse Proxy Agent is not removed automatically, which can cause problems when you reinstall IBM Tivoli Monitoring. To prevent these problems, manually remove the ODBC data source after you uninstall IBM Tivoli Monitoring.

For example, to remove the DB2 for Linux, UNIX, and Windows data source from the DB2 command-line, run the following command:

```
DB2 UNCATALOG SYSTEM ODBC DATA SOURCE datasource_name
```

If you are using a Microsoft SQL database or an Oracle database, use the Windows ODBC Data Source Administrator utility to remove the ODBC data source.

Uninstalling components and agents silently

Use the following procedures to uninstall components and agents silently:

- “Performing a silent uninstallation on a Windows computer” on page 860
- “Performing a silent uninstallation on a Linux or UNIX computer” on page 861

Performing a silent uninstallation on a Windows computer

Like silent installation, silent uninstallation on Windows uses a response file. Sample response files are shipped with IBM Tivoli Monitoring components and all monitoring agents that use the IBM Tivoli Monitoring services. The sample files can be found in one of the following locations:

- On the product installation media
- After installation, in the *install_dir*samples directory (if present)

Note: The name of the response file varies by installation media and release. The file is typically found on the media under the WINDOWS directory. For Windows on Itanium, there is a separate WIA64 directory with an Itanium specific response file. If IBM Tivoli Monitoring was used to generate the response file for a configured agent, the response file is located where it was generated (*install_dir*\response is the default) and the name of the file is silent_install_*pc*.txt. See "Automatically creating agent response files on Windows" on page 790 for more information.

Note:

Complete the following steps to edit the response file as appropriate for your uninstallation:

1. Locate the appropriate file. For example:

   silent_server.txt
   For a server image

   silent_agent.txt
   For an agent image

   silent_WIA64.txt
   For an agent image for 64-bit Windows Itanium
2. Copy this file to a temporary directory on your system.
3. Open your copy of the file in a text editor.
4. Change the parameters as appropriate for the uninstallation:
   - To remove all components and the installation directory, uncomment (by removing the semicolon) the following line in the ACTION TYPE section:

     ```
     ;REMOVEALL=Yes
     ```
   - To select a component to uninstall, uncomment the following line in the ACTION TYPE section

     ```
     ;UNINSTALLSELECTED=Yes
     ```

     and uncomment the component or components to be removed in the FEATURE section. For example:

     ```
     ;**********************************************************************
     ;
     ;               TIVOLI ENTERPRISE MONITORING AGENT
     ;                   TEMA INSTALLATION SECTION
     ;
     ; Any Feature selected that ends in CMA will cause the TEMA Framework
     ; and specific Agent to be installed.
     ;
     ;**********************************************************************
     ;KGLWICMA=Tivoli Enterprise Monitoring Agent Framework
     ;KNTWICMA=Monitoring Agent for Windows OS
     ;KNT64CMA=Monitoring Agent for Windows OS (86-x64 only)
     ;KR2WICMA=Agentless Monitoring for Windows Operating Systems
     ;KR3WICMA=Agentless Monitoring for AIX Operating Systems
     ;KR4WICMA=Agentless Monitoring for Linux Operating Systems
     ;KR5WICMA=Agentless Monitoring for HP-UX Operating Systems
     ;KR6WICMA=Agentless Monitoring for Solaris Operating Systems
     ;KUMWICMA=Universal Agent
     ;KAC64CMA=32/64 Bit Agent Compatibility Package
     ;KUEWICMA=Tivoli Enterprise Services User Interface Extensions
     ```

5. Save the file and close the editor.

Take the following steps to run the silent uninstallation:

1. Open a DOS command prompt.
2. In the prompt, change to the directory containing this installation (where setup.exe and setup.ins files are located).
3. Run the setup as follows, specifying the parameters in the order listed:

   ```
   start /wait setup /z"/sfresponse_file" /s /f2"C:\temp\silent_setup.log"
   ```

   where:

   /z"/sf
   Specifies the fully qualified name of the response file you edited. For example: `/z"/sfC:\temp\myresponse.txt"`

   /s
   Specifies that this is a silent installation. This causes nothing to be displayed during installation.

   /f2
   Specifies the name of the InstallShield log file. If you do not specify this parameter, the default is to create Setup.log in the same location as the setup.iss file. In either case, the Setup program must be able to create and write to this file. If the specified directory does not exist, the Setup program cannot create and write to the file. Therefore, the specified directory path must exist.

When the uninstallation is complete, setup.exe returns to the command prompt.

If the uninstallation is unsuccessful, check the installation log in the *install_dir*\InstallITM\ *product_nametime_stamp*.log directory. If all components and directories have been removed, the log will be in the root of the C: drive. The name of the file is the name of the product being uninstalled with date and time appended to the end of the product name.

Performing a silent uninstallation on a Linux or UNIX computer

Complete the following steps to uninstall IBM Tivoli Monitoring components and monitoring agents unattended (that is, without having to specify parameters interactively):

1. Stop all agents and servers that you are going to remove. Use the ./cinfo -r command to see a list of running process, and then use the itmcmd command to stop the appropriate processes.
2. Change to the installation directory bin directory:

   ```
   cd install_dir/bin
   ```
3. To select individual components or agents, enter the following command:

   ```
   uninstall.sh [-f] [-i] [-h install_directory] [product platformCode]
   ```

   where

   -f
   Forces delete, suppressing confirmation messages and prompts.

   -i
   Ignores all running processes.

   product
   Is the two-letter code for the product to be uninstalled.

   platformCode
   Is the platform code for the product (such as aix513, sol286, hp11, and so forth: see Appendix D, “IBM Tivoli product, platform, and component codes,” on page 815).

   Repeat the command for each agent or component you want to remove on the target computer.
4. To remove all components and agents enter the following command:

   ```
   uninstall.sh REMOVE EVERYTHING
   ```

When the uninstallation is complete, the uninstall command returns to the command prompt. Some messages may be written to the screen. There may be additional steps, depending on the component being uninstalled. For example, if you uninstall the Warehouse Proxy, the warehouse database is not removed and historical situations on the agent are not stopped (see “Uninstalling the Warehouse Proxy” on page 859).

If the uninstallation is unsuccessful, some messages may be written to the screen. See the installation log in the `install_dir/logs/product_nametime_stamp.log` directory or the *IBM Tivoli Monitoring: Troubleshooting Guide* for more information. If all components have been removed, the log is at the root.

Note: If for any reason, the UNIX uninstallation is not successful, run the following command to remove all Tivoli Monitoring directories:

```
rm -r install_dir
```

Uninstalling the event synchronization component

Use the following steps to uninstall the event synchronization from your event server:

Note: On Windows 2003 you must run the **change user /install** command from a command prompt before you begin. This puts the computer into the required “install” mode. After the uninstallation, run the **change user /execute** command to return the computer to its previous mode.

1. Set the Tivoli environment:
   - On Windows:
     ```
     C:\windows\system32\drivers\etc\Tivoli\setup_env.cmd
     ```

     or
     ```
     C:\winnt\system32\drivers\etc\Tivoli\setup_env.cmd
     ```
   - On operating systems like UNIX and Linux:
     ```
     . /etc/Tivoli/setup_env.sh
     ```
2. Run the following uninstallation program:
   - On Windows:
     ```
     %BINDIR%\TME\TEC\OM_TEC\_uninst\uninstaller.exe
     ```
   - On operating systems like UNIX and Linux:
     ```
     $BINDIR/TME/TEC/OM_TEC/_uninst/uninstaller.bin
     ```

   You can run this uninstallation program in silent mode (by running the program from the command-line with the **-silent** parameter) or in console mode (by using the **-console** parameter).
3. Follow the prompts in the uninstallation program.

When the uninstallation is completed, you can tell the installer what rule base should be loaded. If initial installation created a new rule base, the value shown in “Rule base name of rule base to be loaded on completion of this uninstall” will be Default, meaning that the Default rule base will be loaded. If the initial installation updated an existing rule base, that rule base name is provided as the value for “Rule base name of rule base to be loaded on completion of this uninstall”. You can override this value by typing in the name of the rule base you want to have loaded.

You can also tell the uninstaller to stop and restart the event server.

You can run the silent uninstallation using default processing or create a template to change the default values. The default processing will load the Default rule base (or the existing rule base was chosen during installation) and will *not* restart the TEC server.

To create and use a template:

1. Create the template:

- On Windows:

```
%BINDIR%\TME\TEC\OM_TEC\_uninst\uninstaller.exe –options-template
  itmeventsynchU.txt
```

- On operating systems like UNIX or Linux:

```
$BINDIR/TME/TEC/OM_TEC/_uninst/uninstaller.bin –options-template
  itmeventsynchU.txt
```

2. Modify the template as desired:
   - To specify which rule base to load, modify the `restartTECU.uRBN` file.
   - To automatically restart the event server, modify the `restartTECU.restartTECU` file.
3. Set the Tivoli environment.
4. Run the uninstallation program as follows:
   - On Windows:

```
%BINDIR%\TME\TEC\OM_TEC\_uninst\uninstaller.exe –options
  itmeventsynchU.txt –silent
```

   - On operating systems like UNIX or Linux:

```
$BINDIR/TME/TEC/OM_TEC/_uninst/uninstaller.bin –options
  itmeventsynchU.txt –silent
```

If your event server is running on an HP-UX computer, ensure that the _uninst and _jvm directories are successfully removed by the uninstallation program. If they are not, manually delete these directories.

Uninstalling event synchronization manually

Use the instructions for the appropriate operating system to remove event synchronization manually.

HP11

1. Stop the situation update forwarder long-running process if it is still running:
   a. Find the long-running process using:

```
ps –ef
```

   b. Use the kill command to remove the process:

```
kill –9 process_number
```

2. Run the following command to determine whether the operating system still knows that the event synchronization component is there:

```
swlist -v TecEvntSyncInstaller
```

   If it is there but all the code has been deleted, or just the uninstaller is deleted, you can try this command:

```
swremove TecEvntSyncInstaller
```

3. If errors are returned saying that TecEvntSyncInstaller cannot be removed due to consistency or dependency checks, create a file named something like "remove_EvntSync.txt" and add these two lines:

```
enforce_dependencies=false
enforce_scripts=false
```

   Then run the **swremove** command as follows:

```
swremove -X remove_EvntSync.txt TecEvntSyncInstaller
```

   The `-X` option tells the **swremove** command to ignore checks and dependencies and remove the files regardless.
4. Remove any event synchronization directories that are left behind.

Remove any directories found in OM_TEC including OM_TEC itself. OM_TEC is found in $BINDIR/TME/TEC. To use $BINDIR you must run the following command:

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

If the installation was for Netcool/OMNIbus, remove files from the location indicated during installation.

Windows

1. Stop the situation update forwarder long running process if it is still running:
   a. In the Control Panel, open **Administrative Tools**, then **Services**.
   b. Find the Tivoli Situation Update Forwarder service, and right-click it and select **Stop**.
2. Go to operating system directory (C:\windows or C:\winnt) and open the vpd.properties file.
3. Remove all lines that have itmTecEvntSyncProduct, EvntSyncForwarder, itmTecEvntSyncLapComp or EvntSyncForwarderWin in them.
4. Remove any event synchronization directories that are left behind.

   Remove any directories found in OM_TEC including OM_TEC itself. OM_TEC is found in %BINDIR%/TME/TEC. To use %BINDIR% you must run the C:\windows\system32\drivers\etc\Tivoli\setup_env.cmd command. If the installation was for Netcool/OMNIbus, remove the files from the location indicated during installation.

AIX

1. Stop the situation update forwarder long running process if it is still running:
   a. Find the long running process using:

      ```
      ps –ef
      ```

   b. Use the kill command to remove the process:

      ```
      kill –9 process_number
      ```

2. Go to operating system directory (this is typically /usr/lib/objrepos) and open the vpd.properties file.
3. Remove all lines that have itmTecEvntSyncProduct, EvntSyncForwarder, itmTecEvntSyncLapComp or EvntSyncForwarderWin in them.
4. Remove any event synchronization directories that remain.

   Remove any directories found in OM_TEC including OM_TEC itself. OM_TEC is found in $BINDIR/TME/TEC. To use $BINDIR you must run the following command:

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

   If the installation was for Netcool/OMNIbus, remove the files from the location specified during installation.

Linux

1. Stop the situation update forwarder long running process if it is still running:
   a. Find the long running process using:

      ```
      ps –ef
      ```

   b. Use the kill command to remove the process:

      ```
      kill –9 process_number
      ```

2. Go to operating system directory (this is typically / or /root) and open the file vpd.properties.
3. Remove all lines that have itmTecEvntSyncProduct, EvntSyncForwarder, itmTecEvntSyncLapComp or EvntSyncForwarderWin in them.
4. Remove any event synchronization directories that are left behind.

   Remove any directories found in OM_TEC including OM_TEC itself. OM_TEC is found in $BINDIR/TME/TEC. To use $BINDIR you must run the following command:

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

   If the installation was for OMNIbus, remove the files from the location specified during installation

Solaris

1. Stop the situation update forwarder long running process if it is still running:
   a. Find the long running process using:

      ```
      ps –ef
      ```

   b. Use the kill command to remove the process:

      ```
      kill –9 process_number
      ```

2. Run the following command to remove the situation update forwarder:

   ```
   pkgrm –A ISitmTecE
   ```

3. Remove any event synchronization directories that are left behind.

   Remove any directories found in OM_TEC including OM_TEC itself. OM_TEC is found in $BINDIR/TME/TEC. To use $BINDIR you must run the following command:

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

   If the installation was for OMNIbus, remove the files from the location specified during installation.

Appendix I. Documentation library

This appendix contains information about the publications related to IBM Tivoli Monitoring and to the commonly shared components of Tivoli Management Services. These publications are listed in the following categories:

- IBM Tivoli Monitoring library
- Related publications

See *IBM Tivoli Monitoring and OMEGAMON XE Products: Documentation Guide*, SC23-8816, for information about accessing and using the publications. You can find the *Documentation Guide* in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/. To open the *Documentation Guide* in the information center, select **Using the publications** in the **Contents** pane.

To find a list of new and changed publications, click **What's new** on the Welcome page of the IBM Tivoli Monitoring and OMEGAMON XE Information Center. To find publications from the previous version of a product, click **Previous versions** under the name of the product in the **Contents** pane.

IBM Tivoli Monitoring library

The following publications provide information about IBM Tivoli Monitoring and about the commonly shared components of Tivoli Management Services:

- *Quick Start Guide*,

  Introduces the components of IBM Tivoli Monitoring.
- *Installation and Setup Guide*, GC32-9407

  Provides instructions for installing and configuring IBM Tivoli Monitoring components on Windows, Linux, and UNIX systems.
- *Upgrading from V5.1.2*, GC32-1976

  Gives instructions for migrating your site's custom resource models from IBM Tivoli Monitoring V5.1.2 to IBM Tivoli Monitoring V6.2.
- *Program Directory for IBM Tivoli Management Services on z/OS*, GI11-4105

  Gives instructions for the SMP/E installation of the Tivoli Management Services components on z/OS.
- *Configuring the Tivoli Enterprise Monitoring Server on z/OS*, SC32-9463

  Provides instructions for preparing, configuring, and customizing your monitoring servers on z/OS. This guide complements the *IBM Tivoli OMEGAMON XE and IBM Tivoli Management Services on z/OS Common Planning and Configuration Guide* and the *IBM Tivoli Monitoring Installation and Setup Guide*.
- *Administrator's Guide*, SC32-9408

  Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.
- *High-Availability Guide for Distributed Systems*, SC23-9768

  Gives instructions for several methods of ensuring the availability of the IBM Tivoli Monitoring components.
- Tivoli Enterprise Portal online help

  Provides context-sensitive reference information about all features and customization options of the Tivoli Enterprise Portal. Also gives instructions for using and administering the Tivoli Enterprise Portal.

- *Tivoli Enterprise Portal User's Guide*, SC32-9409

  Complements the Tivoli Enterprise Portal online help. The guide provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.
- *Command Reference*, SC32-6045

  Provides detailed syntax and parameter information, as well as examples, for the commands you can use in IBM Tivoli Monitoring.
- *Troubleshooting Guide*, GC32-9458

  Provides information to help you troubleshoot problems with the software.
- *Messages*, SC23-7969

  Lists and explains messages generated by all IBM Tivoli Monitoring components and by z/OS-based Tivoli Management Services components (such as Tivoli Enterprise Monitoring Server on z/OS and TMS:Engine).
- *IBM Tivoli Universal Agent User's Guide*, SC32-9459

  Introduces you to the IBM Tivoli Universal Agent, an agent of IBM Tivoli Monitoring. The IBM Tivoli Universal Agent enables you to use the monitoring and automation capabilities of IBM Tivoli Monitoring to monitor any type of data you collect.
- *IBM Tivoli Universal Agent API and Command Programming Reference Guide*, SC32-9461

  Explains the procedures for implementing the IBM Tivoli Universal Agent APIs and provides descriptions, syntax, and return status codes for the API calls and command-line interface commands.
- *Agent Builder User's Guide*, SC32-1921

  Explains how to use the Agent Builder for creating monitoring agents and their installation packages, and for adding functions to existing agents.
- *Performance Analyzer User's Guide*, SC27-4004

  Explains how to use the Performance Analyzer to understand resource consumption trends, identify problems, resolve problems more quickly, and predict and avoid future problems.

Documentation for the base agents

If you purchased IBM Tivoli Monitoring as a product, you received a set of *base* monitoring agents as part of the product. If you purchased a monitoring agent product (for example, an OMEGAMON XE product) that includes the commonly shared components of Tivoli Management Services, you did not receive the base agents.

The following publications provide information about using the base agents.
- Operating system agents:
  - *Windows OS Agent User's Guide*, SC32-9445
  - *UNIX OS Agent User's Guide*, SC32-9446
  - *Linux OS Agent User's Guide*, SC32-9447
  - *i5/OS Agent User's Guide*, SC32-9448
  - *UNIX Log Agent User's Guide*, SC32-9471
- Agentless operating system monitors:
  - *Agentless Monitoring for Windows Operating Systems User's Guide*, SC23-9765
  - *Agentless Monitoring for AIX Operating Systems User's Guide*, SC23-9761
  - *Agentless Monitoring for HP-UX Operating Systems User's Guide*, SC23-9763
  - *Agentless Monitoring for Solaris Operating Systems User's Guide*, SC23-9764
  - *Agentless Monitoring for Linux Operating Systems User's Guide*, SC23-9762
- Warehouse agents:
  - *Warehouse Summarization and Pruning Agent User's Guide*, SC23-9767
  - *Warehouse Proxy Agent User's Guide*, SC23-9766

- System P agents:
  - *AIX Premium Agent User's Guide*, SA23-2237
  - *CEC Base Agent User's Guide*, SC23-5239
  - *HMC Base Agent User's Guide*, SA23-2239
  - *VIOS Premium Agent User's Guide*, SA23-2238
- Other base agents:
  - *Systems Director base Agent User's Guide*, SC27-2872
  - *Tivoli Log File Agent User's Guide*, SC14-7484
  - *Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint User's Guide*, SC32-9490

Related publications

You can find useful information about related products in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/.

Other sources of documentation

You can also obtain technical documentation about IBM Tivoli Monitoring and related products from the following sources:

- IBM Tivoli Integrated Service Management Library

  http://www.ibm.com/software/tivoli/opal

  Tivoli Integrated Service Management Library is an online catalog that contains integration documentation and other downloadable product extensions.
- Redbooks

  http://www.redbooks.ibm.com/

  IBM Redbooks and Redpapers include information about products from platform and solution perspectives.
- Technotes

  Technotes provide the latest information about known product limitations and workarounds. You can find Technotes through the IBM Software Support Web site at http://www.ibm.com/software/support.
- Tivoli wikis on the IBM developerWorks® Web site

  Tivoli Wiki Central at http://www.ibm.com/developerworks/wikis/display/tivoli/Home is the home for interactive wikis that offer best practices and scenarios for using Tivoli products. The wikis contain white papers contributed by IBM employees, and content created by customers and business partners.

  Two of these wikis are of particular relevance to IBM Tivoli Monitoring:
  - Tivoli Distributed Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivolimonitoring/Home provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Management products.
  - Tivoli System z Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivoliomegamon/Home provides information about the OMEGAMON XE products, NetView for z/OS, Tivoli Monitoring Agent for z/TPF, and other System z monitoring and application management products.

Appendix J. Additional resources

The following sections include resources and information you can use during your Tivoli Monitoring planning and deployment.

IBM Tivoli Monitoring 6 Welcome Kit

The purpose of this guide is to provide guidelines and reference materials to assist you in getting yourself familiar with the Tivoli Monitoring version 6 product. It was produced with the following objectives in mind:

- Help you to effectively use Tivoli Monitoring
- Obtaining Tivoli Monitoring assistance online
- Provide additional resources for working with Tivoli Monitoring
- Introduce you to the Tivoli Monitoring Support
- Provide an image containing documents and presentations on Tivoli Monitoring

Documents can be viewed at http://www.ibm.com/support/docview.wss?rs=2366&uid=swg21253835 .

General education and support Web sites

Main Tivoli Page
http://www.ibm.com/tivoli

This is the main Tivoli page (formerly www.tivoli.com)

Main Tivoli Support Page
http://www.ibm.com/software/sysmgmt/products/support/

This is the main Tivoli Support page. All Tivoli products are listed in the drop down box in the middle. Choosing a product takes you to the support page dedicated to the product you need help with.

Main Tivoli Education Page
http://www.ibm.com/software/tivoli/education/

IBM Tivoli offers a variety of course types such as: online, on-site and instructor-led education. The courses cover aspects of the Tivoli software portfolio.

Main IBM Redbook Page
http://www.redbooks.ibm.com/redbooks.nsf/redbooks/

IBM Redbooks are developed and published by IBM's International Technical Support Organization. They deliver skills, technical know-how, and materials to technical professionals of IBM, Business Partners, and users and to the marketplace generally.

IBM Certified Advanced Deployment Professional - Tivoli Enterprise Management Solutions
http://www.ibm.com/certify/certs/tvaden04.shtml

An IBM Certified Advanced Deployment Professional - Tivoli Enterprise Management Solutions 2004 is an individual who has demonstrated a higher level of implementation knowledge and skill both in breadth and in depth in the IBM Tivoli Enterprise Management solutions area.

Product documentation and IBM Redbooks

IBM Tivoli Monitoring product documentation
http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/

*Infrastructure Solutions: Building a Smart Bank Operating Environment***, SG24-7113**
http://www.redbooks.ibm.com/abstracts/sg247113.html?Open

Implementing OMEGAMON XE for Messaging 6.0, **SG24-7357**
http://www.redbooks.ibm.com/abstracts/sg247357.html?Open

Best Practices for SOA Management, **REDP-4233**
http://www.redbooks.ibm.com/abstracts/redp4233.html?Open

Deployment Guide Series: IBM Tivoli Monitoring 6.2, **SG24-7444**
http://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg247444.html?Open

Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments, **SG24-7143**
http://www.redbooks.ibm.com/abstracts/sg247143.html?Open

Tivoli Management Services Warehouse and Reporting, **SG24-7290**
http://www.redbooks.ibm.com/abstracts/sg247290.html?Open

IBM Tivoli OMEGAMON XE V3.1 Deep Dive on z/OS, **SG24-7155**
http://www.redbooks.ibm.com/Redbooks.nsf/ad6437adb3f17484852568dd006f956e/b8e023248b2d90718525706c00609acd?OpenDocument

IBM Tivoli Monitoring: Implementation and Performance Optimization for Large Scale Environments, **SG24-7443**
http://www.redbooks.ibm.com/redpieces/abstracts/sg247443.html?Open

Education offerings

A listing of all the current Tivoli Monitoring training can be found at http://www.ibm.com/software/tivoli/education/edu_prd.html#M.

The training road maps for Tivoli Monitoring can be found at http://www.ibm.com/software/tivoli/education/eduroad_prod.html#2.

Support Technical Exchange (STE) Seminars

Expand your technical understanding of your current Tivoli products, in a convenient format hosted by IBM Tivoli Worldwide Support and Services. These live seminars are support oriented discussions of product information, deployment and trouble-shooting tips, common issues, problem solving resources and other support and service recommendations. Tivoli engineers and consultants who are subject matter experts for the product(s) discussed lead each STE. Each STE is recorded and playback is available at anytime. To attend a live STE or review a previously recorded STE go to http://www.ibm.com/software/sysmgmt/products/support/supp_tech_exch.html.

Service offerings

There are several Services offerings for the Tivoli Monitoring product. Access the Services offerings and additional details on some of the offerings at the following link:

http://www.ibm.com/software/tivoli/services/consulting/offers-availability.html#monitoring

IBM QuickStart Services for Tivoli Monitoring
This offering is designed to facilitate ease of deployment and rapid time to value for Tivoli Monitoring, allowing you to begin monitoring and reporting on your essential system resources by providing architecture and design recommendation for production, implementation plan, hands-on training and working test lab prototype using up to six standard resource models.

IBM Migration Assistance Services for Tivoli Monitoring
This new packaged service offering is tailored to help you obtain a clear and applicable understanding in order to migrate an existing Tivoli based monitoring environment to the new Tivoli Monitoring technology.

Tivoli Monitoring On-site

Learn from the experts! This workshop provides three days of formal education course delivery on-site, plus two days of customized training or mentoring specific to your business needs.

Tivoli Common Reporting

The Tivoli Common Reporting tool (TCR) is a reporting feature available to users of Tivoli products. Use Tivoli Common Reporting to gather, analyze, and report important trends in your managed environment in a consistent and integrated manner. A set of predefined reports is provided for the Tivoli Monitoring OS Agents and other products for monitoring individual, multiple, and enterprise resources.

To start using reports complete the following steps:

1. Install and configure the Tivoli Data Warehouse and warehouse agents: Warehouse Proxy Agent and Summarization and Pruning Agent. For more information, see Chapter 24, “Tivoli Data Warehouse solutions: common procedures,” on page 595.
2. Configure historical collection. See *Historical collection configuration* in the *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide*.
3. Prepare the database. See *Creating shared dimensions tables and populating the time dimensions table* and *Creating and populating the resource dimensions table* in the *IBM Tivoli Monitoring: Administrator's Guide*.
4. Install the reports. See *Importing reports using the report installer* in the *IBM Tivoli Monitoring: Administrator's Guide*.
5. Review the reports. See the monitoring agent user's guide for report content specifics.

Complete documentation for the Tivoli Common Reporting tool is located at http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.tcr_cog.doc/tcr_welcome.html.

Other resources

AA&BSM Enablement Best Practices website

http://www.ibm.com/software/tivoli/features/monitoring-best-practices/index.html

Tivoli AA&BSM Technical Exchange Wiki

http://www.ibm.com/developerworks/wikis/display/aabsmenbl/Home

IBM Tivoli Monitoring 6 Forum

http://www.ibm.com/developerworks/forums/dw_forum.jsp?forum=796&cat=15

Appendix K. Support for problem solving

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- "Using IBM Support Assistant"
- "Obtaining fixes" on page 876
- "Receiving weekly support updates" on page 876
- "Contacting IBM Software Support" on page 877

Using IBM Support Assistant

The IBM Support Assistant is a free, stand-alone application that you can install on most workstations and also use to perform remote troubleshooting of other workstations. You can enhance the application by installing product-specific add-ons for the IBM products you use.

The IBM Support Assistant saves you the time it takes to search product, support, and educational resources. Several troubleshooting features are provided, including the ability to perform guided troubleshooting to aid in problem resolution and the ability to collect diagnostic information. The collected diagnostic information can then be used to self-diagnose the problem, or it can be included in an *Electronic Service Request* (ESR) submitted to IBM Support engineers. The ESR tool is used to open, update, and report on PMRs (Problem Management Records) online. See http://www.ibm.com/software/support/help.html for assistance in using the ESR tool.

For more information, and to download the IBM Support Assistant, see http://www.ibm.com/software/support/isa. Currently, the add-on for this product is supported by IBM Support Assistant V4.0.1, or later. After you download and install the IBM Support Assistant, follow these steps to install the IBM Support Assistant add-on for the IBM Tivoli Monitoring product that you are using:

1. Start the IBM Support Assistant application.
2. From the **File → Preferences → Updater preferences** menu, provide the URL to update the site under **Specify an Update Site → Location**.
3. Select **http** from the list.
4. Validate the site and click **OK** to confirm changes.
5. Run **Update → Find new → Product Add-ons**.
6. Select the appropriate plug-in
7. Read the license and description, and if you comply, select **I accept the terms in the license agreements** and click **Next**.
8. Click **Finish** to proceed with the installation, and when prompted, restart the IBM Support Assistant to complete the installation.

To collect the diagnostic files and include them in an ESR that can be sent to IBM Support engineers, view the help files from the Help menu bar. To perform the collection of diagnostic files for self-diagnosis only, complete the following steps:

1. Start the IBM Support Assistant application.
2. From the Home screen, select **Analyze Problem**.
3. In the Select A Collector dialog box, expand the appropriate product name, and select the agent for which you want to collect diagnostic information. Choose **Add**.
4. After the agent or agents are added to the Collector Queue, choose **Collect All** to begin the collection.
5. Enter the information requested in the dialog boxes.

6. The final dialog box requests whether or not you want to upload the collection file to IBM Support or another FTP location. If you only want to view the collected files on your computer, choose **Do Not FTP the Logs.**
7. The collection has finished. You can view the collected files by clicking the compressed file in the **Collector Status** dialog box.

Obtaining fixes

A product fix might be available to resolve your problem. To determine which fixes are available for your Tivoli software product, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/software/support.
2. Under **Select a brand and/or product**, select **Tivoli**.
3. Click the right arrow to view the Tivoli support page.
4. Use the **Select a category** field to select the product.
5. Select your product and click the right arrow that shows the **Go** hover text.
6. Under **Download**, click the name of a fix to read its description and, optionally, to download it.

   If there is no **Download** heading for your product, supply a search term, error code, or APAR number in the field provided under **Search Support (this product)**, and click the right arrow that shows the **Go** hover text.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/handbook.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/software/support.
2. Click **My support** in the far upper-right corner of the page under **Personalized support**.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. The **Edit profile** tab is displayed.
5. In the first list under **Products**, select **Software**. In the second list, select a product category (for example, **Systems and Asset Management**). In the third list, select a product sub-category (for example, **Application Performance & Availability** or **Systems Performance**). A list of applicable products is displayed.
6. Select the products for which you want to receive updates.
7. Click **Add products**.
8. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
9. In the **Documents** list, select **Software**.
10. Select **Please send these documents by weekly email**.
11. Update your e-mail address as needed.
12. Select the types of documents you want to receive.
13. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

Online
   Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

By phone
Call 1-800-IBM-4You (1-800-426-4968).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects. The easiest way to obtain that assistance is to open a PMR or Electronic Service Request (ESR) directly from the IBM Support Assistant (see "Using IBM Support Assistant" on page 875).

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, and DB2 and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage in one of the following ways:

  Online
  Go to the Passport Advantage Web site at http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm .

  By phone
  For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, OS/390, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. "Determining the business impact"
2. "Describing problems and gathering information" on page 878
3. "Submitting problems" on page 878

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Use the following criteria to understand and assess the business impact of the problem that you are reporting:

Severity 1
The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3

The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4

The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- Which software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Submit and track problems** on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. Type your information into the appropriate problem submission form.

By phone

For the phone number to call in your country, go to the contacts page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Appendix L. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM‘s application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Appendix M. Accessibility features for IBM Tivoli Monitoring

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Tivoli Monitoring:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices

The Tivoli Monitoring Information Center and its constituent publications are accessibility-enabled. The accessibility features of the information center are described at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility:

`http://www.ibm.com/able`

Glossary

A

activity. One phase within a sequence of predefined steps called a **policy** that automate system responses to a **situation** that has fired (that is, become true).

administration mode. See “workspace administration mode” on page 895.

Advanced Encryption Standard. An encryption algorithm for securing sensitive but unclassified material designed by the National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce. AES is intended to be a more robust replacement for the **Data Encryption Standard**. The specification calls for a symmetric algorithm (in which the same key is used for both encryption and decryption), using block encryption of 128 bits and supporting key sizes of 128, 192 and 256 bits. The algorithm was required to offer security of a sufficient level to protect data for the next 20 to 30 years. It had to be easily implemented in hardware and software and had to offer good defenses against various attack techniques. AES has been published as Federal Information Processing Standard (FIPS) 197, which specifies the encryption algorithm that all sensitive, unclassified documents must use.

AES. See “Advanced Encryption Standard.”

affinity. A label that classifies **objects** by **managed system**.

agent. Software installed on systems you want to monitor that collects data about an operating system, subsystem, or application running on each such system. Because an executable file gathers information about a **managed system**, there is always a one-to-one correspondence between them. Also called a Tivoli Enterprise Monitoring Agent.

agentless monitor. An agentless monitor uses a standard **API** (such as **SNMP** or **CIM**) to identify and notify you of common problems with the operating system running on a remote computer. Thus, as their name implies, the agentless monitors can retrieve monitoring and **performance** data without requiring OS agents on the computers being monitored. The agentless monitors provide monitoring, data gathering, and event management for Windows, Linux, AIX, HP-UX, and Solaris systems.

agentless monitoring server. A computer that has one or more **agentless monitors** running on it. Each agentless monitoring server can support up to 10 active instances of the various types of agentless monitors, in any combination. Each instance can communicate with up to 100 remote nodes, which means a single agentless monitoring server can support as many as 1000 monitored systems.

alert. A warning message or other indication that appears at a console to indicate that something has occurred or is about to occur that may require intervention.

alert monitor. A **monitoring agent** that monitors and relays alert information to the monitoring server. Sources of alerts include message logs, system consoles, and network and system management products.

algorithm. A set of well-defined rules for the solution of a problem in a finite number of steps. For example, a full statement of an arithmetic procedure for evaluating `sin(x)` to a stated precision.

API. See “Application Programming Interface.”

application. A software component or collection of software components that performs specific user-oriented work (a **task**) on a computer. Examples include payroll, inventory-management, and word-processing applications.

Application Programming Interface. A set of multiple subprograms and data structures and the rules for using them that enables application development via a particular language and, often, a particular operating environment. An API is a functional interface supplied by the operating system or by a separately licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

arithmetic expression. A statement containing any combination of values joined together by one or more arithmetic operators in such a way that the statement can be processed as a single numeric value.

arithmetic operator. A symbol representing a mathematical operation (addition, subtraction, multiplication, division, or exponentiation), such as +, -, *, /, or ^.

associate. The process of linking a **situation** with a **Navigator** item that enables a light to go on and a sound to play for an open **event**. Predefined situations are associated automatically, as are situations created or edited through the Navigator item pop-up menu. When you open the Situation editor from the toolbar, any situations you create cannot be associated with a Navigator item during this editing session. You need to close the Situation editor, and then open it again from the pop-up menu of the Navigator item with which the situation is associated.

attribute. (1) A system or application element being monitored by the **monitoring agent**, such as Disk Name and Disk Read/Writes Per Second. (2) A characteristic of a **managed object**; that is, a field in the data structure of a managed object or in the **workspace** associated with that managed object. (3) A field in an **ODBC**-compliant **database**.

attribute group. A set of related **attributes** that can be combined in a data **view** or a **situation**. When you open the view or start the situation, data samples of the selected attributes are retrieved. Each type of **monitoring agent** has its own set of attribute groups.

B

baroc files. Basic Recorder of Objects in C files define event classes for a particular IBM Tivoli Enterprise Console server. Baroc files also validate event formats based on these event-class definitions.

browser client. The software installed with the **Tivoli Enterprise Portal Server** that is downloaded to your computer when you start the **Tivoli Enterprise Portal** in browser mode. The browser client runs under the control of a Web browser.

C

Candle Management Workstation. The client component of a CandleNet Command Center environment; it provides the graphical user interface. It is replaced by the **Tivoli Enterprise Portal** user interface in the **IBM Tivoli Monitoring** environment.

capacity planning. The process of determining the hardware and software configuration required to accommodate the anticipated **workload** on a system.

chart. A graphical view of data returned from a **monitoring agent**. A data point is plotted for each **attribute** chosen and, for bar and pie charts, a data series for each row. Types of charts include pie, bar, plot, and gauge.

CIM. See “Common Information Model.”

class file. A file containing Java **object** code for a single Java object class.

class loader. A Java component that loads Java **class files**.

client. An application that receives requested data from a **server**.

client/server architecture. An architecture in which the **client** (usually a personal computer or workstation) is the machine requesting data or services and the **server** is the machine supplying them. Servers can be microcomputers, minicomputers, or mainframes. The client provides the user interface and may perform application processing. In **IBM Tivoli Monitoring** the **Tivoli Enterprise Portal** is the client to the **Tivoli Enterprise Portal Server**, whereas the portal server is the client to the **Tivoli Enterprise Monitoring Server**.

A **database server** maintains the databases and processes requests from the client to extract data from or to update the database. An **application server** provides additional business-support processing for the clients.

Common Information Model. An **XML**-based standard for defining device and application characteristics so that system administrators and management programs can monitor and control them using the same set of tools, regardless of their differing architectures. CIM provides a more comprehensive toolkit for such management functions than the **Simple Network Management Protocol**.

Common Object Request Broker Architecture. An industry specification for the design and standardization of different types of object request brokers (ORBs). ORBs allow different computers to exchange object data; CORBA enables ORBs from different software vendors (often running under dissimilar computer systems and operating systems) to exchange **object** data. CORBA facilitates communication among program components in a network using objects. The **Tivoli Enterprise Portal Server** is a CORBA implementation.

condition. An expression that evaluates to either true or false. It can be expressed in natural language text, in mathematically formal notation, or in a machine-readable language.

Configure History permission. Your user ID must have Configure History permission to open the History Collection Configuration window for setting up history files and data **rolloff**. If you do not have this permission, you cannot see the menu item or tool for **historical configuration**.

Configuration Tool, z/OS (ICAT). A REXX-based tool for configuring OMEGAMON XE products running on zSeries systems, after they have been installed using the System Modification Program/Extended (SMP/E) tool.

CORBA. See “Common Object Request Broker Architecture.”

critical state. The indication that a **situation** associated with a **Navigator** item is in an unacceptable state and that you must take corrective action. The critical state is represented by the color red.

Custom Navigator Views permission. Your user ID has a Modify check box for the Custom Navigator Views feature. This permission must be enabled for you to open the **Navigator** view editor to maintain and update Navigator **views**.

D

Data Encryption Standard. A widely used method of private-key data encryption that originated at IBM in 1977 and was adopted by the U.S. Department of Defense. DES supports 72 quadrillion or more possible encryption keys; for each message, the key is chosen at random from among this enormous number of possible keys. Like all other private-key cryptographic methods, both the sender and the receiver must know and use the same private key.

DES applies a 56-bit key to each 64-bit block of data. Although this is considered strong encryption, many companies use triple DES, which applies three keys in succession.

data source name. The name that is stored in the **database server** and that enables you to retrieve information from the database through **ODBC**. The DSN includes such information as the database name, database driver, user ID, and password.

data sources. Data pertaining to J2EE data sources, which are logical connections to **database** subsystems.

data warehouse. A central repository for all or significant parts of the data that an organization's business systems collect.

database. A collection of both interrelated and independent data items that are stored together on a computer disk to serve multiple applications.

DB2 for Linux, UNIX, and Windows. IBM's DB2 Database for Linux, UNIX, and Windows systems is a relational database management system that runs on desktop computers. You install a DB2 **database** on the same system as the **Tivoli Enterprise Portal Server**; it stores the portal server's queries, customized workspaces, user IDs, and custom **Navigator** views. DB2 for Linux, UNIX, and Windows can also serve as the data repository for the **Tivoli Data Warehouse**, which stores historical monitoring information.

default. Pertaining to an **attribute**, **value**, or option that is assumed when none is explicitly specified.

Demilitarized Zone. The area of a World Wide Web application that a company can use to host Internet services without allowing unauthorized access.

Derby. An open-source, public-domain, relational database management system implemented in Java and designed to conform to accepted **database** standards (such as **SQL** and **JDBC**). Derby came about when IBM contributed its Cloudscape database manager to the Apache project and features a small machine footprint. IBM Tivoli Monitoring implements Derby as an embedded database within its **Tivoli Enterprise Portal Server**; in other words, the database is installed with the portal server, and it runs within the portal server's Java virtual machine.

DES. See “Data Encryption Standard.”

desktop client. Software supplied with IBM Tivoli Monitoring that you install on a workstation that you plan to use for interacting with the **Tivoli Enterprise Portal Server** and the **Tivoli Enterprise Monitoring Server**. The desktop **Tivoli Enterprise Portal** client provides the graphical user interface into the **IBM Tivoli Monitoring** network.

detailed attribute name. The name used in formulas, **expert advice**, **Take Action commands**, and headers and footers when referencing a **monitoring agent attribute**. In the Properties and Situation editors, you click **Show Formula**, and then check **Show detailed formula** to see the detailed attribute name.

display item. An **attribute** designated to further qualify a **situation**. With a display item set for a multiple-row **attribute group**, the situation continues to look at the other rows in the **sample** and opens more **events** if other rows qualify. The value displays in the event workspace and in the message log and situation event console views. You can select a display item when building a situation with a multiple-row attribute group.

distribution. The **managed systems** on which the **situation** is running.

DLL. See “Dynamic Link Library.”

DMZ. See “Demilitarized Zone.”

drill down. To access information by starting with a general category and moving through the hierarchy of information, for example, in a database, to move from file to record to field.

DSN. See “data source name.”

Dynamic Link Library. A composite of one or more executable objects that is bound together by a linking procedure and loaded at run time (rather than when the **application** is linked). The code and data in a dynamic link library can be shared by several applications simultaneously. DLLs apply only to Windows operating environments.

E

EIB. See “Enterprise Information Base.”

EIF. See “Event Integration Facility” on page 888.

endcode. You assign endcodes in a **policy** when you connect one **activity** to another. The endcode indicates the result of this activity that triggers the next activity.

Enterprise Information Base. A database used by the **Tivoli Enterprise Monitoring Server** that serves as a repository of shared **objects** for all systems across your enterprise. The EIB stores all **persistent data**, including

situations, **policies**, user definitions, and **managed-object** definitions.

enterprise situation. A **situation** that is created for a Tivoli Enterprise Monitoring Agent that reports **events** to the **Tivoli Enterprise Monitoring Server** to which it connects. Enterprise situations are centrally defined at the monitoring server and distributed at **agent** startup. See also “situation” on page 893.

event. An action or some occurrence, such as running out of memory or completing a transaction, that can be detected by a **situation**. Events cause a change in the state of a **managed object** associated with a situation, thereby make the situation true and causing an **alert** to be issued.

event indicator. The colored icon that displays over a **Navigator** item when an **event** opens for a **situation** running on that item.

Event Integration Facility. An **application programming interface** that external applications can invoke to create, send, or receive IBM Tivoli Enterprise Console **events**. These events are referred to as either **EIF events** or **TEC/EIF events**.

event item. A **Navigator** item that shows when you open the event **workspace** for a true **situation** (by selecting it from the **event** flyover listing or from the situation event console pop-up menu).

event sound. The sound file that plays when an event opens. This sound file is set in the Situation editor when the **situation** is associated with a **Navigator** item and can differ for different Navigator items.

expert advice. A description within the Situation editor of each **situation** provided with a **monitoring agent** to help you quickly understand and interpret **events** arising from it.

Extensible Markup Language. A data-description language derived from Standard Generalized Markup Language (SGML); also a tool for encoding messages so they describe their own fields. You use XML to format a document as a data structure. As program **objects**, such documents can have their contents and data hidden within the object, which allows you to control who can manipulate the document and how. In addition, documents can carry with them the object-oriented procedures called **methods**. The XML standard aids in exchanging data between applications and users.

F

filter criteria. These criteria limit the amount of information returned to the data view in response to a **query**. You can apply a prefilter to the query to collect only certain data, or apply a postfilter to the **view** properties to show only certain data from the information collected.

fix pack. A tested collection of all cumulative maintenance for a product, up to the release of the fix pack. It can also contain fixes that have not been shipped previously, but it might contain no new function.

G

georeferenced map. A special type of graphic that has built-in knowledge of latitude and longitude and can be zoomed into and out of quickly. The **Tivoli Enterprise Portal** uses proprietary .IVL files generated with the map-rendering component. These files cannot be opened or saved in a graphics editor.

GSKit. The Global Security Toolkit provides **SSL** (Secure Sockets Layer) processing within protocols such as **SPIPE** and **HTTPS**. On z/OS systems, GSKit is known as the Integrated Cryptographic Service Facility, or **ICSF**.

H

historical collection. A definition for collecting and storing data samples for historical reporting. The historical collection identifies the **attribute group**, any row filtering you have assigned, the **managed system distribution**, frequency of data collection, where to store it for the short term, and whether to save data long term (usually to the **Tivoli Data Warehouse**).

historical data management. The procedures applied to short-term binary history files that **roll off** historical data to either the **Tivoli Data Warehouse** or to delimited text files (the krarloff utility on UNIX or Windows systems; ddname KBDXTRA for the z/OS **Persistent Datastore**), and then delete entries in the short-term history files over 24 hours old, thereby making room for new entries.

hot standby. A redundant **Tivoli Enterprise Monitoring Server** that, if the primary or **hub monitoring server** should fail, assumes the responsibilities of the failed monitoring server.

HTTP. The Hypertext Transfer Protocol is a suite of Internet protocols that transfer and display hypertext documents within Web browsers.

HTTP sessions. Data related to invocations of specific World Wide Web sites.

HTTPS. The Secure Hypertext Transport Protocol is an implementation of the Hypertext Transport Protocol (**HTTP**) that relies on either the Secure Sockets Layer (**SSL**) API or the Transport Layer Security (TLS) API to provide your users with secure access to your site's

Web server. These APIs encrypt and then decrypt user page requests as well as the pages returned by the Web server.

hub Tivoli Enterprise Monitoring Server. (1) A central host system that collects the status of situations running on your systems. (2) The monitoring server that your site has selected to act as the focal point to which all portal servers and remote monitoring servers in this monitored network connect. A **remote** monitoring server passes its collected data to the hub to be made available to clients, creating an enterprise-wide view.

I

IBM Tivoli Monitoring. A **client/server** implementation for monitoring enterprise-wide computer networks that comprises a **Tivoli Enterprise Monitoring Server**, an application server known as the **Tivoli Enterprise Portal Server**, one or more **Tivoli Enterprise Portal** clients, and multiple **monitoring agents** that collect and distribute data to the monitoring server.

IIOP. See " Internet Inter-ORB Protocol."

input data. Data provided to the computer for further processing. See also "output data" on page 891.

integral Web server. A proprietary Web server developed for **IBM Tivoli Monitoring** that is installed and configured automatically with the **Tivoli Enterprise Portal Server**. You enter the URL of the integral Web server to start the **Tivoli Enterprise Portal** client in **browser mode**.

Internet Inter-ORB Protocol. An Internet communications protocol that runs on distributed platforms. Using this protocol, software programs written in different programming languages and running on distributed platforms can communicate over the Internet.

IIOP, a part of the **CORBA** standard, is based on the **client/server** computing model, in which a **client** program makes requests of a **server** program that waits to respond to client requests. With IIOP, you can write client programs that communicate with your site's existing server programs wherever they are located without having to understand anything about the server other than the service it performs and its address (called the Interoperable Object Reference, **IOR**, which comprises the server's port number and IP address).

Interoperable Object Reference. Connects **clients** to the **Tivoli Enterprise Portal Server**. The IOR identifies a remote **object**, including such information as name, capabilities, and how to contact it. The URL may include an IOR because it goes through the Web server; the portal server uses it to tell the client which IOR to fetch. After it does that, the portal server extracts the host and port information and tells the client where to route the request.

interval. The number of seconds that have elapsed between one **sample** and the next.

IOR. See "Interoperable Object Reference."

J

Java Database Connectivity. A standard **API** that application developers use to access and update relational **databases** (RDBMSes) from within Java programs. The JDBC standard is based on the X/Open SQL Call Level Interface (CLI) and complies with the SQL-92 Entry Level standard; it provides a DBMS-independent interface that enables **SQL**-compliant database access for Java programmers.

Java Management Extensions. A set of Java classes for application and network management in J2EE environments. JMX provides Java programmers a set of native Java tools called **MBeans** (managed beans) that facilitate network, device, and application management. JMX provides a Java-based alternative to the **Simple Network Management Protocol**.

JDBC. See "Java Database Connectivity."

JMX. See "Java Management Extensions."

L

LDAP. See "Lightweight Directory Access Protocol."

Lightweight Directory Access Protocol. A protocol that conforms to the International Standards Organization's X.500 directory standard that uses **TCP/IP** to access directory **databases** where **applications** can store and retrieve common naming and location data. For example, applications can use LDAP to access such directory information as email addresses, service configuration parameters, and public keys.

location broker. The component that manages connections for the hub monitoring server, enabling it to find all other **Tivoli Management Services** components, including **remote monitoring servers**, the **Tivoli Enterprise Portal Server**, and **monitoring agents**.

M

managed object. An icon created in the **Tivoli Enterprise Portal** from a managed object template that represents resources you monitor using **situations**. Managed objects are converted to items in the **Navigator's** Logical view.

managed system. A particular operating system, subsystem, or application in your enterprise where a

monitoring agent is installed and running. A managed system is any system that IBM Tivoli Monitoring is monitoring.

managed system group. (Formerly **managed system list**.) A named, heterogeneous group of both similar and dissimilar **managed systems** organized for the distribution of **historical collections**, **situations**, and **policies**, and for assignment to **queries** and items in custom **Navigator** views. For example, you might create a managed system group named IT_London for a geographic region and another named Credit_Approval for a functional area of your organization.

If a managed system group is updated (usually when a constituent managed system is added or deleted), then all the historical collections, situations, and policies that use that group are redistributed to all managed systems in the group. Managed system groups are created, modified, or deleted either by the **Tivoli Enterprise Portal's** Object Group editor or via the **tacmd** CLI command with the **createsystemlist**, **editsystemlist**, or **deletesystemlist** keywords; they are maintained by the **Tivoli Enterprise Monitoring Server**.

MBeans. Managed beans are Java **objects** that represent managed resources such as devices, services, and **applications**. The management functions are provided by the **MBean server**.

Microsoft Management Console. This feature of Microsoft's various Windows Server environments provides a centralized, consistent, and extensible interface to Windows' various monitoring and management utilities. In particular, MMC manages directory services, job scheduling, event logging, performance monitoring, and user environments.

middleware. Software that enables the exchange of information between components in a distributed computing environment. The middleware is the data-exchange and communications channel that allows programs to cooperate with each other without having to know details about how they are implemented or where they are deployed. Middleware typically provides a range of related facilities such as persistence, auditing, and the ability to build a transactional unit of work. IBM's CICS and WebSphere MQ are examples of middleware.

method. In object-oriented programming, the software that implements an object's behavior as specified by an operation.

migrating. Preserving your customized configuration data so you can use it again after installing a newer version of the product.

MMC. See "Microsoft Management Console."

monitor. An entity that performs measurements to collect data pertaining to the **performance**, availability, reliability, or other **attributes** of **applications** or the systems on which those applications rely. These measurements can be compared to predefined **thresholds**. If a threshold is exceeded, administrators can be notified, or predefined automated responses can be performed.

monitor interval. A specified time, scalable to seconds, minutes, hours, or days, for how often the monitoring server checks to see if a **situation** has become true. The minimum monitor interval is 30 seconds; the default value is 15 minutes.

monitoring. Running a hardware or software tool to **monitor** the **performance** characteristics of a system.

N

NAT. See "Network Address Translation."

Navigator. The upper-left pane of the **Tivoli Enterprise Portal** window. The Navigator Physical view shows your network enterprise as a physical hierarchy of systems grouped by platform. You can also define other views that create logical hierarchies grouped as you specify, such as by department or function.

Network Address Translation. A scheme used by local-area networks (LANs) to establish an internal and external set of IP addresses. Internal IP addresses are kept private and must be translated to and from the external addresses for outbound and inbound communications. NAT is often used in firewall configurations.

Network File System. A **client/server** file system developed by Sun Microsystems that, once mounted (that is, made accessible), allows a user on an NFS **client** to view, store, and update files on a remote computer (the NFS **server**) as though they were on the user's own computer. The portion of the mounted file system that each user can access and in what ways is determined by the user's own file-access privileges and restrictions.

Both the NFS server and client use **TCP/IP's User Datagram Protocol** as the mechanism for sending file contents and updates back and forth. NFS has been designated a file server standard; it uses the **Remote Procedure Call** method of communication between computers.

NFS. See "Network File System."

node. (1) In networking, a point capable of sending and receiving data. A node can be a device, such as printer or workstation, a system, a storage location on a disk, or a single computer. (2) Any managed system, such as an AIX-based pSeries **server**, that IBM Tivoli Monitoring is monitoring. A node can also be a **managed system** of subnodes, all of which are being managed as components of the primary node.

non-agent bundles. You can use these custom bundles to **remotely deploy** components that need not connect to a **Tivoli Enterprise Monitoring Server**, such as those that support other Tivoli products like IBM Tivoli Netcool/OMNIbus.

O

object. An instance of a **class**, which comprises an implementation and an interface. An object reflects its original, holding data and methods and responds to requests for services. **CORBA** defines an object as a combination of state and a set of methods characterized by the behavior of relevant requests.

ODBC. See “Open Database Connectivity.”

OMEGAMON Monitoring Agent. The software process that probes a managed z/OS system or subsystem (such as CICS) for data. The **monitoring agent** sends that monitoring information back to the **Tivoli Enterprise Monitoring Server** and then on to the **Tivoli Enterprise Portal Server** to be formatted into table and chart **views** for display on a **Tivoli Enterprise Portal client**.

OMEGAMON Dashboard Edition (OMEGAMON DE). The OMEGAMON implementation that includes all the features of the **Tivoli Enterprise Portal** provided with **OMEGAMON XE**, plus application-integration components that facilitate an enterprise-wide view of your computing environment. OMEGAMON DE's **workspaces** integrate the data from multiple **OMEGAMON Monitoring Agents** into one network-wide view.

OMEGAMON Extended Edition (OMEGAMON XE). The IBM Tivoli Monitoring implementation of a single **OMEGAMON Monitoring Agent**. OMEGAMON XE displays the monitoring data from each OMEGAMON Monitoring Agent independently, without integrating it into the enterprise-wide **workspaces** provided by OMEGAMON DE.

OMEGAMON Tivoli Event Adapter. Invokes the **Event Integration Facility** API to synchronize IBM Tivoli Monitoring events with the IBM Tivoli Enterprise Console product. OTEA is a component of the **Tivoli Enterprise Monitoring Server**; it forwards IBM Tivoli Monitoring **events** to Tivoli Enterprise Console and maps them to their corresponding Tivoli Enterprise Console event classes based on the **situation** name's suffix, either `_Warning` or `_Critical`.

Integrating these products requires two parts: a Tivoli Enterprise Monitoring Server piece (included with IBM Tivoli Monitoring version 6.1 and subsequent releases) called the OMEGAMON Tivoli Event Adapter, and a Tivoli Enterprise Console piece called the **Situation Update Forwarder** that is installed on the Tivoli Enterprise Console server.

Open Database Connectivity. A standard **API** for accessing data in both relational and nonrelational **database** systems using procedural, non-object-based languages such as C. Using this API, database **applications** can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and programming interface.

Tivoli Enterprise Portal users can access the Query editor to write custom **SQL** queries for creating **views** that retrieve data from ODBC-compliant databases.

OTEA. See “OMEGAMON Tivoli Event Adapter.”

output data. Data resulting from computer processing. See also “input data” on page 889.

parameter. A **value** or reference passed to a function, command, or program that serves as **input** or to control actions. The value is supplied by a user or by another program or process.

P

PDS. See “Persistent Datastore.”

PerfMon. See “Performance Monitor”

performance. A major factor in measuring system productivity. Performance is determined by a combination of throughput, response time, and availability.

Performance Monitor (PerfMon). The Windows Performance Monitor is an SNMP-based performance-monitoring tool for Windows environments. PerfMon monitors network elements such as computers, routers, and switches.

Persistent Datastore. A set of z/OS data sets where IBM Tivoli Monitoring running on z/OS systems stores **historical monitoring data**.

platform. The operating system on which the **managed system** is running, such as z/OS or Linux. The **Navigator's** Physical mapping places the platform level under the Enterprise level.

policy. A set of automated system processes that can perform actions, schedule work for users, or automate manual **tasks**, frequently in response to **events**. Policies are the IBM Tivoli Monitoring automation tool; they comprise a series of automated steps, called **activities**, whose order of execution you control.

In most cases, a policy links a **Take Action command** to a **situation** that has turned true. When started, the policy's **workflow** progresses until all activities have been completed or until the **Tivoli Enterprise Portal** user manually stops the policy. You can create both policies that fully automate workflow strategies and those that require user intervention. As with situations, policies are distributed to the **managed systems** you want to monitor and to which you are sending commands.

private situation. A situation that is defined in an XML-based private configuration file for the local Tivoli Enterprise Monitoring Agent or Tivoli System Monitor Agent and that does not interact with a **Tivoli Enterprise Monitoring Server**. Such **events** can be sent via either **EIF** or **SNMP alerts** to a receiver such as IBM Tivoli Enterprise Console or Netcool/OMNIbus. See also “situation” on page 893.

product code. The three-letter code used by IBM Tivoli Monitoring to identify the product component. For example, the product code for IBM Tivoli Monitoring for WebSphere Application Server is KWE.

Properties editor. A multi-tabbed window for specifying the properties of the individual **views** that make up a **workspace**, as well as the general workspace properties.

pure event. A pure event is one that occurs automatically, such as when a paper-out condition occurs on the printer or when a new log entry is written. **Situations** written to notify you of pure events remain true until they are manually closed or automatically closed by an UNTIL clause. See also “event” on page 888.

Q

query. A particular view of specified **attributes** of selected instances of a set of **managed-object** classes, arranged to satisfy a user request. Queries are written using **SQL**.

R

remote deployment. Using IBM Tivoli Monitoring software, you can deploy agents and other non-agent, **Tivoli Management Services**-based components to remote **nodes** without your having to sign onto those nodes and perform the installation and configuration steps yourself. Remote deployment requires two pieces on the destination node: (1) a bundle containing the component code and the instructions for installing and configuring it and (2) an operating-system agent to read the bundle and perform the installation and configuration steps.

Remote Procedure Call. A protocol based on the Open Software Foundation's Distributed Computing Environment (DCE) that allows one program to request services from a program running on another computer in a network. RPC uses the **client/server** model: the requesting program is the **client**, and the responding program is the **server**. As with a local procedure call (also known as a **function call** or a **subroutine call**), an RPC is a synchronous operation: the requesting program is suspended until the remote procedure returns its results.

remote Tivoli Enterprise Monitoring Server. A remote monitoring server collects **monitoring** data from a subset of your site's monitoring agents and passes its collected data to the **hub Tivoli Enterprise Monitoring Server** to be made available to one or more **Tivoli Enterprise Portal** clients through the **Tivoli Enterprise Portal Server**, thereby creating an enterprise-wide view.

rolloff. The transfer of **monitoring** data to a **data warehouse**.

RPC. See “Remote Procedure Call.”

RTE. See “runtime environment.”

runtime environment. A group of execution libraries that provide an operational environment on a z/OS system. RTEs execute OMEGAMON products on a z/OS image.

runtime libraries. Libraries in the **runtime environment** that the product uses when it is started and running.

S

sample. The data that the **monitoring agent** collects for the monitoring server instance. The **interval** is the time between data samplings.

sampled event. Sampled events happen when a **situation** becomes true. Situations sample data at regular intervals. When the situation becomes true, it opens an **event**, which gets closed automatically when the situation goes back to false (or when you close it manually). See also “event” on page 888.

Secure Sockets Layer. A security protocol for communication privacy that provides secure **client/server** conversations. SSL provides transport layer security (authenticity, integrity, and confidentiality) for a secure connection between a **client** and a **server**.

seed data. The product-provided **situations**, templates, **policies**, and other **sample** data included with a **monitoring agent** to initialize the **Tivoli Enterprise Monitoring Server's Enterprise Information Base**. Before you can use a monitoring agent, the monitoring server to which it reports must be seeded, that is, initialized with **application** data.

server. An application that satisfies data and service requests from **clients**.

SELinux. The National Security Agency's security-enhanced Linux (SELinux) is a set of patches to the Linux kernel plus utilities that together incorporate a strong, flexible mandatory access control (MAC) architecture into the kernel's major subsystems.

SELinux enforces the separation of information based on confidentiality and integrity requirements, which allows attempts to tamper with or bypass application security mechanisms to be recorded and enables the confinement of damage caused by malicious or flawed applications.

Simple Network Management Protocol. A TCP/IP transport protocol for exchanging network management data and controlling the monitoring of network **nodes** in a **TCP/IP** environment. The SNMP software protocol facilitates communications between different types of networks. IBM Tivoli Monitoring uses SNMP messaging to discover the devices on your network and their availability.

Simple Object Access Protocol. The Simple Object Access Protocol is a lightweight, **XML**-based interface that vendors use to bridge **remote procedure calls** between competing systems. SOAP makes it unnecessary for sites to choose between **CORBA**/Java/EJB and Microsoft's COM+.

Because XML and SOAP are platform- and language-neutral, users can mix operating systems, programming languages, and **object** architectures yet maintain business-component interoperability across platforms: using SOAP, **applications** can converse with each other and exchange data over the Internet, regardless of the **platforms** on which they run.

situation. The set of monitored conditions running on a **managed system** that, when met, creates an **event**. A situation comprises an **attribute**, an operator such as greater than or equal to, and a **value**. It can be read as "If *system_condition* compared_to *value* is_true". An example of a situation is: If CPU_usage > 90% TRUE. The expression "CPU_usage > 90%" is the **situation condition**.

Situation Update Forwarder. The Situation Update Forwarder is a Java- and **CORBA**-based background process for communication between IBM Tivoli Enterprise Console and a particular **Tivoli Enterprise Monitoring Server** running under IBM Tivoli Monitoring version 6.1 and subsequent releases. Your site must install this component on the Tivoli Enterprise Console server; for instructions, see the *IBM Tivoli Enterprise Console Installation Guide*.

SNMP. See "Simple Network Management Protocol."

SOAP. See "Simple Object Access Protocol."

sockets. Refers to the sockets method of passing data back and forth between a networked **client** and **server** or between program layers on the same computer.

sound. The **WAV file** that plays whenever a **situation** becomes true for the current **Navigator** item. Sound is assigned to the Navigator item for a situation in the same way a **state** is assigned.

SPIPE. A secure pipe is an implementation of the Internet Protocol's pipe specification that uses the **Secure Sockets Layer** API. Using SPIPE, your corporate Web server can securely access internal servers that are not based on the **HTTPS** protocol, while retaining their ability to process HTTP requests.

SQL. See "Structured Query Language."

SSL. See "Secure Sockets Layer" on page 892.

SSM. See "System Service Monitors" on page 894.

state. The severity of the **situation event**: critical, warning, or informational. Indicated by a colored event indicator, state is set in the Situation editor and can be different for different **Navigator** items.

status. The true or false condition of a **situation**.

Structured Query Language. A standards-based programming language for extracting information from and updating information within a relational **database**. The **Query editor** enables you to write SQL **queries** to **ODBC data sources** for retrieval and display in table and **chart views**.

subnetwork. A configuration wherein a single IP network address is split up so it can be used on several interconnected local networks. Subnetworking is a local configuration; outside it appears as a single IP network.

SUF. See "Situation Update Forwarder."

Summarization and Pruning Agent. One of the IBM Tivoli Monitoring base agents, the Summarization and Pruning Agent keeps the **data warehouse** from growing too large by summarizing and pruning your stored historical data at intervals you set. For every **attribute group** that has data collection configured, you specify how often to aggregate (summarize) data in the **Tivoli Data Warehouse** and the length of time to delete (prune) data from the warehouse.

symbol. Represents a variable that can be added to header or footer text for data **views**, **expert-advice** text, or **query** specification. The detailed **attribute** name is enclosed in dollar signs, such as **$ORIGINNODE$**, and resolves to the attribute's value. For **Tivoli Enterprise Monitoring Server** queries, `== $NODE$` specifies the **managed systems** from which to retrieve data. For queries to be used in link target **workspaces**, you can create symbols for attributes using the *$symbolname$* format.

System Monitor Agent. These **agents** were introduced with IBM Tivoli Monitoring V6.2.2 for **nodes** that run the desktop operating systems (Windows, Linux, UNIX). These agents operate only autonomously (that is, they run without a connection to a **Tivoli Enterprise Monitoring Server**) and pass **SNMP** trap data about operating system **performance** to an SNMP Event Collector such as IBM Tivoli Netcool/OMNIbus's MTTRAPD receiver.

No other IBM Tivoli Monitoring agents or other components should be installed on the same node as a System Monitor Agent. The only exception to this rule is agents created with the Agent Builder tool for V6.2.2 or subsequent.

System Service Monitors. The IBM Tivoli Netcool/OMNIbus product provides System Service Monitors that support basic system-level monitoring of network components such as operating systems. In addition, OMNIbus provides ISMs (Internet Service Monitors) and ASMs (Application Service Monitors).

T

Take Action. A **Tivoli Enterprise Portal** dialog box from which you can enter a command or choose from a list of predefined commands. It also lists systems on which to effect the command, which is usually a response to an **event**.

Take Action command. A Take Action command allows you to send commands to your **managed systems**, either automatically, in response to a **situation** that has fired (that is, turned true), or manually, as the **Tivoli Enterprise Portal** operator requires. With Take Action commands, you can enter a command or select one of the commands predefined by your product and run it on any system in your managed network. Thus you can issue Take Action commands either against the managed system where the situation fired or a different managed system in your network.

target libraries. SMP/E-controlled libraries that contain the data installed from the distribution media.

task. A unit of work representing one of the steps in a process.

TCP/IP. See “Transmission Control Protocol/Internet Protocol.”

TDW. See “Tivoli Data Warehouse.”

telnet. A terminal emulation program used on **TCP/IP** networks. You can start a telnet session with another system and enter commands that execute on that system. A valid user ID and password for that remote system are required.

threshold. (1) A level set in the system at which a message is sent or an error-handling program is called. For example, in a user auxiliary storage pool, the user can set the threshold level in the system values, and the system notifies the system operator when that level is reached. (2) A customizable value for defining the acceptable tolerance limits (maximum, minimum, or reference limit) for an application resource or system resource. When the measured value of the resource is greater than the maximum value, less than the minimum value, or equal to the reference value, an exception is raised.

Tivoli Data Warehouse. This member of the IBM Tivoli Monitoring product family stores Tivoli Monitoring **agents' monitoring** data in separate relational **database** tables so you can analyze historical trends using that enterprise-wide data. Reports generated from Tivoli Data Warehouse data provide information about the availability and **performance** of your monitored environment over different periods of time.

Tivoli Enterprise Monitoring Server. The host data-management component for IBM Tivoli Monitoring. It receives and stores data from either the **agents** or other monitoring servers.

Tivoli Enterprise Portal. The **client** component of a Tivoli Enterprise Portal Server. The Tivoli Enterprise Portal provides the graphical user interface into monitoring data collected by the Tivoli Enterprise Monitoring Server and prepared for user display by the portal server. The Tivoli Enterprise Portal comes in two versions: the **desktop client** and the **browser client**.

Tivoli Enterprise Portal Server. The server you log onto and connect to through the **Tivoli Enterprise Portal client**. The portal server connects to the **hub Tivoli Enterprise Monitoring Server**; it enables retrieval, manipulation, and analysis of data from your enterprise's **monitoring agents**.

Tivoli Enterprise Web Services. An open standards-based interface to the monitoring server that uses **SOAP** requests. Using SOAP, any **monitoring agent** can be dynamically **queried**, which means that its **performance** and availability data can be processed by external **applications** not a part of IBM Tivoli Monitoring.

Tivoli Management Services. An integrated, layered architecture consisting of data-access, communication, and presentation components that enable cross-platform operation and integration of enterprise-wide data for systems-management applications. The software foundation that supports the development and operations of the **Tivoli Enterprise Monitoring Server**, the **Tivoli Enterprise Portal Server** and **Tivoli Enterprise Portal**, and their **monitoring agents**.

Transmission Control Protocol/Internet Protocol. An open, portable communications protocol that is the software basis for the Internet.

TSO. Time Sharing Option, the interactive interface into the **z/OS** operating system.

U

User Datagram Protocol. A **TCP/IP** communications protocol that exchanges messages ("datagrams") between networked computers linked by the Internet Protocol (IP). UDP is an alternative to the Transmission Control Protocol (TCP), which, like UDP, uses IP to move a message from one computer to another. Unlike TCP, however, UDP does not divide the message into packets and reassemble them at the other end.

The **Network File System** uses UDP to move file contents and file updates between the NFS **server** and the NFS **client**.

UDP. See “User Datagram Protocol” on page 894.

V

value of expression. A function in a **situation** condition, **query** specification, or data **view filter** or **threshold** that uses the raw value of an **attribute**. A value can be a number, text string, attribute, or modified attribute. Use this function with any operator.

view. A window pane, or frame, in a **workspace**. It may contain data from an **agent** in a chart or table, or it may contain a terminal session or browser, for example. A view can be split into two separate, autonomous views.

W

Warehouse Proxy Agent. One of the IBM Tivoli Monitoring base agents, the Warehouse Proxy Agent passes **historical monitoring data** from either a **monitoring agent** or the **Tivoli Enterprise Monitoring Server** to the **Tivoli Data Warehouse**. This multithreaded **server** process can handle concurrent requests from multiple data sources to **roll off** data from their short-term history files to the **data warehouse**.

WAV file. Waveform audio format for storing sound in files, developed jointly by Microsoft and IBM.

wildcard. An asterisk (*) used to represent any characters that may follow or precede those entered, such as Sys* to find System and SysTray. Used in formulas with the VALUE function or MISSING function (in the Missing Task List). Used also with the SCAN function, but at the beginning of the text as in *Z to find markZ and typeZ.

Windows Management Instrumentation. Microsoft's Windows Management Instrumentation **API** provides a toolkit for managing devices and **applications** in a network of Windows-based computers. WMI provides both the data about the **status** of local or remote computer systems and the tools for controlling the data. WMI is included with the Windows XP and Windows Server 2003 operating systems.

WMI. See “Windows Management Instrumentation.”

workload. A percentage that shows how much of its resources a **managed system** is using. Workload is calculated using a weighted combination of resource use statistics.

workspace. The viewing area of the **Tivoli Enterprise Portal** window, excluding the **Navigator**. Each workspace comprises one or more **views**. Every Navigator item has its own default workspace and may have multiple workspaces.

workspace administration mode. A global parameter set in the Administer Users editor but which is available only for user IDs with administrator authority. When enabled for a user ID, customization of **workspaces**, links, and terminal-session scripts automatically becomes available to all users connected to the same **Tivoli Enterprise Portal Server**.

X

XML. See “Extensible Markup Language” on page 888.

Z

z/OS. IBM's operating system for its line of mainframe, zSeries computers known for its processing speed and its ability to manage large amounts of memory, direct-access storage, and data.

Index

Printed in USA

GC32-9407-05