IBM Tivoli Monitoring
Version 6.3 Fix Pack 2

Administrator's Guide

SC22-5446-01

IBM Tivoli Monitoring

Version 6.3 Fix Pack 2

Administrator's Guide

SC22-5446-01

Note

Before using this information and the product it supports, read the information in "Notices" on page 609.

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

Contents

Chapter 16. Managing historical data 457

Chapter 17. Tivoli Common Reporting 519

Chapter 18. Replicating the Tivoli Enterprise Portal Server database . . 547

Figures

Tables

About this information

The *IBM®Tivoli® Monitoring Administrator's Guide* describes the administration of your IBM Tivoli Monitoring infrastructure, Tivoli Management Services.

The chapter topics cover the following tasks:

- Configuring, customizing, and maintaining the Tivoli Enterprise Portal clients and server
- Setting up asymmetric encryption using public-private key files
- Enabling user authentication on the hub monitoring server system registry or an external LDAP registry
- Maintaining user IDs and user groups on the Tivoli Enterprise Portal
- Integrating the situation event activities between the IBM Tivoli Enterprise Console® event server or the Netcool/OMNIbus Probe for Tivoli EIF and the hub monitoring server
- Configuring connectors for the event systems that send event information to the Tivoli Enterprise Portal
- Using the Tivoli Enterprise Portal to maintain agents that support the remote agent deployment feature
- Configuring Tivoli Enterprise Monitoring Agents for autonomous operation
- Setting up and enabling Centralized Configuration
- Managing historical data collection and the Tivoli Data Warehouse
- Importing reports for Tivoli Common Reporting that are unique to products that run on theTivoli Enterprise Portal and use the Tivoli Data Warehouse as the source of historical data for generating reports. This information is intended for the administrator who sets up Tivoli Common Reporting and installs report packages
- Replicating the Tivoli Enterprise Portal Server database to another computer or to keep as a backup
- Using IBM Tivoli Monitoring Web Services SOAP methods to query and control your monitored environment

Users of this book should be familiar with performance monitoring concepts and administration. If you use the Tivoli Data Warehouse, you must be familiar with the operating system that hosts the warehouse. To learn more about this family of products, see Tivoli solutions for Service Availability and Performance Management.

Chapter 1. Introduction

This chapter reviews the new features and enhancements to the Tivoli Enterprise Portal interface and Tivoli Management Services administrative features, followed by a list of the administrative tasks you can expect to perform.

For information on how to use the Version 6.3 Tivoli Enterprise Portal features, please consult the integrated help (Help → Contents and Index) or the *Tivoli Enterprise Portal User's Guide*.

New in this release

Review the latest enhancements to the Tivoli Enterprise Portal and to the Tivoli Management Services components that are relevant to the *IBM Tivoli Monitoring Administrator's Guide*.

New in Version 6.3 Fix Pack 2

The following enhancements to the Tivoli Management Services components affect the system administrator for Version 6.3 Fix Pack 2.

Using analytics on your historical data

You can now run external applications against your historical data, such as an analytic engine, to be alerted and help predict potential problems quickly. You must configure your Warehouse Proxy agent and Tivoli Enterprise Monitoring Agent to store historical data in comma-separated (CSV) files that can be processed by an analytic engine. For more information, see "Using your historical data for analytic usage" on page 510.

Update the EIF configuration file and EIF event mapping file in the configuration load list

You can now use the Activate="YES" variable to replace the existing configuration file with a new configuration file that is downloaded from a central configuration server. You can use this variable to replace the EIF configuration file (EIFCNFG) and EIF event mapping file (EIFMAP) in the configuration load list. For more information, see "Configuration load list XML specification" on page 429.

Delete all existing private situations and add new private situations

In the <PRIVATECONFIGURATION> element, you can use the REFRESH= variable to stop and delete all existing private situations and then add only the newly inputted private situations. For more information, see "Private situation XML specification" on page 336.

Configure summarization and pruning for managed system groups

You can now configure summarization and pruning differently for managed system groups. Previously, it was only possible to configure summarization and pruning for a given attribute group, so summarization and pruning had to be global. This new feature allows you to have configurations that address specific business requirements, such as development and test machines that might need to be monitored with smaller retention settings than production systems. For more information, see "Configuring summarization and pruning for managed system groups" on page 488.

Creating and maintaining the dimension tables required for Tivoli Common Reporting and IBM Tivoli Monitoring reports

- Using the Summarization and Pruning agent

  In IBM Tivoli Monitoring V6.3, you had to use the Schema Publication Tool to create the dimension tables in the Tivoli Data Warehouse. Now you can configure the Summarization and Pruning agent to create and maintain the dimension tables. For more information, see "Configuring the Summarization and Pruning agent to create and maintain the dimension tables" on page 525.
- Using the Schema Publication Tool

  You can use the SQL scripts created by the Schema Publication Tool to create both the time dimension tables and the resource dimension tables, or you can create just the time dimension tables, or just the resource dimension tables. Each set of tables can be created using a different connection to the Tivoli Data Warehouse. For more information, see the "Creating the dimension tables using the Schema Publication Tool" on page 528.

Enabling NIST SP800-131a encryption

The National Institute of Standards and Technology (NIST) Special Publications (SP) 800-131a standard strengthens algorithms and increases the key lengths to improve security. To enable SP800-131a encryption, see "Enabling SP800-131a for IBM Tivoli Monitoring" on page 232.

Setting up load balancing for a high availability dashboard environment

If you want to create a high availability dashboard environment, you can use an HTTP server configured for load balancing between your dashboard users and a cluster of Dashboard Application Services Hub nodes, and also between Dashboard Application Services Hub nodes and multiple Tivoli Enterprise Portal Servers. For more information, see "Setting up load balancing for a high availability dashboard environment" on page 53.

Infrastructure Management Dashboards for Servers new tabs and historical reporting

To broaden the Server Dashboards reporting features, several tabs have been added to the managed system dashboard: Properties, CPU, Memory, Disk, and Network. The CPU, Memory, Disk, and Network tabs have a new time selector bar for changing from real time to showing historical data; the situation event results dashboard Details tab has a time selector bar for setting a time range of data samples before or after the event time.

For more information, click the "Learn more" link from the Server Dashboards or see the Infrastructure Management Dashboards for Servers Reference on IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm).

Agent autonomy and centralized configuration information reorganization

The agent autonomy and centralized configuration information previously appeared in separate chapters. This information is now consolidated into the Chapter 15, "Agent-based services," on page 325 chapter.

New in Version 6.3

The following enhancements to the Tivoli Management Services components affect the system administrator for Version 6.3.

Jazz™ for Service Management

Jazz for Service Management brings together the Open Services for Lifecycle Collaboration (OSLC) community's open specifications for linking data, shared administrative services, dashboard and reporting services. Through these facets, Jazz for Service Management accelerates deployment, integration, and workflow automation across IBM, partner, and third party tools. Jazz for Service Management is included with IBM Tivoli Monitoring.

Jazz for Service Management has a number of integration services: Administration, Registry, IBM Tivoli Common Reporting, Security, and IBM Dashboard Application Services Hub. These integration services provide key features including:

- Shared data repository for products integrating through Jazz for Service Management.
- Consistent UI experience through Dashboard Application Services Hub in Jazz for Service Management.
- Simplified administration of products and solutions integrating through Jazz for Service Management.
- Ad hoc, self-service reporting through Tivoli Common Reporting in Jazz for Service Management.

For more information about Jazz for Service Management, go to the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html).

IBM Tivoli Monitoring dashboard data provider for retrieving monitoring data for display in IBM Dashboard Application Services Hub dashboards

The IBM Tivoli Monitoring dashboard data provider retrieves monitoring agent data for display in the IBM Dashboard Application Services Hub component of Jazz for Service Management. The dashboard data provider is optionally installed during the Tivoli Enterprise Portal Server configuration. With the dashboard data provider enabled, Dashboard Application Services Hub users can retrieve read-only data from the hub monitoring server and monitoring agent for display in dashboards provided by the agents or in custom dashboards. IBM Tivoli Monitoring V6.3 includes the Infrastructure Management Dashboards for Servers that displays data for the OS agents. These server dashboards use the dashboard data provider to retrieve data. A connection to the dashboard data provider must be configured in Dashboard Application Services Hub. See Chapter 3, “Preparing your dashboard environment,” on page 29 and “Creating a connection to the IBM Tivoli Monitoring dashboard data provider” on page 58.

IBM Infrastructure Management Dashboards for Servers running on the Dashboard Application Services Hub V3.1 or later

With the IBM Tivoli Monitoring dashboard data provider enabled, Dashboard Application Services Hub users can retrieve managed system groups and events for all monitoring agents and Linux OS agent, UNIX OS agent, and Windows OS agent health metrics using the Infrastructure Management Dashboards for Servers application. This application is installed and configured into Dashboard Application Services Hub V3.1 or later using IBM Installation Manager. For more information, see “Installing and configuring the IBM Infrastructure Management Dashboards for Servers” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Open Services Lifecycle Collaboration Performance Monitoring service provider

The Tivoli Enterprise Monitoring Automation Server component contains the Open Services Lifecycle Collaboration Performance Monitoring (OSLC-PM) service provider and is installed on the same systems as your hub Tivoli Enterprise Monitoring Server. The service provider registers monitoring resources with the Jazz for Service Management Registry Services component and supports integration with other products using OSLC linked data interfaces. For more information, see "Performance Monitoring service provider" on page 11.

Role-based authorization policies

The Tivoli Authorization Policy Server feature provides you with greater access control capabilities than possible with the existing Tivoli Enterprise Portal Server authorization model. You can protect your resources from unauthorized access by users of monitoring dashboards in the IBM Dashboard Application Services Hub. IBM Tivoli Monitoring V6.3 with the Authorization Policy Server feature enabled provides the following capabilities:

- The ability to restrict access for dashboard users to specific managed system groups and to individual managed systems.
- The ability to assign role-based policies to users and user groups in a federated LDAP user registry to simplify policy management.
- A new command-line interface that is highly automatable.
- Central management of authorization policies for multiple IBM Tivoli Monitoring environments, also called domains.

To implement the feature you must install IBM Installation Manager packages for the Tivoli Authorization Policy Server and the tivcmd Command-Line Interface for Authorization Policy. The Authorization Policy Server is installed with your Dashboard Application Services Hub and the tivcmd CLI is installed on the computers used by the administrators who will be creating authorization policies. After successful installation of these two packages, you can execute various CLI commands as required to create roles, grant permissions, exclude permissions, and so on. For information on working with policies, see Chapter 7, "Using role-based authorization policies," on page 181.

OS Agents Report Prerequisites Scanner report

The OS Agents Report Prerequisites Scanner report delivered and installed through the OS agent report package, can be leveraged to check that your system's IBM Tivoli Monitoring prerequisites are configured correctly to use Tivoli Common Reporting without errors. See "Running a prerequisites scan" on page 539.

Creating and maintaining the dimension tables required for Tivoli Common Reporting using the Summarization and Pruning agent

You no longer have to periodically run the Tivoli Common Reporting and OS agent scripts to maintain the IBM_TRAM schema and populate the MANAGEDSYSTEM table. You can configure the Summarization and Pruning agent to create, populate, and maintain the dimension tables. See "Creating and maintaining the dimension tables" on page 523.

Tivoli Data Warehouse range partitioning

Range partitioning is a database data organization feature that can significantly improve pruning and query performance in large Tivoli Data Warehouse databases. You can migrate your existing tables to a partitioned table to take advantage of the performance improvements provided with partitioned tables. Range partitioning allows the database to limit the scope of queries when the column that is part of the partitioning key is used in the WHERE clause. See “Tivoli Data Warehouse range partition migrations” on page 470.

Take Action identity auditing

You can now audit any commands that are executed on a system at the agent level. The originator's user ID and network information are securely transferred to the agent and then recorded in the agent's audit log. The audit log can be historically collected. You can create situations and monitor centrally from the Tivoli Enterprise Portal. See “Take Action and command execution audit logging” on page 255.

AAGP authorization controls

The Access Authorization Group Profile (AAGP) authorization framework is now integrated with the Take Action identity auditing. The AAGP policies now selectively allow specific users to execute take actions from Tivoli Enterprise Portal or using **tacmd executeaction**, to execute commands using **tacmd executecommand**, or to create and modify situations and workflow policies that specify a take action command. The AAGP policy no longer requires the Central Configuration server to deliver the AAGP policy. The policy can be configured from the Agent Service Interface and stored locally on the agent itself. See “Access Authorization Group Profile” on page 398 and “Centralized Configuration” on page 424.

SOAP security enhancements

You can now enable security for CT_EMail and CT_Export requests using the SOAP_IS_SECURE environment variable on the monitoring server. See “Enabling SOAP security” on page 560.

Duper process optimization

The duper process now supports situations that contain reflex actions or display items. See “Duper process for optimizing situations” on page 85.

Changes to default self-describing agent behavior and new tacmd commands

You can now specify what products and versions are installed on your monitoring server and portal server by the automatic self-describing agent process. See “Self-describing monitoring agents” on page 297 and “Dynamically updating the self-describing installation options” on page 306.

Updates for private situations

- *REGEX predicate function

  IBM Tivoli Monitoring frequently requires text scan and pattern matching upon event and sample data, such as name, address, message, and log record. You can add the Regular Expression predicate filter to private situations to enhance agent monitoring event detection.
- Dynamically delete a private situation

  You can now use the DELETE= attribute in a private situation to dynamically remove a private situation without recycling the agent or deleting the local private situation XML file.

For more information, see “Private situation XML specification” on page 336.

Ability to clear the Deployment Status table transactions

Each time you issue an IBM Tivoli Monitoring **tacmd** command or use the Tivoli Enterprise Portal navigator to remotely manage a Tivoli Enterprise Monitoring Agent, information about the transaction is preserved in the

Tivoli Enterprise Monitoring Server Deployment Status table. To make it easier to manage the contents of this table, especially in large environments, you can schedule the periodic removal of completed transactions from the table. See “Clearing the Deployment Status table” on page 294.

Use of login daemon scripts available on IBM Service Management Connect
In IBM Tivoli Monitoring V6.3 or later, monitoring servers can now use the IBM Tivoli Monitoring login daemon solution that is available on IBM Service Management Connect to change the monitoring server an agent connects to. See “Changing the monitoring server an agent connects to” on page 296.

Setting the locale for the browser client
Administrators can no longer set the locale for the Tivoli Enterprise Portal browser client Enterprise-wide. The language can be changed through the Java™ control panel at the client computer if the underlying OS platform has been installed using a different locale than the one you want to use with the Tivoli Enterprise Portal. See the **`user.language`** and **`user.region`** parameters in “Portal client parameter list” on page 70.

Tivoli Integrated Portal name change
The V3.1 release of Tivoli Integrated Portal is now referred to as the Dashboard Application Services Hub.

i5/OS™ agent name change
The i5/OS monitoring agent is now referred to as the IBM i monitoring agent.

IBM Tivoli Monitoring family of products

The following information provides a brief overview of the applications of the IBM Tivoli Monitoring family of products.

IBM Tivoli Monitoring products help you manage 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 provides security, data transfer and storage, notification mechanisms, user interface presentation, and communication services in an agent-server-client architecture. These services are common to many product suites such as IBM Tivoli OMEGAMON® XE mainframe monitoring and IBM Tivoli Composite Application Manager.

After you have installed and initially configured Tivoli Management Services and the products that rely on them, consult this guide to apply further customization in a distributed environment. (*Configuring the Tivoli Enterprise Monitoring Server on z/OS* is provided in the guide of the same name.) It also has general administrative information for the managed systems that share these common services. Product-specific administrative information is given in the guides for the individual products.

Tivoli Management Services components

The following Tivoli Management Services components provide the infrastructure for yourTivoli Enterprise Monitoring Agents.

For a complete list of components, see the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm).

Client The IBM Tivoli Monitoring client, Tivoli Enterprise Portal is a Java-based user interface for viewing and monitoring your enterprise network. Depending on how it was installed, you can start Tivoli Enterprise Portal as a desktop application or through your browser as a web application.

Presentation server

The Tivoli Enterprise Portal client connects to the Tivoli Enterprise Portal Server. The Tivoli Enterprise Portal Server is a collection of software services for the client that enables retrieval, manipulation and analysis of data from the monitoring agents on your enterprise.

The Tivoli Enterprise Portal Server also includes the optional dashboard data provider which is used to retrieve read-only monitoring data for display in monitoring dashboards.

Management server

The Tivoli Enterprise Portal Server connects to the main, or *hub*, Tivoli Enterprise Monitoring Server. The monitoring server acts as a collection and control point for alerts received from the enterprise monitoring agents, and collects performance and availability data from them. The hub monitoring server correlates the monitoring data collected by monitoring agents and any remote monitoring servers and passes it to the portal server for presentation in the portal console.

The **automation server**, Tivoli Enterprise Monitoring Automation Server, is an optional component that can be installed on the same system as the hub monitoring server. It extends the functionality of the hub monitoring server. The automation server includes the Open Services Lifecycle Collaboration Performance Monitoring (OSLC-PM) service provider. For more information, see “Performance Monitoring service provider” on page 11.

Dashboard server

IBM Dashboard Application Services Hub is a Jazz for Service Management component that provides dashboard visualization and reporting services. Operators of the dashboard access it through a web browser interface. IBM Dashboard Application Services Hub uses the dashboard data provider component of the portal server to retrieve monitoring data.

You can install the **IBM Infrastructure Management Dashboards for Servers application** into Dashboard Application Services Hub to display situation event information, managed system groups and key health metrics for Windows OS agents, Linux OS agents, and UNIX OS agents. You can also create custom dashboard pages that display monitoring data.

You can also install the **Authorization Policy Server** and **tivcmd Command-Line Interface for Authorization Policy (tivcmd CLI)** to use role-based authorization policies to control what monitored resources are displayed in dashboards. For more information, see Chapter 7, “Using role-based authorization policies,” on page 181.

Help server

The IBM User Interface Help System built on Eclipse is installed with the portal server and provides presentation and search features for the integrated help system.

tacmd Command-Line Interface (tacmd CLI)

The tacmd CLI is used to manage your monitoring environment and can also be used to automate many of the administrative functions performed using the Tivoli Enterprise Portal. The CLI commands either send requests to the hub monitoring server or to the portal server.

Agents

Tivoli Enterprise Monitoring Agents are installed on the systems or subsystems whose applications and resources you want to monitor. An agent collects monitoring data from the *managed system* and passes it to the monitoring server to which it is connected. The portal client and dashboard server gathers the current values of the attributes and produces reports formatted into tables, charts, and relational table-based topology views. The agents and monitoring servers can also test the values of the current attributes against a threshold. When a threshold is exceeded or a value is matched, an alert icon can be displayed in the portal client or monitoring dashboard and the hub monitoring server can forward an event to an event server such as IBM Tivoli Netcool/OMNIbus. The attribute value conditions to test are called *situations*.

OS agents can be installed outside the enterprise as *Tivoli System Monitor Agents*. They do not connect to nor have any reliance on the Tivoli Enterprise Monitoring Server. They can run *private situations*, which are independent of the monitoring server, save data samples for attribute groups as *private history*, and can send SNMP alerts or EIF events to IBM Tivoli Netcool/OMNIbus.

Data warehouse

The Tivoli Data Warehouse is an optional component for storing historical data collected from agents in your environment. The data warehouse is located on a supported database (such as DB2®, Oracle, or Microsoft® SQL).

Shared user registry

A shared user registry is an LDAP server such as Tivoli Directory Server or Microsoft Active Directory that can be used to authenticate portal server users, IBM Dashboard Application Services Hub users, and other application users such as Netcool/OMNIbus Web GUI users. When a shared user registry is used, users are authenticated by the first server that they access and authentication tokens are passed to the other servers so that the user is not required to re-enter their credentials.

Event synchronization

The event synchronization component is optional. It is configured to send situation event updates that were forwarded to a IBM Tivoli Enterprise Console Event Server or a Netcool/OMNIbus ObjectServer back to the monitoring server.

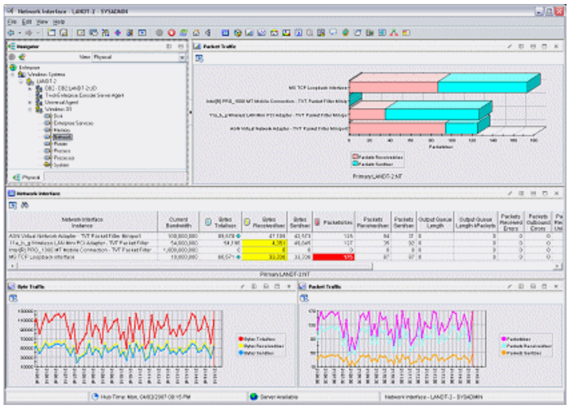
Tivoli Enterprise Portal client

Tivoli Enterprise Portal is one of the user interfaces for your IBM Tivoli Monitoring-based products. In the same way you use your browser's home page as a starting point for navigating the Internet, you use Tivoli Enterprise Portal to get a high level overview of your network environment.

One section of the window displays the Navigator, a tree-like view of your monitored network, from the top level down to individual groupings of information collected by monitoring agents. The rest of the window is filled with views pertinent to the chosen item in the Navigator tree. From the top level or from your home workspace, you can navigate to specific locations to check activity and investigate problems.

This workspace was customized for the select item in the tree. This workspace was designed with a bar chart, two plot charts, and a table that displays a background color for cell values that exceed a certain threshold. You can create and customize additional workspaces for every item in the tree.

The event indicators that display in the tree, or Navigator, are the results of tests, called situations, that run on your monitored systems. When the condition described in the situation is true, a colored icon overlays the affected items in the tree. Use the Situation editor to set up conditional alerts that monitor your environment automatically. Use the Workflow editor to set up policies to automate your environment.



Desktop, Browser, and Java Web Start clients

The Tivoli Enterprise Portal client can be deployed in three ways, as described briefly here and in more detail in the *Installation and Setup Guide*.

Desktop

The desktop client requires that you load and run the installation software on each computer where the desktop client will be run. Users start Tivoli Enterprise Portal the same way they do their other locally installed applications. With the desktop client, you can also create multiple instances for connecting to different portal servers.

Browser

The browser client installation software resides on the Tivoli Enterprise Portal Server. The client software is downloaded from there to your computer the first time you log on to the portal server from your browser, and thereafter only when there are software updates.

You can start the browser client from any browser-enabled computer by entering the URL for the portal server. In this mode of operation, each portal workspace has a URL, so you can save a workspace to your Favorites list.

With the browser client you can 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.

Java Web Start

With Java Web Start, like the browser client, the client software is accessed through a URL and downloaded from the portal server. Unlike the browser client, which is always run inside the browser, the Web Start client is run as a desktop application. Whenever updates to the client software are available, they are downloaded from the portal server automatically. References to *desktop client* behavior in this guide also assumes the Java Web Start client unless otherwise stated. Single sign-on is an example: As well as the browser client, you can use single sign-on with the Web Start client

Historical data collection

In addition to the real-time reports offered by Tivoli Enterprise Portal workspaces, you can configure historical data collection to store the data being collected by your monitoring agents for historical reports and situations. You can specify the following:

- Attribute groups for historical data collection
- Data collection interval.
- Data warehousing intervals if you choose to write data to the Tivoli Data Warehouse
- How data samples are grouped for pruning from the Tivoli Data Warehouse
- Pruning schedule of warehoused data.
- Storage location for the short-term history files before they are sent to the data warehouse. Data samples can be stored at the monitoring agent or on the Tivoli Enterprise Monitoring Server.

To ensure that data samplings are saved to populate your predefined historical workspaces, you must first configure and start historical data collection. Real-time workspaces are available whether you start historical collection or not.

System administrator tasks

A system administrator has the highest level of authority and can access all IBM Tivoli Monitoring features in the Tivoli Enterprise Portal.

This list represents the types of tasks a system administrator might perform:

- Establishes Tivoli Enterprise Portal user IDs and user groups with the appropriate permissions for their jobs.
- Designs workspaces for Navigator items and makes these workspaces available to users based on their established permissions.
- Defines queries that can be applied to table and chart views to specify the attributes and attribute value ranges to retrieve from the monitoring server.
- Writes definitions for launching applications and makes them available to users based on their established permissions.
- Creates command line actions that can run at the specified managed system from the portal client, and makes them available to users who have been granted authority.
- Creates situations using the visual programming facilities
- Sets the severity of a situation for a particular Navigator item and what, if any, sound plays when the situation is true and an event opens
- Decides which situations apply to which managed systems, a process called distribution

- Provides expert advice to display when certain situations evaluate true
- Creates policy workflows, which are actions to take when situations evaluate true
- Creates, installs, upgrades, distributes and configures agents on remote hosts from a central location
- Starts, stops, and recycles agent processes

Performance Monitoring service provider

The Tivoli Enterprise Monitoring Automation Server component contains the Open Services Lifecycle Collaboration Performance Monitoring (OSLC-PM) service provider and is installed on the same systems as your hub Tivoli Enterprise Monitoring Server.

The Performance Monitoring service provider registers monitoring resources with the Registry Services. Registry Services is a Jazz for Service Management integration service that provides a shared data repository for products in an integrated service management environment. Products that discover and manage shared IT resources can register these IT resources and the services they offer with Registry Services. Other products can consume data by querying Registry Services for the managed resources or the associated service providers of interest.

The Performance Monitoring service provider registers resources types such as Computer System, Software Server, Software Module, Database, IPAddress, and ServerAccessPoint on behalf of monitoring agents. These resources types are defined using the OSLC Common Resource Type Vocabulary (CRTV). Agents provide a template that maps their monitoring data to CRTV resources. The template is installed with the agent's monitoring server application support.

The Performance Monitoring service provider also supports the OSLC-PM RESTful API for retrieving linked data about monitored resources. It accommodates the RDF/XML, compact XML and HTML content types in HTTP GET requests. When RDF/XML and HTML content is requested, the API returns resource health metrics defined by the OSLC-PM domain and by the IBM Tivoli Monitoring private namespace.

To discover the resources that have health metrics available from the Performance Monitoring service provider, you must query Registry Services since the Performance Monitoring service provider does not provide OSLC query capability. Registry Services provides a query interface for retrieving service providers records, resource registration records, and reconciled resource records. The reconciled resource records and registration records contain HTTP URLs that can be used to retrieve information about the resource from the service provider that registered the resource. The *Jazz for Service Management Integration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) contains information on querying Registry Services to learn about service providers and resources.

Security Services is an optional Jazz for Service Management component that enables non-WebSphere based applications such as the Performance Monitoring service provider to participate in LTPA based single sign-on. It should be installed and configured if you want the Performance Monitoring service provider to authenticate requests it receives from OSLC clients. This is the only authentication method supported by the service provider. See “Authentication through the Tivoli Enterprise Monitoring Automation Server” on page 128 for more details.

You can use the Performance Monitoring service provider with the Tivoli Business Service Manager V6.1.1 dashboard server to display key health metrics from monitoring agents in the service tree without launching in context to the Tivoli Enterprise Portal. The health metrics are available if a resource such as a ComputerSystem or SoftwareServer has been registered with Registry Services by the Performance Monitoring service provider and the resource has also been discovered by the IBM Tivoli Monitoring Discovery Library Adapter or by a Tivoli Application Dependency Discovery Manager sensor when the TADDM OSLC service provider is also being used. The metrics are displayed in a hover preview dialog that also displays information from other service providers that have registered the same resource. For example, the OSLC service provider for Tivoli Application Dependency Discovery Manager provides configuration and change history information for registered resources in the hover preview dialog. For more information on how to setup the integration between the Performance Monitoring service provider, Tivoli Business Service Manager, Registry Services, and other supported service providers, see the Cross Product Integration Scenarios wiki.

You can also create your own OSLC client applications that retrieve resource information from Registry Services and health metrics for those resources from the Performance Monitoring service provider. Alternatively you can create your own OSLC service provider implementation to augment the information available for registered resources. For more information on creating these types of applications, see Getting started with Registry Services on the Jazz for Service Management wiki.

Note: If you create your own OSLC client application, take note that the `rr:sourceRecord` properties that are retrieved from Registry Services for registration or reconciliation records can change if you have a Hot Standby monitoring environment. The `rr:sourceRecord` property contains the URL that is used to query an OSLC service provider for more details about a resource. When Hot Standby is used, the `rr:sourceRecord` URL for a resource points to the Performance Monitoring service provider for the acting hub monitoring server. When the standby hub monitoring server becomes the acting hub, the URLs are updated to specify the host name and port number of the Performance Monitoring service provider for the new acting hub. Therefore, one of the following scenarios must take place:

- If your OSLC client application caches the `rr:sourceRecord` property values, it must periodically query Registry Services to determine whether the cached URLs must be updated.
  OR
- If your OSLC client application tries to use a cached `rr:sourceRecord` URL for a resource and a response is not received from the Performance Monitoring service provider, then it can query Registry Services to determine whether the `rr:sourceRecord` value changed.

If the Performance Monitoring service provider for the acting and standby monitoring servers are both up and an OSLC client sends a request to the Performance Monitoring service provider for the standby hub monitoring server, the service provider returns an HTTP 301 redirect response if it can determine the host name of the acting hub and Performance Monitoring service provider; otherwise, it returns an HTTP 404 response.

For more information about OSLC and Loosely Coupled Integration, see the following links:

- OSLC community
- Performance Monitoring working group
- Reconciliation working group and Common Resource Type Vocabulary
- Loosely coupled integration at ISM Connect
- IBM Tivoli Monitoring OSLC private namespace schema

For information on installing and configuring the Tivoli Enterprise Monitoring Automation Server and Performance Monitoring service provider, see the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm).

Chapter 2. Preparing your Tivoli Enterprise Portal environment

Review these topics for additional configuration of the Tivoli Enterprise Portal client environment.

Browser client

Users start the browser client by entering the URL for the integral HTTP server on the Tivoli Enterprise Portal Server.

The advantages of the browser client are:

- Easy deployment. The browser client is installed the first time users log on to the URL for the Tivoli Enterprise Portal integral HTTP server.
- Software upgrades are automatic. When users log on, their browser client is checked against the one at the Tivoli Enterprise Portal Server; if a newer version is detected, it is downloaded from the server.
- Global parameter settings are set for all users connected to the same Tivoli Enterprise Portal Server.
- Workspaces have identifying URLs that can be referenced in Web pages and when launching from another Web-enabled application.
- Includes a banner that can be customized with your company logo and URL.

Java runtime environment (JRE) versions

The Tivoli Enterprise Portal Server and client run Java-based software. When you install the portal server or the desktop client, IBM Java 7 is installed automatically.

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 the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm).)
- IBM 32-bit or 64-bit Java Runtime Environment for Windows, version 7.0 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. 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.

When you log on from a browser, a check is done for the level of Java associated with the browser. The required version of Java is controlled at the portal server and you might be prompted to upgrade to IBM Java 7 when you connect.

First time logon

The first time the URL for Tivoli Enterprise Portal is entered from a system, the Java Plug-in transfers the required files from the Tivoli Enterprise Portal Server (on

Windows, the files reside in the <*install_dir*>\cnb branch; on operating systems such as UNIX, they are in the <*install_dir*>/cw branch).

From then on the browser client software does not need to be downloaded again until a new version has been installed. The Java plug-in maintains the version levels of the files on users' computers and compares them with the version levels on the integral HTTP server. If it detects files that are older than the ones on the HTTP server, it downloads the latest files.

Be sure you have sufficient free space for the downloaded files. If the disk runs out of space during the download, you are not warned.

Internet Explorer security settings

About this task

If you have the Internet Explorer security level set to high, you must adjust the settings to run the Tivoli Enterprise Portal. Otherwise, the Tivoli Enterprise Portal browser client cannot run.

Check the security settings

The following procedure should be used to check your current security settings.

Procedure

1. In Internet Explorer, select **Tools** → **Internet Options**
2. Select the **Security** tab.
3. Click **Internet** if you are running Tivoli Enterprise Portal through the Internet; or **Intranet** if you are running Tivoli Enterprise Portal through your intranet.
4. Change your security settings to **Default Level**
5. Click **OK** to save.

Keep current security settings

You can integrate the Tivoli Enterprise Portal website with Internet Explorer without changing your security settings. If you wish to keep your current security settings, you can add the Tivoli Enterprise Portal website to your Trusted Sites zone.

Procedure

1. In Internet Explorer, select **Tools** → **Internet Options**
2. Select the **Security** tab.
3. Click **Trusted Sites** → **Sites**, and enter the URL for Tivoli Enterprise Portal.
4. Clear the check box that checks for (https:) for all sites at this zone, click **Add** .

   Choose the medium security level or lower for all sites in the **Trusted Sites** zone.
5. Click **OK** to save your changes.

Windows write and delete privileges

Starting with Windows 2000, write and delete privileges for certain folders and registry keys were removed from the Users group. These privileges are required for anyone intending to use the Java WebStart client or the browser client. Otherwise, Java exception errors are encountered during attempts to start the product.

Before users can download the Java WebStart client or start the browser client, the Windows administrator must assign the required permissions to individual user IDs or the Users group, or create a new group with the required permissions and assign users to this group in addition to the Users group. The required permissions are:

- Write and Delete permissions on the directory where Windows is installed, such as C:\WINDOWS.
- Set Value, Create Subkey, and Delete permissions on registry key HKEY_LOCAL_MACHINE\SOFTWARE.

Note: The Windows permissions scheme affects the Tivoli Enterprise Portal browser mode and other third-party software installed through Internet Explorer.

Adding your company logo and URL

The Tivoli Enterprise Portal browser application looks much as it does in desktop mode, except that it also has a banner with a link to ibm.com®. You can customize the Tivoli Enterprise Portal browser client by replacing the logo and URL with your organization's.

About this task

Take these steps to customize the portal client banner:

Procedure

1. On the computer where you installed the Tivoli Enterprise Portal Server, open the following file in an HTML editor or text editor:

   `install_dir\cnb\bannerimage.html`

2. Edit the HREF and IMG SRC tags for your organization's URL and logo graphic file:
   a. Replace the **href** `' + URL + '` placeholder with your organization's URL.
   b. Replace the **img src** `' + URL + '` placeholder with the name of your organization's logo GIF or JPG file.
   c. Replace the **alt** `' + URL + '` placeholder with the text that should display when the mouse pointer is over the image, such as the URL.
3. Save the file and exit the editor.
4. Copy the logo graphic to the `install_dir\cnb\ directory`.

Results

Users now see your logo on the right-hand side of the banner the next time they start browser mode.

Starting the Tivoli Enterprise Portal client

Log on to the Tivoli Enterprise Portal Server to start a Tivoli Enterprise Portal work session.

Before you begin

The hub Tivoli Enterprise Monitoring Server and the portal server must be running for the portal client to start successfully. You also must have a valid user ID.

About this task

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 portal using either the desktop client or the browser client. The default user ID is sysadmin.

The Tivoli Enterprise Portal client supports multiple monitor configurations. For example, if you have two display monitors, you can launch the Tivoli Enterprise Portal client in the secondary monitor and the dialogs, message pop-ups, and context menus are displayed in the same secondary monitor.

Procedure

- Start the desktop client:
  - Windows Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Tivoli Enterprise Portal**. When the logon window is displayed, enter your user ID and password and click **OK**.
  - Linux UNIX Enter `./itmcmd agent start cj` at the command line.
- Start the browser client:
  1. Start the browser.
  2. Type the URL for the Tivoli Enterprise Portal Server into the **Address** field of the browser, where the *systemname* is the host name of the computer where the portal server is installed and *15200* is the port number for the browser client: `http://systemname:15200`
  3. Click **Yes** on the Warning - Security window.
  4. When the logon window is displayed, enter your user ID and password and 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.

This section is reproduced from the *IBM Tivoli Monitoring Installation and Setup Guide* for your convenience.

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 the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm).)
- IBM 32-bit or 64-bit Java Runtime Environment for Windows, version 7.0 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. 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.

Installing the IBM JRE

About this task

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. You download an installer from the computer where the Tivoli Enterprise Portal Server is installed.

Windows: Installing the IBM JRE

Install the IBM Java Runtime Environment on the computer where you plan to start the desktop client using Java Web Start.

About this task

Take these steps to download the IBM JRE installer from the Tivoli Enterprise Portal Server and install the JRE on a Windows computer.

Note: The following procedure assumes you are installing the IBM JRE on a 32-bit Windows platform. If you are using a 64-bit Windows platform, the name for the IBM JRE installer executable is `ibm-java7_64.exe`.

Procedure

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, where <*portal_server_host_name*> is the fully qualified host name of the computer where the portal server is installed (for example, `myteps.itmlab.company.com`):

   ```
   http://<portal_server_host_name>:15200/java/ibm-java7.exe
   ```
3. When prompted, save the **ibm-java7.exe** file to a directory on your hard drive.
4. Change to the directory where you saved the **ibm-java7.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. If you have no other system JVM's installed, click **YES** on the message that asks if you want to install this JRE as the system JVM. Otherwise, click **NO**.
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 might 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

Install the IBM Java Runtime Environment on the computer where you plan to start the desktop client using Java Web Start.

About this task

Complete the following steps to download the IBM JRE installer from the Tivoli Enterprise Portal Server and install the JRE on a Linux computer, or install the JRE without downloading the installer by supplying the URL to the rpm in the command.

```
rpm -ivh http://portal_server_host_name:15200/java/ibm-java7.rpm
```

Note: The following procedure assumes you are installing the IBM JRE on a 32-bit Linux platform. If you are using a 64-bit Linux platform, the name for the IBM JRE `.rpm` file is `ibm-java7_64.rpm`.

Procedure

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://portal_server_host_name:15200/java/ibm-java7.rpm
   ```

   where *portal_server_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 installer to disk.
4. Change to the directory where you saved the `ibm-java7.rpm` file and launch the installer to start the installation program using the following command:

   ```
   rpm -ivh ibm-java7.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.

Before you begin

The logs for the Web Start client are located in a different place than logs for the browser client and for 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` or `%USERPROFILE%\AppData\LocalLow\IBM\Java\Deployment\log` directory. On Linux and UNIX computers, the logs are located in the `.java/deployment/log` 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.

About this task

Complete the following steps to enable tracing:

Procedure

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\Java70\jre\bin\javacpl.exe"**.
   - On Linux, change to <*install_dir*>/jre/<*platform*>/bin and run Control Panel: ./ControlPanel
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

The Tivoli Enterprise Portal can be launched as a desktop application or as a web application. You have three ways to install the desktop application: from a browser by entering the URL of the Java Web Start client on the Tivoli Enterprise Portal Server, launching the desktop client from the IBM Java Control Panel, or launching the desktop client using Java Web Start from the command line.

Before you begin

These are the basic instructions for downloading and running the desktop client using Java Web Start. The complete instructions, with configuration notes are given in the *IBM Tivoli Monitoring Installation and Setup Guide*.

About this task

Complete one of these steps to install and launch the desktop client using Java Web Start:

Procedure

- Enter the URL of the portal server in a browser:
  1. Start the browser on the computer where you want to use the desktop client.
  2. Enter the following URL in the **Address** field of the browser, where <*portal_server_host_name*> is the fully qualified host name of the computer where the Tivoli Enterprise Portal Server is installed.

     ```
     http://<portal_server_host_name>:15200/tep.jnlp
     ```
  3. Click **Run** on the security message.
  4. If you want to create a shortcut on your desktop for the Tivoli Enterprise Portal, click **Yes** when prompted. The desktop client starts and displays the logon window. If IBM Java 1.7 is not the system JVM, you cannot use this shortcut. You must create your own, as described in the topic on “Manually creating a shortcut for the Web Start client” in *IBM Tivoli Monitoring Installation and Setup Guide*.
  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*.

  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 where the client was launched. On Windows this defaults to \Documents and Settings\<*userid*>\Desktop.
- Launch the desktop client from IBM Java Control Panel:
  1. Launch the IBM Java Control Panel:
     Windows In the Windows control panel, double-click **IBM Java Control**

    Panel. You must be in the Classic view to see **IBM Java Control Panel**.
    Linux Change to <*install_dir*>/jre/<*platform*>/bin directory and enter `./ControlPanel`.
  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, then click **Launch Online**.

  Java Web Start downloads and starts the desktop client. When the application is launched, you can close the Control Panel windows.

- Launch the desktop client using Java Web Start from the command line:
  1. Open a command line window and change to the directory where Java Web Start is installed.

     Windows

     ```
     C:\Program Files\IBM\Java70\jre\bin
     ```

     Linux

     ```
     <install_dir>/jre/<platform>/bin
     ```

  2. Enter the following command, where <*portal_server_host_name*> is the fully qualified host name of the computer where the Tivoli Enterprise Portal Server is installed.

     Windows

     ```
     javaws http://<portal_server_host_name>:15200/tep.jnlp
     ```

     Linux

     ```
     ./javaws http://<portal_server_host_name>:15200/tep.jnlp
     ```

  Java 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 1.7, the shortcut will not launch the desktop client. You must create a shortcut manually.

About this task

To create a shortcut to use to launch the desktop client using Web Start, complete the following procedure:

Procedure

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\Java70\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.

Starting the desktop client on another portal server

When installing the desktop client, you designate a home Tivoli Enterprise Portal Server. If your monitoring environment has a multiple portal servers, you can define a separate desktop instance to point to another portal server.

Before you begin

The typical scenario for having multiple portal servers is where there is a test and production portal server, or where there are multiple managed networks with a portal server connected to each hub monitoring server.

About this task

Take these steps to create another portal client instance that connects to a different portal server.

Procedure

- Windows
  1. On the computer where the desktop client is installed, select **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.
  2. Right-click **Tivoli Enterprise Portal Desktop Client** and click **Create Instance**. If other instances of the Tivoli Enterprise Portal have been created, you see more than one in the list. **Create Instance** is disabled for all but the original Tivoli Enterprise Portal instance.
  3. In the Tivoli Enterprise Portal window, enter a name to identify the instance and click **OK**.
  4. In the Configure Application Instance window, enter the host name of the Tivoli Enterprise Portal Server that you want to connect to.
  5. Click **OK**.
- Linux UNIX Using the command line:
  1. Change directory (cd) to *install_dir*/bin.
  2. Create a new instance using the following command:

     ```
     ./itmcmd config -A cj
     ```
  3. Launch the new instance using the following command:

     ```
     ./itmcmd agent -o <instance_name> start cj
     ```

     For full syntax information see the *IBM Tivoli Monitoring Command Reference*.

  Using the GUI:
  1. Change directory (cd) to *install_dir*/bin.
  2. To start the Manage Tivoli Enterprise Monitoring Services, use the following command:

     ```
     ./itmcmd manage
     ```
  3. Right-click **Tivoli Enterprise Portal Desktop Client** and click **Configure**.
  4. Enter the instance name and portal server hostname, then click **Save**.
  5. To start the instance, right-click **Tivoli Enterprise Portal Desktop Client** and click **Start Service** and enter the instance name.

Results

The new Tivoli Enterprise Portal instance is added to the list.

What to do next

You can now start the instance at any time by double-clicking its entry.

If you no longer need a Tivoli Enterprise Portal instance, you can delete it: right-clicking the entry and click **Remove Instance**.

Starting the browser client on another portal server

Start a separate instance of your browser and log on to the Tivoli Enterprise Portal Server of a different managed network to see two managed networks from the same computer.

Before you begin

Your managed network can have one portal server and one hub Tivoli Enterprise Monitoring Server. You can log on to the portal server through the Windows Internet Explorer or Mozilla Firefox.

About this task

Before starting the browser client instances, take these steps on each computer where a portal server that you want to connect to is installed.

Procedure

- Windows
  1. In the Manage Tivoli Enterprise Monitoring Services, right-click the **Tivoli Enterprise Portal Server** entry and select **Reconfigure**.
  2. In the Configure Tivoli Enterprise Portal Browser window that opens, double-click the **`cnp.browser.installdir`** parameter.
  3. In the Edit Tivoli Enterprise Portal Browser Parm window that opens, enter the path to the directory where the browser files should be installed, for example, `C:\\temp\\cnpBrowserFiles`.
  4. Select the ☑ **In Use** check box and click **OK**.
  5. Click **OK** to save your changes.
- Linux
  1. Change to the directory where `applet.html` is located: `*install_dir*/platform/cw`, where platform is the current type of operating system.
  2. Open `applet.html` in a text editor.
  3. Find the line, **<!--END OF PARAMS-->** and add a new line above it.
  4. On the new line, add this parameter where *browser_install_dir* is the path to the directory where the browser files are installed.

     ```
     document.writeln( '<PARAM NAME= "cnp.browser.installdir"
     VALUE="browser_install_dir">' )
     ```
  5. Save and close `applet.html`.

What to do next

If you are using Internet Explorer, launch each instance of the portal client that you want.

If you are using the Firefox browser, you must create a separate profile for each instance that you intend to start. The Mozilla support site has a topic on Managing Profiles (http://support.mozilla.com/en-US/kb/Managing+Profiles) that you can refer to for help with setting up profiles. After creating the profiles, launch each instance with this command **<*full_path_to_firefox*> -p <*profile_name*> -no-remote**

Related reference:

"Portal client parameter list" on page 70
Most of the Tivoli Enterprise Portal client parameters are left unchanged from their default values. Edit the client parameters to effect a specific behavior.

Specifying the browser used for Launch Application and 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 the default, specify to the portal server the location of the browser you want to use.

About this task

Complete these steps to specify a different browser to use for the online help and launch application:

Procedure

- Windows
  1. Launch Manage Tivoli Enterprise Monitoring Services (**Start > (All) Programs > IBM Tivoli Monitoring > Manage Tivoli Enterprise 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. Click **OK** to close the editing window and save the change.
  7. Click **OK** to close the reconfiguration window.
- Linux UNIX
  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` 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
```

- Java Web Start:

  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 PortalTivoli Enterprise Portal is installed, the file is located in <*itminstall_dir*>\Config (for example: c:\IBM\ITM\Config). On UNIX computers, the file is located in <*itminstall_dir*>/config (for example, `/opt/IBM/ITM/config`).

  In order to add or modify JVM arguments (such as maximum heap size) or other Tivoli Enterprise Portal-based properties (such as RAS1 trace options), it is necessary to 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` file if the change will be permanent (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.

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>" >
```

Add operating platforms to the Navigator view

Edit the Tivoli Enterprise Portal Server `osnames` file to create additional branches in the Tivoli Enterprise Portal Navigator view for other operating system names.

The Navigator Physical view in the Tivoli Enterprise Portal shows the operating platform below the enterprise level. The operating platform name is followed by the word *Systems* as in Linux Systems or z/OS® Systems. Some operating platforms can be aggregated further. If your environment has such platforms and you want each to have its own Navigator item, with all systems of that type contained there, you can add them to the `osnames` file in the portal server directory (for example, C:\IBM\ITM\CNPS and /opt/IBM/ITM/config).

Chapter 3. Preparing your dashboard environment

Review these topics for additional configuration of your dashboard environment.

Roadmaps

Tasks for setting up your environment depend on many factors. There are two main types of dashboard environments you might have: a basic environment without single sign-on or authorization controls per user, or an advanced environment with single sign-on and authorization controls per user.

If you originally created an environment without single sign-on or authorization controls per user, you can later change your settings to use single sign-on and authorization controls per user. Additionally, if you want to create a high availability dashboard environment, you can use an HTTP server configured for load balancing between your dashboard users and a cluster of Dashboard Application Services Hub nodes, and also between Dashboard Application Services Hub nodes and multiple Tivoli Enterprise Portal Servers.

Setting up a basic monitoring environment without single sign-on and without per user authorization controls

Setup a basic dashboard environment if you want to use IBM Dashboard Application Services Hub with monitoring dashboard applications such as IBM Infrastructure Management Dashboards for Servers and IBM Infrastructure Management Dashboards for VMware or with custom dashboards, without using single sign-on or authorization controls per user.

Your environment must meet the following requirements:

- Your Dashboard Application Services Hub and Tivoli Enterprise Portal Server are not configured to use a federated LDAP user registry for user authentication.
- Your dashboard users will not launch the Tivoli Enterprise Portal browser client from IBM Dashboard Application Services Hub pages, or if they do, they must provide their credentials when the browser client is started.
- All of your dashboard users can be authorized to see the same managed systems and managed system groups in the monitoring dashboard pages.

If these requirements are not met in your environment, follow the steps for "Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls" on page 34.

Note: You can also start with a basic dashboard environment to become familiar with IBM Dashboard Application Services Hub with monitoring dashboards and later add single sign-on and per user authorization by following the steps in "Migrating a basic monitoring dashboard environment to a dashboard environment with single sign-on and per user authorization controls" on page 45.

The Dashboard Application Services Hub uses a HTTP or HTTPS connection to the dashboard data provider component of the portal server to retrieve monitoring data. Real-time monitoring data is retrieved from the hub monitoring server and monitoring agents and historical monitoring data is retrieved from the Tivoli Data Warehouse. Not all monitoring dashboard applications support retrieving historical data from the Tivoli Data Warehouse.

You configure a dashboard data provider connection in the Dashboard Application Services Hub that specifies the hostname, protocol, port, username, and password of the portal server. The user ID configured for the data provider connection is included in all HTTP requests to the dashboard data provider instead of the user who is logged into Dashboard Application Services Hub and using a dashboard application. The connection user must be defined as a Tivoli Enterprise Portal user ID and be assigned the monitoring applications whose data will be displayed in monitoring dashboards. Dashboard Application Services Hub uses roles, for users or user groups, to control what pages a user can access. However, the dashboard data provider performs the authorization of the monitoring resources that are displayed in those pages. Because the dashboard data provider is only sent the credentials of the user configured for the data provider connection, it enforces the Tivoli Enterprise Portal permissions and monitoring application assignments of the connection user and not the dashboard users. For this reason, all monitoring dashboard users will see monitoring data from the same set of managed systems and managed system groups.

Prerequisites

- Install and configure the base IBM Tivoli Monitoring monitoring server, portal server, and portal client components using the instructions in the *IBM Tivoli Monitoring Installation and Setup Guide*. When configuring the portal server, enable the dashboard data provider.
- Install and configure the monitoring agents whose data will be displayed in the monitoring dashboards. Install their application support in the monitoring servers, portal server, and desktop portal client if it is being used, using the instructions in the *IBM Tivoli Monitoring Installation and Setup Guide*.
- Install and configure Dashboard Application Services Hub and your dashboard monitoring applications, see “Required software and memory requirements for a dashboard environment” in the *IBM Tivoli Monitoring Installation and Setup Guide*. Also see “Installing and configuring the IBM Infrastructure Management Dashboards for Servers” in the *IBM Tivoli Monitoring Installation and Setup Guide*, if you will be installing that dashboard application.

Roadmap

Use the following roadmap to help you get started:

Table 1. Roadmap for setting up a basic monitoring environment without single sign-on and without per user authorization controls

Step	Description	Where to find information
1 (required)	Verify the dashboard data provider is enabled in your portal server configuration.	For detailed steps, see “Verifying the dashboard data provider is enabled” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2 (required)	Determine which Tivoli Enterprise Portal user should be configured for the dashboard data provider connection and ensure the user has this authorization: • The user must be assigned the monitoring applications whose data will be displayed in monitoring dashboards. • If your dashboard applications display situation events, the user must have permission to view situation events.	“Administer Users” on page 164 and “Managing user IDs” on page 169

Table 1. Roadmap for setting up a basic monitoring environment without single sign-on and without per user authorization controls (continued)

Step	Description	Where to find information
3 (required)	Login to IBM Dashboard Application Services Hub as an administrative user and create a dashboard data provider connection that uses the HTTP protocol and does not require single sign-on configuration. By default, the URL is `http://`*`hostname`*`:`*`port_number`*`/ibm/console/`. The default port number for HTTP is 16310 and for HTTPS is 16311. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.	"Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 When creating your connection, **do not** select the box **Use the credentials of the user (requires SSO Configuration)**.
4 (optional best practice)	Login to IBM Dashboard Application Services Hub as an administrative user and create a role that controls access to your dashboard application pages and assign dashboard users or user groups to the role. **Note:** Some dashboard applications such as IBM Infrastructure Management for VMware automatically create a role for its pages when the dashboard application is installed. However, other dashboard applications such as IBM Infrastructure Management Dashboards for Servers do not create a role during installation so you must create one or assign the dashboard pages to an existing role.	Refer to the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with roles that control access to dashboard pages.
5 (required)	Login to IBM Dashboard Application Services Hub as a user who has permission to view your dashboard pages, launch the dashboard applications, and verify data is displayed.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

Table 1. Roadmap for setting up a basic monitoring environment without single sign-on and without per user authorization controls (continued)

Step	**Description**	**Where to find information**
6 (optional)	If you want to use HTTPS between Dashboard Application Services Hub and the dashboard data provider, perform these tasks:	
	1. Configure TLS/SSL between the dashboard hub and data provider.	"Configuring TLS/SSL communication between Dashboard Application Services Hub and the dashboard data provider" on page 213
	2. Login to IBM Dashboard Application Services Hub as an administrative user who has been assigned the `administrator` and `iscadmins` roles and delete the dashboard data provider connection that you previously created.	Refer to the IBM Dashboard Application Services Hub online help and the *Jazz for Service Management Integration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with data provider connections.
	3. While still logged into IBM Dashboard Application Services Hub as an administrative user, create the connection again and this time specify HTTPS as the protocol.	"Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 When creating your connection, **do not** select the box **Use the credentials of the user (requires SSO Configuration)**.
	4. Login to IBM Dashboard Application Services Hub as a user who has permission to view your dashboard pages, then launch the dashboard application again and verify data is displayed.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

After you have the basic dashboard monitoring environment setup, you might also need to perform the following tasks:

Table 2. Additional tasks required to setup your basic monitoring environment without single sign-on and without per user authorization controls

Task	**Where to find information**
Create situation definitions for events that your dashboard users will monitor.	See Situations for event monitoring in the *Tivoli Enterprise Portal User's Guide* and also see the *IBM Tivoli Monitoring Command Reference* for information on the **`tacmd`** commands used to work with situations.
Create managed system groups that can be used to group managed systems for display in dashboard pages.	See Managing the environment in the *Tivoli Enterprise Portal User's Guide* and also see the *IBM Tivoli Monitoring Command Reference* for information on the **`tacmd`** commands used to work with system lists.
Configure historical data collection if you want to display historical data in your dashboard pages. **Note:** Not all monitoring dashboard applications support retrieving historical data from the Tivoli Data Warehouse.	Chapter 16, "Managing historical data," on page 457

Table 2. Additional tasks required to setup your basic monitoring environment without single sign-on and without per user authorization controls (continued)

Task	**Where to find information**
Authorize a new user or user group access to dashboard pages in Dashboard Application Services Hub.	Refer to the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with roles that control access to dashboard pages. **Tip:** Best practice is to create a user group, add the users to the group, and then assign the group a Dashboard Application Services Hub role that has permission to view the appropriate dashboard pages.
For each dashboard user who will also use the Tivoli Enterprise Portal client, create a Tivoli Enterprise Portal user ID and assign it monitoring applications and any other permissions that they might require when using the portal client. To launch the Tivoli Enterprise Portal client from Dashboard Application Services Hub, the user must be assigned either the Tivoli Enterprise Portal <All Application> or Tivoli Monitoring Server application. Otherwise, they might see the KFWITM388E message after performing the launch.	"Administer Users" on page 164 and "Managing user IDs" on page 169
Create custom dashboard pages and ensure the dashboard users are assigned a Dashboard Application Services Hub role with permission to view the custom pages.	"Creating custom dashboard pages that display monitoring data" on page 61
Install a new monitoring dashboard application in Dashboard Application Services Hub, assign the dashboard's pages to a new or existing role, and assign users or user groups to the role that controls access to the pages. If an HTTP server is being used with Dashboard Application Services Hub for load balancing, regenerate the web server plugin. Verify the Tivoli Enterprise Portal user configured for the dashboard data provider connection is assigned the monitoring applications whose data will be displayed in the new dashboard application and is assigned view permission for events if the dashboard application displays situation event data. **Note:** The application support for the agent must be installed in the portal server and monitoring server before you can see the agent's data in the new dashboards.	Follow the dashboard application installation documentation. Then, refer to the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with roles that control access to dashboard pages. If required, regenerate the web server plugin by following the steps in the "Generating the plugin-cfg.xml file" topic in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). Also see "Administer Users" on page 164 for details on how to assign monitoring applications to a Tivoli Enterprise Portal user. The *IBM Tivoli Monitoring Installation and Setup Guide* includes information on how to install application support.
Migrate from a basic dashboard environment to a dashboard environment with single sign-on and per user authorization.	"Migrating a basic monitoring dashboard environment to a dashboard environment with single sign-on and per user authorization controls" on page 45
Determine if you want to control UISolutions imports. (New and updated dashboard applications automatically import their UISolutions definitions into the dashboard data provider.)	"Controlling UISolutions imports" on page 67

Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls

Setup an advanced dashboard environment if you want to use IBM Dashboard Application Services Hub with monitoring dashboard applications such as IBM Infrastructure Management Dashboards for Servers and IBM Infrastructure Management Dashboards for VMware or with custom dashboards, using single sign-on and permissions that control what monitoring resources a user can access in the dashboards.

By using single sign-on, your IBM Dashboard Application Services Hub users can launch the Tivoli Enterprise Portal browser client without entering credentials when the portal browser client is started. You can use either authorization policies or Tivoli Enterprise Portal permissions to control what managed systems and managed system groups individual users or members of user groups can access in the dashboards and whether they can display situation events.

To use single sign-on, you must install and configure an LDAP user registry that will contain the credentials of users who will login to IBM Dashboard Application Services Hub and the portal server. Then you configure IBM Dashboard Application Services Hub and the portal server to use the same LDAP user registry to authenticate users and to perform single sign-on using Lightweight Third Party Authentication (LTPA) tokens. You can also use the same LDAP user registry to authenticate users of other applications such as Netcool/OMNIbus WebGUI or Tivoli Business Service Manager if those users will launch the portal client browser or Dashboard Application Services Hub.

Next you configure a dashboard data provider connection from the Dashboard Application Services Hub to the portal server and indicate that single sign-on should be used. The Dashboard Application Services Hub uses a HTTP or HTTPS connection to the dashboard data provider component of the portal server to retrieve monitoring data. Real-time monitoring data is retrieved from the hub monitoring server and monitoring agents and historical monitoring data is retrieved from the Tivoli Data Warehouse. Not all monitoring dashboard applications support retrieving historical data from the Tivoli Data Warehouse.

Dashboard Application Services Hub uses roles, for users or user groups, to control what pages a user can access. However, the dashboard data provider performs the authorization of the monitoring resources that are displayed in those pages. You have two options for authorizing the monitoring resources that can be accessed by dashboard users:

- Use the Tivoli Authorization Policy Server and tivcmd Command-Line Interface for Authorization Policy to create roles and permissions, which are collectively called authorization policies.

  These authorization policies control which managed systems and managed system groups a dashboard user can access. Roles are created for job functions with permissions to view specific managed systems or managed system groups. Users acquire permissions based on the role (or roles) that user belongs to. Users can be assigned to roles directly or the user groups that they are members of can be assigned to roles. The permissions also specify the type of object that can be accessed for a managed system or managed system group. The supported object types are event (for situation events) and attribute group (for monitoring data retrieved from an agent).

  OR

- Use Tivoli Enterprise Portal authorization permissions and monitored application assignments for your dashboard users. This is the default authorization method if authorization policies are not enabled in the portal server configuration.

  With this option, you create Tivoli Enterprise Portal users for each of your dashboard users using the Tivoli Enterprise Portal User Administration dialog. Using the same dialog, you can grant a user permission to view events and assign the user one or more monitored applications that they can view. These steps can also be performed using the **tacmd** CLI.

  Tivoli Enterprise Portal authorization is less granular than authorization policies. While authorization policies allow you to grant a dashboard user permission to view only specific managed systems or members of specific managed system groups, Tivoli Enterprise Portal authorization is at the monitored application level. In other words, a user is assigned permission to view all managed systems of a particular agent application type, for example all Windows OS agents.

Authorization polices only control which monitored resources can be accessed in monitoring dashboards. If your dashboard users will also use the Tivoli Enterprise Portal client then Tivoli Enterprise Portal permissions and agent application assignments will control what monitored resources can be accessed in the portal client. The set of monitored resources that users can view in dashboards might be different than the monitored resources they can view in the Tivoli Enterprise Portal client. This can occur if the permissions are inconsistent or if the authorization policies are more restrictive.

Example of more restrictive authorization policies

Assume the user is granted permission to view a subset of Windows OS agents in Dashboard Application Services Hub using authorization policies and the user is assigned the Windows OS application type in their Tivoli Enterprise Portal permissions. In this scenario, the user will only see the authorized Windows OS agents in the dashboards but they will see all Windows OS agents when they access the Tivoli Enterprise Portal client.

Example of inconsistent permissions

Assume the user is granted permission to view a subset of Windows OS agents in Dashboard Application Services Hub using authorization policies but the user is not assigned the Windows OS application type in their Tivoli Enterprise Portal permissions. In this scenario, the user will see their authorized Windows OS agents in the dashboards but they will not see any Windows OS agents when they access the Tivoli Enterprise Portal client.

When you are initially setting up your monitoring and dashboard environment, best practice is that you start with Tivoli Enterprise Portal permissions and monitored application assignments. After you are able to see monitoring data in Dashboard Application Services Hub and your administrators have created authorization policies, then reconfigure the portal server if you want to start using authorization policies.

Note: Tivoli Enterprise Portal permissions and authorization policies only control access to monitored resources in the dashboards. They do not control access to monitored resources displayed in reports using Tivoli Common Reporting.

Prerequisites

- Install and configure the base IBM Tivoli Monitoring monitoring server, portal server, and portal client components using the instructions in the *IBM Tivoli Monitoring Installation and Setup Guide*. When configuring the portal server, enable the dashboard data provider.
- Install and configure the monitoring agents whose data will be displayed in the monitoring dashboards. Install their application support in the monitoring servers, portal server, and desktop portal client if it is being used, using the instructions in the *IBM Tivoli Monitoring Installation and Setup Guide*.
- Install and configure Dashboard Application Services Hub and your dashboard monitoring applications, see "Required software and memory requirements for a dashboard environment" in the *IBM Tivoli Monitoring Installation and Setup Guide*. Also see "Installing and configuring the IBM Infrastructure Management Dashboards for Servers" in the *IBM Tivoli Monitoring Installation and Setup Guide*, if you will be installing that dashboard application.
- Determine if you will use authorization policies or Tivoli Enterprise Portal authorization for your dashboard users. If you plan to use authorization policies, install and configure the Tivoli Authorization Policy Server and tivcmd Command-Line Interface for Authorization Policy components using the instructions in the "Installing and configuring the Tivoli Authorization Policy Server and tivcmd Command-Line Interface for Authorization Policy" topic in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Roadmap

Use the following roadmap to help you get started:

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls

Step	**Description**	**Where to find information**
1 (required)	Setup an LDAP server such as Tivoli Directory Server or Microsoft Active Directory to authenticate Dashboard Application Services Hub and portal server users and add your users to this registry.	See "Prerequisites for configuring LDAP authentication on the portal server" on page 99, then refer to the documentation for your LDAP server.
2 (required)	Ensure the time is synchronized to UTC on your portal server and Dashboard Application Services Hub.	For more information and for planning considerations for using single sign-on, see "About single sign-on" on page 102.

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls (continued)

Step	**Description**	**Where to find information**
3 (required)	Use the WebSphere® Administrator Console of IBM Dashboard Application Services Hub to configure the Dashboard Application Services Hub application server to use the LDAP user registry to authenticate users and to enable single sign-on. **Note:** During the configuration, specify a realm name and a domain name. These same values must be specified when configuring the portal server and any other applications that perform single sign-on with the portal server or the dashboard server. • The domain name is the Internet or Intranet domain for which SSO is configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for SSO. • A realm identifies a set of federated repositories used by the portal server and other application servers. You can choose your own realm name, but this value must be the same across all applications that are configured for SSO within the specified domain.	Refer to the *Jazz for Service Management Configuration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on configuring Jazz for Service Management to use a central user registry, configuring SSO, configuring the LTPA token timeout values, and configuring a TLS/SSL connection to the LDAP server.

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls (continued)

Step	Description	Where to find information
4 (required)	Configure the portal server to use an LDAP user registry and specify the realm name and domain used for single sign-on. To configure the portal server to use LDAP, you can use the following options: • IBM Manage Tivoli Enterprise Monitoring Services utility • **itmcmd** command line interface on Linux and UNIX • TEPS/e administration console You use either IBM Manage Tivoli Enterprise Monitoring Services or the **itmcmd** command to enable LDAP user validation for the portal server. You can also use these utilities to configure the LDAP connection parameters unless: • You want to use a server besides Microsoft Active Directory or Tivoli Directory Server • You want to configure TLS/SSL between the portal server and the LDAP server • You need to specify advanced LDAP configuration parameters For these scenarios, you specify the type of **Other** when configuring the portal server and then use the TEPS/e administration console to complete the LDAP connection configuration. **Note:** You can also export the portal server's LTPA key or import the LTPA key from another application at the same time as configuring LDAP user authentication or you can perform these steps after you have verified the portal server's LDAP authentication is working.	Use the instructions in one of the following topics to enable LDAP user validation on the portal server: • "Using Manage Tivoli Enterprise Monitoring Services to configure the portal server for LDAP authentication" on page 107 • "Using the Linux or UNIX command line to configure the portal server for LDAP authentication" on page 111 Then, follow the instructions in "Using the TEPS/e administration console" on page 113 if you specified an LDAP server type of **Other** when enabling LDAP user validation for the portal server. **Usage notes:** If you are using Microsoft Active Directory, see "LDAP user authentication using Microsoft Active Directory" on page 129 for planning and configuration information specific to this type of LDAP server. If you are using Tivoli Directory Server, see Understanding single sign-on between IBM Tivoli Monitoring and Tivoli Integrated Portal using Tivoli Directory Server in the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring). These instructions explain how to map entries configured in Tivoli Directory Server to the information configured using the TEPS/e administration console. Ignore the steps provided for Tivoli Integrated Portal.
5 (required)	Login to the Tivoli Enterprise Portal client as sysadmin, then map your existing Tivoli Enterprise Portal user IDs to LDAP distinguished names except for sysadmin. If you do not have any Tivoli Enterprise Portal user IDs besides `sysadmin`, create a Tivoli Enterprise Portal user ID for at least one of your LDAP users and, when creating the user ID, enter the user's LDAP distinguished name. You will login as one of your LDAP users in a later task to verify that data can be displayed in your monitoring dashboards. Use the Tivoli Enterprise Portal client to assign this user the monitoring applications that will be displayed in the dashboards and permission to view events if situation event data is displayed in the dashboard.	If you have existing Tivoli Enterprise Portal users, see "Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120. If you need to create a new Tivoli Enterprise Portal user ID, see "Adding a user ID" on page 170. See "Administer Users" on page 164 for details on assigning monitoring applications and permissions to Tivoli Enterprise Portal users. See "Reconfiguring the browser client for SSO" on page 122 if Dashboard Application Services Hub and the portal server are on the same computer.

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls (continued)

Step	Description	Where to find information
6 (optional best practice)	Verify that you can login to the Tivoli Enterprise Portal client as an LDAP user who has been mapped to a Tivoli Enterprise Portal user ID.	N/A
7 (optional best practice)	Configure a TLS/SSL connection between the portal server and LDAP server if you want to secure this communication.	"Configuring TLS/SSL communication between the portal server and the LDAP server" on page 119
8 (optional best practice)	Verify that you can login to the Tivoli Enterprise Portal client as an LDAP user who has been mapped to a Tivoli Enterprise Portal user ID.	N/A
9 (required)	You must ensure the following applications are using the same LTPA key as the portal server: • A web-based or web-enabled application that launches the Tivoli Enterprise Portal • A web-based or web-enabled application that can be launched from the Tivoli Enterprise Portal client • IBM Dashboard Application Services Hub • Another application such as Tivoli Integrated Portal that uses the IBM Tivoli Monitoring charting web service Determine which application will be the source of the LTPA key for all of the other participating SSO applications and export its LTPA key. The key file and the password used to encrypt the key must be provided to the administrators of the other participating applications.	If you decide that the portal server will be the source of the LTPA key, export its LTPA key using the export instructions in "Importing and exporting LTPA keys" on page 123. If IBM Dashboard Application Services Hub will be the source of the LTPA key, see "Exporting LTPA keys" in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). Otherwise, refer to the documentation of the application whose LTPA key will be exported to determine how to perform the export operation.
10 (required)	The administrators of the other participating SSO applications must import the LTPA key that was exported in the previous step. They need the key file and the password that was used to encrypt the key.	To import an LTPA key into the portal server, see the import instructions in "Importing and exporting LTPA keys" on page 123. To import an LTPA key into IBM Dashboard Application Services Hub see "Importing LTPA keys" in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). See the documentation for the other participating SSO applications for instructions on importing the LTPA key.

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls (continued)

Step	Description	Where to find information
11 (required)	Login to IBM Dashboard Application Services Hub as an LDAP user who is also a dashboard hub administrative user and create a dashboard data provider connection using the URL `http://`*`hostname`*`:`*`port_number`*`/ibm/console/`. Because single sign-on has been enabled, you must specify the fully qualified hostname in this URL. The default port number for HTTP is 16310 and for HTTPS is 16311. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.	"Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 When creating your connection, select the box **Use the credentials of the user (requires SSO Configuration)**.
12 (required)	While logged into IBM Dashboard Application Services Hub as an administrative user, create a role that controls access to your dashboard application pages and assign dashboard users or user groups to the role. **Note:** Some dashboard applications such as IBM Infrastructure Management for VMware automatically create a role for its pages when the dashboard application is installed. However, other dashboard applications such as IBM Infrastructure Management Dashboards for Servers do not create a role during installation so you must create one or assign the dashboard pages to an existing role.	Refer to the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with roles that control access to dashboard pages.
13 (optional best practice)	Login to IBM Dashboard Application Services Hub as an LDAP user who has permission to view your dashboard pages and who has a Tivoli Enterprise Portal user ID that is assigned monitoring applications and permissions to view events. Then launch the dashboard applications, and verify data is displayed. By default, the URL is `http://`*`hostname`*`:`*`port_number`*`/ibm/console/`. Because single sign-on has been enabled, you must specify the fully qualified hostname in this URL. The default port number for HTTP is 16310 and for HTTPS is 16311. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls (continued)

Step	Description	Where to find information
14 (optional best practice)	If you want to use HTTPS between Dashboard Application Services Hub and the dashboard data provider, perform these tasks:	
	1. Configure TLS/SSL between the dashboard hub and data provider.	"Configuring TLS/SSL communication between Dashboard Application Services Hub and the dashboard data provider" on page 213
	2. Login to IBM Dashboard Application Services Hub as an administrative user who has been assigned the `administrator` and `iscadmins` roles and delete the dashboard data provider connection that you previously created.	Refer to the IBM Dashboard Application Services Hub online help and the *Jazz for Service Management Integration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with data provider connections.
	3. While still logged into IBM Dashboard Application Services Hub as an administrative user, create the connection again and this time specify HTTPS as the protocol.	"Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 When creating your connection, select the box **Use the credentials of the user (requires SSO Configuration)**.
	4. Login to IBM Dashboard Application Services Hub as a user who has permission to view your dashboard pages, then launch the dashboard application again and verify data is displayed.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

Table 3. Roadmap for setting up a monitoring dashboard environment with single sign-on and with per user authorization controls (continued)

Step	Description	Where to find information
15 (optional)	If you want to use authorization policies, perform these tasks:	
	1. Use the **tivcmd** CLI to assign authorization policy administrators, assign a user permission to distribute authorization policies, and create authorization policies to control which monitored resources your dashboard users can access. **Note:** After you have verified that you can use the **tivcmd** CLI to login to the Authorization Policy Server, configure TLS/SSL between the **tivcmd** CLI and the Authorization Policy Server so that subsequent commands are secured.	"Preparing to enable authorization policies" on page 186 and"Configuring TLS/SSL communication with the Authorization Policy Server" on page 216
	2. Enable authorization policy checking in the portal server. **Note:** Once this task is performed, only dashboard users who are assigned an authorization policy role will be able to view monitored resources in your dashboards.	"Enabling authorization policies in the portal server" on page 194
	3. Login to IBM Dashboard Application Services Hub as an LDAP user who has permission to view your dashboard pages and who has been assigned one or more authorization policy roles that give the user permission to view attribute group data, situation event data, or both for the managed systems or managed system groups that they can be displayed in your dashboard pages. Launch the dashboard pages and verify that the user can only see the monitored resources that they have been authorized for.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.
	4. Configure the portal server to use TLS/SSL when retrieving authorization policies from the Dashboard Application Services Hub where the Authorization Policy Server is installed.	"Configuring TLS/SSL communication with the Authorization Policy Server" on page 216

After you have the advanced dashboard monitoring environment setup, you might also need to perform the following tasks:

Table 4. Additional tasks required to setup your advanced monitoring environment with single sign-on and with per user authorization controls

Task	Where to find information
Review the roadmap if you plan to set up load balancing for a high availability dashboard environment.	"Setting up load balancing for a high availability dashboard environment" on page 53
Create situation definitions for events that your dashboard users will monitor.	See Situations for event monitoring in the *Tivoli Enterprise Portal User's Guide* and also see the *IBM Tivoli Monitoring Command Reference* for information on the **tacmd** commands used to work with situations.
Create managed system groups that can be used to group managed systems for display in dashboard pages.	See Managing the environment in the *Tivoli Enterprise Portal User's Guide* and also see the *IBM Tivoli Monitoring Command Reference* for information on the **tacmd** commands used to work with system lists.

Table 4. Additional tasks required to setup your advanced monitoring environment with single sign-on and with per user authorization controls (continued)

Task	**Where to find information**
Configure historical data collection if you want to display historical data in your dashboard pages. **Note:** Not all monitoring dashboard applications support retrieving historical data from the Tivoli Data Warehouse.	Chapter 16, "Managing historical data," on page 457
For each new dashboard user: Ensure the dashboard user has permission to access the dashboard pages that they will work with. Determine if the user can be added to an existing LDAP group that is assigned to a Dashboard Application Services Hub role. If there is not an existing LDAP group that the user can be assigned to, complete one of the following tasks: • Best practice is to create a new LDAP group, add the user to the group, and then assign the group to a Dashboard Application Services Hub role that has permission to view the appropriate dashboard pages. OR • Assign the dashboard user directly to a Dashboard Application Services Hub role that has permission to view the appropriate dashboard pages.	Refer to the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with roles that control access to dashboard pages. See your LDAP server documentation for details on adding users to LDAP groups.
For each new dashboard user: If authorization polices are being used, ensure the dashboard user is assigned to one or more authorization policy roles that give the user permission to view attribute group data, situation event data, or both for the managed systems or managed system groups that they will be monitoring. Determine if the user can be added to an existing LDAP group that is already assigned authorization policy roles with the required permissions. If the user cannot be added to an existing LDAP group, complete one of the following tasks: • Best practice is to create a new LDAP group, add the user to the group, and then assign the group to the authorization policy roles. OR • Assign the dashboard user directly to the authorization policy roles.	See your LDAP server documentation for details on adding users to LDAP groups. See "Policy management scenarios" on page 187 and the *IBM Tivoli Monitoring Command Reference* chapter on the **`tivcmd`** CLI for details on creating and working with authorization policies.

Table 4. Additional tasks required to setup your advanced monitoring environment with single sign-on and with per user authorization controls (continued)

Task	**Where to find information**
For each new dashboard user: If Tivoli Enterprise Portal authorization is being used to control what monitored resources can be accessed in your dashboards, or if the new dashboards user will use the Tivoli Enterprise Portal client, then ensure the Tivoli Enterprise Portal user has the correct permission. First ensure there is a Tivoli Enterprise Portal user ID mapped to the dashboard user's LDAP distinguished name. Then determine if the Tivoli Enterprise Portal user should be assigned to an existing Tivoli Enterprise Portal group that is assigned the permissions and monitoring applications required by the new dashboard user. If there is not an existing group that can be used, complete one of the following tasks: • Best practice is to create a new Tivoli Enterprise Portal group, add the user to the group, and assign the group the appropriate permissions and application types. • Assign the Tivoli Enterprise Portal user the appropriate permissions and monitoring applications directly. If a dashboard user will not use the Tivoli Enterprise Portal client, they only need permission to view events and should be assigned the monitoring applications that they will be monitoring in the dashboard pages. For example, if the dashboard user will be using the Infrastructure Management Dashboards for Servers then they need to be assigned one or more of these application types: Linux OS, UNIX OS, or Windows OS. If the dashboard user will also use the Tivoli Enterprise Portal client, they might need additional permissions. To launch the Tivoli Enterprise Portal client from Dashboard Application Services Hub, the user must be assigned either the Tivoli Enterprise Portal <All Application> or Tivoli Monitoring Server application. Otherwise, they might see the KFWITM388E message after performing the launch.	See “Managing user IDs” on page 169 for details on creating new Tivoli Enterprise Portal user IDs. See “Managing user groups” on page 173 for details on adding Tivoli Enterprise Portal user IDs to groups. See “Administer Users” on page 164 for details on assigning monitoring applications and permissions to Tivoli Enterprise Portal users and groups.
Create custom dashboard pages and ensure the dashboard users are assigned a Dashboard Application Services Hub role with permission to view the custom pages.	“Creating custom dashboard pages that display monitoring data” on page 61

Table 4. Additional tasks required to setup your advanced monitoring environment with single sign-on and with per user authorization controls (continued)

Task	Where to find information
Install a new monitoring dashboard application in Dashboard Application Services Hub, assign the dashboard's pages to a new or existing role, and assign LDAP users or user groups to the role that controls access to the pages. **Note:** Some dashboard applications automatically create a role for its pages when the dashboard application is installed. If an HTTP server is being used with Dashboard Application Services Hub for load balancing, regenerate the web server plugin. If authorization policies are being used, ensure the dashboard users that have access to the new pages are assigned to one or more authorization policy roles that give the users permission to view attribute group data, situation event data, or both for the managed systems or managed system groups that they will be monitoring using the new dashboard application. Assign the dashboard user's Tivoli Enterprise Portal user ID or group the monitoring applications that will be displayed in the new dashboard application, if the dashboard user will also use the Tivoli Enterprise Portal client, or if Tivoli Enterprise Portal authorization are being used instead of authorization policies. **Note:** The application support for the agent must be installed in the portal server and monitoring server before you can see the agent's data in the new dashboards.	Follow the dashboard application's installation documentation. Then, refer to the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with roles that control access to dashboard pages. If required, regenerate the web server plugin by following the steps in the "Generating the plugin-cfg.xml file" topic in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). See Chapter 7, "Using role-based authorization policies," on page 181 and the *IBM Tivoli Monitoring Command Reference* chapter on the **tivcmd** CLI for details on creating and working with authorization policies. Also see "Administer Users" on page 164 for details on how to assign agent applications to a Tivoli Enterprise Portal user. The *IBM Tivoli Monitoring Installation and Setup Guide* includes information on how to install application support.
Determine if you want to control UISolutions imports. (New and updated dashboard applications automatically import their UISolutions definitions into the dashboard data provider.)	"Controlling UISolutions imports" on page 67

Migrating a basic monitoring dashboard environment to a dashboard environment with single sign-on and per user authorization controls

Migrate your basic dashboard environment to an advanced dashboard environment.

After you have setup a basic monitoring environment as described in "Setting up a basic monitoring environment without single sign-on and without per user authorization controls" on page 29, you can migrate to an advanced dashboard environment with single sign-on.

Roadmap

Use the following roadmap to help you get started:

Table 5. Roadmap for migrating to an advanced dashboard environment

Step	Description	Where to find information
1 (required)	Setup an LDAP server such as Tivoli Directory Server or Microsoft Active Directory to authenticate Dashboard Application Services Hub and portal server users and add your users to this registry.	See "Prerequisites for configuring LDAP authentication on the portal server" on page 99, then refer to the documentation for your LDAP server.
2 (required)	Ensure the time is synchronized to UTC on your portal server and Dashboard Application Services Hub.	For more information and for planning considerations for using single sign-on, see "About single sign-on" on page 102.
3 (required)	Use the WebSphere Administrator Console of IBM Dashboard Application Services Hub to configure the Dashboard Application Services Hub application server to use the LDAP user registry to authenticate users and to enable single sign-on. **Note:** During the configuration, specify a realm name and a domain name. These same values must be specified when configuring the portal server and any other applications that perform single sign-on with the portal server or the dashboard server. • The domain name is the Internet or Intranet domain for which SSO is configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for SSO. • A realm identifies a set of federated repositories used by the portal server and other application servers. You can choose your own realm name, but this value must be the same across all applications that are configured for SSO within the specified domain.	Refer to the *Jazz for Service Management Configuration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on configuring Jazz for Service Management to use a central user registry, configuring SSO, configuring the LTPA token timeout values, and configuring a TLS/SSL connection to the LDAP server.

Table 5. Roadmap for migrating to an advanced dashboard environment (continued)

Step	Description	Where to find information
4 (required)	Configure the portal server to use an LDAP user registry and specify the realm name and domain used for single sign-on. To configure the portal server to use LDAP, you can use the following options: • IBM Manage Tivoli Enterprise Monitoring Services utility • **itmcmd** command line interface on Linux and UNIX • TEPS/e administration console You use either IBM Manage Tivoli Enterprise Monitoring Services or the **itmcmd** command to enable LDAP user validation for the portal server. You can also use these utilities to configure the LDAP connection parameters unless: • You want to use a server besides Microsoft Active Directory or Tivoli Directory Server • You want to configure TLS/SSL between the portal server and the LDAP server • You need to specify advanced LDAP configuration parameters For these scenarios, you specify the type of **Other** when configuring the portal server and then use the TEPS/e administration console to complete the LDAP connection configuration. **Note:** You can also export the portal server's LTPA key or import the LTPA key from another application at the same time as configuring LDAP user authentication or you can perform these steps after you have verified the portal server's LDAP authentication is working.	Use the instructions in one of the following topics to enable LDAP user validation on the portal server: • "Using Manage Tivoli Enterprise Monitoring Services to configure the portal server for LDAP authentication" on page 107 • "Using the Linux or UNIX command line to configure the portal server for LDAP authentication" on page 111 Then, follow the instructions in "Using the TEPS/e administration console" on page 113 if you specified an LDAP server type of **Other** when enabling LDAP user validation for the portal server. **Usage notes:** If you are using Microsoft Active Directory, see "LDAP user authentication using Microsoft Active Directory" on page 129 for planning and configuration information specific to this type of LDAP server. If you are using Tivoli Directory Server, see Understanding single sign-on between IBM Tivoli Monitoring and Tivoli Integrated Portal using Tivoli Directory Server in the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring). These instructions explain how to map entries configured in Tivoli Directory Server to the information configured using the TEPS/e administration console. Ignore the steps provided for Tivoli Integrated Portal.
5 (required)	Login to the Tivoli Enterprise Portal client as sysadmin, then map your existing Tivoli Enterprise Portal user IDs to LDAP distinguished names except for sysadmin. If you do not have any Tivoli Enterprise Portal user IDs besides `sysadmin`, create a Tivoli Enterprise Portal user ID for at least one of your LDAP users and, when creating the user ID, enter the user's LDAP distinguished name. You will login as one of your LDAP users in a later task to verify that data can be displayed in your monitoring dashboards. Use the Tivoli Enterprise Portal client to assign this user the monitoring applications that will be displayed in the dashboards and permission to view events if situation event data is displayed in the dashboard.	If you have existing Tivoli Enterprise Portal users, see "Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120. If you need to create a new Tivoli Enterprise Portal user ID, see "Adding a user ID" on page 170. See "Administer Users" on page 164 for details on assigning monitoring applications and permissions to Tivoli Enterprise Portal users. See "Reconfiguring the browser client for SSO" on page 122 if Dashboard Application Services Hub and the portal server are on the same computer.

Table 5. Roadmap for migrating to an advanced dashboard environment (continued)

Step	Description	Where to find information
6 (optional best practice)	Verify that you can login to the Tivoli Enterprise Portal client as an LDAP user who has been mapped to a Tivoli Enterprise Portal user ID.	N/A
7 (optional best practice)	Configure a TLS/SSL connection between the portal server and LDAP server if you want to secure this communication.	"Configuring TLS/SSL communication between the portal server and the LDAP server" on page 119
8 (optional best practice)	Verify that you can login to the Tivoli Enterprise Portal client as an LDAP user who has been mapped to a Tivoli Enterprise Portal user ID.	N/A
9 (required)	You must ensure the following applications are using the same LTPA key as the portal server: • A web-based or web-enabled application that launches the Tivoli Enterprise Portal • A web-based or web-enabled application that can be launched from the Tivoli Enterprise Portal client • IBM Dashboard Application Services Hub • Another application such as Tivoli Integrated Portal that uses the IBM Tivoli Monitoring charting web service Determine which application will be the source of the LTPA key for all of the other participating SSO applications and export its LTPA key. The key file and the password used to encrypt the key must be provided to the administrators of the other participating applications.	If you decide that the portal server will be the source of the LTPA key, export its LTPA key using the export instructions in "Importing and exporting LTPA keys" on page 123. If IBM Dashboard Application Services Hub will be the source of the LTPA key, see "Exporting LTPA keys" in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). Otherwise, refer to the documentation of the application whose LTPA key will be exported to determine how to perform the export operation.
10 (required)	The administrators of the other participating SSO applications must import the LTPA key that was exported in the previous step. They need the key file and the password that was used to encrypt the key.	To import an LTPA key into the portal server, see the import instructions in "Importing and exporting LTPA keys" on page 123. To import an LTPA key into IBM Dashboard Application Services Hub see "Importing LTPA keys" in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). See the documentation for the other participating SSO applications for instructions on importing the LTPA key.

Table 5. Roadmap for migrating to an advanced dashboard environment (continued)

Step	Description	Where to find information
11 (required)	Login to IBM Dashboard Application Services Hub as an LDAP user who is also a dashboard hub administrative user using the URL `http://`*`hostname`*`:`*`port_number`*`/ibm/console/`. Because single sign-on has been enabled, you must specify the fully qualified hostname in this URL. The default port number for HTTP is 16310 and for HTTPS is 16311. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment. Then **delete** the existing dashboard data provider connection and **create a new** dashboard data provider connection that supports single sign-on.	“Creating a connection to the IBM Tivoli Monitoring dashboard data provider” on page 58 When creating your connection, select the box **Use the credentials of the user (requires SSO Configuration)**.
12 (optional best practice)	Login to IBM Dashboard Application Services Hub as an LDAP user who has permission to view your dashboard pages and who has a Tivoli Enterprise Portal user ID that is assigned monitoring applications and permissions to view events. Then launch the dashboard applications, and verify data is displayed. By default, the URL is `http://`*`hostname`*`:`*`port_number`*`/ibm/console/`. Because single sign-on has been enabled, you must specify the fully qualified hostname in this URL. The default port number for HTTP is 16310 and for HTTPS is 16311. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

Table 5. Roadmap for migrating to an advanced dashboard environment (continued)

Step	Description	Where to find information
13 (optional best practice)	If you have **not** already configured HTTPS between Dashboard Application Services Hub and the dashboard data provider, perform these tasks.	
	1. Configure TLS/SSL between the dashboard hub and data provider.	"Configuring TLS/SSL communication between Dashboard Application Services Hub and the dashboard data provider" on page 213
	2. Login to IBM Dashboard Application Services Hub as an administrative user who has been assigned the `administrator` and `iscadmins` roles and delete the dashboard data provider connection that you previously created.	Refer to the IBM Dashboard Application Services Hub online help and the *Jazz for Service Management Integration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for details on how to work with data provider connections.
	3. While still logged into IBM Dashboard Application Services Hub as an administrative user, create the connection again and this time specify HTTPS as the protocol.	"Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 When creating your connection, select the box **Use the credentials of the user (requires SSO Configuration)**.
	4. Login to IBM Dashboard Application Services Hub as a user who has permission to view your dashboard pages, then launch the dashboard application again and verify data is displayed.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

Table 5. Roadmap for migrating to an advanced dashboard environment (continued)

Step	Description	Where to find information
14 (optional)	Create authorization policies and enable authorization policy checking if you want role-based control rather than Tivoli Enterprise Portal permissions and monitoring application assignment.	
	1. Use the **`tivcmd`** CLI to assign authorization policy administrators, assign a user permission to distribute authorization policies, and create authorization policies to control which monitored resources your dashboard users can access. **Note:** After you have verified that you can use the **`tivcmd`** CLI to login to the Authorization Policy Server, configure TLS/SSL between the **`tivcmd`** CLI and the Authorization Policy Server so that subsequent commands are secured.	"Preparing to enable authorization policies" on page 186 and"Configuring TLS/SSL communication with the Authorization Policy Server" on page 216
	2. Enable authorization policy checking in the portal server. **Note:** Once this task is performed, only dashboard users who are assigned an authorization policy role will be able to view monitored resources in your dashboards.	"Enabling authorization policies in the portal server" on page 194
	3. Login to IBM Dashboard Application Services Hub as an LDAP user who has permission to view your dashboard pages and who has been assigned one or more authorization policy roles that give the user permission to view attribute group data, situation event data, or both for the managed systems or managed system groups that they can be displayed in your dashboard pages. Launch the dashboard pages and verify that the user can only see the monitored resources that they have been authorized for.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.
	4. Configure the portal server to use TLS/SSL when retrieving authorization policies from the Dashboard Application Services Hub where the Authorization Policy Server is installed.	"Configuring TLS/SSL communication with the Authorization Policy Server" on page 216

Table 5. Roadmap for migrating to an advanced dashboard environment (continued)

Step	Description	Where to find information
15 (optional)	For each new dashboard user: If Tivoli Enterprise Portal authorization is being used to control what monitored resources can be accessed in your dashboards, or if the new dashboards user will use the Tivoli Enterprise Portal client, then ensure the Tivoli Enterprise Portal user has the correct permission. First ensure there is a Tivoli Enterprise Portal user ID mapped to the dashboard user's LDAP distinguished name. Then determine if the Tivoli Enterprise Portal user should be assigned to an existing Tivoli Enterprise Portal group that is assigned the permissions and monitoring applications required by the new dashboard user. If there is not an existing group that can be used, complete one of the following tasks: • Best practice is to create a new Tivoli Enterprise Portal group, add the user to the group, and assign the group the appropriate permissions and application types. • Assign the Tivoli Enterprise Portal user the appropriate permissions and monitoring applications directly. If a dashboard user will not use the Tivoli Enterprise Portal client, they only need permission to view events and should be assigned the monitoring applications that they will be monitoring in the dashboard pages. For example, if the dashboard user will be using the Infrastructure Management Dashboards for Servers then they need to be assigned one or more of these application types: Linux OS, UNIX OS, or Windows OS. If the dashboard user will also use the Tivoli Enterprise Portal client, they might need additional permissions. To launch the Tivoli Enterprise Portal client from Dashboard Application Services Hub, the user must be assigned either the Tivoli Enterprise Portal <All Application> or Tivoli Monitoring Server application. Otherwise, they might see the KFWITM388E message after performing the launch.	See “Managing user IDs” on page 169 for details on creating new Tivoli Enterprise Portal user IDs. See “Managing user groups” on page 173 for details on adding Tivoli Enterprise Portal user IDs to groups. See “Administer Users” on page 164 for details on assigning monitoring applications and permissions to Tivoli Enterprise Portal users and groups.
16 (optional)	Review the roadmap if you plan to set up load balancing for a high availability dashboard environment.	“Setting up load balancing for a high availability dashboard environment” on page 53

Setting up load balancing for a high availability dashboard environment

If you want to create a high availability dashboard environment, you can use an HTTP server configured for load balancing between your dashboard users and a cluster of Dashboard Application Services Hub nodes, and also between Dashboard Application Services Hub nodes and multiple Tivoli Enterprise Portal Server.

Work load is distributed by session, not by request. When one of the servers fails, new and existing user sessions are directed to an operational server.

When you have multiple portal servers in a monitoring environment, you must designate one of the portal servers as the *read-write* portal server and the others must be *read-only* portal servers. Tivoli Enterprise Portal client users and tacmd CLI users who make customization changes must connect to the read-write portal server to make their updates. The content from the database for the read-write portal server must be periodically exported and then imported into the database for each read-only portal server. Because the dashboards in Dashboard Application Services Hub do not make customization changes, the HTTP server can redirect dashboard requests to any of the portal servers for the monitoring environment. In a load balancing environment, you might not know which portal server (the read-write portal server or a read-only portal server) is serving your requests when you launch the portal browser client from Dashboard Application Services Hub. Because of this limitation, best practice is to *not* use the portal browser client to make configuration changes when the portal client is launched from the dashboard server. If you need to make configuration changes, configure the portal client to connect to the read-write portal server directly and then start the client.

Tip: When using launch in context in a load balancing environment you might receive a message if, you have already launched into the portal client and the portal server that was serving its requests goes down and a secondary portal server begins servicing requests. The portal client cannot register this switch and you might receive the message `KFWITM094W: The Tivoli Enterprise Client has lost contact with the Tivoli Enterprise Portal Server`. You must close the browser window and log in again to clear this message.

Important: If you have multiple Dashboard Application Services Hub nodes and use the Authorization Policy Server, the policy server can only be installed on one of the dashboard nodes because it does not support load balancing. When you enable authorization policy checking for each portal server, you specify the hostname and port for the Dashboard Application Services Hub where the policy server is installed. For information on what happens if the Dashboard Application Services Hub where the policy server is installed is not available, see “Managing the authorization policy store” on page 201.

The following figures show two types of load balancing environments with default port numbers:

1. An environment that includes a single Dashboard Application Services Hub node and multiple portal servers.
2. An environment with load balancing for both Dashboard Application Services Hub and Tivoli Enterprise Portal Server.

Note: The portal server also has a local IBM HTTP server. When the load balancing HTTP server sends requests to the portal server, it is sending those requests to the portal server's local HTTP server on port 15200 (for HTTP) or 15201 (for HTTPS). Then the local HTTP server is responsible for forwarding the requests to the other portal server components (such as the dashboard data provider or the components that handle portal client requests).

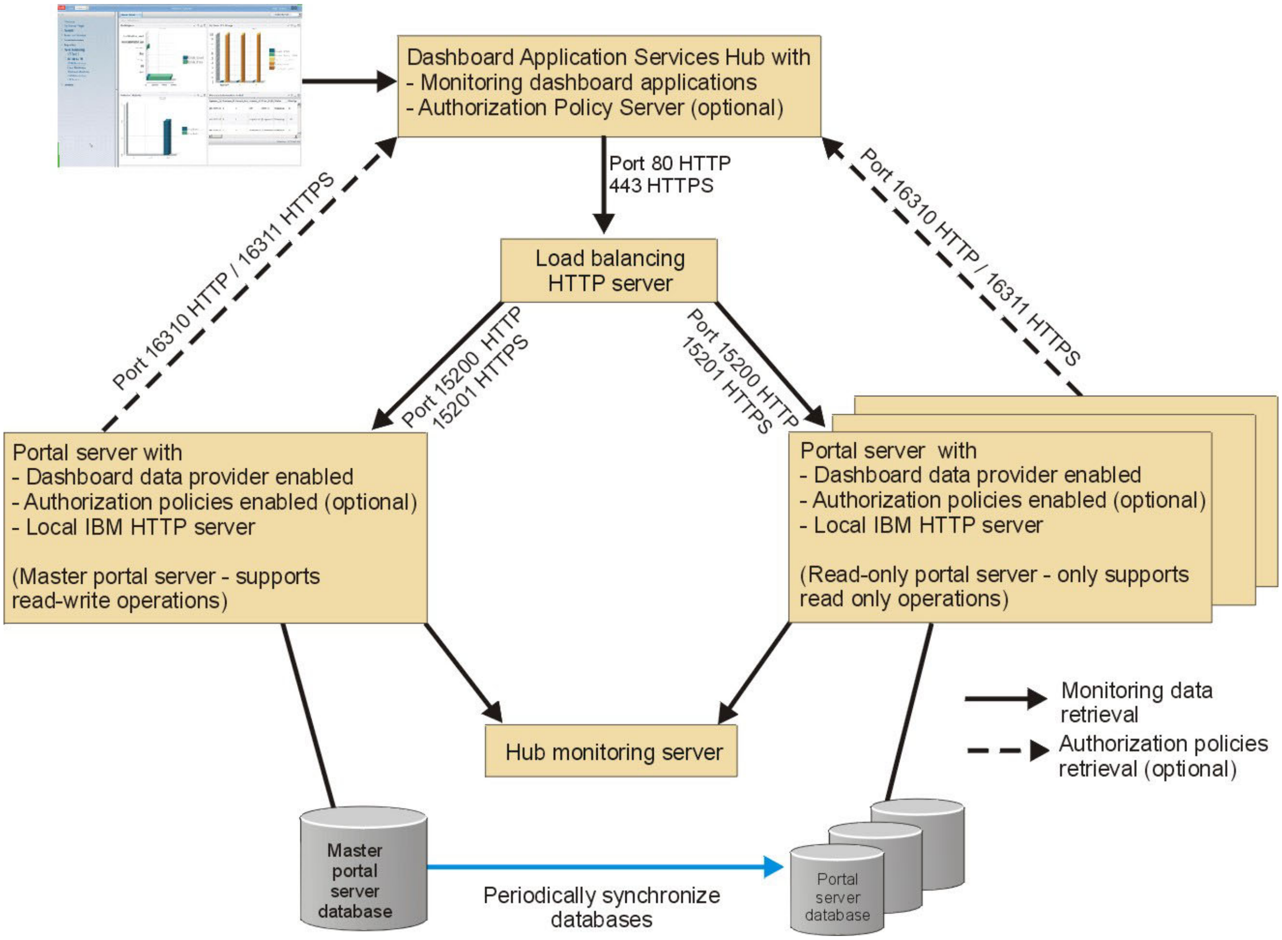

Dashboard Application Services Hub with
- Monitoring dashboard applications
- Authorization Policy Server (optional)
Port 80 HTTP
443 HTTPS
Load balancing HTTP server
Port 16310 HTTP / 16311 HTTPS
Port 16310 HTTP / 16311 HTTPS
Port 15200 HTTP
15201 HTTPS
Port 15200 HTTP
15201 HTTPS
Portal server with
- Dashboard data provider enabled
- Authorization policies enabled (optional)
- Local IBM HTTP server
(Master portal server - supports read-write operations)
Portal server with
- Dashboard data provider enabled
- Authorization policies enabled (optional)
- Local IBM HTTP server
(Read-only portal server - only supports read only operations)
Hub monitoring server
Monitoring data retrieval
Authorization policies retrieval (optional)
Master portal server database
Periodically synchronize databases
Portal server database

Figure 1. An environment that includes a single Dashboard Application Services Hub node and multiple portal servers

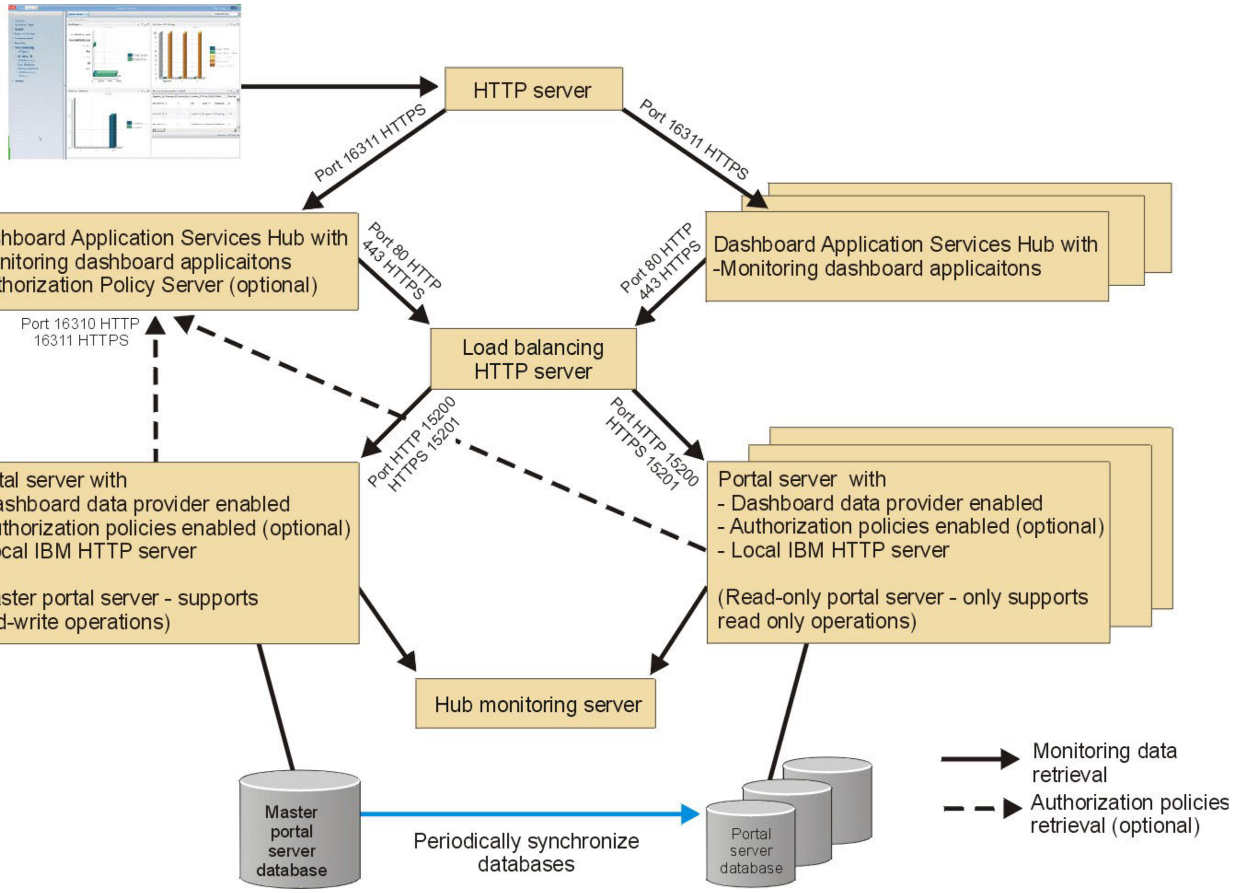


Figure 2. An environment with load balancing for both Dashboard Application Services Hub and Tivoli Enterprise Portal Server

The following roadmap describes the steps for configuring load balancing for your dashboard environment. It assumes that you have an existing dashboard environment with single sign-on support and now you are adding load balancing to the environment. The roadmap takes you through the steps to add multiple portal servers to your environment and then it optionally adds additional Dashboard Application Services Hub nodes.

Table 6. Roadmap for setting up load balancing for your dashboard environment

Step	Task	Where to find information
1	Install and configure the additional portal servers. Also install application support on each portal server. When you configure each portal server, enable the dashboard data provider. If you already have authorization policies enabled in the dashboard environment, enable authorization policy checking in each portal server specifying the hostname and port number of the Dashboard Application Services Hub node where the Authorization Policy Server is installed. **Note:** When you enable authorization policies and specify HTTPS for the protocol used to retrieve policies from the Authorization Policy Server, you must also configure TLS/SSL communication between the portal server and the Authorization Policy Server.	"Installing IBM Tivoli Monitoring" in the *IBM Tivoli Monitoring Installation and Setup Guide*. "Configuring TLS/SSL communication between the portal server and the Authorization Policy Server" on page 222

Table 6. Roadmap for setting up load balancing for your dashboard environment (continued)

Step	Task	Where to find information
2	Determine which portal server you want to use for read-write operations and the approach you want to use to periodically export data from the database for the read-write portal server into the database for each read-only portal server. 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. **Note:** If the various portal server databases are not running on the same OS version, then the RDBMS backup and restore utilities might not work.	See your database server documentation if you decide to use RDBMS backup utilities. If you choose to use the portal server **migrate-export** and **migrate-import** commands, see Chapter 18, "Replicating the Tivoli Enterprise Portal Server database," on page 547.
3	Ensure the time is synchronized to UTC on the new portal servers and configure single sign-on on each new portal server. **Note:** Ensure that all of the portal servers and Dashboard Application Services Hub nodes are using the same LTPA key.	For each new portal server, perform steps 4-10 in "Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls" on page 34. **Note:** Perform step 5 only on the read-write portal server. Before performing steps 6-10 on a read-only portal server, ensure that the customization updates for step 5 are exported from the database for the read-write portal server and then imported into the database of each read-only portal server.
4	Install and configure an HTTP server to load balance across the multiple portal servers. **Note:** Best practice is to install an HTTP server that is dedicated to Tivoli Enterprise Portal Server load balancing.	"Configuring an HTTP Server to load balance multiple Tivoli Enterprise Portal Servers" on page 63
5	If dashboard users will launch the Tivoli Enterprise Portal client, you must configure the browser client to only use HTTP requests and instruct existing portal client browser users to clear their browser's application and jar file cache so that they can pick up the configuration change. In addition, the portal client user IDs must be assigned either the Tivoli Enterprise Portal <All Application> or Tivoli Monitoring Server application.	See "Configuring HTTP communication between the portal client and server" in the *IBM Tivoli Monitoring Installation and Setup Guide* and "Clearing the JAR cache" in the *IBM Tivoli Monitoring Troubleshooting Guide*.

Table 6. Roadmap for setting up load balancing for your dashboard environment (continued)

Step	Task	Where to find information
6	If you are using HTTPS in your dashboard environment, perform these steps:	
	1. Configure TLS/SSL communication between the load balancing HTTP server and each portal server's HTTP server.	“Configuring TLS/SSL communication between the load balancing HTTP Server and each portal server's local HTTP server” on page 223
	2. Configure TLS/SSL communication between the Dashboard Application Services Hub and the HTTP server used for load balancing multiple portal servers.	“Configuring TLS/SSL communication between Dashboard Application Services Hub and an HTTP server used for load balancing multiple portal servers” on page 226
7	Login to IBM Dashboard Application Services Hub as an LDAP user who is also a dashboard hub administrative user. Delete the dashboard data provider connection in Dashboard Application Services Hub and add a new connection that specifies the fully qualified hostname and port number of the load balancing HTTP server in place of the fully qualified hostname and port number of a portal server. **Note:** Typically, the HTTP port number is 80 and the HTTPS port number is 443. Ensure the box **Use the credentials of the user (requires SSO configuration)** is selected.	“Creating a connection to the IBM Tivoli Monitoring dashboard data provider” on page 58
8	Login to IBM Dashboard Application Services Hub as an LDAP user who has permission to view your dashboard pages and their content. Then launch the dashboard applications, and verify data is displayed.	See your dashboard application's user guide for details on how to launch and use the dashboard. **Tip:** First select **System Status and Health** > **Dashboard Health Checks** to verify your environment is working correctly. Then if you are using Infrastructure Management Dashboards for Servers, select **System Status and Health** > **Server Dashboards**. For more information on using Infrastructure Management Dashboards for Servers, see the OS agent user's guides.

Table 6. Roadmap for setting up load balancing for your dashboard environment (continued)

Step	Task	Where to find information
9	If you want to use multiple Dashboard Application Services Hub nodes in your dashboard environment, perform these steps:	
	1. Configure load balancing for Dashboard Application Services Hub and ensure that your monitoring dashboard applications such as Infrastructure Management Dashboards for Servers are installed on each dashboard node. **Note:** Install the Authorization Policy Server only on a single node in the Dashboard Application Services Hub cluster.	See the *Jazz for Service Management Configuration Guide* for details on configuring load balancing across multiple Dashboard Application Services Hub nodes.
	2. If you are using HTTPS in the dashboard environment, configure TLS/SSL communication between each new Dashboard Application Services Hub node and the HTTP server used for load balancing multiple portal servers.	"Configuring TLS/SSL communication between Dashboard Application Services Hub and an HTTP server used for load balancing multiple portal servers" on page 226
	3. On each new Dashboard Application Services Hub node, login to the IBM Dashboard Application Services Hub node as an LDAP user who is also a dashboard hub administrative user. Add a new connection that specifies the fully qualified hostname and port number of the load balancing HTTP server in place of the fully qualified hostname and port number of a portal server. **Note:** Typically, the HTTP port number is 80 and the HTTPS port number is 443. Ensure the box **Use the credentials of the user (requires SSO configuration)** is selected.	"Creating a connection to the IBM Tivoli Monitoring dashboard data provider"

Creating a connection to the IBM Tivoli Monitoring dashboard data provider

To retrieve metrics about your managed systems and situation events from the monitoring dashboard application in IBM Dashboard Application Services Hub, such as IBM Infrastructure Management Dashboards for Servers, or IBM Infrastructure Management Dashboards for VMware, or in custom dashboards, you must first have a connection that is established to the IBM Tivoli Monitoring dashboard data provider that is on your Tivoli Enterprise Portal Server.

This connection procedure is one that you do not need to repeat unless the configuration of the portal server changes.

Before you begin

Connections are defined in the Dashboard Application Services Hub console. Before you create the connection, you must ensure the following steps have already been performed:

- Ensure that the dashboard data provider is enabled in the portal server configuration. See "Verifying the dashboard data provider is enabled" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

  If your IBM Tivoli Monitoring environment is configured for hot standby, you should configure a domain override value for the dashboard data provider in the portal server configuration before creating a connection to the dashboard data provider. The domain override value ensures the data provider's connection ID value does not change when the portal server is reconfigured to connect to the standby hub monitoring server.

- You must log into the Dashboard Application Services Hub as a user who has been assigned the `administrator` and `iscadmins` Dashboard Application Services Hub roles. These roles are required in order to create and manage connections. For information on assigning these roles to users, see the *Jazz for Service Management Administrator's Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html).

  If you plan to use single sign-on, you must also be logged into the Dashboard Application Services Hub as a user who is a member of the federated LDAP user registry shared by the dashboard hub server and by the portal server when you create the connection. For additional steps that must be preformed before creating a data provider connection that uses single sign-on, see "Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls" on page 34.
- If you plan to use HTTPS as the protocol connection, you must configure TLS/SSL between Dashboard Application Services Hub and the portal server before creating the data provider connection. For more information, see "Configuring TLS/SSL communication between Dashboard Application Services Hub and the dashboard data provider" on page 213.

  Tip: You can create the connection with HTTP protocol for your initial testing. Once your environment is working, then you can configure TLS/SSL between the servers, delete the connection, and re-add it with the HTTPS protocol.
- To define a connection to the dashboard data provider, you must know which network protocol is required, the host name and port number for the portal server, credentials to authenticate with the portal server, and whether single sign-on should be used.

Procedure

1. In the Dashboard Application Services Hub console, click **Console Settings** and select **Connections** (under **General**).
2. Click **Create new remote provider**. Fields are displayed for specifying the connection to the dashboard data provider.
3. In the **Protocol** field, select the application protocol for connecting to the portal server computer: `HTTP`, `HTTPS-SSL` (Secure Socket Layer), or `HTTPS-TLS` (Transport Layer Security).
4. Click inside the **Host name** field and enter the IP address or the fully qualified hostname of the portal server computer.

   Note: If you plan to enable single sign-on for your dashboard environment, you must enter the fully qualified hostname of the portal server. The hostname is included in the LTPA token that is sent from Dashboard Application Services Hub to the portal server so that it can verify that the request came from a server in the same Internet or Intranet domain.
5. Click inside the **Port** field and enter the port number for the portal server: 15200 for HTTP or 15201 for HTTPS. If using an HTTP server for load balancing between Dashboard Application Services Hub and multiple portal servers, enter your HTTP server port number, 80 for HTTP and 443 for HTTPS.
6. If your configuration has a firewall between the Dashboard Application Services Hub Server and the portal server, select the **Connection goes through**

a firewall check box and enter the fully qualified host name and port number (the default port is 16324) of the computer where Dashboard Application Services Hub is installed.

7. In the **Name** and **Password** fields, enter a user name and password that can authenticate with the portal server.

   If you are setting up a basic dashboard environment without single sign-on and per user authorization, enter a user who has been granted the Tivoli Enterprise Portal authorization described in the first table in "Setting up a basic monitoring environment without single sign-on and without per user authorization controls" on page 29. This user will be used to create the connection and to send all subsequent requests to the dashboard data provider on behalf of your dashboard users.

   If you are setting up a dashboard environment with single sign-on and per user authorization, enter your username and password. Your username and password will be used to send the request to the portal server to get the list of available data providers on the portal server. After that point, all subsequent requests to the data provider will include the name of the user who is logged into Dashboard Application Services Hub.
8. Click **Search** to populate the table with the data providers that are on the portal server computer.
9. Click the radio button for the dashboard data provider to select it and complete the remaining fields.
10. Edit the entries in the following fields:
    - **Name** is the same as the original data provider name. You can leave this as is.
    - **Description** is the same as the original data provider description. You can leave this as is.
    - **Provider ID** is, by default, initially `itm.`*`hub_monitoring_server_name`*`.`*`portal_server_host_name`* for dashboard data provider connections. If a domain override value was configured during portal server configuration, the override value replaces the *hub_monitoring_server_name* portion of the original ID string.

      You must change the Provider ID to `ITMSD` if you are using one of the following applications:
      - IBM Infrastructure Management Dashboards for Servers provided with IBM Tivoli Monitoring
      - IBM Infrastructure Management Dashboards for VMware
      - IBM Infrastructure Management Capacity Planner for VMware
      - IBM Infrastructure Management Capacity Planner for PowerVM®

      If you are not using any of the dashboard applications listed above, but have installed dashboard applications for other monitoring agents, check the documentation for those agents to determine if their dashboards require you to use `ITMSD`. If you are not using one of the dashboard applications mentioned above, but might in the future, or if you are not sure what Provider ID to use, best practice is to use `ITMSD`.
11. Select the **Use the credentials of the user (requires SSO Configuration)** check box if you are setting up a dashboard environment that uses single sign-on and per user authorization. When this option is checked, the LTPA token of the user who is logged into the Dashboard Application Services Hub is included in requests to the dashboard data provider when retrieving monitoring data.

12. After you are finished defining the connection, click **OK** to save it and connect to the dashboard data provider.

Results

The connection is made to the data provider and the connections table **Status** column shows the progress: `Pending`, `Working`, `Failed`, `No data sources`, or `Not configured`.

If an error occurs when creating the data provider connection, see the *IBM Tivoli Monitoring Troubleshooting Guide*.

See "Setting up a basic monitoring environment without single sign-on and without per user authorization controls" on page 29 or "Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls" on page 34 for additional steps to perform.

Creating custom dashboard pages that display monitoring data

IBM Dashboard Application Services Hub allows you to create custom dashboard pages.

A *page* is an arrangement of one or more widgets in the work area of the console. A *widget* is a user interface component that displays information in a console dashboard. Dashboard Application Services Hub provides a set of predefined widgets. Each widget is configured to retrieve information from a data provider that has a connection defined in Dashboard Application Services Hub. Each data provider divides its information into data sets.

Before you begin

For detailed information on predefined widgets, how to edit and customize each widget type, and how to create catalogs and pages with widgets, see the Dashboard Application Services Hub online help or the *Jazz for Service Management Integration Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html).

About this task

You can add the predefined widgets to dashboard pages or you can customize the predefined widgets to change their appearance. There are widgets for tables, lists, gauges, bar charts, pie charts, topology, and more. Widgets can also be logically organized by placing them in catalogs.

The IBM Tivoli Monitoring dashboard data provider's data sets correspond to agent attribute groups. These are the same attribute groups that you create queries for when customizing Tivoli Enterprise Portal workspace views. The dashboard data provider also has topology data sets if they are provided by a monitoring agent such as the IBM Tivoli Monitoring for VMware monitoring agent. Refer to the agent user guides to see a description of the agent's attribute groups and to determine if the agent provides a topology data set.

When you create or edit a page, you choose the widgets and their placement on the page. You edit each widget to choose a data set and select what information from the data set should be displayed in the widget. When editing a widget, provide the following information:

Procedure

1. Choose a data set.

   When you edit a widget, you choose a data set from the list of data providers configured in Dashboard Application Services Hub. You can either elect to see all data set names or enter search criteria for the data set name. If you list all of the data set names, you see the data sets for all agents whose application support is installed in the Tivoli Enterprise Portal Server as well as data sets available from other data providers. Since this can be a large list of data sets, you can filter the data sets by searching on a portion of the data set name. For example, to see all of the data sets (attribute groups) for the Linux OS agent, enter "`Linux`" in the search field.
2. Map the data set columns to the widget visualization attributes.

   Depending on the widget type, you might be asked to specify which column(s) from the data set you want displayed in the widget. For example, if you are editing an analog gauge widget to show disk utilization from a Linux OS agent, select the **Disk Used Percent** column for the gauge value.
3. Configure the widget visualization options.

   You can also configure the visualization options of the widget such as, labels, units of measure, and so forth.
4. Specify the data set configuration parameters.

   If you selected a data set that maps to an agent's attribute group, you must enter the name of the managed system or managed system group from which the data is retrieved. You can also specify a time filter value to retrieve historical data if the widget supports showing data over a time frame and you have configured historical data collection.

   The data set may have other configuration parameters to further filter what information is displayed in the widget. For example, if you are editing a widget such as a gauge that displays values from a single row in an agent attribute group, the data set configuration parameters allow you to specify other data set columns (attributes) that uniquely identify which row of data is displayed in the widget. For example, if you are displaying disk utilization for a Linux OS agent, in a gauge widget the Linux Disk data set configuration parameters allow you to specify which disk and mount point to show the utilization of.

   You can also specify how often the dashboard data provider provides refreshes of the data if the widget supports auto-refresh and you have not selected the events data set.

   Many of the Dashboard Application Services Hub widgets can support connections, or wires, between widgets so that they can exchange messages with each other. When an action occurs in a source widget, it creates an event, which contains information that can be sent to other widgets. The dashboard data provider only supports incoming connections when the source widget has the managed system name in either an ORGINNODE or NODE property.

   Dashboard Application Services Hub uses roles, for users or user groups, to control which users can create pages and work with widgets and which pages a user can display. However, the dashboard data provider performs the authorization of the monitoring resources that are displayed in a widget that uses one of the data provider's data sets. Custom dashboard pages use the authorization type that is configured for your monitoring dashboard environment, either authorization policies or Tivoli Enterprise Portal event permission and monitoring application assignments.

What to do next

For examples of how to create custom dashboard pages with monitoring data, see Creating custom monitoring dashboard pages in the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring).

Configuring an HTTP Server to load balance multiple Tivoli Enterprise Portal Servers

You can install and configure an HTTP Server to load balance multiple Tivoli Enterprise Portal Servers.

Before you begin

Use an IBM HTTP Server Version 8.5 or later.

Procedure

1. Complete the steps described in the “Setting up load balancing for a high availability dashboard environment” on page 53 roadmap that occur before the step for configuring the load balancing HTTP server.
2. Install and configure an HTTP server.

   The IBM HTTP Server can be installed using a Jazz for Service Management image that is provided with IBM Tivoli Monitoring. See the "Download instructions" in the IBM Tivoli Monitoring Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/welcome.htm) for information on the WebSphere Application Server Supplement images that contain the IBM HTTP Server. For instructions on installing the IBM HTTP Server Version 8.5, see IBM HTTP Server Version 8.5 Information Center (http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.ihs.doc/ihs/welcome_ihs.html). The IBM HTTP Server Information Center also provides information on how to download and install the IBM HTTP Server from an external IBM Installation Manager repository.

   Note: The HTTP Server Plug-in for IBM WebSphere Application Server must be installed into the HTTP server. The plug-in is provided with the IBM HTTP Server. However, if you are using a different HTTP Server, the plug-in is included on a Jazz for Service Management image that is provided with IBM Tivoli Monitoring. See the "Download instructions" in the IBM Tivoli Monitoring Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/welcome.htm) for information on the WebSphere Application Server Supplement images that contain the HTTP Server Plug-in for IBM WebSphere Application Server.
3. Assign a unique clone ID to each Tivoli Enterprise Portal Server. For more information, see “Setting Clone IDs on each Tivoli Enterprise Portal Server” on page 64.
4. Generate the `plugin-cfg.xml` file for each Tivoli Enterprise Portal Server and configure the HTTP server to use the plugin details. For more information, see “Generating the `plugin-cfg.xml` file” on page 64.

Setting Clone IDs on each Tivoli Enterprise Portal Server

Assign a unique clone ID to each Tivoli Enterprise Portal Server. The load balancing HTTP server uses the clone ID to ensure that requests for a session are always sent to the same portal server.

Assign the clone ID by following these steps for each portal server:

Procedure

1. Edit the `server.xml` file in the following location:

   Windows *install_dir*`\CNPSJ\profiles\ITMProfile\config\cells\ITMCell\nodes\ITMNode\servers\ITMServer`

   Linux UNIX *install_dir*`/`*interp*`/iw/profiles/ITMProfile/config/cells/ITMCell/nodes/ITMNode/servers/ITMServer`
2. In the `server.xml` file, locate the entry `<components xmi:type="applicationserver.webcontainer:WebContainer`.

   Within the `components` element, add the following entry:

   ```
   <properties xmi:id="WebContainer_1183077764084"
   name="HttpSessionCloneId" value="12345" required="false"/>
   ```

   Where `value` is set to the clone ID you want to assign for this portal server. This clone ID must be unique for each portal server.
3. Save and close the file.
4. Restart the Tivoli Enterprise Portal Server.

Generating the `plugin-cfg.xml` file

Generate the `plugin-cfg.xml` file for each Tivoli Enterprise Portal Server and then configure the load balancing HTTP server to use the plugin information. The `plugin-cfg.xml` file specifies the URIs that must be redirected and which servers to redirect requests to.

Procedure

1. On each Tivoli Enterprise Portal Server, run the following command:

   Windows *install_dir*`\CNPSJ\bin\GenPluginCfg.bat`

   Linux UNIX *install_dir*`/`*interp*`/iw/bin/GenPluginCfg.sh`

   This command generated a file called `plugin-cfg.xml` and saves it to the *install_dir*`\CNPSJ\profiles\ITMProfile\config\cells` directory on Windows and the *install_dir*`/`*interp*`/iw/profiles/ITMProfile/config/cells` directory on Linux and UNIX.
2. On the computer where the load balancing HTTP Server is installed, replace the existing `plugin-cfg.xml` with the `plugin-cfg.xml` file generated for one of the portal servers. You will copy a subset of the information from the `plugin-cfg.xml` files for the other portal servers into the `plugin-cfg.xml` file that is used by the HTTP server.
3. Customize the `plugin-cfg.xml` file for your load balancing HTTP Server environment and each of your Tivoli Enterprise Portal Server instances:
   a. On the computer where the IBM HTTP Server is installed, change to the directory where the `plugin-cfg.xml` is located (such as `cd c:\Program Files(x86)\IBM\WebSphere\Plugins_1\config\webserver1` on Windows or `/opt/IBM/WebSphere/Plugins/config/webserver1` on Linux and UNIX).
   b. Open the `plugin-cfg.xml` file in a text editor and edit the file to provide details for your IBM HTTP Server and all portal server instances. Use the

example provided at the end of this topic for reference. The example file includes the following variables which must be replaced with values specific to your environment:

- **PLUGIN_PATH** is the directory path where the WebSphere plugin is installed.

  On Linux and UNIX the default is `/opt/IBM/WebSphere/Plugins`

  On Windows the default is `C:\Program Files(x86)\IBM\WebSphere\Plugins_1`

  If your portal servers are installed on different operating system types than the HTTP server, you might need to change the directory path and also make sure you use the correct directory separator character for the system where the HTTP server is installed (for example, on Windows use \ and on Linux or UNIX use /).
- **HTTP_SERVER_PORT** is the port number for the HTTP server.
- **SERVER1_NAME** is a name that identifies the server definition for the first portal server. For example, ITMNODE_ITMSERVER1.
- **SERVER1** is the fully qualified hostname of the computer where the first portal server is installed and started.
- **SERVER2_NAME** is a name that identifies the server definition for the second portal server. For example, ITMNODE_ITMSERVER2.
- **SERVER2** is the fully qualified hostname of the computer where the second portal server is installed and started.
- **CLONE_ID** is the unique clone ID assigned to a particular portal server.

Note: The HTTP and HTTPS port values for all portal servers must be the same so check the `plugin-cfg.xml` file for each portal server to ensure they are using the same port numbers.

c. Add the following line to the end of the UriGroup section to handle requests for launching the Tivoli Enterprise Portal from Dashboard Application Services Hub:

```
<Uri Name="/*"/>
```

d. In the Log section, ensure the path for the `Name` attribute and `PluginInstallRoot` property match your load balancing HTTP server environment. The following values are typical:

   The typical value for `Name` attribute is `PLUGIN_PATH/logs/webserver1/http_plugin.log`.

   The typical value for the `PluginInstallRoot` property is `PLUGIN_PATH/`.

e. In the `ServerCluster` section, ensure the paths for the keyring and stashfile properties match your load balancing HTTP server environment. The following values are typical:

   The typical `keyring` property value is `PLUGIN_PATH/config/webserver1/plugin-key.kdb`.

   The `stashfile` properties is `PLUGIN_PATH/config/webserver1/plugin-key.sth`.

f. For each additional portal server:

   - Add a `Server` entry which specifies its clone ID, name, and fully qualified hostname. You can obtain the clone ID from the portal server's `plugin-cfg.xml` file. Assign each portal server a unique server name.

     Ensure you specify ports 15200 (for HTTP) and 15201 (for HTTPS) instead of ports 15210 and 15211. Ports 15200 and 15201 are the ports for the local

HTTP server for each portal server, and these ports must be used to support launching the portal client from Dashboard Application Services Hub.

Follow the same pattern shown in the example at the end of this topic.

- Specify the server name in the PrimaryServers section.

g. Add the CloneID and LoadBalanceWeight attributes for every Server entry.

Important: For more information on web server plug-in workload management policies and to help you determine the appropriate values for the elements LoadBalance and LoadBalanceWeight, refer to the following articles:

- http://www.redbooks.ibm.com/abstracts/TIPS0235.html
- http://www-01.ibm.com/support/docview.wss?rs=180 &uid=swg21219567

Example

The following plugin-cfg.xml example for Windows includes values in bold that must be customized for your load balancing HTTP server environment.

```
<?xml version="1.0" encoding="ISO-8859-1"?><!--HTTP server plugin config file for
the cell ITMCell generated on 2013.07.10 at 06:59:43 AM CDT-->
<Config ASDisableNagle="false" IISDisableNagle="false"
        IgnoreDNSFailures="false" RefreshInterval="60"
        ResponseChunkSize="64" AcceptAllContent="false"
        IISPluginPriority="High"  FIPSEnable="false"
        AppServerPortPreference="HostHeader" VHostMatchingCompat="false"
        ChunkedResponse="false">
  <Log LogLevel="Trace" Name="PLUGIN_PATH\logs\webserver1\
    http_plugin.log"/>
    <Property Name="ESIEnable" Value="true"/>
    <Property Name="ESIMaxCacheSize" Value="1024"/>
    <Property Name="ESIInvalidationMonitor" Value="false"/>
    <Property Name="PluginInstallRoot"  Value="PLUGIN_PATH\"/>
   <VirtualHostGroup Name="default_host">
      <VirtualHost Name="*:9080"/>
      <VirtualHost Name="*:80"/>
      <VirtualHost Name="*:9443"/>
      <VirtualHost Name="*:5060"/>
      <VirtualHost Name="*:5061"/>
      <VirtualHost Name="*:443"/>
      <VirtualHost Name="*:15200"/>
      <VirtualHost Name="*:15201"/>
      <VirtualHost Name="*:HTTP_SERVER_PORT"/>
   </VirtualHostGroup>
  <ServerCluster Name="ITMServer_ITMNode_Cluster" CloneSeparatorChange="false"
   LoadBalance="Round Robin" PostBufferSize="64" IgnoreAffinityRequests="true"
    PostSizeLimit="-1" RemoveSpecialHeaders="true" RetryInterval="60">
      <Server CloneID="CLONE_ID" ConnectTimeout="0" ExtendedHandshake="false"
     MaxConnections="-1" Name="SERVER1_NAME" ServerIOTimeout="0"
     WaitForContinue="false">
         <Transport Hostname="SERVER1" Port="15200" Protocol="http"/>
         <Transport Hostname="SERVER1" Port="15201" Protocol="https">
            <Property Name="keyring" Value="PLUGIN_PATH\config\
         webserver1\plugin-key.kdb"/>
            <Property Name="stashfile" Value="PLUGIN_PATH\config\
         webserver1\plugin-key.sth"/>
         </Transport>
      </Server>
     <Server CloneID="CLONE_ID" ConnectTimeout="0" ExtendedHandshake="false"
    MaxConnections="-1" Name="SERVER2_NAME" ServerIOTimeout="0"
    WaitForContinue="false">
         <Transport Hostname="SERVER2" Port="15200" Protocol="http"/>
         <Transport Hostname="SERVER2" Port="15201" Protocol="https">
            <Property Name="keyring" Value="PLUGIN_PATH\config\
         webserver1\plugin-key.kdb"/>
            <Property Name="stashfile" Value="PLUGIN_PATH\config\
         webserver1\plugin-key.sth"/>
         </Transport>
      </Server>
      <PrimaryServers>
        <Server Name="SERVER1_NAME/>
        <Server Name="SERVER2_NAME"/>
      </PrimaryServers>
   </ServerCluster>
   <UriGroup Name="default_host_ITMServer_ITMNode_Cluster_URIs">
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/ivt/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/IBM_WS_SYS_RESPONSESERVLET/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsp"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsv"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsw"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/IBM_WS_SYS_RESPONSESERVLET/j_security_check"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/IBM_WS_SYS_RESPONSESERVLET/ibm_security_logout"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/RedirectServletWeb/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/LICServletWeb/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/TEPSREST/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/ITMRESTProvider/*"/>
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
     Name="/ibm/tivoli/*"/>
      <Uri Name="/*"/>
   </UriGroup>
   <Route ServerCluster="ITMServer_ITMNode_Cluster"
   UriGroup="default_host_ITMServer_ITMNode_Cluster_URIs"
   VirtualHostGroup="default_host"/>
   <RequestMetrics armEnabled="false" newBehavior="false"
        rmEnabled="false" traceLevel="HOPS">
        <filters enable="false" type="URI">
            <filterValues enable="false" value="/servlet/snoop"/>
            <filterValues enable="false" value="/webapp/examples/HitCount"/>
        </filters>
        <filters enable="false" type="SOURCE_IP">
            <filterValues enable="false" value="255.255.255.255"/>
            <filterValues enable="false" value="254.254.254.254"/>
        </filters>
      </RequestMetrics>
</Config>
```

Controlling UISolutions imports

When a dashboard application such as IBM Infrastructure Management Dashboards for Servers is launched, the dashboard application sends a request to the dashboard data provider to import the application's UISolutions. UISolutions in the dashboard data provider define the characteristics of what data the dashboards can display. The import occurs once per dashboard application and it happens automatically. When a dashboard application is updated and contains updated UISolutions, the import also happens automatically.

You can disable the importing of UISolutions all together after you have installed and launched all of your dashboard applications, or you can control which users can import UISolutions by configuring the portal server.

About this task

If you disable UISolutions altogether, you must re-enable UISolutions importing when you either install a new dashboard application or update an existing dashboard applications in IBM Dashboard Application Services Hub. After the dashboard application install or update is complete, the dashboard application has been launched, and you have verified that data can be displayed, then you can disable importing of the UISolutions again.

If you want to control which user can import UISolutions then be aware that new or updated dashboard applications with UISolutions will not be usable until the user specified in the portal server's environment file launches the dashboard application.

The `KD8_VM_IMPORT_ID` variable is optional and not set by default, which means any dashboard user that has been authenticated can trigger the request to import UISolutions when they launch the dashboard application.

Procedure

1. Open the Tivoli Enterprise Portal Server environment variable file.

   Windows *`install_dir`*`\CNPS\kfwenv`

   Linux UNIX *`install_dir`*`/config/cq.ini`
2. Set the `KD8_VM_IMPORT_ID` environment variable.

   To control which user is allowed to do the import, set this variable to a particular ID such as `KD8_VM_IMPORT_ID=user1`.

   To disable all users from doing an import of UISolutions, set `KD8_VM_IMPORT_ID=$nouser@` or any name that you know does not match a Dashboard user ID.
3. Restart the Tivoli Enterprise Portal Server to implement the changes.

Chapter 4. Editing your environment configuration settings

The Tivoli Enterprise Portal client has several dozen parameters that you can set to affect behavior and performance at user computers. As well, the Tivoli Enterprise Portal Server has an environment file that you can edit to adjust or add variables to affect all portal clients connected to it and its interaction with the hub monitoring server. You can also control environment variables at the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Monitoring Automation Server.

The following topics include information that pertains to the environment variables referenced within the *Administrator's Guide*. For a complete list of environment variables, see "Environment variables" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Tivoli Enterprise Portal client configuration settings

The Tivoli Enterprise Portal client has parameters that affect its performance, such as the maximum size of files attached to event acknowledgements and for how long to keep the common event list in the cache.

Editing the client parameters

Changes you make to the browser client are applied globally because they are downloaded automatically through the HTTP server that is installed with the portal server. If users are deploying the desktop client themselves through Java WebStart, the changes will also be applied globally. Otherwise, desktop client changes must be made on each computer where it is installed if you want the change to affect all users.

About this task

Complete these steps to adjust the client parameters:

Procedure

1. Start Manage Tivoli Enterprise Monitoring Services. For the browser client and Java WebStart, this is the computer where the portal server is installed; otherwise, it is where the desktop client is installed.
   - Windows Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.
   - Linux Change to the *install_dir*`/bin` directory and enter: `./itmcmd manage`.
2. Right-click **Tivoli Enterprise Portal – Desktop** or **Tivoli Enterprise Portal – Browser**, and click **Reconfigure**. The Configure Application Instance window is displayed for the desktop client (also used for Java WebStart); the Configure Tivoli Enterprise Portal Browser window is displayed for the browser client.
3. Double-click the parameter value you want to change.
4. To activate the parameter, type a value and select **In Use** in the Edit Tivoli Enterprise Portal Parm window.

5. After you are finished editing the parameters, click **OK** to save your changes. Your changes will take effect the next time users log on to the portal server. Users already logged on will see no change until they exit, and log on again.

Related reference:

"Portal client parameter list"
Most of the Tivoli Enterprise Portal client parameters are left unchanged from their default values. Edit the client parameters to effect a specific behavior.

Portal client parameter list

Most of the Tivoli Enterprise Portal client parameters are left unchanged from their default values. Edit the client parameters to effect a specific behavior.

Some parameters pertain to the desktop client only, to the desktop client and Java WebStart client only, or to the browser client only and are noted as such.

browser.cache.memory.capacity

Indicates the maximum amount of memory in KB to be used to cache decoded images and other features by Browser views (a positive non-zero integer). Specify a value of 0 to disable memory caching. Default: **-1**, whereby the capacity value is automatically decided based on the total amount of memory.

Physical memory	Memory cache in KB
32 MB	2048
64 MB	4096
128 MB	6144
256 MB	10240
512 MB	14336
1 GB	18432
2 GB	24576
4 GB	30720
8 GB and beyond	32768

cnp.agentdeploy.timeout

This is the time that should pass before the agent deploy request times out. Default: **1800** seconds (30 minutes).

cnp.attachment.segment.maxsize

For transmission across the network, file attachments are broken into segments then reassembled at the Tivoli Enterprise Portal Server. For example, an 8 MB file is transmitted in eight segments of 1 MB. Adjust this parameter for the segment size that best suits your environment. Enter the maximum size in bytes, such as 250000 for 250 KB. Default: **1000000** (1 MB).

This parameter is also available as a portal server environment variable. See "Controlling the size of event attachments" on page 82.

cnp.attachment.total.maxsize

Use this parameter to set the maximum size of each file attached to an acknowledgement. Enter the maximum size in bytes, such as 2500000 for 2.5 MB. Default: **10000000** (10 MB).

This parameter is also available as a portal server environment variable. See “Controlling the size of event attachments” on page 82.

cnp.authentication.skip_dns

Value: “N”. This determines whether the server certificate validation tries to resolve and match the host DNS name.

cnp.browser.installdir

The WebRenderer Java browser component is used for browser view functionality in the portal client. The first time a user creates a browser view, a subdirectory is created automatically on the user's computer.

Windows *%HOMEPATH%*\wr*WebRendererVersion*\.webrendererswing. Example: C:\Documents and Settings\Administrator\wr4.2.14\.webrendererswing

Linux %HOME/wr*WebRendererVersion*/.webrendererswing

This subdirectory is where the browser jar files are extracted to and where certificates and other WebRenderer artifacts are created for browser views. Use this parameter to specify a different path for the browser view files to be saved on user computers. A different path is required if users will be running multiple instances of the portal client and possibly logging on to different versions of the portal server.

cnp.commonevent.cache.timeout

Number of minutes to retain the cache for the common event console when the user has switched to a workspace that does not contain the common event console view (which means the cache is not being used). If this time period ends before the cache is used again, the cache is cleared. The cache is then rebuilt when it is needed by a common event console view.

A value of -1 means always retain the cache, even when it is not being used. A value of 0 means immediately clear the cache when the user has switched to a workspace that does not contain the common event console view. Default: **30**.

cnp.databus.pageSize

In the Tivoli Enterprise Portal user interface, the Properties editor has a field for adjusting the page size for individual query-based views. This parameter sets the number of rows to fetch in single logical page for all query-based views. Default: **100** rows. Although there is no limit to what you can set here, the larger the page size, the more memory required at the portal client and server.

You might, for example, want to set a larger page size for the searching in the table view over a larger number of rows. Or you might want fewer pages to scroll through when interacting with views that retrieve a large number of rows (or instances). You must make sure, however, that you have sufficient resources on the portal client and server to handle the additional data being packaged, transported, and ultimately rendered as a result of increasing the page size value. Probably the best way to find the right number here is to increase it gradually (such as increments of 100) until response time across a good sampling of workspaces begins to suffer. At that point, you might want to reduce the number by the last increment (such as 100 rows fewer) as that will be close to the optimal value for the environment.

Another setting that affects query-based view response time is KFW_REPORT_NODE_LIMIT, which is a portal server environment variable.

cnp.drag.sensitivity
Number of pixels the mouse must move before drag operation begins. Default: **7**.

cnp.encoding.codeset
String encoding code set identifier.

cnp.eventcon.autoresume.delay
The number of seconds to wait before automatically resuming updates to the Situation Event Console and the Common Event Console after they have been paused due to scrolling. Default: **60** seconds.

cnp.heartbeat.interval
Heartbeat ping interval between the Tivoli Enterprise Portal client and server. An increase in the interval means that the client will take longer to detect when the portal server is offline. A shorter interval means the client will be notified sooner but it also increases the traffic between client and server. Default: **30** seconds.

cnp.history.depth
Number of workspaces to maintain in the back / forward history navigation stack. Default: **20**.

cnp.http.proxy.password
Password used for proxy authentication using Browser view.

cnp.http.proxy.user
Userid used for proxy authentication using Browser view.

cnp.http.url.host
Desktop client and Java WebStart client only: URL host for IOR fetch.

cnp.http.url.path
Desktop client and Java WebStart client only: URL path for IOR fetch.

cnp.http.url.port
Desktop client and Java WebStart client only: URL port for IOR fetch.

cnp.http.url.protocol
Desktop client and Java WebStart client only: URL protocol for IOR fetch.

cnp.http.url.DataBus
Desktop client and Java WebStart client only: The URL for the `cnps.ior` file, which is required for the portal server to locate the graphic view images and style sheets. The default setting, which does not show, assumes the integral HTTP server. If it has been disabled for some reason, you must enter the URL for the integral HTTP server. See the *IBM Tivoli Monitoring Troubleshooting Guide* for details. When this parameter is set, it overrides the settings of the other cnp.http.url parameters for protocol, port, and path.

cnp.pipeline.factor
Databus to Server Pipeline monitoring factor (in Heartbeat cycles). Default: **2**.

cnp.playsound.interval
Number of seconds before the same sound file can be played again. If events open frequently, this setting provides sound pause. Default: **10** seconds.

cnp.publishurl.delay
Browser client only: When you make a workspace switch, allows the user interface rendering to complete before the browser initializes the new applet and destroys the old applet. Default: **1** second.

Important: Modify this parameter only after consulting IBM Software Support.

cnp.systemtray.offset
Tivoli Enterprise Portal factors in the Windows task bar at the bottom of the screen when sizing menus and windows for display. Default: **true**.

cnp.terminal.cache.entries
Maximum number of active terminal emulator sessions. Default: **50**.

cnp.terminal.host
Default terminal emulator host name.

cnp.terminal.port
Default terminal emulator port number. Default: **23**.

cnp.terminal.script.entries
Maximum number of user terminal emulator scripts that can be saved. Default: **256**.

cnp.terminal.type
Default terminal emulator type. When specifying a terminal type, enclose the terminal type within double quotes and enter one of these supported names:

IBM 3270 (24x80)

IBM 3270 (32x80)

IBM 3270 (43x80)

IBM 3270 (27x132)

IBM 5250 (24x80)

VT100 (24x80)

cnp.view.change_remove.warning
Warning message when the user is about to change or remove a view.

Default: **True**. The message is displayed. Change the setting to False to stop the message from being displayed.

cnp.workspace.switch.rate
The minimum amount of time that must pass before the workspace can be replaced by the next one selected. Default: **1000** (1 second).

cnp.workspacerender.delay
Browser mode only: Workspace post render delay in milliseconds.

http:agent
Defines the name of the integral HTTP server. If it or its proxy requires a different browser identity before it enables the browser view to access the Internet, you can enter a one-word name for the browser. It can be any name so long as it is not rejected by the proxy server. You normally do not need to add an http name definition unless users get an error when they attempt to access the Internet through a workspace browser view.

http.nonproxyhosts
When ☑ **Enable HTTP Proxy Server Requests** is selected, the servers in

this list bypass the proxy. Separate each server name with a vertical line (|). See “Enabling the HTTP proxy server” on page 76.

http.proxyHost
Browser client: Used to specify the host name or IP address of the http proxy server if one is used.

http.proxyPort
Browser client: Used with the http.proxyHost parameter to specify the listening port number for the HTTP proxy server. Port **80** is the default for third-party HTTP servers.

kjr.browser.default
This is the path and name of the browser application to use when launching contextual help. To open the help with a specific browser or one other than the default, enter the path and the application name, such as `C:\Program Files\Mozilla Firefox\firefox.exe`.

kjr.trace.file
File name of RAS1 trace log if trace mode is LOCAL.

kjr.trace.mode
The RAS1 tracing option. Default: LOCAL.

kjr.trace.params
RAS1 trace options. Default: ERROR.

kjr.trace.qdepth
Sets the tracing thread queue depth to **15000** by default.

kjr.trace.thread
Determines whether trace calls are threaded. Default: **true**.

sun.java2d.noddraw
When the Tivoli Enterprise Portal is run as a client image in an emulation environment that does not support the DirectDraw screen-writing function, turn off the function by setting this variable to true in both the browser and desktop clients. Otherwise, users encounter conditions of high CPU usage because the Java process attempts to write to the screen. Default: **true**.

user.language
Desktop client and Java Web Start client only: Specifies the language code of the user's locale preference (such as cs, de, en, es, fr, hu, it, ja, ko, pl, pt, ru, th, and zh). You can create another instance of the desktop client and change this variable (and user.region) to another locale. In this way, you can have two or more instances of the desktop client running on the same computer, each in a different language. If you specify an unsupported locale, the failover is to en_US.

Browser client only: On the client computer, enter the text below directly into the Java plug-in runtime parameters used by the browser, where *xx* is the language and *XX* is the locale.

```
-Duser.language=xx -Duser.region=XX
```

Note: The portal client uses cascading style sheets to render the application text. If no localized version of a style sheet, such as `ws_press.css`, is available, the English version will be used.
To edit the runtime parameters complete the following tasks:

1. Open the Java control panel:

Windows Launch the **IBM Control Panel for Java** or the **Java** Control Panel.

Linux Find the Java **ControlPanel** executable under your *jre_install_dir* and launch it. For example: `/opt/IBM/ibm-java2-i386-70/jre/bin/ControlPanel`.

2. Click the **Java** tab.
3. In the Java Applet Run time Settings area, click **View**.
4. If you have multiple Java versions, verify that you have the correct control panel open by reading the **Location** column to confirm the Java Run time and that the JRE is in the correct path. For example: `C:\Program Files\IBM\Java70\jre\bin` for IBM Java on Windows.
5. Edit the parameter you want to change.
6. Save your changes.

user.region

Specifies country code of user's locale preference (such as BR, CN, CZ, DE, ES, FR, HU, IT, JP, KR, PL, RU, TH, TW, and US). See also the description for **user.language**.

Use the following table to find language codes and locale codes:

Table 7. Language and locale codes

Language	Language code (xx)	Locale code (XX)
Chinese, Simplified	zh	CN
Chinese, Traditional	zh	TW
Czech	cs	CZ
English	en	US
French	fr	FR
German	de	DE
Hungarian	hu	HU
Italian	it	IT
Japanese	ja	JP
Korean	ko	KR
Polish	pl	PL
Portuguese, Brazilian	pt	BR
Russian	ru	RU
Spanish	es	ES
Thai	th	TH

Related tasks:

"Editing the client parameters" on page 69
Changes you make to the browser client are applied globally because they are downloaded automatically through the HTTP server that is installed with the portal server. If users are deploying the desktop client themselves through Java WebStart, the changes will also be applied globally. Otherwise, desktop client changes must be made on each computer where it is installed if you want the change to affect all users.

"Controlling the size of event attachments" on page 82
By default, the maximum size of each file attached to an event acknowledgement is 10 MB, and 1 MB for the size of information segments sent across the network. Environment variables are provided that enable you to change the maximum at the Tivoli Enterprise Portal or at the Tivoli Enterprise Portal Server. The event attachment settings that are changed at the desktop client override those for the portal server.

"Starting the browser client on another portal server" on page 24
Start a separate instance of your browser and log on to the Tivoli Enterprise Portal Server of a different managed network to see two managed networks from the same computer.

Related reference:

"Portal server environment variables" on page 80
The environment configuration file for the Tivoli Enterprise Portal Server can be edited to add certain environment settings and to change the values of others.

Enabling the HTTP proxy server

Environments that use an HTTP proxy server require additional client configuration to enable URL access from the browser view in a Tivoli Enterprise Portal workspace.

About this task

To enable the HTTP proxy server, complete these steps on every computer where the Tivoli Enterprise Portal client is used that also uses an HTTP proxy for the browser view:

Procedure

1. Open a workspace that contains a browser view or add a browser view to the current workspace.
2. In the browser view's address box, type: `about:config`
3. In the filter field that appears at the top of the page, enter the following to see the network proxy fields: `network.proxy`
4. Out of the reduced set shown, the following three entries are of interest. Double-click an entry or select it and press Enter to modify its values:

   network.proxy.http
   Enter the DNS identifier or the IP address of the proxy host to use for the HTTP protocol.

   network.proxy.http_port
   Enter `80`, the default port number, or a different number used by the proxy host.

   network.proxy.no_proxies_on
   Append any fully qualified host names or IP addresses that should be accessed without the proxy. For example, this setting bypasses the

proxy server for any files on your local system and on the portal server (myteps.uk.ibm.com) that are accessed from the browser view: `localhost,127.0.0.1, myteps.uk.ibm.com`.

Results

After you click **OK** on the property edit panel, the change is saved on the Tivoli Enterprise Portal client.

Setting application properties for Linux and UNIX systems

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.

About this task

You might have to update one or more of the following files:

Note: All file paths are relative to your *install_dir* directory where you installed IBM Tivoli Monitoring.

Table 8. File locations for changing application properties for UNIX and Linux systems

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 only change 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.

You can also set instance name, browser, and Tivoli Enterprise Portal Server properties on Linux. Refer to the *IBM Tivoli Monitoring Command Reference* for details.

To change the location of the web browser you must change the above file or files to include a new property by completing the following procedure:

Procedure

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
   ```

Setting the environment variable when the hub is on a z/OS system

On z/OS, GSKit is known as the Integrated Cryptographic Service Facility, or ICSF. The Tivoli Enterprise Monitoring Server supports secure password encryption through ICSF, which provides a robust encryption and decryption scheme for stored passwords and is the preferred method of password encryption.

About this task

If the hub Tivoli Enterprise Monitoring Server is on a z/OS system that does not have ICSF installed, an alternative, less secure encryption scheme is used. The hub monitoring server and the portal server both must be using the same scheme. Therefore, if the hub system does not use ICSF, you must configure the Tivoli Enterprise Portal to use the less secure scheme (EGG1) as well. This involves editing the Tivoli Enterprise Portal Server environment file to add a new line.

To add the new line to the environment file, complete the following steps:

Procedure

- Windows
  1. On the system where the Tivoli Enterprise Portal Server is installed, select **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.
  2. Right-click Tivoli Enterprise Portal Server, point to **Advanced** and select **Edit ENV File** from the list.
  3. If the Tivoli Enterprise Portal Server message displays, click **OK** to close it.
  4. Add a new line: `USE_EGG1_FLAG=1`.
  5. Click **Save**.
  6. Click **Yes** to implement your changes and recycle the service.
- Linux UNIX

1. Change directory (cd) to *install_dir*/`config`
2. Add the following line to the cq.ini file: `USE_EGG1_FLAG=1`
3. Save the file.
4. Recycle the portal server.

What to do next

See *Configuring the Tivoli Enterprise Monitoring Server on z/OS* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/ztemsconfig/ztemsconfig.htm).

Tivoli Enterprise Portal Server configuration settings

The Tivoli Enterprise Portal Server runs a process called KfwServices, which has a set of environment variables that can be edited and enabled for certain configuration requirements. This can be done through the Manage Tivoli Enterprise Monitoring Services application or at the command line using `itmcmd manage`.

For example, when you have security enabled, you can control the number of log in attempts before a user is locked out of the portal.

If you want to set the application properties for advanced configuration functions on UNIX or Linux, such as the location of the web browser that the Tivoli Enterprise Portal browser client launches, this has to be done manually.

If the portal server connects to a hub monitoring server that is on a z/OS system that does not have the Integrated Cryptographic Service Facility (ICSF) installed, you must edit the environment file to add a new line.

Note: Any customizations made within the TEPS/e administration console, such as to configure TLS/SSL communications to the LDAP server, are overwritten and cleared any time the portal server is reconfigured unless you chose the LDAP type of **Other** during portal server configuration through Manage Tivoli Monitoring Services. To prevent this from occurring, choose the LDAP type of **Other** during portal server configuration. When **Other** is chosen, the LDAP user registry information is handled by TEPS/e and is not affected by Tivoli Management Services directly. See step 5 on page 109.

Editing the portal server environment file

Edit the Tivoli Enterprise Portal Server environment file, KFWENV, to reconfigure the portal server parameters.

About this task

Take these steps to edit the portal server environment file:

Procedure

1. Open the environment file on the computer where the portal server is installed:
   - Windows From Manage Tivoli Monitoring Services (**Start → Programs→ IBM Tivoli Monitoring→ Manage Tivoli Monitoring Services**), right-click **Tivoli Enterprise Portal Server** and click **Advanced → Edit ENV File** to open the `kfwenv` file.
   - Linux UNIX Change to the *install_dir*/`config` directory and open the `cq.ini` file in a text editor.

2. Edit the file to enable (delete # at the beginning of the line), disable (type # at the beginning of the line) or modify any of the environment variables.
3. Save `kfwenv` (Windows) or `cq.ini` (Linux and operating systems such as UNIX) and exit the text editor.
4. Click **Yes** when a message asks if you want to recycle the service. Click **No** if you prefer to have the changes you made take effect later by manually recycling the portal server.

Related reference:

"Portal server environment variables"
The environment configuration file for the Tivoli Enterprise Portal Server can be edited to add certain environment settings and to change the values of others.

Portal server environment variables

The environment configuration file for the Tivoli Enterprise Portal Server can be edited to add certain environment settings and to change the values of others.

The file shows a number of environment variables that have been enabled and others that are disabled by default or as a result of the way you configured the portal server. Other variables in this list must be added manually to enable them.

`KFW_AUTHORIZATION_MAX_INVALID_LOGIN=0`

You can control the number of attempts a user can make to log on to the Tivoli Enterprise Portal Server by setting this environment variable to a value from 0 to 15. The default value, 0, indicates that there is no limit to the number of failed attempts a user can make before being locked out.

This configuration setting is effective only when you have enabled security through the hub Tivoli Enterprise Monitoring Server as described in the topic, "Controlling the number of logon attempts" on page 83.

`KFW_CMW_DETECT_AGENT_ADDR_CHANGE=N`

The Navigator function detects when the IP address for an agent is discovered. If the agent environment is constantly changing or has improper configurations that generate excessive Navigator tree rebuilding, consider adding this environment variable to have any discovery of changes or additions of IP address ignored.

`KFW_CMW_DETECT_AGENT_HOSTNAME_CHANGE=N`

This variable is like the one for detect agent address change except that it prevents the Navigator rebuilding if an agent hostname is changed.

`KFW_CMW_DETECT_AGENT_PROPERTY_CHANGE=N`

This is like the detect agent address change except that it prevents the Navigator rebuilding if an agent affinity or affinity version changes.

`KFW_CMW_SITUATION_ADMIN_SUPPRESS=N`

When a situation is stopped, no message is sent to the situation event console. If you prefer to have the message written to the situation event console for each system the situation was distributed to, enable (remove the # at the beginning of the line) this environment variable. The `Stopped` message alerts the user that the situation has been stopped, thus, its state is unknown.

`KFW_CMW_SPECIAL_HUB_ENTERPRISE=Y`

Associates situations to the Navigator Physical view root item, **Enterprise**. The default value is **Y** to allow association of Managed System Online and Offline situations to the Enterprise Navigator item. A setting of **N** disables the automatic assignment of the *HUB managed system group to the Enterprise Navigator item.

`KFW_ECLIPSE_HELP_SERVER_PORT=9999`
The default port number for the Eclipse help server is 9999. If 9999 is already used by another device, add this variable and specify a port number from 1 to 65535. This value will be passed as a property from the portal server to the client at logon time.

`KFW_FIPS_ENFORCED=N`
The monitoring server and agent components of the Tivoli Management Services are already FIPS compliant. This variable specifies whether the encryption methods used by the portal server should comply with the Federal Information Processing Standard (FIPS) 140–2 specification. If your environment must conform to the FIPS 140–2 standard, specify `Y`.

`KFW_REPORT_NODE_LIMIT=200`
When a workspace that contains a query-based view is opened or refreshed, the view's query requests data from the managed systems that are assigned to that Navigator item (unless you have edited the view's query definition to assign specific managed systems or managed system groups). The number of managed systems from which a query can retrieve data can be up to 200. This limitation is provided to keep traffic and resource usage of your managed environment at an acceptable level. You can adjust the maximum number with this variable but keep in mind that if you increase the maximum number of managed systems being queried, the longer it can take to render the view.

Consider creating filtered queries, managed system groups, or custom Navigator views with managed systems assignments on Navigator items that limit the number of managed systems to retrieve data from. These features are described in the Tivoli Enterprise Portal online help and user's guide.

Another setting that affects query-based view response time is the `cnp.databus.pageSize` client parameter.

`KFW_REPORT_TERM_BREAK_POINT=86400`
Adjust this setting to change the point, in seconds, where a historical request selects from short-term or long-term history data. The default is for short-term history data to be collected from *now* to 24 hours ago, and long-term from 24 hours onward. Set to **`0`** to select only from long-term history data.

Related tasks:

"Editing the portal server environment file" on page 79
Edit the Tivoli Enterprise Portal Server environment file, KFWENV, to reconfigure the portal server parameters.

Related reference:

"Portal client parameter list" on page 70
Most of the Tivoli Enterprise Portal client parameters are left unchanged from their default values. Edit the client parameters to effect a specific behavior.

Pruning events on the portal server database

Event information is stored in the KFW tables in the Tivoli Enterprise Portal Server (TEPS) database. Because this information can grow in the amount of space it consumes, it is automatically pruned.

About this task

By default, closed events are removed from the TEPS database one day after they are closed, within the hours of 12:00 AM and 4:00 AM on the local portal server. You can control the pruning of this data by changing the following environment variables in the Tivoli Enterprise Portal Server configuration file.

Procedure

1. Open the Tivoli Enterprise Portal Server environment file for editing:
   - Windows In Manage Tivoli Monitoring Services, right-click **Tivoli Enterprise Portal Server** and click **Advanced** → **Edit ENV File**.
   - Linux UNIX Change to the *install_dir*/`config` directory and open **cq.ini** in a text editor.
2. Locate and edit the TEPS database event pruning parameters as needed:
   - `KFW_EVENT_RETENTION=0` is the number of days to keep a closed event. For example, to prune an event 2 days after it is closed, specify `2`.
   - `KFW_PRUNE_START=00:00` is the time of day to start pruning data, in 24-hour notation. For example, to begin pruning data at 11:00 PM, specify `23:00`.
   - `KFW_PRUNE_END=04:00` is the time of day to stop pruning data, specified in 24-hour notation. For example, to end pruning data at 1:00 AM, specify `01:00`.
3. Save and close the environment file.
4. Click **Yes** when a message asks if you want to recycle the service; or click **No** if you prefer to have your changes take effect later by recycling the portal server.

Controlling the size of event attachments

By default, the maximum size of each file attached to an event acknowledgement is 10 MB, and 1 MB for the size of information segments sent across the network. Environment variables are provided that enable you to change the maximum at the Tivoli Enterprise Portal or at the Tivoli Enterprise Portal Server. The event attachment settings that are changed at the desktop client override those for the portal server.

About this task

Complete the steps for editing the environment settings of the Tivoli Enterprise Portal or of the Tivoli Enterprise Portal Server.

Procedure

- Edit the Tivoli Enterprise Portal environment file:
  1. Start Manage Tivoli Enterprise Monitoring Services:
     Windows Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.
     Linux Change to the *install_dir*/`bin` directory and enter: `./itmcmd manage`.
  2. Right-click **Tivoli Enterprise Portal – Desktop** or **Tivoli Enterprise Portal – Browser**, and click **Reconfigure**. The Configure Application Instance window is displayed for the desktop client (also used for Java WebStart); the Configure Tivoli Enterprise Portal Browser window is displayed for the browser client.

3. Double-click **cnp.attachment.total.maxsize** and enter the maximum size in bytes for individual files that get attached to an event acknowledgemen (such as 2500000 for 2.5 MB), and select ☑ **In Use**.
4. If you want to change the segment size (the default 1000000 is 1 MB, thus a 5 MB attachment would be transmitted in 5 x 1 MB segments), double-click **cnp.attachment.segment.maxsize** and enter a new segment size in bytes, and select ☑ **In Use**.
5. Click **OK** to save your changes. Your changes will take effect the next time a user logs on to the portal server. Users already logged on will see no change until they exit, then log on again.

- Edit the Tivoli Enterprise Portal Server environment file:
  1. Open the Tivoli Enterprise Portal Server environment file for editing:
     Windows In Manage Tivoli Enterprise Monitoring Services, right-click **Tivoli Enterprise Portal Server** and click **Advanced** → **Edit ENV File** .
     Linux UNIX Change to the *install_dir*/config directory and open **cq.ini** in a text editor.
  2. Delete the # pound symbol at the beginning of the two KFW_ATTACHMENT lines and edit the settings as needed.
     KFW_ATTACHMENT_MAX=10000000 is 10 MB. Specify the new maximum file attachment size.
     KFW_ATTACHMENT_SEGMENT_MAX=1000000 is 1 MB. Specify the new maximum size for file segments that the attachment file is broken into for transmission.
  3. Save and close the environment file.
  4. Click **Yes** when a message asks if you want to recycle the service; or click **No** if you prefer to have your changes take effect later by recycling the portal server.

Related reference:

"Portal client parameter list" on page 70
Most of the Tivoli Enterprise Portal client parameters are left unchanged from their default values. Edit the client parameters to effect a specific behavior.

Controlling the number of logon attempts

You can specify the number of attempts a user can make to log into the Tivoli Enterprise Portal by setting the KFW_AUTHORIZATION_MAX_INVALID_LOGIN environment variable.

About this task

See the procedures in **What to next** at the end of this topic to disable a user from accessing the portal, regardless of the KFW_AUTHORIZATION_MAX_INVALID_LOGIN setting. Complete these steps to control the number of logon attempts to the portal server:

Procedure

1. Open the Tivoli Enterprise Portal Server environment file for editing:
   - Windows In Manage Tivoli Monitoring Services, right-click **Tivoli Enterprise Portal Server** and click **Advanced** → **Edit ENV File** .
   - Linux UNIX Change to the *install_dir*/config directory and open **cq.ini** in a text editor.

2. Locate `KFW_AUTHORIZATION_MAX_INVALID_LOGIN=0` and specify a value between 0 and 15. The default value of **0** indicates that there is no limit to the number of failed attempts a user can make before they are locked out.
3. Save and close the environment file.
4. Click **Yes** when a message asks if you want to recycle the service; or click **No** if you prefer to have your changes take effect later by recycling the portal server.

Results

The next time a user attempts to log on to the portal server, the number of logon attempts will be restricted by the value you set KFW_AUTHORIZATION_MAX_INVALID_LOGIN to in the environment file.

What to do next

☑ **Security: Validate User**

The invalid login setting is effective only when you have enabled security through the hub monitoring server.

Linux UNIX You must also enable the **Login Lockout** feature by turning on the validation setting in the monitoring server configuration file: `KDS_VALIDATE_EXT="Y"`.

The monitoring server configuration files are named *`hostname`*`_ms_`*`address`*`.config` and `ms.ini`, and are located in the *`install_dir`*`/config/` directory.

Restoring user access

If a user is locked out, you have two options to restore their access to the Tivoli Enterprise Portal:

- In the Tivoli Enterprise Portal , click **Administer Users** and select the user ID. In the Permissions tab, click **User Administration** and enable ☑ **Logon Permitted**.
- On the computer where the Tivoli Enterprise Portal Server is installed, run this command line utility to enable or disable access:

  Windows Change directory to *`install_dir`*`\cnps\` and enter

  ```
  KfwAuthorizationAccountClient.exe ENABLE|DISABLE
    user_id
  ```

  For example, `KfwAuthorizationAccountClient.exe disable guest01` locks out the guest01 user until you re-enable the user ID.

  Linux UNIX Change directory to *`install_dir`*`/bin` and enter

  ```
  ./itmcmd execute cq "KfwAuthorizationAccountClient
     enable|disable user_name"
  ```

Tivoli Enterprise Monitoring Server configuration settings

The Tivoli Enterprise 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. Environment variables control the monitoring server's behavior.

Editing the monitoring server environment file

Edit the Tivoli Enterprise Monitoring Server environment file, KBBENV, to reconfigure the monitoring server parameters.

About this task

Take these steps to edit the monitoring server environment file:

Procedure

1. Open the environment file on the computer where the monitoring server is installed:
   - Windows From Manage Tivoli Enterprise Monitoring Services (Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services), right-click Tivoli Enterprise Monitoring Server and click **Advanced** → **Edit ENV File** to open the `KBBENV` file.
   - Linux UNIX Change to the *`install_dir`*`/config` directory and open the `ms.ini` file in a text editor.
2. Edit the file to enable (delete # at the beginning of the line), disable (type # at the beginning of the line), or modify any of the environment variables.
3. Save the file and exit the text editor.
4. Click **Yes** when a message asks if you want to recycle the service. You must recycle the monitoring server to implement the changes.

Duper process for optimizing situations

The Tivoli Enterprise Monitoring Server has a mechanism called *duper* that optimizes the activation of multiple situations when they are evaluating the same data at the same sampling interval. This topic describes how the duper process works, how to identify situations that use it, why you might want to disable it, and how to configure the Tivoli Enterprise Monitoring Server environment file to disable it.

Duper process

A duper situation is created and runs on the agent to collect data from the attribute group once and upload it to the monitoring server. The monitoring server evaluates the multiple situations using the data collected by the duper situation. Because the situation evaluation is taking place at the monitoring server, when the agent is disconnected, these situations are no longer evaluated.

If agents are routinely offline or disconnected from the monitoring server and running autonomously, they are probably sending events directly from the agent to an event receiver other than the monitoring server. It might be preferable to define *private situations* at the agent rather than using enterprise situations that are defined at the monitoring server.

Duper eligibility

For a situation to qualify for the duper process, it must have these qualities:

- Monitors the same attribute group on the same managed system and with the same monitoring interval as at least one other situation
- Uses only the VALUE formula function
- Does not specify persistence, an Until clause, or dynamic thresholding with expression override
- Defined to AutoStart
- Does not embed another situation
- Match another situation's display item (applies only if display item is used)

Reflex automation consideration

In IBM Tivoli Monitoring V6.3, duper is automatically applied to situations with reflex actions. No additional configuration is required.

If you have a large number of situations with reflex actions set in the Tivoli Enterprise Portal to ◉ **Execute the Action at the Managed System (Agent)**, then you can set the monitoring server environment variable KMS_EVAL_REFLEX_AT_TEMS=Y in the KBBENV file to increase your level of duper optimization.

Set the KMS_EVAL_REFLEX_AT_TEMS environment variable at each monitoring server in your environment to cause the evaluations of the actions to be handled by duper on the monitoring server, while the actions are still directed to the managed systems. If you want to implement this variable, your managed systems must be connected to a monitoring server in order for the actions to execute.

Attention: Use the KMS_EVAL_REFLEX_AT_TEMS environment variable with caution.

Note: The KMS_EVAL_REFLEX_AT_TEMS variable has no affect on your situation if you have set ◉ **Execute the Action at the Managing System (TEMS)**.

Duper situation _Z_ identifier

You can verify that a duper situation is collecting data from the agent by examining the LG0 log on the agent, such as `C:\ibm\ITM\TMAITM6\logs\Primary_IBM_MyComputer_NT.LG0`. There will be an entry starting with **_Z_** that shows the agent is starting a situation on the attribute group that the multiple situations monitor. Example: `Starting _Z_WTSYSTEM0 <3207594896,3996125040> for KNT.WTSYSTEM`.

Disable duper

By adding a parameter to the monitoring server, you can disable the duper process. This is done by adding this line to the KBBENV file: `CMS_DUPER=NO`

When the monitoring server is recycled, the duper is skipped.

To edit the monitoring server environment variable file, KBBENV on Windows and KDSENV on z/OS, follow these steps:

Windows

Use Manage Tivoli Enterprise Monitoring Services (Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services) to edit environment files. Right-click the component you want to modify and click **Advanced** → **Edit ENV File**. You must recycle the component to implement the changes.

Linux UNIX

Edit the environment file directly. Edit environment variables in the `<install_dir>/config/ms.ini` file and then reconfigure and recycle the monitoring server to implement the changes.

z/OS

See *Configuring the Tivoli Enterprise Monitoring Server on z/OS* for more information.

Tivoli Enterprise Monitoring Automation Server configuration settings

The Tivoli Enterprise Monitoring Automation Server extends the functionality of the hub Tivoli Enterprise Portal Server and includes the Open Services Lifecycle Collaboration Performance Monitoring (OSLC) service provider.

Editing the Tivoli Enterprise Monitoring Automation Server

Edit the Tivoli Enterprise Monitoring Automation Server environment file, KASENV, to reconfigure the automation server parameters.

About this task

Take these steps to edit the automation server environment file:

Procedure

1. Open the environment file on the computer where the hub monitoring server is installed:
   - Windows From Manage Tivoli Enterprise Monitoring Services (Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services), right-click Tivoli Enterprise Monitoring Automation Server and click **Advanced** → **Edit ENV File** to open the `KASENV` file.
   - Linux UNIX Change to the *`install_dir`*`/config` directory and open the `as.ini` file in a text editor.
2. Edit the file to enable (delete # at the beginning of the line), disable (type # at the beginning of the line), or modify any of the environment variables.
3. Save the file and exit the text editor.
4. Click **Yes** when a message asks if you want to recycle the service. You must recycle the automation server to implement the changes.

Chapter 5. Enabling user authentication

Login access to the Tivoli Enterprise Portal client is controlled by user accounts that are defined to the Tivoli Enterprise Portal Server. Password authentication is controlled by a registry, either the operating system user registry of the hub monitoring server or an external LDAP user registry that is configured at the hub monitoring server or at the portal server.

tacmd CLI login access and SOAP client requests to the hub Tivoli Enterprise Monitoring Server are controlled by user accounts that are defined to the hub monitoring server using either the operating system registry of the monitoring server or an external LDAP server that is configured at the hub monitoring server.

Login access to the IBM Dashboard Application Services Hub is controlled by the operating system user registry, an LDAP user registry, or a custom standalone user registry. If you plan to use monitoring dashboard applications or custom monitoring dashboards in IBM Dashboard Application Services Hub then you must configure the Tivoli Enterprise Portal Server and Dashboard Application Services Hub to use a federated LDAP user registry and single sign-on, if you want your dashboard users to launch the Tivoli Enterprise Portal client without being prompted for their credentials and if you want to control authorization to monitored resources on a per user basis. See the roadmaps in Chapter 3, “Preparing your dashboard environment,” on page 29 to determine if you want to use a federated LDAP user registry and single sign-on.

Login access to the Open Services Lifecycle Collaboration Performance Monitoring service provider component of the Tivoli Enterprise Monitoring Automation Server is controlled by an LDAP user registry and using the Security Services component of Jazz for Service Management.

The sysadmin user ID

An initial **sysadmin** user ID with full administrator authority is provided at installation so that you can log on to the Tivoli Enterprise Portal client and add more user accounts. No password is required to log on to the portal client unless the hub monitoring server was configured to enable ☑ **Security: Validate User**.

Tivoli Enterprise Portal user profile

To login using a Tivoli Enterprise Portal client, a user must be authenticated by the portal server and have a Tivoli Enterprise Portal user ID. Each user ID that is defined in the Tivoli Enterprise Portal is assigned a set of permissions that determine the portal client features the 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.

User IDs that will have the same permissions can be organized into user groups so that changes to the permissions are applied to all member user IDs.

When the Dashboard Application Services Hub and portal server are configured for single sign-on, a Tivoli Enterprise Portal user ID must exist for each monitoring dashboard user. The first time a dashboard user accesses monitoring data, a Tivoli Enterprise Portal user ID is automatically created for the user if there is not already a user ID mapped to the user's

LDAP distinguished name. In this case, the Tivoli Enterprise Portal user ID is a randomly generated ID and the user is not assigned any permissions. If Tivoli Enterprise Portal permissions are being used to control access to monitored resources in the dashboards instead of authorization policies, or if the dashboard user can launch the Tivoli Enterprise Portal, assign the user ID permissions and the monitored applications that can be accessed.

For more information on assigning Tivoli Enterprise Portal permissions and monitoring applications, see Chapter 6, "Using Tivoli Enterprise Portal user authorization," on page 163.

Authentication through the hub monitoring server

User IDs authenticated through the hub monitoring server can be authenticated by either the local operating system registry or an external LDAP-enabled central user registry.

User IDs that use the **tacmd** commands which send requests to the hub monitoring server or that make SOAP Server requests, must be authenticated through the hub monitoring server.

Limitations:

1. LDAP authentication is not supported for hub monitoring servers on z/OS.
2. The Tivoli Directory Server LDAP client used by the Tivoli Enterprise Monitoring Server does not support LDAP referrals, such as those supported by Microsoft Active Directory.
3. When the hub monitoring server is installed on a distributed operating system and is used to authenticate Tivoli Enterprise Portal users, the Tivoli Enterprise Portal user IDs must be 10 characters or less. However, for SOAP client users and **tacmd** CLI users that are authenticated by the hub monitoring server, the user IDs can be up to 15 characters.
4. When the hub monitoring server is installed on z/OS, the user ID length is limited to 8 characters if authentication uses the RACF® (Resource Access Control Facility) security for z/OS.

LDAP authentication through the portal server

The portal server authenticates Tivoli Enterprise Portal users, Dashboard Application Services Hub users who access monitoring data, IBM Tivoli Monitoring charting web service users, and **tacmd** CLI users who use commands that send requests to the portal server.

By default, the portal server contacts the hub monitoring server to perform the authentication. However, it is best practice to configure the portal server to perform its own authentication through a federated LDAP user registry for these scenarios:

- The Tivoli Enterprise Portal is launched from other web-based applications and you don't want users to re-enter their credentials.
- The Tivoli Enterprise Portal is used to launch other web-based or web-enabled applications and you don't want users to re-enter their credentials.
- IBM Dashboard Application Services Hub is used to display monitoring data retrieved using the dashboard data provider component of the portal server. Best practice is to use single sign-on in this case, so that dashboard users can launch the Tivoli Enterprise Portal and user don't

have to re-enter their credentials. Additionally, single sign-on must be used if you want to control authorization to monitored resources on a per user basis.

- The IBM Tivoli Monitoring charting web service is being used by another application such as Tivoli Integrated Portal.

When the portal server is configured to authenticate with an LDAP server, users login to Tivoli Enterprise Portal using their LDAP relative distinguished name (which normally maps to the `cn=` or `uid=` value) and not their Tivoli Enterprise Portal user ID. Because the portal server uses Tivoli Enterprise Portal user IDs to control permissions, you must map LDAP distinguished names to Tivoli Enterprise Portal user IDs. Although the Tivoli Enterprise Portal user IDs are limited to 10 characters, the LDAP distinguished names can be much longer.

You can configure the portal server to use an LDAP user registry by using the Manage Tivoli Enterprise Monitoring Services utility, the **itmcmd** command line interface on Linux and UNIX, or the TEPS/e administration console (ISCLite). If you configure LDAP using the TEPS/e administration console, you must manually restart ISCLite through the Manage Tivoli Enterprise Monitoring Services after each portal server restart.

Authentication through the hub monitoring server and the portal server

The hub monitoring server and portal server can connect to the same LDAP server if you have users who need login access to both servers. You can use the same user ID to log on to the Tivoli Enterprise Portal client that you use for the **tacmd login** command. To do this, you must go to **Administer Users** in the portal client to map the Tivoli Enterprise Portal user ID to the distinguished name used by the portal server's LDAP user registry which, by default, uses `o=ITMSSOEntry` and *not* the distinguished name that uses `o=DEFAULTWIMITMBASEDREALM`.

Migrating LDAP authentication from the hub to the portal server

If your hub Tivoli Enterprise Monitoring Server has already been configured to authenticate users to an LDAP user registry, and you now want to configure the portal server to use an LDAP user registry, you must change the Distinguished Name that is set for the user IDs in the Administer Users window of the Tivoli Enterprise Portal.

Roadmap for user authentication

Use the following roadmap to get you started with user authentication.

Table 9. Roadmap for user authentication

Task	Where to find information
Setup user authentication through the hub monitoring server using either the local operating system user registry or an LDAP user registry.	“User authentication through the hub monitoring server” on page 92
Setup the portal server to use an LDAP user registry to authenticate users when single sign-on is used with IBM Dashboard Application Services Hub or other applications.	If the hub monitoring server is not using an LDAP user registry, see “LDAP user authentication through the portal server” on page 99. If the hub monitoring server is using an LDAP user registry, see “Migrating LDAP authentication from the monitoring server to the portal server” on page 126.

Table 9. Roadmap for user authentication (continued)

Task	Where to find information
Setup the Tivoli Enterprise Monitoring Automation Server and its Performance Monitoring service provider to authenticate HTTP GET requests from OSLC clients.	"Authentication through the Tivoli Enterprise Monitoring Automation Server" on page 128

User authentication through the hub monitoring server

User authentication through the hub monitoring server uses either the local operating system user registry or an external LDAP-enabled central registry.

Prerequisites for configuring authentication on the hub monitoring server

Complete the following tasks before enabling user authentication on the hub monitoring server.

About this task

Table 10. Tasks to complete before configuring authentication

Task	Where to find information
Set up Tivoli Enterprise Portal user accounts.	"Adding a user ID" on page 170
Set up user accounts in the authenticating registry.	See the documentation for setting up user accounts on the local operating system or LDAP directory server. For information on setting up users on z/OS, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS*. **Note:** • When the hub monitoring server is installed on a distributed operating system and is used to authenticate Tivoli Enterprise Portal users, the Tivoli Enterprise Portal user IDs must be 10 characters or less. However, hub monitoring users who only use the **tacmd** CLI commands that send requests to the hub or who send SOAP requests, can have user IDs up to 15 characters. The passwords of SOAP and tacmd command users are also limited to 15 characters or less. • When the hub monitoring server is installed on z/OS, the user ID length is limited to 8 characters if authentication uses the RACF (Resource Access Control Facility) security for z/OS.
Setup TLS/SSL communication between the hub and an LDAP server.	"Configuring TLS/SSL communication between the hub monitoring server and the LDAP server" on page 212

If you intend to authenticate using the hub Tivoli Enterprise Monitoring Server, make sure that user accounts for the Tivoli Enterprise Portal Server log-in IDs are set up in the authenticating registry before authentication is enabled. At a minimum, add the **sysadmin** user ID to the local operating system user registry on the hub computer, so that **sysadmin** can log in after authentication has been enabled.

Note: On Windows, the installer creates a **sysadmin** user account in the Windows user registry and asks you to specify a password for that ID. The password is not required unless password authentication is enabled.

Tip: The Windows installer does not set the "Password never expires" option when it creates the **sysadmin** account. If you do not set this option, the password will expire according to the security policy on the hub computer, and you will not be able to log in to the portal server. Use the Windows Administrative Tools to ensure that the "Password never expires" option is selected for the **sysadmin** user account.

Before you enable authentication, obtain the following information:

Procedure

- If you are using an external LDAP server for authentication, obtain the information shown in the following table from the LDAP administrator before configuring user authentication.

Table 11. LDAP configuration parameters

Parameter	Description

LDAP User Filter	The attributes used to map Tivoli Enterprise Portal user IDs to LDAP log-in IDs. The attribute must contain the same name as the Tivoli Enterprise Portal log-in ID. The portal user ID will usually become the "%v" in the LDAP user filter. For example:
	`IBM Tivoli Directory Server: (&(mail=%v@yourco.com)`
	` (objectclass=inetOrgPerson))`
	`Microsoft Windows Active Directory: (&(mail=%v@yourco.com)`
	` (objectclass=user))`
	`Sun Java System Directory Server: (&(mail=%v@yourco.com)`
	`(objectclass=inetOrgPerson)`
	Not all LDAPs have the mail attribute for the person. For example, the LDAP administrator might only set the common name, in which case the filter would look like the following:
	`(&(cn=%v) (objectclass=inetOrgPerson))`
	The Tivoli Enterprise Portal administrator should verify exactly which LDAP attribute must be used to search for the user. With Active Directory, for example, the `cn` equals the Full Name of the Active Directory user, and this *must* be exactly the same as the Tivoli Monitoring user, and cannot have spaces (for example, "S Smith" must be "SSmith").

LDAP base	The LDAP base node in the LDAP user registry that is used in searches for users. For example:
	`IBM Tivoli Directory Server: dc=yourdomain,dc=yourco,dc=com`
	`Microsoft Windows Active Directory: dc=yourdomain,dc=yourco,dc=com`
	`Sun Java System Directory Server: dc=yourdomain,dc=yourco,dc=com`

LDAP bind ID	The LDAP user ID for bind authentication, in LDAP notation. This LDAP user ID must be authorized to search for LDAP users. This value can be omitted if an anonymous user can search for LDAP users.
LDAP bind password	The password for LDAP bind authentication. This value can be omitted if an anonymous user can bind to your LDAP server. This value is encrypted by the installer.

Table 11. LDAP configuration parameters (continued)

Parameter	Description
LDAP host name	The LDAP server host name. This value can be omitted if your LDAP server is on the same host as the Tivoli Enterprise Monitoring Server. (The default is localhost.)
LDAP port number	The LDAP server port number. This value can be omitted if your LDAP server is listening on port 389.

- If you are using Microsoft Active Directory, see "LDAP user authentication using Microsoft Active Directory" on page 129 for planning and configuration information specific to this type of LDAP server.
- If you intend to use TLS/SSL communication between the hub Tivoli Enterprise Monitoring Server and the LDAP server, obtain the information described in the following table.

Table 12. TLS/SSL parameters for communication between hub and LDAP server

Parameter	Description
LDAP key store file	The location of GSKit key store data base file. You can specify any location. For example: `C:\IBM\ITM\keyfiles`
LDAP key store stash	The location of the GSKit database password file. For example: `C:\IBM\ITM\keyfiles\keyfile.sth`
LDAP key store label	The key store label. For example: `IBM_Tivoli_Monitoring_Certificate`
LDAP key store password	The password required to access the key store.

Configuration procedure

Configure user authentication on the Windows-, Linux-, or UNIX-based hub monitoring server.

For instructions to configure authentication on a hub monitoring server installed on z/OS, see*IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*. Authentication by an external LDAP registry is not supported on a z/OS hub.

Windows: Configuring user authentication through the hub

Configure a hub Tivoli Enterprise Monitoring Server on Windows to authenticate users.

About this task

To configure user authentication through the hub on a Windows computer, complete the following procedure:

Procedure

1. Select **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services**
2. Right-click the hub monitoring server and select **Reconfigure**.
3. In the configuration window that displays, select **Security: Validate User**. The option **LDAP Security: Validate User with LDAP** becomes available.

4. If you want to use LDAP for user authentication, check the **Validate User with LDAP** option, then click **OK** to open the LDAP window. If you want to use the local registry, skip to step 7.
5. Specify the required LDAP values as appropriate for your site.
6. If you want to use TLS/SSL to secure communications between the hub and the LDAP server, check **LDAP SSL Communications: Use SSL?** and provide the appropriate values. If required, provide a password for the keystore.
7. Click **OK** The Hub TEMS Configuration window is displayed.
8. Click **OK** to accept the current settings.
9. In the Manage Tivoli Enterprise Monitoring Services window, restart the hub monitoring server by right-clicking its name and selecting **Start**.

Linux or UNIX: Configuring user authentication through the hub

Configure user authentication for an environment in which the hub is installed on Linux or UNIX.

Configuring user authentication from the command line:

Using the following procedure, you can configure user authentication from the command line.

About this task

To configure the hub from the command line, perform the following procedure:

Procedure

1. Change to the `install_dir/bin` directory and run the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

   where *install_dir* is the installation directory for IBM Tivoli Monitoring, and *tems_name* is the name of the hub monitoring server. The default installation directory on Linux or UNIX is `/opt/IBM/ITM`. You will see the following prompt:

   ```
   Configuring TEMS...
   ```
2. Accept the defaults for the following prompts. The defaults should reflect the selections made during installation.
3. When you see the prompt:

   ```
   Security: Validate User?
   ```

   type 1 and press Enter.
4. If you do not want to use LDAP for authentication, press Enter to select the default (NO). If you want to use LDAP for authentication, type 1 and press Enter. Respond to following prompts by entering the values. To enable TLS/SSL communications between the hub and the LDAP server, provide the appropriate values.
5. Accept the defaults for the Tivoli Event Integration Facility and the Workflow Policy/Tivoli Emitter Agent Forwarding.
6. At the following prompt, type 6 (Save/exit) and press Enter:

   ```
   Hubs
   ##      CMS_Name
   1       ip.pipe:elsrmt1[4441]
   ```
7. Restart the hub Tivoli Enterprise Monitoring Server:

```
./itmcmd server stop tems_name
./itmcmd server start tems_name
```

Configuring authentication by using Manage Tivoli Enterprise Monitoring Services:

Configure authentication by using Manage Tivoli Enterprise Monitoring Services.

About this task

To configure authentication by using Manage Tivoli Enterprise Monitoring Services, complete the following steps:

Procedure

1. 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 on Linux or UNIX is `/opt/IBM/ITM`. The Manage Tivoli Enterprise Monitoring Services window is displayed.
2. Right-click the hub monitoring server and click **Configure**.
3. Click the **Advanced Setting** tab. Select **Security: Validate User**.
4. If you want to use LDAP to authenticate users instead of the system registry, select **LDAP user authentication**.
5. Click **OK**. If you selected the LDAP option, the LDAP configuration panel is displayed.
6. Specify the values, then click **OK**.
7. Click **OK**.
8. Restart the hub monitoring server, using one of the following methods:
   - In the Manage Tivoli Enterprise Monitoring Services window, right-click the hub monitoring server and select **Recycle**.
   - From the command line, enter:

     ```
     ./itmcmd server stop tems_name
     ./itmcmd server start tems_name
     ```

Ldapsearch for LDAP information

Ldapsearch is a command-line tool available from LDAP server vendors that you can use to verify LDAP information before configuration and to troubleshoot problems encountered during configuration. You can save a lot of time by running **ldapsearch** to verify the LDAP information before configuring a hub monitoring server for LDAP authentication.

Note: Use this tool only if you are configuring LDAP authentication through the hub monitoring server. If you are configuring LDAP authentication through the Tivoli Enterprise Portal Server, use the TEPS/e (Tivoli Enterprise Portal Server extension server) administration console to verify configuration parameters.

Ideally, **ldapsearch** is run by the LDAP administrator. The **ldapsearch** command operates similarly to the ping command. If the values used as input to the command are correct, the command returns a version of the values you use in the search. If the values are not correct, the command returns nothing or returns an error message that can help you determine which value is involved, such as an incorrect password or a bad host name.

IBM Tivoli Directory Server **ldapsearch** is best suited for IBM Tivoli Monitoring. The Tivoli Directory Server **ldapsearch** supports GSKit TLS/SSL operations that are used in Tivoli Monitoring and has additional command-line options to support LDAP TLS/SSL searches. Tivoli Monitoring does not include ldapsearch with production installations. For information on Tivoli Directory Server ldapsearch, see "Client utilities" in the *Tivoli Directory Server Command Reference* on the IBM Security Systems Information Center.

The **ldp.exe** is a Microsoft Windows LDAP search tool that has the same basic features as the **ldapsearch** tool. You can download this tool from the Microsoft website for your version of Windows. The **ldp.exe** tool is included in the Windows Server 2003 CD support tools. For information on using the Microsoft Windows **ldp** command, see http://support.microsoft.com/kb/224543.

Another tool that can assist in LDAP configuration is the LDAP Browser tool from Softerra.

ldapsearch command-line options

The following table lists the ldapsearch options in the command-line and their corresponding parameters located in the monitoring server configuration file.

Table 13. ldapsearch command line options and corresponding monitoring server configuration parameters

Option	Description	Corresponding parameter in monitoring server configuration file
-h *host*	The host name of LDAP server.	KGL_LDAP_HOST_NAME
-p *port*	The LDAP port number.	KGL_LDAP_PORT
-D *dn*	The LDAP bind ID Do not use this command-line option if an LDAP bind ID is not required.	KGL_LDAP_BIND_ID
-w password	The LDAP bind password Use the unecrypted value for the **ldapsearch** command-line option. Do not use this command-line option if an LDAP bind ID is not required.	KGL_LDAP_BIND_PASSWORD
-b base_dn	The LDAP base.	KGL_LDAP_BASE
-K keyfile	The LDAP key store file (used only with LDAP SSL).	KGL_KEYRING_FILE
-P key_pw	The LDAP key store password (used only with LDAP TLS/SSL). Use the unecrypted value for the ldapsearch command-line option.	KGL_KEYRING_PASSWORD decrypted value
-N key_name	The LDAP key store label (used only with LDAP SSL).	KGL_KEYRING_LABEL
Filter	LDAP user filter. Replace %v with Tivoli Enterprise Portal, SOAP, or tacmd user ID.	KGL_LDAP_USER_FILTER

Sample `ldapsearch` command (no TLS/SSL)

Here is a sample ldapsearch command and its corresponding output data for a configuration with TLS/SSL disabled.

Use the following values to configure the ldapsearch command in an environment where TLS/SSL is not enabled, and no user ID or password are required:

LDAP host name	ldap.itm62.com
LDAP port name	389
LDAP base	ou=itm62users,o=itm62.com
LDAP user filter	"(mail=%v@us.ibm.com)"

Use the following command syntax for this sample configuration:

```
ldapsearch -h ldap.itm62.com -p 389 -b "ou=itm62users,o=itm62.com"
 -s sub "(mail=sysadmin@itm62.com)"
```

The following output is produced:

```
uid=12345678,ou=itm62users,o=itm62.com
objectClass=person
objectClass=organizationalPerson
...
mail=sysadmin@itm62.com
...
```

Sample `ldapsearch` command (with TLS/SSL)

Here is a sample ldapsearch command and its corresponding output data for a configuration with TLS/SSL enabled.

Use the following values to configure the ldapsearch command in an environment where TLS/SSL is enabled, and a bind ID and password are required:

LDAP host name	ldap.itm62.com
LDAP port name	636
LDAP bind ID	uid=1,ou=itm62users,o=itm62.com
LDAP bind password	itm62
LDAP base	ou=itm62users,o=itm62.com
LDAP key store	C:\IBM\ITM\itm62keyfiles\keyfile.kdb
LDAP key stash	C:\IBM\ITM\itm62keyfiles\keyfile.sth
LDAP keystore label	BM_Tivoli_Monitoring_Certificate
LDAP keystore password	itm62
LDAP user filter	"(mail=%v@us.ibm.com)"

Use the following command syntax for this sample configuration:

```
ldapsearch -h ldap.itm62.com -p 636 -D uid=1,ou=itm62users,o=itm62.com
    -w itm62 -b "ou=itm62users,o=itm62.com" -s sub
    -K C:\IBM\ITM\itm62keyfiles\keyfile.kdb -P itm62
    -N "IBM_Tivoli_Monitoring_Certificate" "(mail=sysadmin@itm62.com)"
```

The following output is produced:

```
uid=12345678,ou=itm62users,o=itm62.com
objectClass=person
objectClass=organizationalPerson
...
mail=sysadmin@itm62.com
...
```

LDAP user authentication through the portal server

You can configure the Tivoli Enterprise Portal Server to use an LDAP user registry to authenticate users.

This is required if you intend to provide single sign-on (SSO) capability for these scenarios:

- The Tivoli Enterprise Portal is launched from other web-based applications and you don't want users to re-enter their credentials.
- The Tivoli Enterprise Portal is used to launch other web-based or web-enabled applications and you don't want users to re-enter their credentials.
- IBM Dashboard Application Services Hub is used to display monitoring data retrieved using the dashboard data provider component of the portal server. Best practice is to use single sign-on in this case, so that dashboard users can launch the Tivoli Enterprise Portal and user don't have to re-enter their credentials. Additionally, single sign-on must be used if you want to control authorization to monitored resources on a per user basis.
- The IBM Tivoli Monitoring charting web service is being used by another application such as Tivoli Integrated Portal.

Prerequisites for configuring LDAP authentication on the portal server

Before configuring LDAP authentication on the Tivoli Enterprise Portal Server, you must create the user accounts in the Tivoli Enterprise Portal and in the authenticating LDAP registry, and have the LDAP registry configuration parameters at hand.

Verify user IDs in the LDAP registry

Add or verify user IDs in the registry, but do *not* create an account for **sysadmin** until after you have enabled authentication and are already logged on to the Tivoli Enterprise Portal.

The default user name for the Tivoli Enterprise Portal Server extended services (TEPS/e) administrator is **wasadmin**. If this UID was added to the registry, have the user registry administrator either change the name or remove the entry. In a federated LDAP user registry, two entries with the same name cause a conflict.

A best practice is to not add **sysadmin** to the LDAP user registry. If the LDAP server is unavailable you cannot log onto the Tivoli Enterprise Portal using LDAP user accounts, but you can still log onto the portal using **sysadmin** because it is mapped to the default Tivoli Monitoring realm that is authenticated by the hub monitoring server.

Set up Tivoli Enterprise Portal user accounts.

Add the user IDs that you intend to authenticate with an LDAP registry. This can be done before or after the portal server has been configured for LDAP authentication. After LDAP configuration, you must return to the Administer Users window in the portal client to associate the user ID with its distinguished name from the LDAP user registry.

Windows **The sysadmin password**

The IBM Tivoli Monitoring Windows installer creates a **sysadmin** user account in the Windows user registry on the hub monitoring server computer and prompts you to specify a password for that ID. The password is not required unless password authentication is enabled.

The installer does not set the “Password never expires” option when it creates the **sysadmin** account. If you do not set this option, the password will expire according to the security policy on the hub Tivoli Enterprise Monitoring Server and you will not be able to log in to the portal server. Use the Windows Administrative Tools to ensure that the “Password never expires” option is selected for the **sysadmin** user account.

LDAP configuration information

Obtain the information shown in the following table from the LDAP administrator before configuring the portal server for LDAP user authentication. The portal server and participating SSO applications must be configured to use the same LDAP user registry.

Table 14. LDAP configuration parameters

Parameter	Description
LDAP type	One of the following types of LDAP servers can be defined to the portal server using the Tivoli Management Services installation and configuration utilities: • Active Directory Server 2000 • Active Directory Server 2003 • Active Directory Server 2008 • Active Directory Server 2008 R2 • Tivoli Directory Server 6.x • Other **Other** is specified if you are configuring a different type of LDAP server, you are planning to enable TLS/SSL between the portal server and LDAP server, or you need to specify advanced LDAP configuration parameters besides those listed in this table. When you select **Other**, you must use the TEPS/e administration console to configure and modify the LDAP user registry details. See “Using the TEPS/e administration console” on page 113.
LDAP base	This parameter specifies distinguished name (DN) for the base entry in the LDAP registry. It is the starting point for user searches in the LDAP server. For example, for a user with a distinguished name of `cn=John Doe,ou=Rochester,o=IBM,c=US`, specify `ou=Rochester,o=IBM,c=US` for this parameter. **Note:** If you use the TEPS/e administration console to configure LDAP, this parameter is called **Distinguished name of the base entry in the repository** in the TEPS/e administration console.

Table 14. LDAP configuration parameters (continued)

Parameter	Description
LDAP DN base entry	The default value is `o=ITMSSOEntry`. However, best practice is to choose a value that is more meaningful for your organization. Typically, you set this parameter to the distinguished name of the base entry in the LDAP registry for the portal server users. For example, for a user with a distinguished name of `cn=John Doe,ou=Rochester,o=IBM,c=US`, specify `ou=Rochester,o=IBM,c=US` for this parameter. However, when multiple LDAP repositories are being configured for the portal server, use this field to define an additional distinguished name (DN) that uniquely identifies the set of LDAP users from this LDAP server. For example, the `LDAP1` registry and the `LDAP2` registry might both use `o=ibm,c=us` as their base entry. In this case, use this parameter to uniquely specify a different base entry for each LDAP server within the realm. For example, specify `o=ibm1,c=us` when configuring the `LDAP1` registry and `o=ibm2,c=us` when configuring the `LDAP2` registry. **Note:** If you have multiple LDAP registries, they cannot contain any overlapping user names. The value of this parameter is displayed in the Tivoli Enterprise Portal Administer Users dialog when you list the distinguished names that can be mapped to Tivoli Enterprise Portal user IDs. **Note:** If you use the TEPS/e administration console to configure LDAP, this parameter is called **Distinguished name of the base entrythat uniquely identifies this set of entries in the realm** in the TEPS/e administration console.
LDAP bind ID	This is the LDAP user ID for bind authentication, in LDAP notation, and must be authorized to search for LDAP users. The bind ID can be omitted if an anonymous user can search for LDAP users.
LDAP bind password	This is the LDAP user password for LDAP bind authentication. This value can be omitted if an anonymous user can bind to your LDAP server. This value is encrypted by the installer.
LDAP port number	This is the port number that the LDAP server is listening on. This value can be omitted if the port is 389.
LDAP host name	This is the hostname or IP address of the LDAP server. It can be omitted if the LDAP server is on the same computer as the portal server. If you are using Microsoft Active Directory, use the hostname of a domain controller within the Active Directory Forest that is hosting the user accounts for the portal server.

Information for SSO configuration

If you intend to configure SSO, work with the administrators for the other applications that plan to use single sign-on with the portal server, to determine the values for the parameters listed in the following table. Each participating SSO application must have the same value for these parameters.

Table 15. SSO parameters

Parameter	Description
Domain name	This is the Internet or Intranet domain for which SSO is configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for SSO. Example: `ibm.com`

Table 15. SSO parameters (continued)

Parameter	Description
Realm name	A realm identifies a set of federated repositories in TEPS/e and other WebSphere Application Servers. You can choose your own realm name, but this value must be the same across all applications that are configured for SSO within the specified domain. Applications configured for the same domain name, but for a different realm name, cannot work as a part of the same SSO infrastructure. Example: `ibm_tivoli_sso`

About single sign-on

The single sign-on (SSO) feature provides users with the ability to start other Tivoli web-based or web-enabled applications from the Tivoli Enterprise Portal, or to start the Tivoli Enterprise Portal from other applications, without having to re-enter their credentials. It is also used when IBM Dashboard Application Services Hub retrieves monitoring data from the portal server or the IBM Tivoli Monitoring charting web service is being used by another application.

Authenticated credentials are shared among participating applications using LTPA (Lightweight Third Party Authentication) tokens. Read this topic to understand SSO usage and requirements.

User logon
Users log onto one of the participating applications, have their user ID and password authenticated, and then start another application from within the original application to view related data or perform required actions without having to re-enter their user ID and password.

Tivoli Enterprise Portal browser client or Java Web Start client
Using a browser client or Java Web Start client, you can start another participating Tivoli web application from the Tivoli Enterprise Portal by using **Launch Application** or by typing the URL of the application into a browser view.

You can start the Tivoli Enterprise Portal browser client from an SSO-enabled web application. SSO is also supported when launching to the Java Web Start client.

Note: If you are using SSO and you want to use the browser client on the same computer as the Tivoli Enterprise Portal Server, you must reconfigure the client to use the fully qualified name of the host computer.

Tivoli Enterprise Portal desktop client
Using the desktop client, you can start another application from a workspace by using SSO. To do this, you must enter the URL of the application in the address field of a browser view. However, you cannot start the Tivoli Enterprise Portal from another application to the desktop client.

Dashboard Application Services Hub
Dashboard users log onto IBM Dashboard Application Services Hub. When they access a dashboard that displays monitoring data, the dashboard hub sends a request to the dashboard data provider component of the portal server and includes the logged in user's LTPA token. The portal server

validates the LTPA token, extracts the LDAP user ID from the LTPA token, and determines what monitored resources the user is allowed to access.

IBM Tivoli Monitoring charting web service

When users log onto Tivoli Integrated Portal, they access a page with a chart configured to use the IBM Tivoli Monitoring charting web service. A request is sent to the charting web service on the portal server and includes the logged in user's LTPA token. The portal server validates the LTPA token, extracts the LDAP user ID from the LTPA token, and determines what monitored resources the user is allowed to access.

SSO-enabled applications belong to the same security domain and realm

For SSO to be enabled, authentication must be configured through the Tivoli Enterprise Portal Server for an external LDAP user registry that is shared by all participating Tivoli applications. This is also called a federated LDAP user registry. All the participating applications must be configured for SSO and must belong to the same Internet or intranet domain and realm.

The domain is the Internet or Intranet domain for which SSO must be configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for the SSO. Users who access applications that support SSO must specify the fully qualified hostname when entering the URL of the application, for example, the URL of Dashboard Application Services Hub. The hostname is included in the LTPA tokens that are sent to the other application servers participating in SSO, and those other servers need the fully qualified hostname so that they can verify the request came from a server in the same domain.

The realm is a parameter shared across different applications that are using the LTPA SSO implementation.

LTPA tokens

Authenticated credentials are shared among participating applications using LTPA tokens. An LTPA token is encrypted data containing the authentication-related data for a user who has already been authenticated using the shared LDAP user registry. Participating SSO applications pass the user's LTPA token using a browser cookie.

LTPA tokens are secure because they are created using secure cryptography. The tokens are both encrypted and signed. The server creating an LTPA token uses a set of cryptographic keys. The cryptographic keys are used to encode the token, so that the encoded token traveling to the user's browser cannot be decoded by someone who does not have the cryptographic keys. The cryptographic keys also are used to validate the token ensuring that the token integrity is verifiable and tampering can be readily detected. When an SSO server receives an HTTP request and sees that the LTPA token is included, the server verifies the token using its copy of the shared cryptographic keys, and the information in the valid token allows the server to recognize the logged-in user.

Accordingly, LTPA keys must be exchanged among participating SSO servers so that all servers are using the same LTPA key. Choose one of the servers to be the source of the LTPA key. Then export its LTPA key and provide it to the administrators of the other servers so that they can import it. When you perform the export step, you must export the key into a key file. You must provide a name for the key file and the password to use to encrypt the key. The key file and password must be provided to the administrators of the other participating SSO applications so that they can import the LTPA key.

For example, if multiple applications can launch the Tivoli Enterprise Portal client, you can export the LTPA key from the portal server and provide the key file and password to the administrators of the other applications so that they can import the LTPA key.

Synchronizing the time across participating servers

LTPA tokens are time sensitive. Verify that the date, time, and time zone on the portal server computer and the computers of the participating SSO applications are correctly set and relative to Coordinated Universal Time (UTC). For example, the portal server in New York is set to UTC -5:00 and the Dashboard Application Services Hub in Paris is set to UTC+1:00.

Related tasks:

"Reconfiguring the browser client for SSO" on page 122
Reconfigure the browser client to specify the fully-qualified name of the Tivoli Enterprise Portal Server if you want to have SSO capability when you log on to the Tivoli Enterprise Portal from the same computer.

Roadmap for setting up the portal server to use an LDAP user registry and single sign-on

After the user IDs available for single sign-on (SSO) have been established in the LDAP user registry, enable SSO by completing the tasks in this topic.

- Verify that all prerequisites for enabling authentication and single sign-on have been met.
- Define Tivoli Enterprise Portal user accounts. (This can also be done after LDAP authentication and SSO have been configured.)
- Configure LDAP authentication and SSO through the portal server.
- Exchange LTPA keys with participating SSO applications.
- Map Tivoli Enterprise Portal user IDs to LDAP distinguished names.

Roadmap

Use the following scenario roadmap to help you set up the portal server to use an LDAP user registry and single sign-on.

Table 16. Roadmap for setting up the portal server to use an LDAP user registry and single sign-on

Step	Task	Where to find information
1	Configure the portal server to use an LDAP user registry and specify the realm name and domain used for single sign-on. To configure the portal server to use LDAP, you can use the following options: • IBM Manage Tivoli Enterprise Monitoring Services utility • **itmcmd** command line interface on Linux and UNIX • TEPS/e administration console You use either IBM Manage Tivoli Enterprise Monitoring Services or the **itmcmd** command to enable LDAP user validation for the portal server. You can also use these utilities to configure the LDAP connection parameters unless: • You want to use a server besides Microsoft Active Directory or Tivoli Directory Server • You want to configure TLS/SSL between the portal server and the LDAP server • You need to specify advanced LDAP configuration parameters For these scenarios, you specify the type of **Other** when configuring the portal server and then use the TEPS/e administration console to complete the LDAP connection configuration. **Note:** You can also export the portal server's LTPA key or import the LTPA key from another application at the same time as configuring LDAP user authentication or you can perform these steps after you have verified the portal server's LDAP authentication is working.	See "Prerequisites for configuring LDAP authentication on the portal server" on page 99. Then, use the instructions in one of the following topics to enable LDAP user validation on the portal server: • "Using Manage Tivoli Enterprise Monitoring Services to configure the portal server for LDAP authentication" on page 107 • "Using the Linux or UNIX command line to configure the portal server for LDAP authentication" on page 111 Then, follow the instructions in "Using the TEPS/e administration console" on page 113 if you specified an LDAP server type of **Other** when enabling LDAP user validation for the portal server. **Usage notes:** If you are using Microsoft Active Directory, see "LDAP user authentication using Microsoft Active Directory" on page 129 for planning and configuration information specific to this type of LDAP server. If you are using Tivoli Directory Server, see Understanding single sign-on between IBM Tivoli Monitoring and Tivoli Integrated Portal using Tivoli Directory Server in the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring). These instructions explain how to map entries configured in Tivoli Directory Server to the information configured using the TEPS/e administration console. Ignore the steps provided for Tivoli Integrated Portal.
2	Configure the other participating SSO applications to use the same LDAP user registry, realm, and Internet or intranet domain name as the portal server and enable SSO. Also, verify that the date, time, and time zone on the portal server computer and the computers of the participating SSO applications are correctly set and relative to Coordinated Universal Time (UTC).	If you are using single sign-on with Dashboard Application Services Hub, see the "Configuring Jazz for Service Management for a central user registry" and "Configuring SSO on the application server" topics in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). For other applications, refer to their product documentation to determine how to configure them to use the LDAP user registry, to enable SSO, and how to specify the realm name and domain name as the portal server.
3	Map Tivoli Enterprise Portal user IDs to LDAP distinguished names.	"Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120

Table 16. Roadmap for setting up the portal server to use an LDAP user registry and single sign-on (continued)

Step	Task	Where to find information
4	Reconfigure the Tivoli Enterprise Portal browser client for SSO if it will be launched by another application on the same computer as the portal server.	"Reconfiguring the browser client for SSO" on page 122
5	Verify your Tivoli Enterprise Portal users can launch the portal client and successfully login. **Note:** The portal client users must specify the value of their relative distinguished name when they login. For example, if their relative distinguished name is `cn=John Doe` then they must specify `John Doe` when prompted for their credentials.	If the Tivoli Enterprise Portal users cannot log into the Tivoli Enterprise Portal, review the TEPS/e log for diagnostic information. This is the `SystemOut.log` located on the computer where the portal server is installed at *install_dir*`\CNPSJ\profiles\ITMProfile\logs`; *install_dir*`/Platform/iw/profiles/ITMProfile/log`. If you encounter authentication errors and cannot resolve them, you can disable LDAP authentication by following the steps in "Disabling LDAP authentication on the portal server" on page 126.
6	Configure TLS/SSL between the portal server and LDAP server if you want to secure this communication.	"Configuring TLS/SSL communication between the portal server and the LDAP server" on page 119
7	Verify your Tivoli Enterprise Portal users can still login.	N/A
8	You must ensure the following applications are using the same LTPA key as the portal server: • A web-based or web-enabled application that launches the Tivoli Enterprise Portal • A web-based or web-enabled application that can be launched from the Tivoli Enterprise Portal client • IBM Dashboard Application Services Hub when it uses the dashboard data provider component of the portal server to retrieve monitoring data • Another application such as Tivoli Integrated Portal that uses the IBM Tivoli Monitoring charting web service Determine which application will be the source of the LTPA key for all of the other participating SSO applications and export its LTPA key. The key file and the password used to encrypt the key must be provided to the administrators of the other participating applications.	If you decide that the portal server will be source of the LTPA key, export its LTPA key using the export instructions in "Importing and exporting LTPA keys" on page 123. If you are using IBM Dashboard Application Services Hub for monitoring dashboards and it will be the source of the LTPA key, see "Exporting LTPA keys" in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). Otherwise, refer to the documentation of the application whose LTPA key will be exported to determine how to perform the export operation.

Table 16. Roadmap for setting up the portal server to use an LDAP user registry and single sign-on (continued)

Step	Task	Where to find information
9	The administrators of the other participating SSO applications must import the LTPA key that was exported in the previous step. They need the key file and the password that was used to encrypt the key.	To import an LTPA key into the portal server, see the import instructions in "Importing and exporting LTPA keys" on page 123. To import an LTPA key into IBM Dashboard Application Services Hub see "Importing LTPA keys" in the *Jazz for Service Management Configuration Guide* on the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). See the documentation for the other participating SSO applications for instructions on importing the LTPA key.
10	Verify that single sign-on is working between the portal server and each participating SSO application by performing the following tasks that apply to your SSO environment: 1. Verify that other application can launch the Tivoli Enterprise Portal and users are not prompted for their credentials. 2. Verify that Tivoli Enterprise Portal can be used to launch another application and that the user is not prompted to re-enter their credentials. 3. Verify that monitored resources can be displayed in monitoring dashboards of Dashboard Application Services Hub after a data provider connection has been created and configured for SSO. 4. Verify that another application can use the IBM Tivoli Monitoring charting web service to retrieve monitoring data. **Note:** When accessing the web interface of an application that supports SSO, enter the fully qualified hostname when specifying the URL of the application. The application servers participating in SSO check the LTPA tokens to verify that the request is coming from a server in the same Internet or Intranet domain.	N/A
11	Create Tivoli Enterprise Portal user IDs when new users are added in the LDAP user registry.	"Managing new LDAP users" on page 125

Using Manage Tivoli Enterprise Monitoring Services to configure the portal server for LDAP authentication

You can use Manage Tivoli Enterprise Monitoring Services to enable LDAP user authentication and single sign-on in the portal server, and optionally, to configure the LDAP server connection details.

You can use this utility to configure the LDAP server connection information, if all the following conditions are met:

- You are using Microsoft Active Directory Server or Tivoli Directory Server for your LDAP server.
- You do not plan to configure TLS/SSL between the portal server and the LDAP server.
- You do not need to configure any LDAP configuration parameters besides those listed in the Table 14 on page 100 table.

For all other scenarios, use Manage Tivoli Enterprise Monitoring Services to enable LDAP user validation and SSO for the portal server and specify server type of **Other**. Then use the TEPS/e administration console to complete the LDAP configuration.

Configuring the portal server to use an LDAP user registry involves adding LDAP information such as the bind ID and port number to the portal server configuration. At the same time, best practice is to enable single sign-on by specifying the realm name and Internet or intranet domain name used by the other applications participating SSO. For more information about these parameters, see "Prerequisites for configuring LDAP authentication on the portal server" on page 99.

You can also export the portal server's LTPA key or import the LTPA key from a participating SSO application if you have already decided which application will be the source of the LTPA key. (All participating SSO applications must use the same key). The export or import steps can also be performed at a later time if you want to concentrate on getting LDAP user authentication working or you don't have an LTPA key to import.

Before you begin

Have the configuration information for the LDAP server at hand, as well as the realm and Internet or intranet domain name for SSO.

If you want to export or import LTPA keys, ensure that the portal server is running before beginning configuration. You will get a message that the portal server will be stopped during configuration, but the server is stopped only at the end of the configuration procedure after you click **OK** to close the last dialog. If you are importing an LTPA key, you need the key file and the password that was used when the key file was generated.

About this task

Take these steps to reconfigure the portal server for user validation with an LDAP registry, enable SSO, and optionally export or import LTPA keys.

Procedure

1. Start Manage Tivoli Enterprise Monitoring Services on the computer where the portal server is installed:
   - Windows Click **Start → Programs →IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services**.
   - Linux UNIX Where *install_dir* is the IBM Tivoli Monitoring installation directory, change to the `install_dir/bin` directory and run `./itmcmd manage [-h install_dir]`.
2. Right-click **Tivoli Enterprise Portal Server:**
   - Windows Click **Reconfigure**, and click **OK** to accept the existing configuration and go to the second TEP Server Configuration window.

   - Linux UNIX Click **Configure**.
3. In the LDAP Security area, select ☑ **Validate User with LDAP?**. On Linux and UNIX, the LDAP Security area is on the **TEMS Connection** tab.
4. Optional: If you plan to use SSO, select ☑ **Enable Single Sign On?**.
5. Select the **LDAP type** from the ▼ list:
   - **AD2000** for Active Directory Server 2000
   - **AD2003** for Active Directory Server 2003
   - **AD2008** for Active Directory Server 2008
   - **IDS6** for IBM Tivoli Directory Server Version 6.x.
   - **Other** if your LDAP server is not one of those listed, you intend to customize the LDAP configuration for the Active Directory Server or Tivoli Directory Server, or you are configuring SSL communications to the LDAP server. After completing this procedure, start the TEPS/e administration console to complete the LDAP server configuration. See "Using the TEPS/e administration console" on page 113.

     Important: If you think you might need to edit the configuration of the Active Directory Server or Tivoli Directory Server at a later time, such as to configure TLS/SSL communications to the LDAP server, be sure to select **Other** and use the TEPS/e administration console to configure the server (skip step 6). Otherwise, any customization done in the TEPS/e administration console is lost the next time you reconfigure the portal server.
6. If you selected **AD2000**, **AD2003**, or **IDS6** as the **LDAP type**, complete the other fields to specify the LDAP server:
   - **LDAP base** is the distinguished name (DN) for the base entry in the LDAP registry.

     It is the starting point for user searches in the LDAP server. For example, for a user with a distinguished name of `cn=John Doe,ou=Rochester,o=IBM,c=US`, specify `ou=Rochester,o=IBM,c=US` for this parameter.
   - **LDAP DN base entry** is typically set to the distinguished name of the base entry in the LDAP registry for portal server users. For example, for a user with a distinguished name of `cn=John Doe,ou=Rochester,o=IBM,c=US`, specify `ou=Rochester,o=IBM,c=US` for this parameter.

     However, when multiple LDAP repositories are being configured for the portal server, use this field to define an additional distinguished name (DN) that uniquely identifies the set of LDAP users from this LDAP server. For example, the `LDAP1` registry and the `LDAP2` registry might both use `o=ibm,c=us` as their base entry. In this case, use this parameter to uniquely specify a different base entry for each LDAP server. For example, specify `o=ibm1,c=us` when configuring the `LDAP1` registry and `o=ibm2,c=us` when configuring the `LDAP2` registry.

     Note: If you have multiple LDAP registries, they cannot contain any overlapping user names.

     The value of this parameter is displayed in the Tivoli Enterprise Portal Administer Users dialog when you list the distinguished names that can be mapped to Tivoli Enterprise Portal user IDs.
   - **LDAP bind ID** is the LDAP user ID for bind authentication, in LDAP notation, and must be authorized to search for LDAP users. The bind ID can be omitted if an anonymous user can search for LDAP users.

- **LDAP bind password** is the LDAP user password for LDAP bind authentication. This value can be omitted if an anonymous user can bind to your LDAP server. This value is encrypted by the installer.
- **LDAP port number** that the LDAP server is listening on. This value can be omitted if the port is 389.
- **LDAP host name**, which can be omitted if the LDAP server is on the same computer as the portal server. Default: **localhost**.

7. Click **OK**.
   - If you selected ☑ **Enable Single Sign On?**, the Single Sign On dialog is displayed with **Realm name** and **Domain name** fields and **Import Keys** and **Export Keys** buttons.
   - If you are not enabling single sign-on at this time, click **OK** to close any other portal server configuration dialogs and go to step 12 on page 111
8. For SSO, specify the realm and domain in the Single Sign On dialog:
   a. **Realm name** is a parameter shared across applications participating in SSO. Applications configured for the same domain name, but for a different realm name will not work as a part of the same SSO infrastructure.

   b. **Domain name** is the Internet or Intranet domain for which SSO is configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for SSO.
9. At this time, you can export the portal server's LTPA key if you want it to be the key used by all other participating SSO applications. Click **Export Keys** and complete the following steps:
   a. Navigate to the directory where you want to create the file or change the file type, or both. The directory displayed initially, on Windows, is *ITM_dir*\InstallITM; and on Linux and UNIX, it is the Root directory.

   b. Type a name for the file that the LTPA key should be placed in and click **Save**.

   c. In the Export keys window, type a password to use to encrypt the file, and click **OK**. You see a console window while the file is created and encrypted, and then you are returned to the Single Sign On window.

   Note: After the LDAP configuration is complete, provide the key file and password to the administrators of the applications that launch Tivoli Enterprise Portal, use the dashboard data provider in IBM Dashboard Application Services Hub, or use the IBM Tivoli Monitoring charting web service.
10. If another participating SSO application is providing the LTPA key, you can import it now if you have the key file and the password that was used to encrypt the key. Click **Import Keys** and complete the following steps:
    a. In the **Open** window that is displayed, navigate to the directory where the key file is located. The directory displayed initially, on Windows, is *ITM_dir*\InstallITM; and on Linux and UNIX, it is the Root directory.

    b. Type the name of the file that you want to import, and click **Open**. You see a console window while the file is created and encrypted, and then you are returned to the Single Sign On window. Repeat the import process to import keys from additional participating servers.

    c. Type the password required to decrypt the file, and click **OK**. You see a console window while the file is created and encrypted, and then you are returned to the Single Sign On window.

   d. Repeat the import process to import keys from additional participating servers.

11. Click **OK**.
12. Windows If you are prompted to reconfigure the warehouse connection information, answer **No**. After some processing of the configuration settings, the Common Event Console Configuration window is displayed. Sometimes this window does not open in the foreground and is hidden by other windows. If processing seems to be taking longer than expected, minimize other windows and look for the configuration window. When the Common Event Console Configuration window is displayed, click **OK**.
13. If necessary, recycle the portal server by selecting **Tivoli Enterprise Portal Server** and clicking **Recycle** or by stopping, then starting the portal server.

What to do next

If you chose **Other** as the LDAP type, the LDAP configuration must be completed in the TEPS/e administration console. See "Using the TEPS/e administration console" on page 113.

Otherwise, for all other LDAP types, follow steps 1 and 2 in the procedure above to check if **Validate User with LDAP?** is still selected. If it is not selected then an error occurred when the configuration utility attempted to connect to the LDAP server and LDAP validation was disabled. If it is disabled, check the `*install_dir*/logs/ConfigureLDAPRepo.log` file.

Once the LDAP registry is completely configured, you can map the Tivoli Enterprise Portal user IDs to the LDAP distinguished names to complete the LDAP configuration. You must log on to the Tivoli Enterprise Portal with the **sysadmin** user ID or a user ID that has the same administrative authority and is not an LDAP user. See "Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120.

If you enabled SSO, you will need to export or import LTPA keys. Refer back to the "Roadmap for setting up the portal server to use an LDAP user registry and single sign-on" on page 104 to determine when to perform these steps.

Using the Linux or UNIX command line to configure the portal server for LDAP authentication

If the Tivoli Enterprise Portal Server is on Linux or UNIX, you can enable LDAP user authentication and single sign-on in the portal server, and optionally, configure the LDAP server connection details, by using the itmcmd command line interface.

You can use the command line to configure the LDAP server connection information, if all the following conditions are met:

- You are using Microsoft Active Directory Server or Tivoli Directory Server for your LDAP server.
- You do not plan to configure TLS/SSL between the portal server and the LDAP server.
- You do not need to configure any LDAP configuration parameters besides those listed in the Table 14 on page 100 table.

For all other scenarios, use the itmcmd command to enable LDAP user validation and SSO for the portal server and specify server type of **Other**. Then use the TEPS/e administration console to complete the LDAP configuration.

Configuring the portal server to use an LDAP user registry involves adding LDAP information such as the bind ID and port number to the portal server configuration. At the same time, best practice is to enable single sign-on by specifying the realm name and Internet or intranet domain name used by the other applications participating SSO. For more information about these parameters, see "Prerequisites for configuring LDAP authentication on the portal server" on page 99.

About this task

Complete these steps to configure the portal server from the command line:

Procedure

1. Log on to the computer where the Tivoli Enterprise Portal Server is installed.
2. At the command line, change to the `*install_dir*/bin` directory, where *install_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**. The message "Agent configuration started..." is displayed, followed by a prompt:

   ```
   Edit "Common event console for IBM Tivoli Monitoring" settings?
   [ 1=Yes, 2=No ] (default is: 1)
   ```
4. Enter **2**. The following prompt is displayed:

   ```
   Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
   ```
5. Accept the default values for this prompt and the prompts that follow it until you see the following prompt. The default values reflect the selections made during the original configuration.

   ```
   LDAP Security: Validate User with LDAP ? (1=Yes, 2=No)(Default is: 2):
   ```
6. Enter 1 to begin configuration of LDAP authentication and provide the values for the LDAP parameters.

   ```
   LDAP type: [AD2000, AD2003, AD2008, IDS6, OTHER](Default is: OTHER):
   ```

   For LDAP type, choose **Other** if your LDAP server is not one of those listed or you intend to customize the LDAP configuration for the Active Directory Server or Tivoli Directory Server or you plan to configure TLS/SSL between the portal server and the LDAP server. After completing this procedure, start the TEPS/e administration console to complete the LDAP server configuration. See "Using the TEPS/e administration console" on page 113.

   Important: If you think you might need to edit the configuration of the Active Directory Server or Tivoli Directory Server at a later time, for example configuring TLS/SSL communications to the LDAP server, be sure to select **Other** and use the TEPS/e administration console to configure the server. Otherwise, any customization done in the TEPS/e administration console is lost the next time you reconfigure the portal server.
7. If you did not specify type of **Other**, you are prompted to enter additional LDAP configuration values. (see Table 14 on page 100 for more information about those parameters):

   ```
   LDAP base: o=IBM
   LDAP DN Base Entry(Default is: o=ITMSSOEntry):  o=IBM
   LDAP bind ID: cn=root
   ```

```
LDAP bind password:
Re-type: LDAP bind password:
LDAP Port number(Default is: 389):
LDAP host name(Default is: localhost): itmxseries04
```

8. If you want to enable single sign-on as well as LDAP authentication, enter 1 at the following prompt; then provide the Realm name and Domain name.

   ```
   Enable Single Sign On ? (1=Yes, 2=No)(Default is: 2):
   ```

   a. **Realm name** is a parameter shared across applications participating in SSO. Applications configured for the same domain name, but for a different realm name will not work as a part of the same SSO infrastructure.

   b. **Domain name** is the Internet or Intranet domain for which SSO is configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for SSO.

   After the installer has completed the configuration, the following message is displayed: `Agent configuration completed...`

9. Recycle the portal server.

   ```
   ./itmcmd agent stop cq
   ./itmcmd agent start cq
   ```

What to do next

If you chose **Other** as the LDAP type, the LDAP configuration must be completed in the TEPS/e administration console. See "Using the TEPS/e administration console."

Once the LDAP registry is completely configured, you can map the Tivoli Enterprise Portal user IDs to the LDAP distinguished names to complete the LDAP configuration. You must log on to the Tivoli Enterprise Portal with the **sysadmin** user ID or a user ID that has the same administrative authority and is not an LDAP user. See "Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120.

If you enabled SSO, you will need to export or import LTPA keys. Refer back to the "Roadmap for setting up the portal server to use an LDAP user registry and single sign-on" on page 104 to determine when to perform these steps.

Using the TEPS/e administration console

The Tivoli Enterprise Portal Server extended services (TEPS/e) has an administrative console (ISCLite) that you can access for configuring an LDAP registry that is not supported by the IBM Tivoli Monitoring installation programs: Manage Tivoli Enterprise Monitoring Services and **itmcmd** command line configuration utilities.

You can also use the TEPS/e administration console to configure SSL for communication with the LDAP server and with other applications that send requests to either the dashboard data provider or the IBM Tivoli Monitoring charting web service.

You must use the TEPS/e administration console to configure portal server LDAP user authentication if you specified **Other** as the LDAP type when you configured the portal server during installation, or when you used the Manage Tivoli Enterprise Monitoring Services utility or **itmcmd** command line interface. You must also use the TEPS/e administration console to configure LDAP connection details if you plan to perform the following tasks:

- Use an LDAP server other than Microsoft Active Directory Server or Tivoli Directory Server.
- Configure TLS/SSL communication between the portal server and the LDAP server.
- Configure advanced LDAP configuration parameters that can be specified in the TEPS/e administration console but not in the other portal server configuration utilities.

Important: Any LDAP customization made within the TEPS/e administration console are overwritten and cleared any time the portal server is reconfigured unless you chose the LDAP type of **Other** during portal server configuration through Manage Tivoli Monitoring Services or the **`itmcmd`** command line interface configuration utilities. When **Other** is chosen, the LDAP user registry information is handled by TEPS/e and is not affected by the other portal server configuration utilities.

Starting the TEPS/e administration console

Use the TEPS/e administration console to configure an LDAP server with a type of **Other**, to configure SSL between the portal server and other applications such as the LDAP server, and to verify your LDAP configuration.

Before you begin

The TEPS/e administration console is disabled by default for security reasons and to save system resources. The Tivoli Enterprise Portal Server must be running before you enable the console.

About this task

Take these steps to enable and then start the TEPS/e administration console:

Procedure

1. Enable the TEPS/e administration console:
   - Windows In the Manage Tivoli Enterprise Monitoring Services window, highlight **Tivoli Enterprise Portal Server** and select **Advanced → TEPS/e Administration → Enable TEPS/e Administration**.
   - Linux UNIX From the command line, change to the `scripts` directory (Intel Linux: *`ITM_dir`*`/li6263/iw/scripts`; zLinux:*`ITM_dir`*`/ls3266/iw/scripts`; AIX®:*`ITM_dir`*`/aix533/iw/scripts`) and enter the following command, where `true` starts the console and `false` stops the console:

     ```
     ./enableISCLite.sh {true/false}
     ```

   The TEPS/e administration console is now enabled for logon and will remain enabled until the portal server is stopped.
2. If this is the first time you are enabling the console, you must set the administration password:
   - Windows In the Manage Tivoli Enterprise Monitoring Services window, highlight **Tivoli Enterprise Portal Server** and select **Advanced → TEPS/e Administration → TEPS/e Administration Password**.
   - Linux UNIX From the `scripts` directory, enter the following command, where *`<username>`* is **wasadmin**, and *`<password>`* is the new password:

     ```
     updateTEPSEPass.sh <username> <password>
     ```

Subsequently, entering a TEPS/e administration password resets the password.

3. Enter the following URL in your Internet Explorer or Firefox browser: `http://localhost:15205/ibm/console` or `https://localhost:15206/ibm/console`
4. Log on to the console using **wasadmin** for the user ID and the password you entered as the TEPS/e administration password.

Results

The Integrated Solutions Console (TEPS/e administration console) window is opened. Even after you log out of the administration console, it remains enabled until the Tivoli Enterprise Portal Server is stopped. You must manually restart the TEPS/e administration console through the Manage Tivoli Enterprise Monitoring Services after each portal server restart.

What to do next

You can now configure an external LDAP server connection, SSL, or verify the configuration.

If the TEPS/e administration console is running when the portal server is recycled, you must log out and enable the console again to resynchronize with the portal server.

Configuring the portal server for LDAP authentication using the TEPS/e administration console

You must use the Tivoli Enterprise Portal Server extended services (TEPS/e) administration console to configure LDAP server connection parameters if you specified **Other** as the LDAP type when you configured the portal server.

Before you begin

Start the TEPS/e administration console.

Attention: Best practice is to select the LDAP type of **Other** in the Manage Tivoli Enterprise Monitoring Services utility of **`itmcmd`** command line interface before using the TEPS/e administration console to change the LDAP server configuration in order for any future changes to persist. For example, if you selected **IDS6** as the LDAP type when you configured the portal server using the itmcmd command and you make changes to the LDAP connection parameters through the TEPS/e administration console, your changes are lost the next time you reconfigure the portal server.

Procedure

1. In the TEPS/e administration console navigation tree, click **Security → Global security**.
2. On the page that is displayed, ensure that `Federated repositories` is selected for **Available realm definition**, and click **Configure**.
3. Configure the federated repository:
   a. Verify or enter the **Realm Name** value. A realm identifies a set of federated repositories in TEPS/e and other WebSphere Application Servers. You can choose your own realm name but this value must be the same across all applications that are configured for SSO within an Internet or intranet domain. If you enabled single sign-on when you configured the portal

server, this field displays the value your specified for the realm name. For details on specifying the domain, see step 9 on page 117

   b. On the same page, click **Add Base entry to Realm**.

4. On the **Repository reference** page, click **Add repository** and choose **LDAP repository** from the dropdown list. The page now displays the properties that can be configured for the portal server to LDAP connection.
5. Provide the appropriate values for the following parameters:
   - For **Repository identifier**, enter a name for the repository that you find meaningful to identify the type of uses in the LDAP repository. For example, `ITMtepUsers`.
   - For **Directory type**, choose the type of LDAP server being used in your environment.
   - For **Primary hostname**, enter the fully qualified hostname or IP address of your LDAP server.
   - For **Port**, enter the port number of the LDAP server. The default value is 389.
   - For **Bind distinguished name**, enter the distinguished name for a user that is authorized to search for LDAP users. For example, `cn=root`. The bind ID can be omitted if an anonymous user can search for LDAP users.
   - For **Bind password**, enter the password for the user specified in the **Bind distinguished name** field. This value can be omitted if an anonymous user can bind to your LDAP server.

   If necessary, you can also customize other parameters on this page to match the capabilities of your LDAP server. For more information about the other parameters that can be configured on this panel, see the TEPS/e administration console online help.
6. Click **OK** to accept the settings.
7. On the **Repository reference** page, enter these values:
   - For **Distinguished name of the base entry that uniquely identifies this set of entries in the realm**, enter a value that uniquely identifies the set of LDAP user entries from the LDAP server for which you are configuring a connection.

     Typically, you set this parameter to the distinguished name of the base entry in the LDAP registry for the portal server users. For example, for a user with a distinguished name of `cn=John Doe,ou=Rochester,o=IBM,c=US`, specify `ou=Rochester,o=IBM,c=US` for this parameter.

     However, when multiple LDAP repositories are being configured for the portal server, use this field to define an additional distinguished name (DN) that uniquely identifies the set of LDAP users from this LDAP server. For example, the `LDAP1` registry and the `LDAP2` registry might both use `o=ibm,c=us` as their base entry. In this case, use this parameter to uniquely specify a different base entry for each LDAP server within the realm. For example, specify `o=ibm1,c=us` when configuring the `LDAP1` registry and `o=ibm2,c=us` when configuring the `LDAP2` registry.

     Note: If you have multiple LDAP registries, they cannot contain any overlapping user names.

     The value of this parameter is displayed in the Tivoli Enterprise Portal Administer Users dialog when you list the distinguished names that can be mapped to Tivoli Enterprise Portal user IDs.\

- For **Distinguished name of the base entry in this repository**, enter the distinguished name (DN) for the base entry in the LDAP registry.

  It is the starting point for user searches in the LDAP server. For example, for a user with a distinguished name of `cn=John Doe,ou=Rochester,o=IBM,c=US`, specify `ou=Rochester,o=IBM,c=US` for this parameter. Typically, this parameter is the same as the LDAP base parameter, unless you customized the distinguished name of the base entry in the realm so that it does not match the distinguished name in the LDAP server.

8. Click **OK** to accept the settings.
9. To enable SSO, return to the **Global security** page and complete the following:
   a. Ensure that **LTPA** is selected as the authentication mechanism.
   b. Expand the **Web Security** option.
   c. Select the **Single sign-on (SSO)** link to complete the SSO configuration.
10. On the **Single sign-on (SSO)** page, complete the following:
    a. Verify that SSO is enabled.
    b. Verify that the **Domain Name** parameter is correct. **Domain name** is the Internet or intranet domain for which SSO is configured, for example mycompany.com. Only applications available in this domain or its sub-domains are enabled for SSO. All participating SSO application must also be configured with the same realm name. If you enabled single sign-on when you configured the portal server, this field displays the value that you specified for the domain name.
    c. Select **OK** to accept the settings.
11. To save the changes, click the **Save** option near the top of the screen, then log out from the administration console.
12. If you want to export or import LTPA keys at this time, see the TEPS/e administration console steps in "Importing and exporting LTPA keys" on page 123.

    Note: If you export or import the keys now, you still need to perform the other steps listed in "Roadmap for setting up the portal server to use an LDAP user registry and single sign-on" on page 104 before attempting to verify that SSO is working.
13. Restart the Tivoli Enterprise Portal Server.

    Note: When the portal server is restarted, the TEPS/e administration console is disabled automatically. You must re-enable it before it can be used again by following the instructions in "Starting the TEPS/e administration console" on page 114.

What to do next

Map the Tivoli Enterprise Portal user IDs to the LDAP distinguished names. See "Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120.

You must enable the administration console after a recycle of the portal server before you can start the console again.

Important: Any LDAP customization made within the TEPS/e administration console are overwritten and cleared any time the portal server is reconfigured unless you chose the LDAP type of **Other** during portal server installation or when using the Manage Tivoli Enterprise Monitoring Services utility of **itmcmd** command line interface to perform portal server configuration. When **Other** is chosen, the registry information is handled by TEPS/e and is not affected by these other configuration utilities. See step 5 on page 109 in "Using Manage Tivoli Enterprise Monitoring Services to configure the portal server for LDAP authentication" on page 107 and 6 on page 112 in "Using the Linux or UNIX command line to configure the portal server for LDAP authentication" on page 111.

Starting and stopping TEPS/e

If you need to start or stop the application server instance of TEPS/e on which the Tivoli Enterprise Portal Server is running, you must do it by starting or stopping the Tivoli Enterprise Portal Server.

You cannot use the TEPS/e start and stop commands to control TEPS/e. If you have already used TEPS/e commands, you can recover by following the procedure below.

Starting and stopping the Tivoli Enterprise Portal Server

- Windows In the Manage Tivoli Monitoring Services window, highlight **Tivoli Enterprise Portal Server** and select **Stop** or **Start**.
- Linux UNIX Use the itmcmd utility, located in `ITM_home/bin`.
  - To start:

    ```
    cd ITM_home/bin
    ./itmcmd agent start cq
    ```

  - To stop:

    ```
    cd ITM_home/bin
    ./itmcmd agent stop cq
    ```

Starting and stopping other server instances of TEPS/e

If you need to start or stop a different application server instance on TEPS/e, for example if you have your own profile, cell, or server created on TEPS/e, you need to use the following two scripts:

- Windows

  ```
  <ITM_home>/CNPSJ/profiles/<name_of_your_profile>/bin/startServer.bat
  <name_of_your_server>
  <ITM_home>/CNPSJ/profiles/<name_of_your_profile>/bin/stopServer.bat
   <name_of_your_server>
  ```

- UNIX

  ```
  <ITM_home>/<arch>/iw/profiles/<name_of_your_profile>/bin/startServer.sh
  <name_of_your_server>
  <ITM_home>/<arch>/iw/profiles/<name_of_your_profile>/bin/stopServer.sh
  <name_of_your_server>
  ```

Examples:

- Windows If you are using a profile created by IBM Tivoli Monitoring, and your own server called <*YourServer*>, you need to use the following command:

  ```
  <ITM_home>/CNPSJ/profiles/<ITMProfile>/bin/startServer.bat
  <YourServer>
  ```

- UNIX If you have your own profile created called <*YourProfile*>, and your own server called TEPS/e, to stop the server on the UNIX platform (for example RHEL4), you need to use the following command:

  ```
  <ITM_home>/<arch>/iw/profiles/<YourProfile>/bin/stopServer.sh
  <YourServer>
  ```

Configuring TLS/SSL communication between the portal server and the LDAP server

Use the TEPS/e administration console to configure TLS (Transport Layer Sockets) or SSL (Secure Socket Layers) between the portal server and the LDAP server.

Before you begin

Ensure that you already have an existing connection to an LDAP server and that Tivoli Enterprise Portal users can login to the portal server and be authenticated by the LDAP server. You must also ensure that the Tivoli Enterprise Portal Server is configured to use an LDAP type of **Other** since the configuration of TLS/SSL for LDAP server communication must be performed using the TEPS/e administration console.

Your LDAP server must be configured to accept TLS/SSL connections and be running on the secured port number, typically port 636. Refer to your LDAP server documentation if you need to create a signer certificate, which as part of this task, must be imported from your LDAP server into the trust store of TEPS/e.

LDAP TLS/SSL requires some actions by an LDAP administrator that are not covered by the Tivoli Monitoring documentation. The following topics in the IBM Security Systems Information Center include information about setting up LDAP servers for TLS/SSL:

- Configuring Microsoft Active Directory for SSL access
- Configuring the Tivoli Directory Server client for SSL access
- Configuring Oracle Java System Directory Server for SSL access

Start the TEPS/e administration console using the instructions in “Starting the TEPS/e administration console” on page 114 before beginning the procedure.

Procedure

1. Perform the following steps to import your LDAP server's signer certificate into the TEPS/e trust store:
   a. Click **Security → SSL certificate and key management**.
   b. In the Related Items area of the page, click the **Key stores and certificates** link and in the table that is displayed, click the **NodeDefaultTrustStore** link.
   c. In the Additional Properties area, click the **Signer certificates** link and click the **Retrieve from port** button.
   d. In the relevant fields provide the hostname, port (typically `636` for SSL connections), SSL configuration details, as well as the alias of the certificate for your LDAP server. Then click the **Retrieve signer information** button and then click **OK**.
2. Follow these steps to enable TLS/ SSL communications to your LDAP server:
   a. Click **Security → Global security**.
   b. In the Related Items area near the bottom of the page, select **Manage repositories**.
   c. In the table of repositories, select the link for the repository identifier for your LDAP server.
   d. Select the **Require SSL communications** check box and select the **Centrally managed** option.

e. Change the port number from 389 to the port number that your LDAP server uses for SSL connections (typically 636).

f. Click **OK**.

g. Save the configuration changes.

3. Restart the portal server.

   Note: When the portal server is restarted, the TEPS/e administration console is disabled automatically. You must re-enable it before it can be used again by following the instructions in "Starting the TEPS/e administration console" on page 114.

What to do next

Verify that your Tivoli Enterprise Portal users can log in and be authenticated by the LDAP server.

Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names

When the portal server is configured to authenticate users using the LDAP user registry, the user logs into the portal server using the unique identifier (UID) value of the relative distinguished name. This name is not necessarily the same as the user ID known to the Tivoli Enterprise Portal. For this reason, Tivoli Enterprise Portal user IDs must be mapped to LDAP distinguished names (which include the UID).

Every entry in the LDAP user registry has a distinguished name (DN). The DN is the name that uniquely identifies an entry in the directory. A DN is made up of attribute=value pairs, separated by commas, for example:

```
cn=Jim Grey,ou=users,ou=SWG,o=IBM,c=US
cn=Sally White,ou=users,ou=SWG,o=IBM,c=US
```

The order of the attribute value pairs is important. The DN contains one component for each level of the directory hierarchy from the root down to the level where the entry resides. LDAP DNs begin with the most specific attribute, usually some sort of name, and continue with progressively broader attributes, often ending with a country attribute. The first component of the DN is referred to as the Relative Distinguished Name (RDN). It identifies an entry distinctly from any other entries that have the same parent. In the examples above, the RDN cn=Jim Grey separates the first entry from the second entry, (with RDN cn=Sally White). These two example DNs are otherwise equivalent. These two users would log into the Tivoli Enterprise Portal as Jim Grey and Sally White.

The default distinguished name for new users you create for the Tivoli Enterprise Portal has the following structure:

```
UID=tep_userid,O=DEFAULTWIMITMBASEDREALM
```

This distinguished name indicates that the user is authenticated by the hub monitoring server. Using the procedure in this topic, update the distinguished name for any Tivoli Enterprise Portal users that are defined in the portal server's LDAP user registry to specify their distinguished name in the LDAP user registry instead of UID=tep_userid,O=DEFAULTWIMITMBASEDREALM.

The default DN suffix for the TEPS/e user registry is `o=defaultWIMFileBasedRealm`. The TEPS/e user registry contains the wasadmin user ID for TEPS/e administration console access: `UID=wasadmin,o=defaultWIMFileBasedRealm`.

Do not update the distinguished names for any Tivoli Enterprise Portal user IDs that are using the `o=defaultWIMFileBasedRealm` suffix.

Before you begin

User IDs are mapped to LDAP distinguished names in the Tivoli Enterprise Portal Administer Users window by a user with administrator authority. The **`tacmd`** command line interface can also be used to preform this mapping. For more information, see the **`tacmd edituser`** command in the *IBM Tivoli Monitoring Command Reference*.

If LDAP authentication is being configured through the Tivoli Enterprise Monitoring Server, user IDs are mapped instead by editing the `KGL_LDAP_USER_FILTER` environment variable in the Tivoli Enterprise Monitoring Server configuration file.

About this task

Complete these steps to map Tivoli Enterprise Portal user IDs to LDAP distinguished names using the Tivoli Enterprise Portal Administer Users dialog window:

Procedure

1. Log on to the portal using **sysadmin** or another user account with full administrative authority.
2. Click **Administer Users**.
3. In the Administer Users window, right-click the row of the user ID to map and select **Modify User**.
4. In the Modify User dialog box, click **Find** to locate the LDAP distinguished name to be associated with the Tivoli Enterprise Portal user ID. Example:`UID=TEPUSER,O=SS`.

   Note:

   - The default suffix for LDAP distinguished names that are configured through the Tivoli Enterprise Portal Server configuration utilities is `o=ITMSSOEntry`, however this value might have been customized when the portal server was configured for LDAP.
   - If the selected LDAP distinguished name contains non-alphanumeric characters, those characters must be escaped with a backslash before the mapping is saved. For example, if a user ID contains a pound sign, `#`, place a backslash before the pound sign, `\#`.
5. Click **OK** to save the mapping and return to the Administer Users window.
6. Repeat steps 3 through 5 until you have mapped all the users that you want to authenticate with the configured LDAP registry.
7. Click **OK** to exit the Administer Users window.

What to do next

Reconfigure the Tivoli Enterprise Portal browser client for SSO if it will be launched by another application on the same computer as the portal server. See "Reconfiguring the browser client for SSO."

Verify that your Tivoli Enterprise Portal users who have IDs that are mapped to LDAP distinguished names, can log into the Tivoli Enterprise Portal client. They must use their LDAP relative distinguished name to login. If the users are not successful at logging into the Tivoli Enterprise Portal, review the TEPS/e log for diagnostic information. This is the `SystemOut.log` located on the computer where the portal server is installed at **Windows** `*install_dir*\CNPSJ\profiles\ITMProfile\logs`; **Linux** **UNIX** `*install_dir*/*Platform*/iw/profiles/ITMProfile/log`.

Refer to the "Roadmap for setting up the portal server to use an LDAP user registry and single sign-on" on page 104 for additional steps to perform after Tivoli Enterprise Portal users can be successfully authenticated by the portal server's LDAP user registry.

Reconfiguring the browser client for SSO

Reconfigure the browser client to specify the fully-qualified name of the Tivoli Enterprise Portal Server if you want to have SSO capability when you log on to the Tivoli Enterprise Portal from the same computer.

Before you begin

By default, the launch URL associated with the browser client running on the same computer as the Tivoli Enterprise Portal Server is `localhost`. If you want to use a browser client on the same computer as the portal server, this value must be the fully-qualified name of the computer, such as `dev1.myco.com`. The suffix `myco.com` is the domain value you enter in the SSO configuration panel. Using the suffix ensures that SSO tokens are visible only to the servers that are under the same domain suffix.

About this task

Complete these steps to reconfigure the browser client:

Procedure

1. Launch the Manage Tivoli Enterprise Monitoring Services utility.
2. Right-click the Tivoli Enterprise Portal Browser entry and click **Reconfigure** to open the Configure Tivoli Enterprise Portal Browser window.
3. In the **Host** field beneath portal server area, type the fully-qualified name of the computer. Example: `myhost.mycompany.com`

Related concepts:

"About single sign-on" on page 102
The single sign-on (SSO) feature provides users with the ability to start other Tivoli web-based or web-enabled applications from the Tivoli Enterprise Portal, or to start the Tivoli Enterprise Portal from other applications, without having to re-enter their credentials. It is also used when IBM Dashboard Application Services Hub retrieves monitoring data from the portal server or the IBM Tivoli Monitoring charting web service is being used by another application.

Importing and exporting LTPA keys

Authenticated credentials are shared among participating applications using LTPA keys.

Ensure that the following applications are using the same LTPA key as the portal server:

- A web-based or web-enabled application that launches the Tivoli Enterprise Portal
- A web-based or web-enabled application that can be launched from the Tivoli Enterprise Portal client
- IBM Dashboard Application Services Hub when it uses the dashboard data provider component of the portal server to retrieve monitoring data
- Another application such as Tivoli Integrated Portal that uses the IBM Tivoli Monitoring charting web service

Determine which application will be the source of the LTPA key for all of the other participating SSO applications and export its LTPA key.

If you decide to export the portal server's LTPA key, you must export the LTPA key into a key file. When you perform the export step, you must provide a name for the key file and the password to use to encrypt the key. The key file and password must be provided to the administrators of the applications listed above so that they can import the LTPA key.

If another application will not provide the LTPA key, the administrator of that application must export the application's LTPA key into a key file and then provide you with the key file and the password that was used to encrypt the key. You must import the LTPA key into the portal server and enter the password.

Before you begin

The Tivoli Enterprise Portal Server must be running for import and export operations to be performed.

If you are using the TEPS/e administration console to import or export keys, you must start the console. See "Starting the TEPS/e administration console" on page 114.

Before you can import an LTPA key, the administrator of the application that exported the key must provide you with a key file containing the LTPA key and the password that was used to encrypt the key.

About this task

Follow the steps for your environment to import or export LTPA keys:

Procedure

- **From Manage Tivoli Enterprise Monitoring Services window, complete the following procedure to export keys:**
  1. Right-click the Tivoli Enterprise Portal Server and click **Advanced → TEPS/e Administration → Export keys**.
  2. Navigate to the directory where you want to create the file or change the file type, or both. The directory displayed initially, on Windows, is *ITM_dir*\InstallITM; and on Linux and UNIX, it is the Root directory.
  3. Type a name for the file that the LTPA key should be placed in and click **Save**.
  4. In the Export keys window, type a password to use to encrypt the file, and click **OK**. You see a console window while the file is created and encrypted, and then you are returned to the Single Sign On window.
- **From Manage Tivoli Enterprise Monitoring Services window, complete the following procedure to import keys:**
  1. Right-click the Tivoli Enterprise Portal Server and click **Advanced → TEPS/e Administration → Import keys**.
  2. In the **Open** window that is displayed, navigate to the directory where the key file is located. The directory displayed initially, on Windows, is *ITM_dir*\InstallITM; and on Linux and UNIX, it is the Root directory.
  3. Type the name of the file that you want to import, and click **Open**. You see a console window while the file is created and encrypted, and then you are returned to the Single Sign On window. Repeat the import process to import keys from additional participating servers.
  4. Type the password required to decrypt the file, and click **OK**. You see a console window while the file is created and encrypted, and then you are returned to the Single Sign On window.
  5. Repeat the import process to import keys from additional participating servers.
- **From the AIX® and Linux command line, to export a key, run ./exportKeys.sh *<filename> <password>*.** The script is installed to *ITM_dir*/*platform*/iw/scripts. Examples: /opt/IBM/ITM/aix533/iw/scripts on AIX, /opt/IBM/ITM/li6263/iw/scripts on Linux, and/opt/IBM/ITM/ls3263/iw/scripts on zLinux.
- **From the AIX and Linux command line, to import a key, run ./importKeys.sh *<filename> <password>*.** The script is installed to *ITM_dir*/*platform*/iw/scripts.
- **From the TEPS/e administration console, complete the following procedure to export the LTPA key:**
  1. Select **Security→ Global Security**.
  2. Select **LTPA**.
  3. In the **Password** and **Confirm password** fields, enter the password to encrypt the key file.
  4. In the **Fully qualified key file name** field, enter fully qualified path and file name for the key file.
  5. Click **Export keys**.
  6. Click **OK** and then **Save**.
- **From the TEPS/e administration console, complete the following procedure to import the LTPA key:**
  1. Select **Security→ Global Security**.
  2. Select **LTPA**.

3. In the **Password** and **Confirm password** fields, enter the password to decrypt the key file.
4. In the **Fully qualified key file name** field, enter fully qualified path and file name for the key file.
5. Click **Import keys**.
6. Click **OK** and then **Save**.
7. Restart the Tivoli Enterprise Portal Server.

   Note: When the portal server is restarted, the TEPS/e administration console is disabled automatically. You must re-enable it before it can be used again by following the instructions in "Starting the TEPS/e administration console" on page 114.

What to do next

If you exported the portal server's LTPA key, provide the key file and password that you used to encrypt it to the administrators of the other participating SSO applications so that they can import the key.

Managing new LDAP users

Whenever new users are added to the portal server's LDAP user registry and those users need to have login access to Tivoli Enterprise Portal or other participating SSO application such as, IBM Dashboard Application Services Hub, you must create a Tivoli Enterprise Portal user ID for the user and map it to their LDAP distinguished name.

The Tivoli Enterprise Portal user ID should also be assigned Tivoli Enterprise Portal permissions and the monitoring applications that can be accessed. See "Managing user IDs" on page 169 and "Administer Users" on page 164. The only Tivoli Enterprise Portal users who do not need any permissions or monitoring application assignments, are monitoring dashboard users who do not use the Tivoli Enterprise Portal client when authorization policies are used.

Note: The first time a dashboard user accesses monitoring data, a Tivoli Enterprise Portal user ID is automatically created for the user if there is not already a user ID mapped to the user's LDAP distinguished name. In this case, the Tivoli Enterprise Portal user ID is a randomly generated ID and the user is not assigned any permissions. If Tivoli Enterprise Portal permissions are being used to control access to monitored resources in the dashboards instead of authorization policies, or if the dashboard user can launch the Tivoli Enterprise Portal, assign the user ID permissions and the monitored applications that can be accessed.

Scripting can be employed to maintain automated synchronization of LDAP user registry and Tivoli Enterprise Portal users. Scripts for managing the LDAP server's user accounts can ensure that modifications to user accounts (for example, users added or deleted) are also made for the corresponding Tivoli Enterprise Portal user ID via the **`tacmd createuser`** and **`tacmd deleteuser`** commands. Run your user synchronization script as a scheduled action as frequently as your environment requires to ensure your Tivoli Enterprise Portal and LDAP user registry users remain synchronized.

Disabling LDAP authentication on the portal server

You might need to disable LDAP authentication to the portal server if errors occur.

About this task

If the LDAP connection is broken and the normal procedure to switch off LDAP-based authentication does not work, use the following procedure.

Procedure

Windows

1. Stop the portal server service by using the Manage Tivoli Enterprise Monitoring Services application.
2. Run the **`disableLDAPRepository.bat`** script from *`candle_home`*`\CNPSJ\` scripts.
3. Reconfigure the portal server by using the Manage Tivoli Enterprise Monitoring Services application and disable the "Validate User with LDAP" option.
4. Start the portal server service by using the Manage Tivoli Enterprise Monitoring Services application. The portal server authentication through the monitoring server is now enabled.
5. If the monitoring server was also configured to use LDAP, and you are using this procedure because the LDAP is out of service, you must also change the monitoring server configuration to not use LDAP to authenticate. Complete these configuration changes by using the monitoring server configuration help.

AIX Linux

1. Stop the portal server by issuing the **`./itmcmd agent stop cq`** command at a command prompt from the installation directory.
2. Run the **`./disableLDAPRepository.sh`** script from *`candle_home`*`/`*`arch`*`/iw/` scripts, where *`arch`* is the machine architecture, for example `li6263` or `aix533`.
3. Reconfigure the portal server and disable LDAP authentication by issuing the **`./itmcmd config -A cq`** command at a command prompt from the installation directory.
4. Start the portal server by issuing the **`./itmcmd agent start cq`** command at a command prompt from the installation directory. The portal server authentication through the monitoring server is now enabled.
5. If the monitoring server was also configured to use LDAP, and you are using this procedure because the LDAP is out of service, you must also change the monitoring server configuration to not use LDAP to authenticate. Complete these configuration changes by using the monitoring server configuration help.

Migrating LDAP authentication from the monitoring server to the portal server

If your environment has already been configured for LDAP authentication using the hub monitoring server and you now want to configure the portal server to use an LDAP user registry for single sign-on, complete the steps in this topic.

Before you begin

Make sure that all users log off the Tivoli Enterprise Portal before you begin the procedure and do not log on again until the procedure is completed.

About this task

Complete these steps to temporarily disable security validation on the hub monitoring server, configure the portal server to use an LDAP user registry, map Tivoli Enterprise Portal user IDs to the distinguished name of the LDAP user registry, and then re-enable security validation on the hub monitoring server.

Procedure

1. Temporarily disable Tivoli Enterprise Monitoring Server security validation:
   - Windows Use the Manage Tivoli Enterprise Monitoring Servicesutility to reconfigure the hub monitoring server:
     a. Right-click the Tivoli Enterprise Monitoring Server and click **Reconfigure** .
     b. On the Tivoli Enterprise Monitoring Server Configuration window, disable □ **Security: Validate User** and click **OK**.
     c. Click **OK** to accept the existing settings on the next window.
     d. Restart the hub monitoring server.
   - Linux UNIX From the command line:
     a. Change to the `/opt/IBM/ITM/bin` directory (or the directory where you installed Tivoli Management Services).
     b. Run the following command, where *`tems_name`* is the name of your monitoring server (for example, `HUB_itmdev17`):

        ```
        ./itmcmd config -S -t tems_name
        ```

     c. Press Enter to accept the existing values until you see the prompt for **Security: Validate User**.
     d. Enter `NO` to disable security.
     e. Continue to press Enter until the configuration is complete.
     f. Restart the hub monitoring server.
2. Rename the **sysadmin** UID in the LDAP registry (for example, sysadmin_tems).
3. Configure LDAP authentication and single sign-on for portal server. Use the Manage Tivoli Enterprise Monitoring Services utility, the **`itmcmd`** command line interface on Linux and UNIX, or the TEPS/e administration console to configure the portal server. For instructions, see "LDAP user authentication through the portal server" on page 99.
4. Start the Tivoli Enterprise Portal Server and log on to the Tivoli Enterprise Portal as **sysadmin**.
5. Adjust all user mappings to LDAP user IDs:
   a. Click **Administer Users** to open the Administer Users window.
   b. Right-click the row of the user ID to remap and click **Modify User**.
   c. Click **Find** to locate the LDAP distinguished name to be associated with the portal server.
   d. Select the distinguished name for the user. If you see multiple entries, select the one with the correct LDAP suffix (parent entry). Examples: `UID=TIVOLIUSER,O=MYCOMPANY` and `uid=myname, dc=tivoli, dc=ibm, dc=us`. If you see an entry with one of these organization values, do not choose it: `O=DEFAULTWIMITMBASEDREALM` is the default suffix for user IDs that authenticate through the hub monitoring server; and o=defaultWIMFileBasedRealm is the default for the TEPS/e user registry.
   e. Click **OK** to save the mapping and return to the Administer Users window, then continue to modify the DN for each user ID.

6. Before logging out of the Tivoli Enterprise Portal, have the LDAP administrator rename the LDAP sysadmin account back to **sysadmin**, then map the Tivoli Enterprise Portal sysadmin user account to the LDAP sysadmin DN.
7. Save your changes and log out of the Tivoli Enterprise Portal.
8. Re-enable Tivoli Enterprise Monitoring Server security validation by performing step 1 again, but this time enable security validation.

Results

At this point, the migration is complete.

What to do next

Verify the authentication changes by performing these steps:

1. Use the **`tacmd`** login command to verify that hub monitoring server security is enabled. Try logging in with a valid username and password and with a username or password that is not valid.
2. Login to the Tivoli Enterprise Portal using the sysadmin user.
3. Login to the Tivoli Enterprise Portal using a user from the LDAP user registry configured for the portal server.

Authentication through the Tivoli Enterprise Monitoring Automation Server

The Tivoli Enterprise Monitoring Automation Server extends the hub monitoring server by providing the Open Services Lifecycle Collaboration Performance Monitoring (OSLC-PM) service provider. The service provider registers monitoring resources such as computer systems, software servers, and databases with the Jazz for Service Management Registry Services component and also responds to HTTP GET requests for resource health metrics from OSLC clients.

By default the Performance Monitoring service provider does not authenticate HTTP GET requests from OSLC clients. If you want the Performance Monitoring service provider to authenticate these requests, you must install and configure the Security Services component of Jazz for Service Management. Security Services enables non-WebSphere based applications such as the Performance Monitoring service provider to participate in Lightweight Third Party Authentication (LTPA) based single sign-on. Registry Services and Security Services must be installed into the same WebSphere Application Server. They should also be configured to use the same LDAP user registry as the OSLC client applications and be configured for single sign-on.

Note: Registry Services and Security Services and the OSLC client applications must be in the same the Internet and Intranet domain, for example mycompany.com, or one of its sub-domains. They also must be configured to use the same realm name which is set when configuring a WebSphere Application Server to use a LDAP repository.

To configure Registry Services and Security Services for single sign-on, generate the LTPA key for the application server where they are installed, export the key, and then import the LTPA key into the OSLC client applications that will be sending HTTP GET requests to the Performance Monitoring service provider. See “Configuring Jazz for Service Management for a central user registry” and “Configuring Jazz for Service Management for SSO” in the Jazz for Service

Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html). These chapters contain instructions for configuring Registry Services and Security Services to use an LDAP user registry and generating and exporting their LTPA key.

The Performance Monitoring service provider must be configured to use Security Services to authenticate OSLC client requests by setting the Tivoli Enterprise Monitoring Automation Server `KAS_SECURITY_SERVICES_ENABLED` environment variable to `YES` and restarting the automation server.

When the Performance Monitoring service provider receives a HTTP GET request from an OSLC client, it forwards the LTPA token to Security Services to authenticate the request. If the request does not contain a LTPA token, or Security Services indicates that the token is not valid or has expired, the Performance Monitoring service provider returns a HTTP 401 status code to indicate that the request could not be authenticated.

Note: The Performance Monitoring service provider uses basic authentication when it sends requests to Registry Services so LTPA tokens are not involved in the service provider's resource registration interactions.

LDAP user authentication using Microsoft Active Directory

Use these topics to assist you in setting up user authentication using Microsoft's LDAP-based Active Directory product.

These topics cover the steps that you must complete to incorporate LDAP as implemented in an Active Directory environment, while presenting the procedures from an Active Directory perspective. Two user scenarios (one illustrating monitoring server integration with Active Directory, the other portal server integration with Active Directory) are provided to show you how this process can help you implement Tivoli Monitoring security in the working environment; see “User scenarios” on page 147.

This procedure uses the TEPS/e Web browser interface to complete the portal server configuration; see “Using the TEPS/e administration console” on page 113.

Note:

1. Configuring the portal server to use an LDAP server to authenticate users has the advantage, that it allows user IDs longer than 10 characters, a limit that is imposed by monitoring server authentication. It also supports SSO (single sign-on), which monitoring server authentication does not.

   Only monitoring server-based user authentication allows user IDs to make SOAP Server requests or to issue CLI commands that invoke SOAP Server methods.
2. The configuration procedures and steps for enabling IBM Tivoli Monitoring LDAP user authentication are the same for all LDAP implementations (Active Directory, Tivoli Directory Server, and so on), but the configuration values you specify will vary. These differences are due to the differences within the LDAP implementations themselves. The most pronounced differences are the syntax for Distinguished Names of objects within the directory. Additionally, the LDAP schema differences between LDAP implementations and any LDAP schema customizations will have a high impact on the LDAP user authentication configuration values provided.

3. Although the scenarios in this set of topics assumes a Microsoft Active Directory version 2003 environment, these instructions and scenarios have also been verified using Active Directory Server 2008 and Active Directory Server 2008 R2.

The configuration uses all information that is provided to connect, bind, query, and filter records from a specified LDAP Base to the targeted LDAP user registry for user authentication. The configurations of the monitoring server and portal server LDAP user authentication are separate operations; these configurations (after completion) can be enabled and disabled independently. Do not consider that the steps for configuring the monitoring server's LDAP user authentication translates to the portal server's LDAP user authentication, nor vice versa.

Before you begin

You must have a working Active Directory environment and be familiar with the following Active Directory concepts:

- Organizational Units

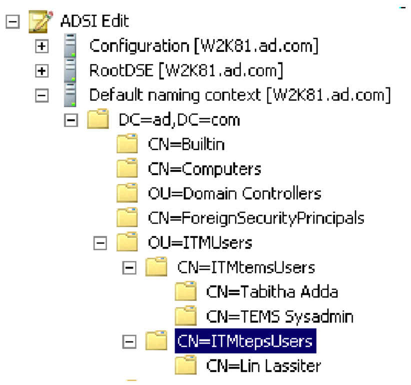
  ADSI Edit MMC snap-in
- Group Policy Management
- User Administration

  Active Directory User Object Schema

You must have installed both the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server as explained in the *IBM Tivoli Monitoring Installation and Setup Guide*. Familiarize yourself with the introductory information in Chapter 5, “Enabling user authentication,” on page 89.

You should work with your site's Active Directory administrator when deciding which LDAP users will be authorized for monitoring server or portal server authentication.

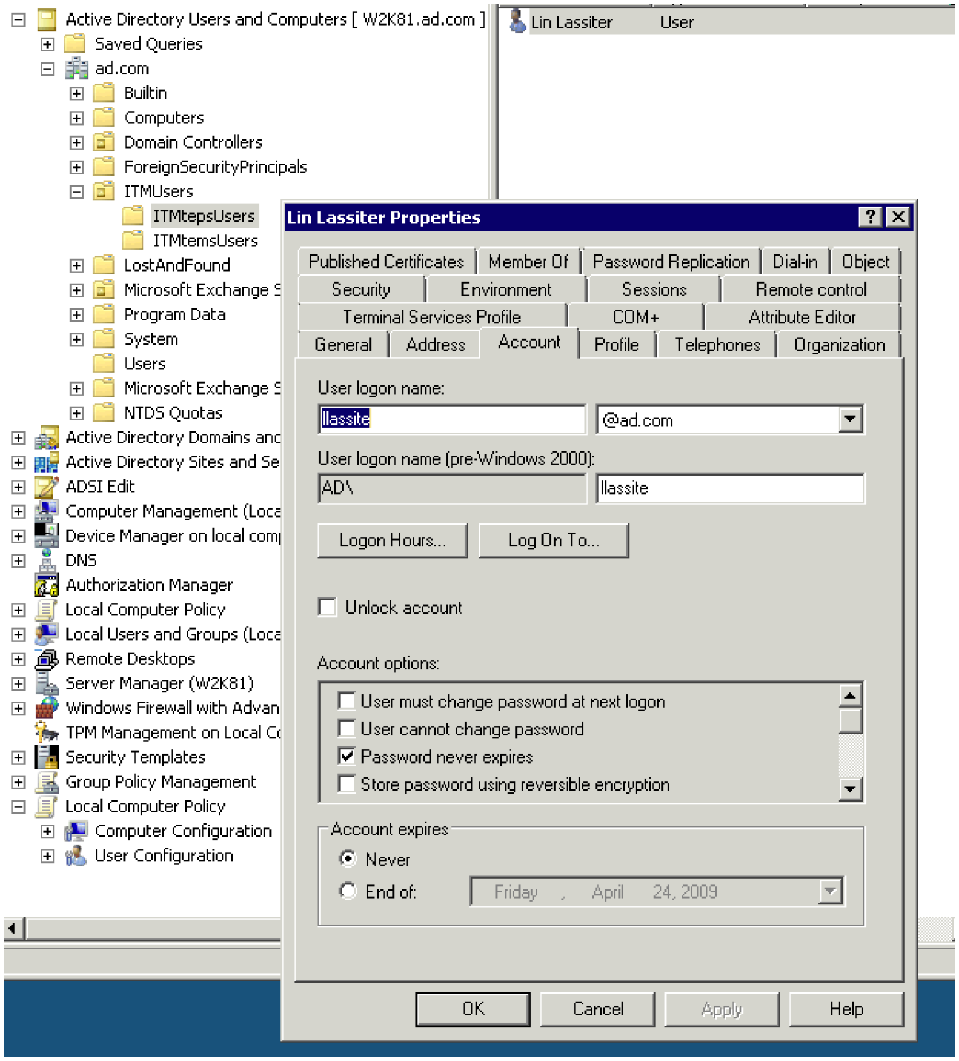
Best practice is that you also create an OU hierarchy that will contain your users. This will facilitate a Base name directory search and limit search time while increasing the performance of Tivoli Monitoring-to-LDAP user authentication. Figure 3 on page 131 shows a sample configuration comprising an `OU=ITMUsers` hierarchy with containers `ITMtepsUsers` and `ITMtemsUsers`. With this schema, the base for searching for monitoring server users to authenticate will be `CN=ITMtemsUsers,OU=ITMUsers,DC=`*`company`*`,DC=com`, and the base for portal server users to authenticate will be `CN=ITMtepsUsers,OU=ITMUsers,DC=`*`company`*`,DC=com`.



```
⊟ ADSI Edit
  ⊞ Configuration [W2K81.ad.com]
  ⊞ RootDSE [W2K81.ad.com]
  ⊟ Default naming context [W2K81.ad.com]
    ⊟ DC=ad,DC=com
        CN=Builtin
        CN=Computers
        OU=Domain Controllers
        CN=ForeignSecurityPrincipals
      ⊟ OU=ITMUsers
        ⊟ CN=ITMtemsUsers
            CN=Tabitha Adda
            CN=TEMS Sysadmin
        ⊟ CN=ITMtepsUsers
            CN=Lin Lassiter
```

Figure 3. Suggest LDAP user hierarchy for Tivoli Monitoring servers

You also need to be aware of your Active Directory user object/attribute schema. This information is required when coding your monitoring server's LDAP filter configuration and for the portal server's TEPS/e Repository Security login property. Figure 4 on page 132 shows one user's possible account settings (this Tivoli Enterprise Portal Server user must also be authorized as a Tivoli Enterprise Monitoring Server user).



User logon name: llassite @ad.com

User logon name (pre-Windows 2000): AD\ llassite

□ Unlock account

Account options:

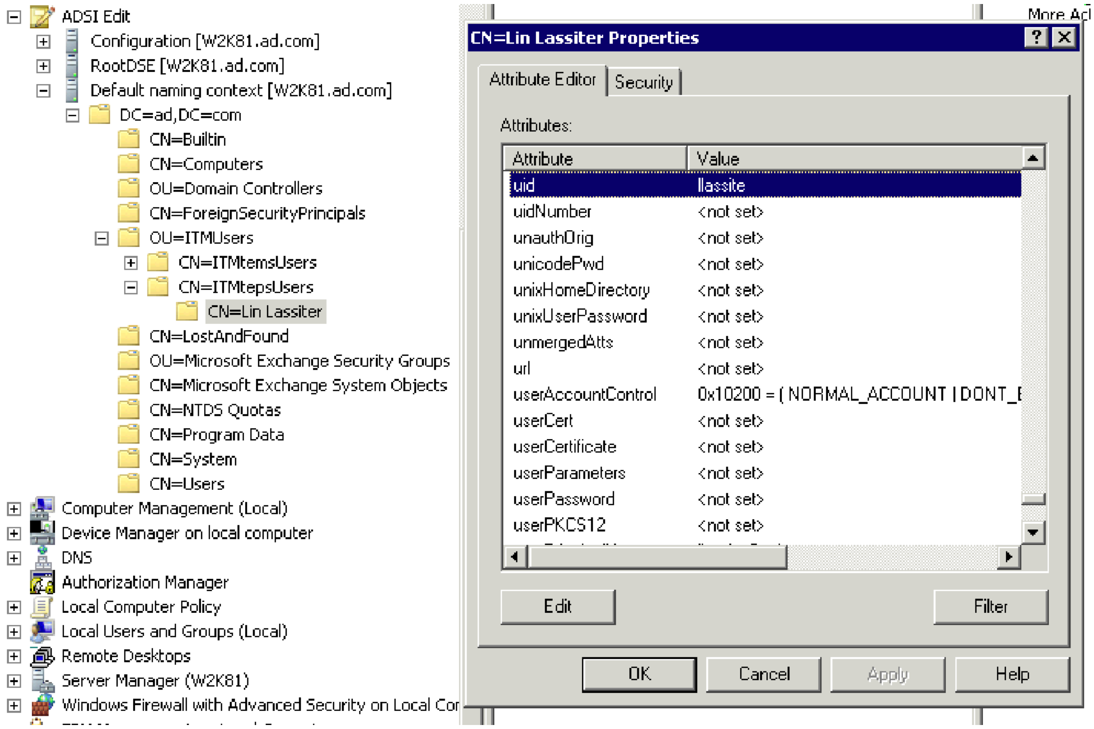
□ User must change password at next logon
□ User cannot change password
☑ Password never expires
□ Store password using reversible encryption

Account expires

● Never
○ End of: Friday , April 24, 2009

Figure 4. Portal server user properties

The configuration for TEPS/e LDAP user authentication requires that you specify Active Directory user object attribute Login Property, which will contain the matching user name (in this example, *llassite*). Figure 5 on page 133 shows the Active Directory user class instance for user *llassite*.



Attribute	Value
uid	llassite
uidNumber	<not set>
unauthOrig	<not set>
unicodePwd	<not set>
unixHomeDirectory	<not set>
unixUserPassword	<not set>
unmergedAtts	<not set>
url	<not set>
userAccountControl	0x10200 = (NORMAL_ACCOUNT | DONT_E
userCert	<not set>
userCertificate	<not set>
userParameters	<not set>
userPassword	<not set>
userPKCS12	<not set>

Figure 5. LDAP user properties

You must make the TEPS/e **uid** LDAP user authentication property match the portal server's user account. To do this, edit the Active Directory's user/uid attribute for the *llassite* user, and set **uid=llassite** so the portal server's user account *llassite* will match *uid=llassite* in the `CN=Lin Lassiter,CN=ITMtepsUsers,OU=ITMUsers,DC=`*`company`*`,DC=com` LDAP object (which can be found by searching the directory beginning with the `CN=ITMtepsUsers,OU=ITMUsers,DC=`*`company`*`,DC=com` base record).

Figure 3 on page 131, Figure 4 on page 132, and Figure 5 are provided to give you an idea of the Active Directory properties that will be used for LDAP authentication. The knowledge of where LDAP users reside within Active Directory (the Base to query or search for Tivoli Monitoring users in the directory) and the User schema (the user object attribute that contains the exact user name used for authentication) are critical to successful configuration of either Tivoli Enterprise Monitoring Server or Tivoli Enterprise Portal Server LDAP user authentication.

Note: The portal server's user account's permissions for such Tivoli Monitoring features as applications, views, and groups will continue to be managed within the portal server's User Administration tool, as shown in Figure 6 on page 134.

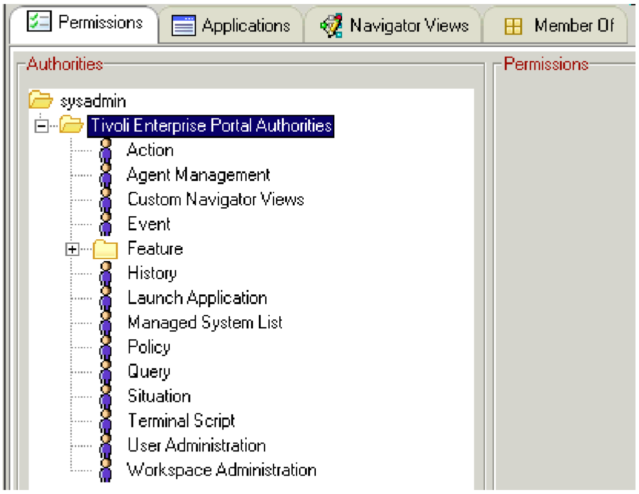


Figure 6. Tivoli Enterprise Portal Server user permissions

LDAP user authentication is available only for individual Tivoli Monitoring users and user groups. The enablement of LDAP authentication for individual Tivoli Monitoring users ensures maximum flexibility on both the IBM Tivoli Monitoring and LDAP sides. Scripting can be employed to maintain automated synchronization of Active Directory and Tivoli Monitoring users. Data-collection scripts for Active Directory user accounts can ensure that modifications to Active Directory accounts (for example, users added or deleted) are reflected back into the corresponding Tivoli Enterprise Portal users via the **tacmd** CLI..

Roadmap overview

To integrate your IBM Tivoli Monitoring user authentication environment with your site's Active Directory implementation, complete the steps outlined in this topic.

1. “Plan and create monitoring server and portal server users within Active Directory” on page 135

   Follow the steps outlined in this topic.
2. “Create and configure the portal server user accounts and permissions, if desired” on page 135

   Skip this step: If you do not want to use an LDAP server to authenticate Tivoli Enterprise Portal users and you do not need to configure single sign-on for integration with other products such as IBM Dashboard Application Services Hub.

   Use either the Tivoli Enterprise Portal Administer Users interface or the **tacmd** command-line interface, if your site requires portal server user authentication.

   These user accounts will use LDAP for authentication; thus, the userids chosen must exactly match those specified in Active Directory. For the attributes you should choose, see Figure 5 on page 133.
3. “Enable and configure LDAP user authentication for the portal server, if desired” on page 136

   Skip this step: If you do not want to use an LDAP server to authenticate Tivoli Enterprise Portal users and you do not need to configure single sign-on for integration with other products such as IBM Dashboard Application Services Hub.

Complete this step if your site requires portal server authentication.

4. “Configure TEPS/e for TLS/SSL, if necessary” on page 143

   Typically, TEPS/e is TLS/SSL enabled by default. If you need to configure TLS/SSL communication between the portal server and the LDAP server use the TEPS/e administration console.

5. “Enable and configure LDAP user authentication for the monitoring server, if desired” on page 143

   Skip this step: If you do not want to use an LDAP user to authenticate your monitoring server users.

   Complete this step if your site requires monitoring server authentication.

Plan and create monitoring server and portal server users within Active Directory

When creating Active Directory users for either the Tivoli Enterprise Monitoring Server or the Tivoli Enterprise Portal Server, do the following:

1. (Optional) Create the OU hierarchy for monitoring server and portal server users.

   See Figure 3 on page 131. Use the Microsoft Management Console's (MMC) snap-in **ADSI Edit**.

2. Create the monitoring server and portal server users (and, optionally, groups) in Active Directory.

   See Figure 4 on page 132. Use the MMC **AD Users and Computers** function.

3. Apply the desired User/Group Policies to the new Active Directory users and groups.

   Use the MMC snap-in for GPO.

No user synchronization currently exists between Tivoli Monitoring and LDAP. User accounts can be synchronized with scripting. Use Active Directory scripting for maintaining an awareness of User account modifications (limited to the OU that applies to IBM Tivoli Monitoring). These detected modifications can then be made to the Tivoli Monitoring users via the CLI **tacmd** command and to the Active Directory users with scripting. You will need to run your user synchronization script as a scheduled action as frequently as your environment requires to ensure your Tivoli Monitoring and Active Directory users remain synchronized.

Create and configure the portal server user accounts and permissions, if desired

Skip this step: If you do not want to use an LDAP server to authenticate Tivoli Enterprise Portal users and you do not need to configure single sign-on for integration with other products such as IBM Dashboard Application Services Hub.

Each Active Directory account previously created requires a matching Tivoli Enterprise Portal user account. The Tivoli Enterprise Portal userid must exactly match the Active Directory's TEPS User Object attribute field planned for use within the TEPS/e configuration (see Figure 5 on page 133).

Configure all required permissions, applications, views, and groups for user account operations within IBM Tivoli Monitoring, see Chapter 6, “Using Tivoli Enterprise Portal user authorization,” on page 163 for more information. (Note that these user accounts' permissions, applications, views, and groups will not be available in Active Directory, nor will they translate from Tivoli Monitoring to Active Directory; see Figure 6 on page 134.)

Note: You could update Active Directory's User Object schema to map the IBM Tivoli Monitoring user permissions, applications, views, and groups into Active Directory. Then you can leverage these new schema attributes to assist you both with user synchronization between Tivoli Monitoring and Active Directory and with Active Directory's management of portal server user properties via Active Directory scripting and the Tivoli Monitoring CLI's **`tacmd`** command.

It is not recommended that you add the default **sysadmin** account to your LDAP directory. The **sysadmin** account should be reserved for local monitoring server **Security: Validate User** authorization, thereby retaining a non-LDAP method for accessing the monitoring server and the portal server.

User ID and **User Description** are freeform, but for good form, you should attempt to match the **User Name** and **User Description** you already created in Active Directory.

The Distinguished Name is critical to binding the Tivoli Monitoring userid to the LDAP User account based on the TEPS/e LDAP configuration. This point is discussed further later; for now, select entry `UID=`*`userid`*`,O=DEFAULTWIMITMBASEDREALM`.

Enable and configure LDAP user authentication for the portal server, if desired

Skip this step: If you do not want to use an LDAP server to authenticate Tivoli Enterprise Portal users and you do not need to configure single sign-on for integration with other products such as IBM Dashboard Application Services Hub.

Now that the Tivoli Enterprise Portal userids have been created with the desired IBM Tivoli Monitoring permissions and these same userids exist within Active Directory, you must enable LDAP user authentication for these portal server users. Use the TEPS/e Web browser interface to set the portal server's LDAP user registry configuration details and the Active Directory Base name configuration. The complete procedure for TEPS/e LDAP configuration is detailed in "Configuring the portal server for LDAP authentication using the TEPS/e administration console" on page 115.

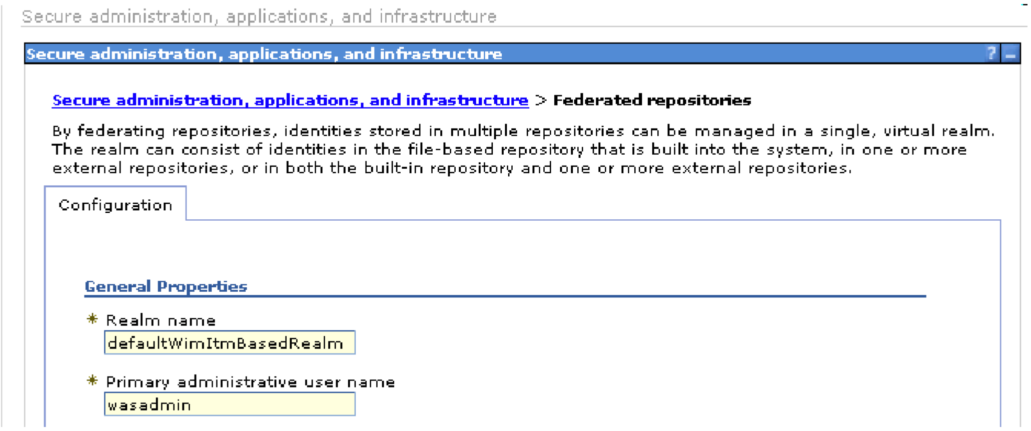
Starting the TEPS/e administration console

If you are having issues accessing `http://localhost:15205/ibm/console`, ensure you have completed the TEPS/e enablement and password-setting steps detailed in "Starting the TEPS/e administration console" on page 114. Also, if you wish to use the TEPS/e interface, you must enable it each time you restart the portal server. (Note that enabling TEPS/e controls only the TEPS/e interface; it does not control the portal server's LDAP enablement.) Log information for this activity can be found here:

- Windows `%CANDLE_HOME%\CNPSJ\profiles\ITMProfile\logs\ITMServer`
- Linux `$CANDLEHOME/$INTERP/iw/profiles/ITMProfile/logs/ITMServer`

When performing the steps described in "Configuring the portal server for LDAP authentication using the TEPS/e administration console" on page 115, observe the notes below.

- A realm identifies a set of federated repositories in TEPS/e and other WebSphere Application Servers. You can choose your own realm name but this value must be the same across all applications that are configured for SSO within an Internet or intranet domain. If you are not configuring SSO, you should accept the default for the **Realm name**, unless this name is already in use in your environment.


Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

Configuration

General Properties

* Realm name
defaultWimItmBasedRealm

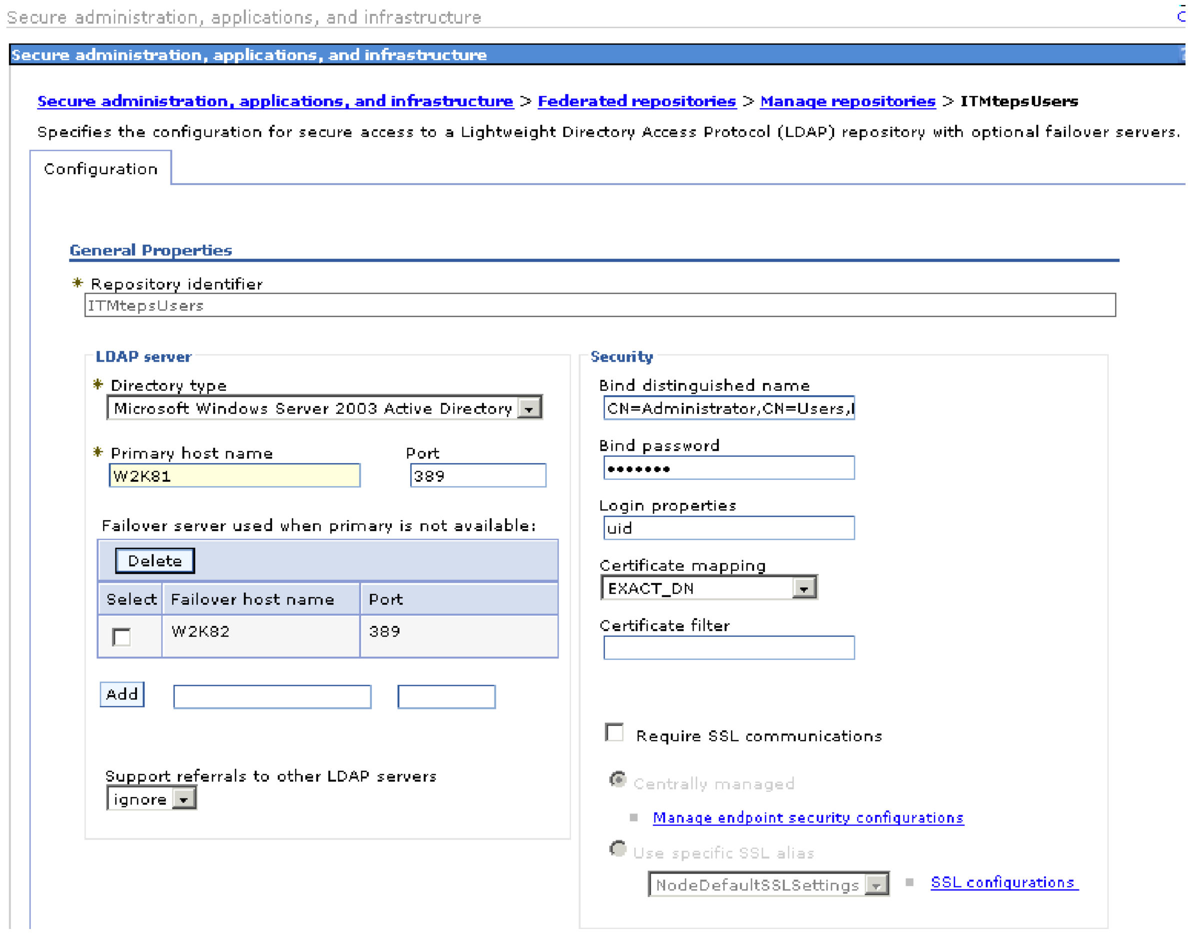
* Primary administrative user name
wasadmin

Figure 7. Accept these default values

- When entering your **Repository identifier** choose a descriptive name.

  The **Repository identifier** will become a reference label for a configuration container within TEPS/e that will hold your LDAP server information, your **LDAP Bind password**, and your **LDAP Login properties**; this is the Active Directory User Class attribute to search for an exactly matching portal server userid (see Figure 5 on page 133).

  Additionally, this Repository will be associated with one or more LDAP Base values (containers) that the portal server's LDAP authentication will search. Again, as pointed out in “Before you begin” on page 130, OU planning is recommended for this configuration step. Good OU planning restricts IBM Tivoli Monitoring LDAP user authentication searches to the Base and below. Grouping users within the ITMtepsUsers OU (see Figure 3 on page 131) is recommended for efficient Base searches and LDAP user authentication performance.



Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > Manage repositories > **ITMtepsUsers**

Specifies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover servers.

General Properties

* Repository identifier: ITMtepsUsers

LDAP server

* Directory type: Microsoft Windows Server 2003 Active Directory

* Primary host name: W2K81

Port: 389

Failover server used when primary is not available:

Select	Failover host name	Port
□	W2K82	389

Support referrals to other LDAP servers: ignore

Security

Bind distinguished name: CN=Administrator,CN=Users,I

Bind password: •••••••

Login properties: uid

Certificate mapping: EXACT_DN

Certificate filter:

□ Require SSL communications

● Centrally managed

Manage endpoint security configurations

○ Use specific SSL alias

NodeDefaultSSLSettings

SSL configurations

Figure 8. Configure the repository

General Property → Repository identifier

This is a freeform field. It is recommended that you choose a meaningful name here that administrators can easily associate to the users defined within your Active Directory environment. In this case ITMtepsUsers is specified, as this will be the only Repository created for portal server LDAP user authentication within our environment. It would also be appropriate to name your repository using a convention that reflects the **LDAP Bind password** account configured within. An example for this case may be to use a derivative of a Forest Name or a Domain Name if you will require different repositories to allow you to configure different **LDAP Bind password**s to meet your requirements.

LDAP Server Property → Directory type

The value provided here must match your Forest level. Our sample Forest is running at an Active Directory 2003 level.

LDAP Server Property → Primary host name

A Domain Controller within the Active Directory Forest that is hosting the User accounts you created earlier for portal server LDAP user authentication. Your choice here should be driven by the hierarchy level within your Forest that owns the IBM Tivoli Monitoring users' OU. Consider your selection here in light of possible issues with IBM Tivoli Monitoring LDAP authentication due to Active Directory connectivity or replication failures of your Active Directory User objects.

LDAP Server Property → Port
The Active Directory LDAP default port value is 389. Match this value to your LDAP environment's port assignment.

LDAP Server Property → Failover server used when primary is not available
Additional Active Directory LDAP (DC) servers can be added here to compensate for replication or connectivity issues. This value should be completed with failover DCs (preferably with GC roles).

LDAP Server Property → Support referrals to other LDAP Servers
This is your choice. Consider whether your environment is closed or open (that is, there are no DCs within your DMZ).

Security → Bind distinguished name
The user specified here requires sufficient authority (that is, an applied policy) for searching the Active Directory directory Base. This user is the account that will connect, bind, and query the specified **Login Properties**. This value can be omitted if anonymous users can search the registry.

Our example uses `CN=Administrator,CN=Users,DC=`*`company`*`,DC=com`, but this is not absolutely required if the user specified exists within the `CN=Users,DC=`*`company`*`,DC=com` container for Active Directory. However, this is not recommended, as it is better to be specific and use a complete Distinguished Name.

Security → Bind password
The password for the **Bind distinguished name** account above.

Security → Login properties
This configuration property is critical for Active Directory in that this is the name of the attribute within the Active Directory Name object that is used to reflect exactly the portal server's user name (see Figure 5 on page 133).

Security → Certificate
The default value **EXACT_DN** is set for searching Active Directory and matching on an exact DN. It is recommended you keep this default value.

Security → Certificate filter
Selecting **CERTIFICATE_FILTER** for the **Certificate** field above requires LDAP filter parameters. These parameters map attributes within the client certificate to entries within the LDAP directory.

Security → Require SSL communications
Select this if you desire TLS/SSL LDAP communications. Click the radio buttons for the following options that apply to your TLS/SSL implementation:
Centrally Managed
Use specific SSL alias

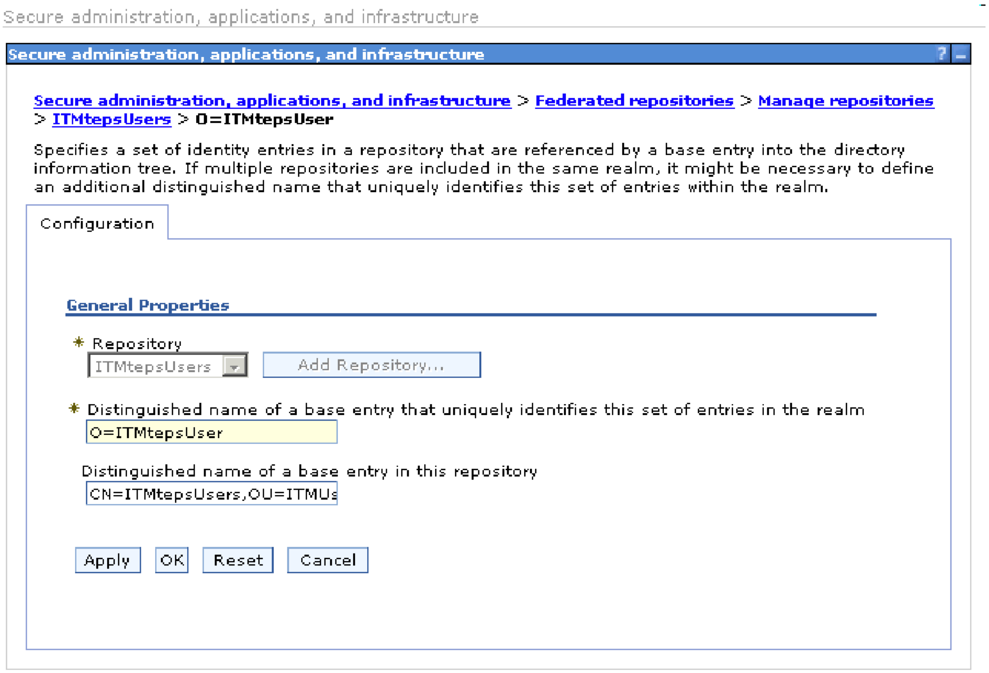
- When specifying a **Distinguished name of the base entry that uniquely identifies this set of entries in the realm** and **Distinguished name of the base entry in this repository** see Figure 9 on page 140 for reference.

Now that you have a Repository configured, you need to add the Base entry as a starting point when searching for your Active Directory-defined portal server users' OU. You can define multiple Base entries for your repository if you require. Multiple Base entries should be required only if you have defined multiple directory Base locations (multiple OU hierarchies) for your Active Directory's portal server users.

The information provided in this panel for the property **Distinguished name of a base entry that uniquely identifies this set of entries in the realm** will be reflected back into your Tivoli Enterprise Portal User Distinguished Name choices, as shown in Figure 12 on page 142. It is this field that is made available from within the TEPS/e portal server configuration to associate the portal server-defined userid to an LDAP connect, bind, query, or select (see the login properties from Step 5 on page 116).

This property is freeform; it is the TEPS/e configuration DN. It is recommended that you use the convention `O=`*`DNofChoice`*. Your choice of DN should be meaningful for its relationship to your Active Directory-defined TEPS Users Base container (shorter is better). The examples here have assigned `O=ITMtepsUser` for clarity when viewing the Tivoli Enterprise Portal users' Distinguished Names in the **Administer Users** interface (see Figure 12 on page 142). The example used here is redundant, since the Repository Name and the Base DN are very similar, but it is clear this Repository and Base are configured for accessing and binding our Active Directory-defined users.

For **Distinguished name of the base entry in this repository**, specify the distinguished name (DN) for the base entry in the LDAP registry. It is the starting point for user searches in the LDAP server.



Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > Manage repositories > ITMtepsUsers > O=ITMtepsUser

Specifies a set of identity entries in a repository that are referenced by a base entry into the directory information tree. If multiple repositories are included in the same realm, it might be necessary to define an additional distinguished name that uniquely identifies this set of entries within the realm.

Configuration

General Properties

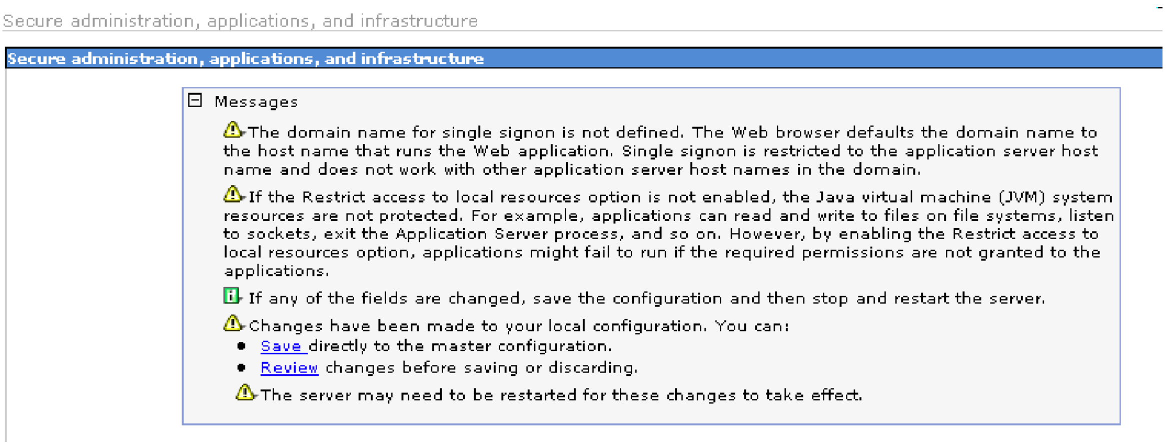
* Repository
ITMtepsUsers

* Distinguished name of a base entry that uniquely identifies this set of entries in the realm
O=ITMtepsUser

Distinguished name of a base entry in this repository
CN=ITMtepsUsers,OU=ITMUs

Figure 9. Adding the Base entry to your realm. Note that this entry is associated with the ITMtepsUsers section of the Repository.

- When saving your Step TEPS/e configuration work use Figure 10 on page 141 as a reference. Click **Save**.


Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure

⊟ Messages

⚠ The domain name for single signon is not defined. The Web browser defaults the domain name to the host name that runs the Web application. Single signon is restricted to the application server host name and does not work with other application server host names in the domain.

⚠ If the Restrict access to local resources option is not enabled, the Java virtual machine (JVM) system resources are not protected. For example, applications can read and write to files on file systems, listen to sockets, exit the Application Server process, and so on. However, by enabling the Restrict access to local resources option, applications might fail to run if the required permissions are not granted to the applications.

ℹ If any of the fields are changed, save the configuration and then stop and restart the server.

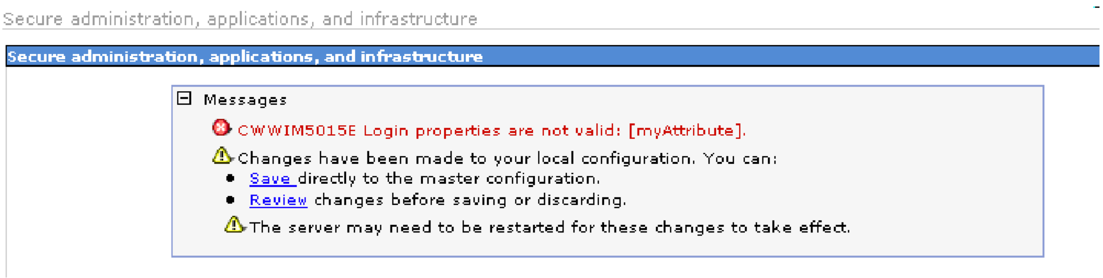
⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Figure 10. Save your TEPS/e configuration updates

Also, while configuring your Repository and Base settings, remember to click OK when asked to verify your configuration settings. Clicking OK or Apply connects, binds, and queries the Active Directory LDAP database using your updated configuration settings. If there is an issue, the configuration panel returns a red error message like the one shown in Figure 11.


Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure

⊟ Messages

⊗ CWWIM5015E Login properties are not valid: [myAttribute].

⚠ Changes have been made to your local configuration. You can:

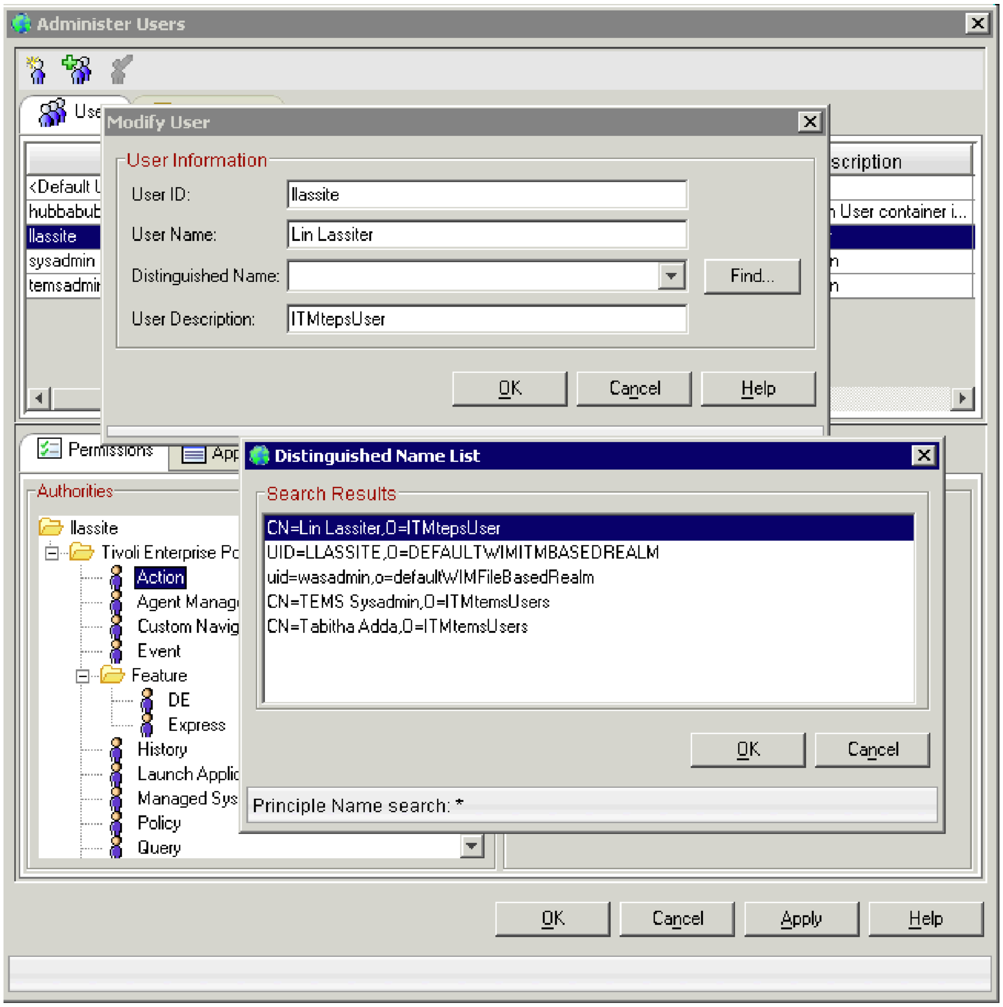
- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Figure 11. TEPS/e configuration error message

You must correct these errors before saving your configuration.

Note: Once the TEPS/e configuration is completed and saved, you must recycle the Tivoli Enterprise Portal Server to see the TEPS/e-configured users listed as Distinguished Names (see the example for *llassite* in Figure 12 on page 142).



User ID: llassite
User Name: Lin Lassiter
Distinguished Name:
User Description: ITMtepsUser

CN=Lin Lassiter,O=ITMtepsUser
UID=LLASSITE,O=DEFAULTWIMITMBASEDREALM
uid=wasadmin,o=defaultWIMFileBasedRealm
CN=TEMS Sysadmin,O=ITMtemsUsers
CN=Tabitha Adda,O=ITMtemsUsers

Principle Name search: *

Figure 12. Tivoli Enterprise Portal Administer Users screen. Distinguished Names resolve to the Base names configured within TEPS/e for the Repository; compare the values presented here with those shown in Figure 4 on page 132.

Mapping portal client userids to LDAP Distinguished Names

"Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 120 provides the steps required to associate your new portal server userids to the Distinguished Names made available by the previous TEPS/e configurations steps. The available Distinguished Names relate directly to the TEPS/e Repository-associated Base configuration property **Distinguished name of a base entry that uniquely identifies this set of entries in the realm**.

To display all available Distinguished Names, first delete any entry in the **Distinguished Name** field; then click the **Find** button. All configured Realm → Repository → Base Distinguished Names that uniquely identify this set of entries in the realm are displayed with the users that were returned by IBM Tivoli Monitoring when performing a query against Active Directory using the **Login Properties** value.

These users will be displayed using their Active Directory CN format (see Figure 12).

Option: The Active Directory LDAP configuration steps recommend the Tivoli Enterprise Portal userids be created prior to the configuration of TEPS/e.

Optionally, the creation of portal server userids can wait until after the Active Directory userids and TEPS/e configuration are completed. The advantage here is that you can use the Create User function in the **Administer Users** interface while having access to the Distinguished Names to ensure the userid you assign matches the available Distinguished Name.

Enable LDAP authentication within the portal server

"Using Manage Tivoli Enterprise Monitoring Services to configure the portal server for LDAP authentication" on page 107 and "Using the Linux or UNIX command line to configure the portal server for LDAP authentication" on page 111 provide the steps required to enable portal server user authentication using an LDAP user registry.

Note: When you are asked to select the LDAP type, select **Other** and do not supply values for any other LDAP parameters.

After you have finished configuring the portal server to use LDAP, refer to topic "Roadmap for setting up the portal server to use an LDAP user registry and single sign-on" on page 104 for additional steps to complete the configuration.

Configure TEPS/e for TLS/SSL, if necessary

Typically, TEPS/e is TLS/SSL enabled by default.

SSL communication between the portal server and the LDAP server is configured using the TEPS/e administration console. Follow the steps in "Configuring TLS/SSL communication between the portal server and the LDAP server" on page 119.

Enable and configure LDAP user authentication for the monitoring server, if desired

Skip this step: If you do not want to use an LDAP user to authenticate your monitoring server users.

User configuration for the Tivoli Enterprise Monitoring Server is completely separate from that for the Tivoli Enterprise Portal Server. TEPS/e is not involved.

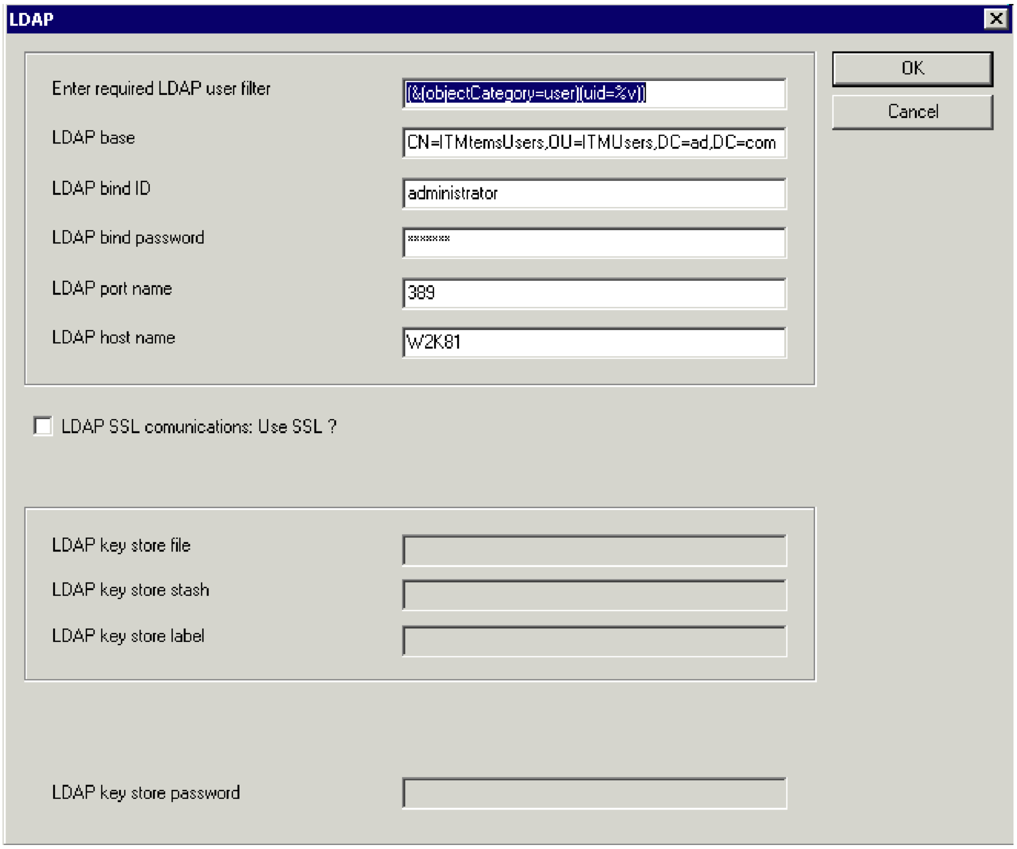
None of the portal server's LDAP configuration or enablement affects the monitoring server's LDAP configuration or enablement. Monitoring server users are not required to be created nor exist within the Tivoli Enterprise Portal **Administer Users** list of users. Monitoring server users are required only if you wish to create userids that can be authenticated using the **Security: Validate User** option or if you wish to enable or prohibit SOAP requests to the monitoring server's SOAP Server (see "Configuring Tivoli Monitoring Web Services (SOAP Server)" on page 555).

"User authentication through the hub monitoring server" on page 92 provides the steps required to enable LDAP user authentication for the Tivoli Enterprise Monitoring Server. Additional comments are provided here for specific steps within this process.

Note: The monitoring server's userids are limited to 10 characters, dictating that the Active Directory user names you choose also not exceed 10 characters.

The monitoring server's LDAP configuration allows only one LDAP Base and one LDAP User filter (to query the LDAP directory for userid attributes). OU planning is recommended for creating the Active Directory Base and OU hierarchy that best meets your requirements. Use a Base that limits directory subtree searches while maximizing Active Directory's LDAP user authentication performance (see Figure 3 on page 131).

- Step 5 on page 95: see Figure 13.



Enter required LDAP user filter: (&(objectCategory=user)(uid=%v))

LDAP base: CN=ITMtemsUsers,OU=ITMUsers,DC=ad,DC=com

LDAP bind ID: administrator

LDAP bind password: *******

LDAP port name: 389

LDAP host name: W2K81

□ LDAP SSL comunications: Use SSL ?

LDAP key store file:

LDAP key store stash:

LDAP key store label:

LDAP key store password:

Figure 13. LDAP configuration panel for monitoring server users

Enter required LDAP user filter

This defines the attribute that will be queried and collected for Tivoli Enterprise Monitoring Server LDAP authentication. The monitoring server ID used for login (**`tacmd login –s tems_name –u username –p password`**) will be checked against the matching, Active Directory-filtered User for authentication.

LDAP user filter

Example: `(&(objectCategory=user)(userPrincipalName=%v@`*`company`*`.com))` where `%v` is a variable that IBM Tivoli Monitoring replaces with the userid entered at login.

This filter queries Active Directory, collecting all User objects from the specified Base. The **userPrincipalName** attribute values returned by this query will be parsed against the string `%v@`*`company`*`.com`, causing a comparison of the monitoring server userid with only the `%v` substitution portion of the **userPrincipalName** (in this case, `userPrincipalName=llassite@`*`company`*`.com | userPrincipalName=%v@`*`company`*`.com == llassite`).

LDAP base
IBM recommends you enter an LDAP Base that gives you visibility to the OU container that contains your Active Directory-defined monitoring server users.

LDAP bind ID
IBM recommends you enter an LDAP ID that can access your Active Directory's OU hierarchy to locate your Active Directory-defined portal server users.

LDAP bind password
The LDAP bind ID's password.

LDAP port name
This value is set for the default Active Directory LDAP port. Enter your LDAP-configured port number.

LDAP host name
A Domain Controller within the Active Directory Forest that is hosting the User accounts you created earlier for monitoring server LDAP user authentication. Your choice here should be driven by the hierarchy level within your Forest that owns the Tivoli Monitoring users' OU. Consider your selection here in light of possible issues with IBM Tivoli Monitoring LDAP user authentication due to Active Directory connectivity or replication failures of your Active Directory User objects.

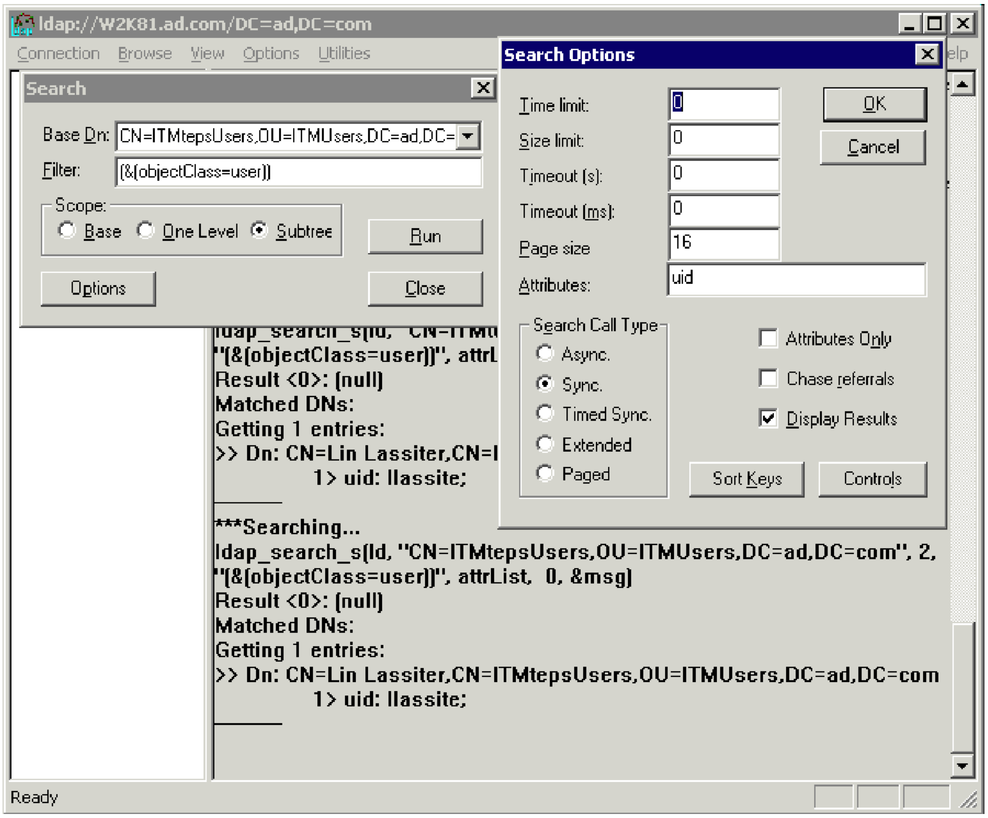
Active Directory LDAP verification tools

Microsoft Active Directory provides several tools for your use in managing your site's LDAP environment; the following two will prove particularly useful when linking it to IBM Tivoli Monitoring:

ADSI Edit
Use this Microsoft Management Console snap-in to view your user object attributes and to confirm that the attributes you are specifying for the Tivoli Enterprise Portal Server **Login properties** and the Tivoli Enterprise Monitoring Server **attributename=%v** substitution parameter are defined and available.

LDP.exe
Use this tool to validate your monitoring server and portal server LDAP configuration's Base settings. This tool allows you to connect, bind, and query your LDAP environment from your workstation; see Figure 14 on page 146.

LDP.exe for Windowx XP is available from Microsoft at this URL: http://www.microsoft.com/downloads/details.aspx?FamilyID=49ae8576-9bb9-4126-9761-ba8011fabf38&displaylang=en



Base Dn: CN=ITMtepsUsers,OU=ITMUsers,DC=ad,DC=

Filter: (&(objectClass=user))

Scope: ○ Base ○ One Level ● Subtree

Time limit: 0

Size limit: 0

Timeout (s): 0

Timeout (ms): 0

Page size: 16

Attributes: uid

Search Call Type: ○ Async. ● Sync. ○ Timed Sync. ○ Extended ○ Paged

□ Attributes Only

□ Chase referrals

☑ Display Results

```
ldap_search_s(ld, "CN=ITMt
"(&(objectClass=user))", attrL
Result <0>: (null)
Matched DNs:
Getting 1 entries:
>> Dn: CN=Lin Lassiter,CN=I
        1> uid: llassite;
-----------
***Searching...
ldap_search_s(ld, "CN=ITMtepsUsers,OU=ITMUsers,DC=ad,DC=com", 2,
"(&(objectClass=user))", attrList,  0, &msg)
Result <0>: (null)
Matched DNs:
Getting 1 entries:
>> Dn: CN=Lin Lassiter,CN=ITMtepsUsers,OU=ITMUsers,DC=ad,DC=com
        1> uid: llassite;
-----------
```

Figure 14. LDP query results

This sample demonstrates the verification of a configuration using:

```
LDAP filter object = (&(objectCategory=user)(uid=%v))
LDAP base = CN=ITMtemsUsers,OU=ITMUsers,DC=company,DC=com
```

Alternatively, this sample demonstrates verification of a configuration using:

```
LDAP base = CN=ITMtepsUsers,OU=ITMUsers,DC=company,DC=com
Login properties = uid
```

To successfully configure Microsoft Active Directory LDAP authentication, either you need the Domain Administrator or you need to get hold of two very useful tools that allow you to look at your LDAP directory from the outside. These tools are:

ldapsearch

Use this tool to test your connect strings from the command line and to verify that you are pointing at the right location inside the LDAP user registry. Figure 19 on page 152 shows sample `ldapsearch` output.

"Ldapsearch for LDAP information" on page 96 contains additional information about this command and its uses and options.

The `ldapsearch` options you specify (see "ldapsearch command-line options" on page 97) are based on your site's Tivoli Enterprise Monitoring Server LDAP configuration:

-h is the LDAP host name.

-p is the LDAP port name.

-b is the LDAP base value.

-D is the LDAP bind ID.

-w is the LDAP bind password.

Note: If you do not specify the -w option, you will be required to enter the LDAP bind password from the keyboard.

Always specify the `ldapsearch` -s *sub* option because the monitoring server's LDAP client uses it when authenticating Tivoli Monitoring users. Replace `%v` with the Tivoli Monitoring user ID when specifying the LDAP user filter (this string is the last part of the `ldapsearch` command line).

Example: To verify user `sysadmin` with the monitoring server LDAP configuration shown in Figure 18 on page 151, specify the following `ldapsearch` command:

```
ldapsearch -h 192.168.1.241 -p 389 -b "DC=bjomain,CN=users,DC=bjomain,
      DC=com"
           -D "CN=Administator,CN=users,DC=bjomain,DC=com" -w admin10admin
           -s sub "(mail=sysadmin@bjomain.com)"
```

Follow this link to download a free version of `ldapsearch`: http://publib.boulder.ibm.com/infocenter/wasinfo/v4r0/index.jsp?topic=/com.ibm.support.was40.doc/html/Security/swg21113384.html

ldapbrowser

Use this tool to graphically traverse the LDAP user registry and to spell out the Distinguished Names and other parameters that you need to complete the configuration. To verify that IBM Tivoli Monitoring can access your LDAP user registry across the network, install the LDAP browser on a Tivoli Monitoring server. Figure 17 on page 150 shows a sample `ldapbrowser` display.

The LDAP browser also enables you to retrieve LDAP information from the portal server itself.

Follow this link to download a free version of `ldapbrowser`: http://www.ldapbrowser.com/download.htm; then click the **LDAP Browser** tab. `ldapbrowser` is also available for both UNIX/Linux and Windows at this URL: http://www.mcs.anl.gov/~gawor/ldap/

User scenarios

In these scenarios it is desired that all user authentication be done via the site's Microsoft Server 2003 Active Directory LDAP user registry. There are two possible ways to configure this authentication for IBM Tivoli Monitoring users:

Monitoring server LDAP authentication

Configure authentication at the Tivoli Enterprise Monitoring Server using a username filter that maps a userid entered at the Tivoli Enterprise Portal (such as *name*) to `*name@company*.com`. The user logs in as *name*, which then gets translated through the filter to match the needed LDAP lookup.

This method is described in "Authenticating monitoring server userids with Microsoft Active Directory" on page 148.

Portal server LDAP authentication

Configure authentication at the Tivoli Enterprise Portal Server so that the userid the user supplies when logging into the Tivoli Enterprise Portal gets looked up and authenticated against the LDAP user registry. In this scenario, the user logs in with `*firstname.lastname@company*.com`.

This method is described in “Authenticating portal server userids with Microsoft Active Directory” on page 152.

Authenticating monitoring server userids with Microsoft Active Directory

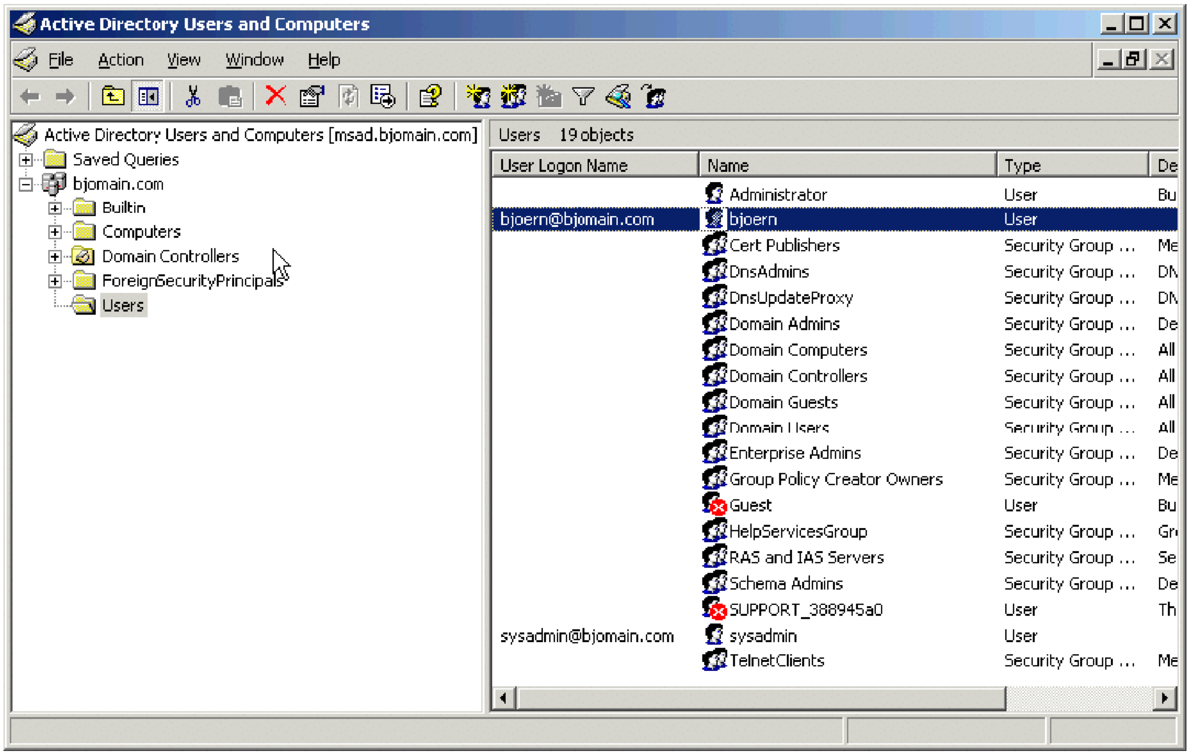
This scenario illustrates how you can configure the Tivoli Enterprise Monitoring Server to use Microsoft Active Directory to authenticate monitoring server users.

This scenario does not require TEPS/e configuration in order to work. The drawback with this solution is that SSO (single sign-on) cannot be implemented; also, userids are restricted to 10 characters maximum. The advantage is that monitoring server-based user authentication allows your users to make SOAP Server requests and use **tacmd** command that send requests to the hub monitoring server.

Environment

The environment comprises two systems, one running the Tivoli Monitoring monitoring server (IP address 192.168.1.240) and one running Microsoft Windows 2003 Advanced Server configured as a Microsoft Active Directory domain controller (IP address 192.168.1.241). Note that the Tivoli Monitoring system is a stand-alone server called **itm6210** and not part of the configured domain. The sample domain is called **bjomain.com**, and the Active Directory server is called **msad**.

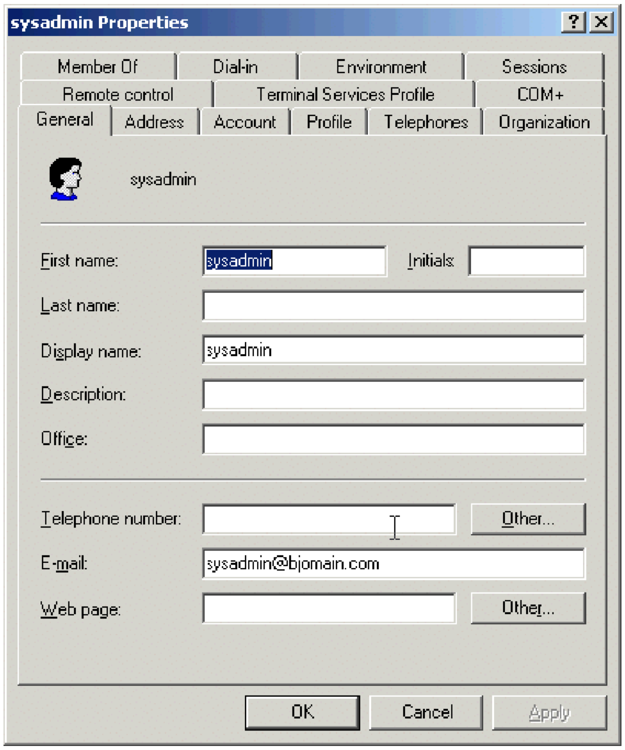
Microsoft Active Directory configuration:


Active Directory Users and Computers
File
Action
View
Window
Help
Active Directory Users and Computers [msad.bjomain.com]
Saved Queries
bjomain.com
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Users
Users 19 objects

User Logon Name	Name	Type	De
	Administrator	User	Bu
bjoern@bjomain.com	bjoern	User	
	Cert Publishers	Security Group ...	Me
	DnsAdmins	Security Group ...	DN
	DnsUpdateProxy	Security Group ...	DN
	Domain Admins	Security Group ...	De
	Domain Computers	Security Group ...	All
	Domain Controllers	Security Group ...	All
	Domain Guests	Security Group ...	All
	Domain Users	Security Group ...	All
	Enterprise Admins	Security Group ...	De
	Group Policy Creator Owners	Security Group ...	Me
	Guest	User	Bu
	HelpServicesGroup	Security Group ...	Gr
	RAS and IAS Servers	Security Group ...	Se
	Schema Admins	Security Group ...	De
	SUPPORT_388945a0	User	Th
sysadmin@bjomain.com	sysadmin	User	
	TelnetClients	Security Group ...	Me

Figure 15. Active Directory users listing

As shown in Figure 15, two users have been configured in Active Directory, *sysadmin* and *bjoern*, both with email addresses, as shown in Figure 16 on page 149 (you will see when the LDAP filter is configured in IBM Tivoli Monitoring why the email address is important). You can use other parameters, but this is the one the instructions in “User authentication through the hub monitoring server” on page 92 recommend.



sysadmin

First name: sysadmin

Initials:

Last name:

Display name: sysadmin

Description:

Office:

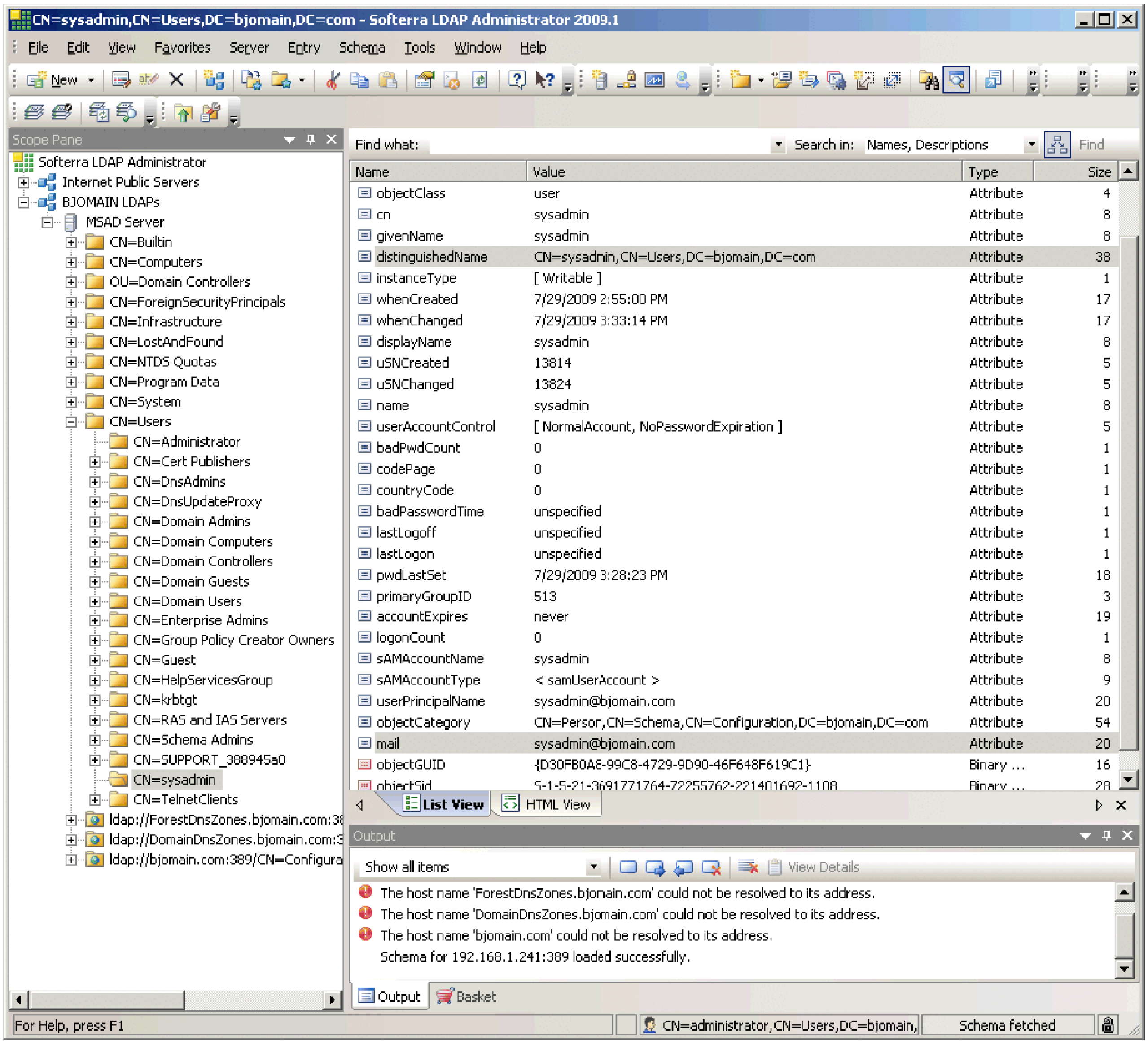
Telephone number:

E-mail: sysadmin@bjomain.com

Web page:

Figure 16. Properties of an individual Tivoli Monitoring user

Browsing Active Directory: Browsing the Active Directory Repository with the GUI browser, `ldapbrowser`, shows all the parameters you need, such as the Distinguished Name and the email address for the *sysadmin* user (see Figure 17 on page 150).



Name	Value	Type	Size
objectClass	user	Attribute	4
cn	sysadmin	Attribute	8
givenName	sysadmin	Attribute	8
distinguishedName	CN=sysadnin,CN=Users,DC=bjomain,DC=com	Attribute	38
instanceType	[Writable]	Attribute	1
whenCreated	7/29/2009 2:55:00 PM	Attribute	17
whenChanged	7/29/2009 3:33:14 PM	Attribute	17
displayName	sysadmin	Attribute	8
uSNCreated	13814	Attribute	5
uSNChanged	13824	Attribute	5
name	sysadmin	Attribute	8
userAccountControl	[NormalAccount, NoPasswordExpiration]	Attribute	5
badPwdCount	0	Attribute	1
codePage	0	Attribute	1
countryCode	0	Attribute	1
badPasswordTime	unspecified	Attribute	1
lastLogoff	unspecified	Attribute	1
lastLogon	unspecified	Attribute	1
pwdLastSet	7/29/2009 3:28:23 PM	Attribute	18
primaryGroupID	513	Attribute	3
accountExpires	never	Attribute	19
logonCount	0	Attribute	1
sAMAccountName	sysadmin	Attribute	8
sAMAccountType	< samUserAccount >	Attribute	9
userPrincipalName	sysadmin@bjomain.com	Attribute	20
objectCategory	CN=Persor,CN=Schema,CN=Configuration,DC=bjomain,DC=com	Attribute	54
mail	sysadmin@bjomain.com	Attribute	20
objectGUID	{D30FB0AE-99C8-4729-9D90-46F648F619C1}	Binary ...	16
objectSid	S-1-5-21-3691771764-72255762-221401692-1108	Binary ...	28

```
The host name 'ForestDnsZones.bjomain.com' could not be resolved to its address.
The host name 'DomainDnsZones.bjomain.com' could not be resolved to its address.
The host name 'bjomain.com' could not be resolved to its address.
Schema for 192.168.1.241:389 loaded successfully.
```

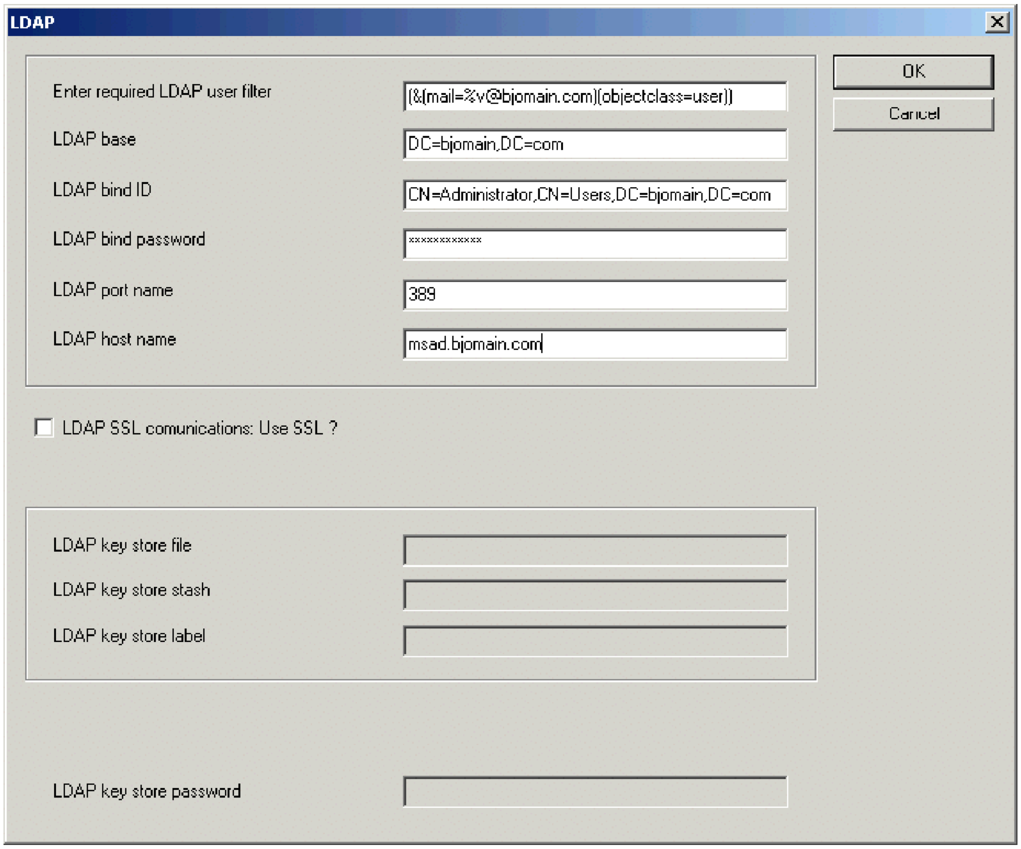
Figure 17. ldapbrowser window

By right-clicking **MSAD Server** at the top of the tree and selecting **Properties**, you can see that the Base DN (the point where Tivoli Monitoring will start searching for users) is **DC=bjomain,DC=com**.

The next step is to match this information with the appropriate fields in the Tivoli Enterprise Monitoring Server configuration dialog.

Putting the pieces together: Figure 18 on page 151 shows the monitoring server's LDAP settings that allow you to log in as either *sysadmin* or *bjoern* (only these users are defined to the monitoring server).

Note: If you need to activate Secure Sockets Layer, SSL, security for your Tivoli Monitoring-to-Active Directory communications, see Chapter 8, “Securing communications,” on page 209. Also ensure you have at hand the parameter values listed in Table 12 on page 94.



Enter required LDAP user filter	(&(mail=%v@bjomain.com)(objectclass=user))
LDAP base	DC=bjomain,DC=com
LDAP bind ID	CN=Administrator,CN=Users,DC=bjomain,DC=com
LDAP bind password	************
LDAP port name	389
LDAP host name	msad.bjomain.com

□ LDAP SSL comunications: Use SSL ?

LDAP key store file	
LDAP key store stash	
LDAP key store label	
LDAP key store password	

Figure 18. Monitoring server's LDAP parameters

The following are some of the more important parameters shown in Figure 18:

`Enter required LDAP user filter`
This parameter says to search for the **mail** parameter within the User object.

This is why you included the email address in the user's Active Directory entry.

`%v` Is a variable that Tivoli Monitoring replaces with the userid entered on the login screen.

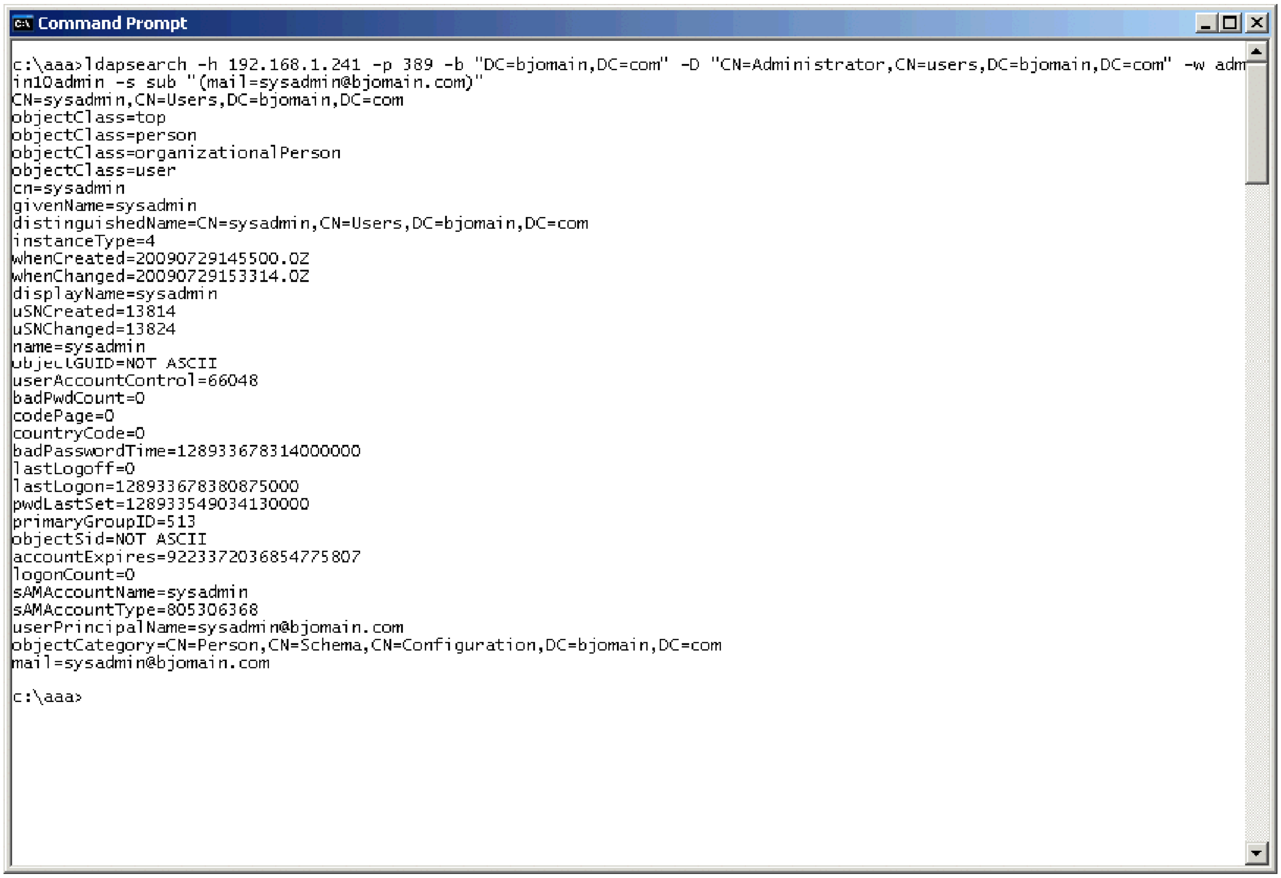
`LDAP base`
Is the complete Base DN listed in "Browsing Active Directory" on page 149.

If IBM Tivoli Monitoring complains that the user entered the wrong password, this is a sign that the wrong LDAP Base DN was specified here, in which case Tivoli Monitoring starts its search at the wrong LDAP location.

`LDAP bind ID`
Enter the Distinguished Name for a user that has read permission to the entire Base DN where Tivoli Monitoring will begin searching for its users.

Note: It is not enough to enter only the user name, for example, *sysadmin*.

Once you have gotten your parameters defined right, use the **grep** command to search for the string `LDAP` in the monitoring server's log file to verify that there are no error messages. Optionally, you can use the `ldapsearch` utility to test your parameters without starting the monitoring server: if `ldapsearch` does not return output similar to that shown in Figure 19 on page 152, your input is incorrect. You should verify your site's LDAP parameters before restarting the monitoring server, as an incorrect LDAP configuration will prevent users from logging in.



```
c:\aaa>ldapsearch -h 192.168.1.241 -p 389 -b "DC=bjomain,DC=com" -D "CN=Administrator,CN=users,DC=bjomain,DC=com" -w adm
in10admin -s sub "(mail=sysadmin@bjomain.com)"
CN=sysadmin,CN=Users,DC=bjomain,DC=com
objectClass=top
objectClass=person
objectClass=organizationalPerson
objectClass=user
cn=sysadmin
givenName=sysadmin
distinguishedName=CN=sysadmin,CN=Users,DC=bjomain,DC=com
instanceType=4
whenCreated=20090729145500.0Z
whenChanged=20090729153314.0Z
displayName=sysadmin
uSNCreated=13814
uSNChanged=13824
name=sysadmin
objectGUID=NOT ASCII
userAccountControl=66048
badPwdCount=0
codePage=0
countryCode=0
badPasswordTime=128933678314000000
lastLogoff=0
lastLogon=128933678380875000
pwdLastSet=128933549034130000
primaryGroupID=513
objectSid=NOT ASCII
accountExpires=9223372036854775807
logonCount=0
sAMAccountName=sysadmin
sAMAccountType=805306368
userPrincipalName=sysadmin@bjomain.com
objectCategory=CN=Person,CN=Schema,CN=Configuration,DC=bjomain,DC=com
mail=sysadmin@bjomain.com

c:\aaa>
```

Figure 19. ldapsearch results for monitoring server userids

Authenticating portal server userids with Microsoft Active Directory

In this scenario, you configure the portal server to use server to authenticate users because you want to use single signon with other applications or you want your Tivoli Enterprise Portal users to login with userids longer than 10 characters.

The site attempted to use the portal sever's Manage Tivoli Enterprise Monitoring Services utility or `itmcmd` command line interface to configure the LDAP authentication. This proved unsuccessful because the built-in authentication mechanism expects to look up an LDAP field named **uid**; this customer's Active Directory LDAP records have no **uid** field.

The rest of this section documents the user authentication steps you should follow via the TEPS/e Administration Console, the portal sever's built-in eWAS server that defines the custom LDAP user mappings for Tivoli Enterprise Portal access at this company.

Required LDAP environment information

To authenticate users against the customer's Active Directory LDAP user registry, several pieces of information are required:

1. The type and location of the LDAP information store.

2. The retrieval method for that information—notably whether or not to use SSL.
3. A **Bind ID** and **Password**, a userid/password combination that the system will use to log into the LDAP store and look up user accounts.

   This ID must be in full LDAP Distinguished Name format.
4. The **Login Properties** to use—namely which LDAP field to look up. This field needs to be something within the LDAP information store that uniquely identifies a user in the environment.

   The instructions in "LDAP user authentication through the portal server" on page 99 assume you have the **uid** field available; this proved not to be the case with this customer's LDAP directory.
5. An **LDAP Base**, the full LDAP Distinguished Name of a Base entry in the LDAP user registry.

In this example, your site's Active Directory administrator provided the following LDAP information:

1. LDAP type: Microsoft Active Directory Server version 2003; location: Hostname `adhost.`*`company`*`.com`
2. Port to use: 636 (which indicates that SSL is required for connection)
3. Bind ID of `svc.tivolisec@`*`company`*`.com`, along with the appropriate password
4. The **Login Properties** field to use: the user's **email address**
5. The **LDAP Base**: `DC=US,DC=GLOBAL,DC=`*`company`*`,DC=COM`

Despite having all the necessary connection information, every attempt to connect to the LDAP user registry fails. You can use the LDAP utilities described in "Active Directory LDAP verification tools" on page 145 to look up and verify your connection information and determine why every connection attempt failed.

In this example, two utilities are used: `LDP.exe` and `ldapbrowser`. These utilities show you that SSL communication is not required in this customer's environment; thus, connecting at the normal unencrypted LDAP port of 389 is valid. The tools also reveal that the full LDAP Distinguished Name associated with the `svc.tivolisec@`*`company`*`.com` address is `CN=svc.tivolisec,OU=ServiceAccount,DC=us,DC=global,DC=`*`company`*`,DC=com`.

Once all these entries have been validated, it is time to use the TEPS/e Administration (eWAS) tools to define the LDAP lookup parameters for Tivoli Monitoring user administration.

Enable TEPS/e administration: Enabling the TEPS/e administration (eWAS) console, the Integrated Solutions Console, requires that you complete the following steps.

Note: Step 1 need be done only once. The remaining steps must be done every time you wish to use the TEPS/e administration console.

Define the `wasadmin` password: Before you can complete the TEPS/e administration, you must set a password for the `wasadmin` account that admits you to the eWAS server. To do this, either invoke script `updateTEPSEPass.sh` in the `$CANDLEHOME/iw/scripts` directory or use the Manage Tivoli Enterprise Monitoring Services interface.

This procedure need be done only once, unless at a later time you desire to change the `wasadmin` user password.

Via the command line

To define the wasadmin password via the command line, invoke script updateTEPSEPass.sh in the $CANDLEHOME/$INTERP/iw/scripts directory:

```
# cd $CANDLEHOME/$INTERP/iw/scripts
# ./updateTEPSEPass.sh wasadmin newpw
WASX7209I: Connected to process "ITMServer" on node ITMNode using SOAP connector;
The type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable: "[wasadmin, newpw]"
```

Via Manage Tivoli Enterprise Monitoring Services

1. Right-click the **Tivoli Enterprise Portal Server** entry, and select menu option **TEPS/e Administration** → **Update TEPS Extension Password**..
2. In the **Enter Password** window supply the new wasadmin password, and press **OK**.

   The effect of your attempted password change is displayed in the Manage Tivoli Enterprise Monitoring Services message pane. For example:

   ```
   Password for user wasadmin was changed
   ```

Enable ISCLite (TEPS/e eWAS server administration): Anytime you need to manage your site's LDAP authentication, you first need to enable TEPS/e administration via the Integrated Solutions Console. Note that whenever you either reconfigure or stop then restart the Tivoli Enterprise Portal Server, the TEPS/e console is automatically disabled.

Via the command line

To enable the TEPS/e console via the command line, invoke the enableISCLite.sh script from the $CANDLEHOME/*platformcode*/iw/scripts subdirectory on the portal server machine:

```
# pwd /apps/TEPS_s11154cdc/li6263/iw/scripts

# ./enableISCLite.sh true
WASX7209I: Connected to process "ITMServer" on node ITMNode using SOAP connector;
The type of process is: UnManagedProcess

WASX7303I: The following options are passed to the scripting environment and are
available as arguments that are stored in the argv variable: "[true]"

ISCLite started
```

Via Manage Tivoli Enterprise Monitoring Services

1. Right-click the **Tivoli Enterprise Portal Server** entry, and select menu option **TEPS/e Administration** → **Enable TEPS/e Administration**.
2. The effect of your enablement attempt is shown in the Manage Tivoli Enterprise Monitoring Services message pane. For example:

   ```
   ISCLite is enabled successfully
   ```

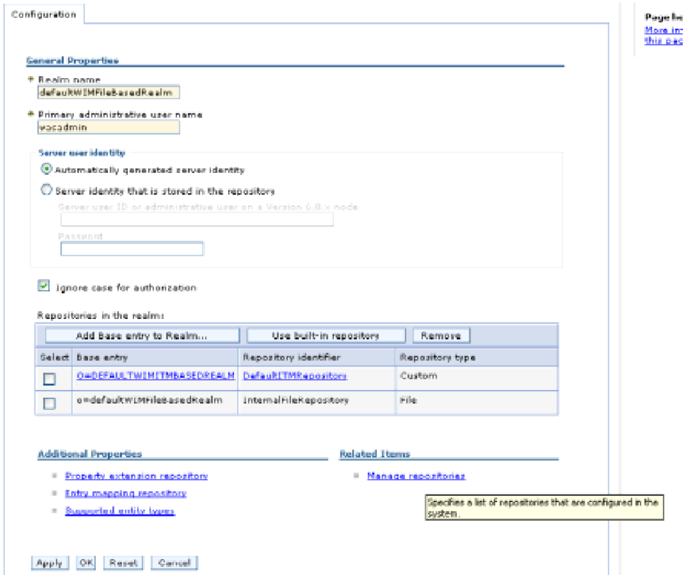
Log into the TEPS/e administration console: Bring up the eWAS Integrated Solutions Console in your browser using this address:

```
http://tepsserver.company.com:15205/ibm/console
```

Log in using the username wasadmin and the password you set for that user in “Define the wasadmin password” on page 153.

Define the LDAP user registry in the Integrated Solutions Console: When using the Integrated Solutions Console to define the LDAP user registry, follow these steps:

1. On the left side of the primary Integrated Solutions Console screen, expand the list of **Security** options, and select **Global security**. The **Global security** panel is displayed.
2. Under the **User account repository** section, click **Configure**.
3. From the **Configuration** tab, click **Manage Repositories** at the bottom under **Related Items** to open the screen where you can define your LDAP user registry:



Configuration

General Properties

* Realm name
defaultWIMFileBasedRealm

* Primary administrative user name
vscadmin

Server user identity

● Automatically generated server identity

○ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

☑ Ignore case for authorization

Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
□	O=DEFAULTWIMITMBASEDREALM	DefaultITMRepository	Custom
□	o=defaultWIMFileBasedRealm	InternalFileRepository	File

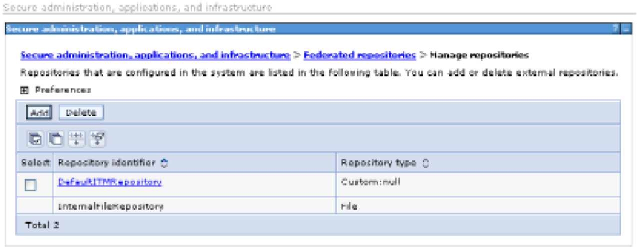
Additional Properties
- Property extension repository
- Entry mapping repository
- Supported entity types

Related Items
- Manage repositories

Specifies a list of repositories that are configured in the system.

Figure 20. The Integrated Solutions Console Configuration notebook tab

4. On the **Manage repositories** screen, click **Add**:



Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > Manage repositories

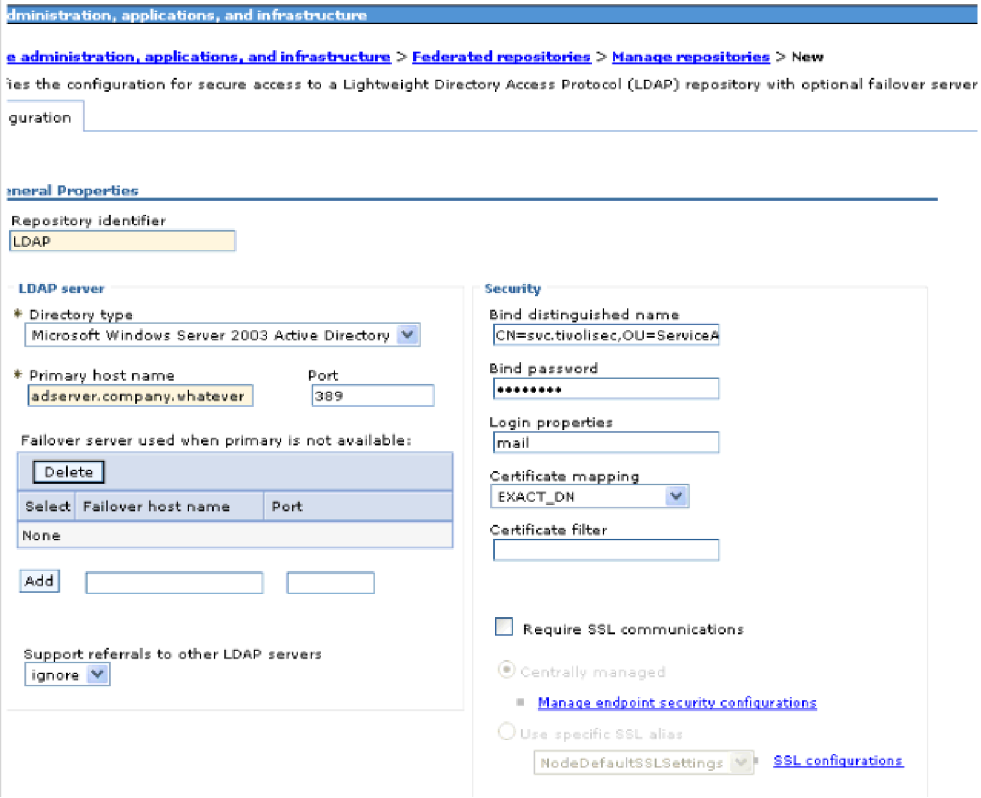
Repositories that are configured in the system are listed in the following table. You can add or delete external repositories.

⊞ Preferences

Select	Repository identifier	Repository type
□	DefaultITMRepository	Custom:null
	InternalFileRepository	File
Total 2		

Figure 21. The Integrated Solutions Console Manage repositories screen

The **General Properties** screen, shown in Figure 22 on page 156, is displayed. This is where you supply the information that defines the location and configuration of your LDAP user registry.



dministration, applications, and infrastructure

e administration, applications, and infrastructure > Federated repositories > Manage repositories > New

ies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover server

guration

neral Properties

Repository identifier: LDAP

LDAP server

* Directory type: Microsoft Windows Server 2003 Active Directory

* Primary host name: adserver.company.whatever

Port: 389

Failover server used when primary is not available:

Select	Failover host name	Port
None		

Support referrals to other LDAP servers: ignore

Security

Bind distinguished name: CN=svc.tivolisec,OU=ServiceA

Bind password: ••••••••

Login properties: mail

Certificate mapping: EXACT_DN

Certificate filter:

□ Require SSL communications

● Centrally managed

Manage endpoint security configurations

○ Use specific SSL alias

NodeDefaultSSLSettings

SSL configurations

Figure 22. The Integrated Solutions Console General Properties screen

5. Complete this screen with the following information:

Repository identifier
A freeform name for the registry, in this case simply **LDAP**.

Primary host name
Hostname of the LDAP server, in this case `adhost.`*`company`*`.com`.

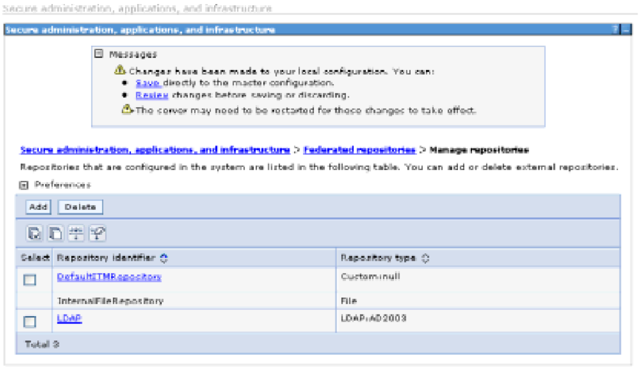
Port
Port the LDAP Server is listening on. In this example, 389 is the valid value.

Bind distinguished name
The full LDAP Distinguished Name of the Bind ID. In this case, the full LDAP Distinguished Name for the `svc.tivolisec` account that your site's LDAP administrator supplied is `CN=svc.tivolisec,OU=ServiceAccount,DC=us,DC=global,DC=`*`company`*`,DC=com`.

Bind password
The password for that Bind ID.

Login properties
The login properties to use for that Distinguished Name, in this case the **mail** property.

When you have completed this page, click **OK**. The verification screen is shown:


Secure administration, applications, and infrastructure
Secure administration, applications, and infrastructure
Preferences
Add
Delete

⊟ Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Secure administration, applications, and infrastructure > Federated repositories > Manage repositories

Repositories that are configured in the system are listed in the following table. You can add or delete external repositories.

Select	Repository identifier	Repository type
□	DefaultITMRepository	Custom:null
	InternalFileRepository	File
□	LDAP	LDAP:AD2003
Total 3		

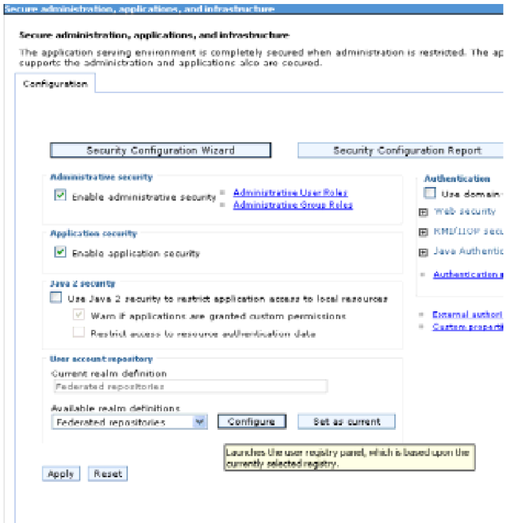
Figure 23. The Integrated Solutions Console verification screen

6. Click **Save**.

Your site's LDAP user registry is now defined.

Add your LDAP user registry to the eWAS realm: The next step is to add the newly defined registry to the eWAS realm so your site can use LDAP to look up userids.

1. On the left side of the primary Integrated Solutions Console screen, under Security options select **Global Security**.
2. In the **User account repository** section, click **Configure** (at the bottom beside Federated Repositories).


Secure administration, applications, and infrastructure
Configuration
Security Configuration Wizard
Security Configuration Report
Administrative User Roles
Administrative Group Roles
Authentication
Web security
RMI/IIOP secu
Java Authentic
Authentication m
External authori
Custom properti
Configure
Set as current
Apply
Reset

Secure administration, applications, and infrastructure

The application serving environment is completely secured when administration is restricted. The ap supports the administration and applications also are secured.

Administrative security

☑ Enable administrative security

Application security

☑ Enable application security

Java 2 security

□ Use Java 2 security to restrict application access to local resources

☑ Warn if applications are granted custom permissions

□ Restrict access to resource authentication data

□ Use domain-

User account repository

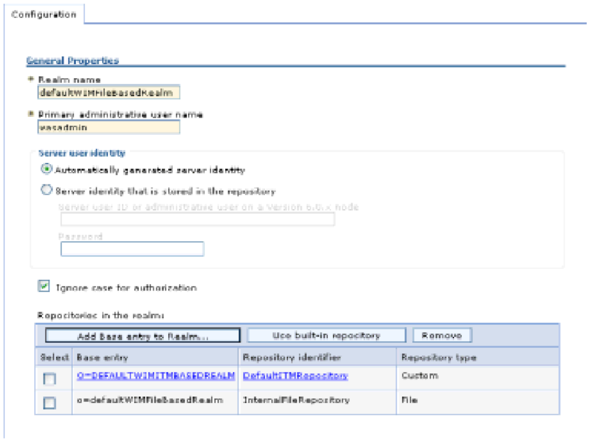
Current realm definition: Federated repositories

Available realm definitions: Federated repositories

Launches the user registry panel, which is based upon the currently selected registry.

Figure 24. The Integrated Solutions Console Configuration notebook tab

This opens the screen that lets you add registries to the realm.



* Realm name: defaultWIMFileBasedRealm

* Primary administrative user name: wasadmin

Server user identity

● Automatically generated server identity

○ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

☑ Ignore case for authorization

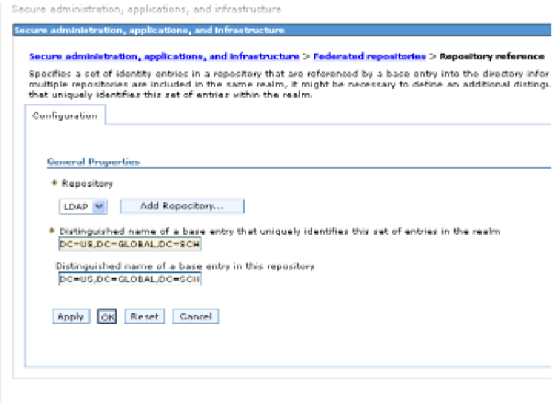
Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
□	O=DEFAULTWIMITMBASEDREALM	DefaultITMRepository	Custom
□	o=defaultWIMFileBasedRealm	InternalFileRepository	File

Figure 25. The Integrated Solutions Console's Configuration tab

3. To add the repository defined in “Define the LDAP user registry in the Integrated Solutions Console” on page 155, click **Add Base entry to Realm**.

   The **Repository reference** screen is shown, where you can add the LDAP user registry to your site's eWAS realm:



Secure administration, applications, and infrastructure
Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > Repository reference

Specifies a set of identity entries in a repository that are referenced by a base entry into the directory infor multiple repositories are included in the same realm, it might be necessary to define an additional disting that uniquely identifies this set of entries within the realm.

* Repository: LDAP

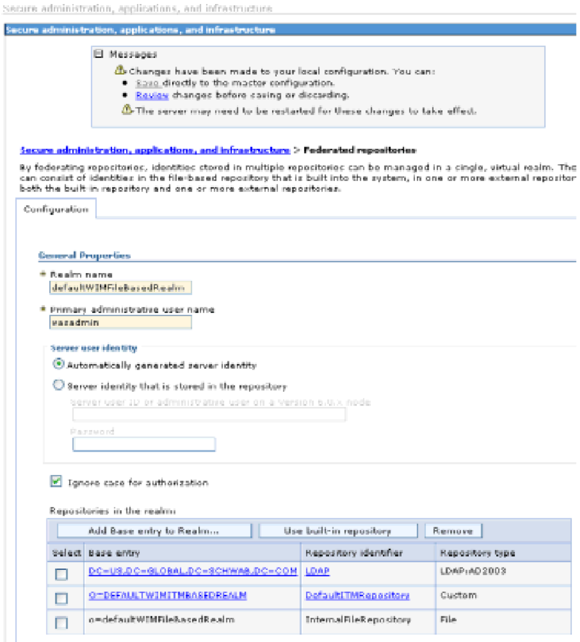
* Distinguished name of a base entry that uniquely identifies this set of entries in the realm: DC=US,DC=GLOBAL,DC=SCH

Distinguished name of a base entry in this repository: DC=US,DC=GLOBAL,DC=SCH

Figure 26. The Integrated Solutions Console's Repository reference screen

   At this screen, ensure that **Repository** is set to **LDAP** (or whatever **Repository identifier** you assigned in “Define the LDAP user registry in the Integrated Solutions Console” on page 155). In the two entry fields, enter the **Bind distinguished name**, which was defined in this instance to be `DC=US,DC=GLOBAL,DC=`*`company`*`,DC=COM`. Then click **OK**.

4. From the Integrated Solutions Console verification screen, click **Save**.


Secure administration, applications, and infrastructure
Configuration
Add Base entry to Realm...
Use built-in repository
Remove

Secure administration, applications, and infrastructure

Messages

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

Secure administration, applications, and infrastructure > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The can consist of identities in the file-based repository that is built into the system, in one or more external repositor both the built in repository and one or more external repositories.

General Properties

* Realm name
defaultWIMFileBasedRealm

* Primary administrative user name
wasadmin

Server user identity

● Automatically generated server identity

○ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

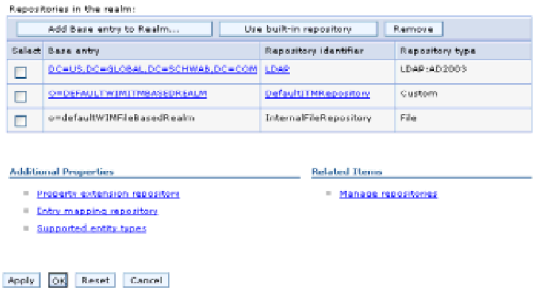
☑ Ignore case for authorization

Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
□	DC=US,DC=GLOBAL,DC=SCHWAB,DC=COM	LDAP	LDAP:AD2003
□	O=DEFAULTWIMITMBASEDREALM	DefaultITMRepository	Custom
□	o=defaultWIMFileBasedRealm	InternalFileRepository	File

Figure 27. The Integrated Solutions Console verification screen

5. This returns you to the list of registries in the current realm.



Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
□	DC=US,DC=GLOBAL,DC=SCHWAB,DC=COM	LDAP	LDAP:AD2003
□	O=DEFAULTWIMITMBASEDREALM	DefaultITMRepository	Custom
□	o=defaultWIMFileBasedRealm	InternalFileRepository	File

Additional Properties

- Property extension repository
- Entry mapping repository
- Supported entity types

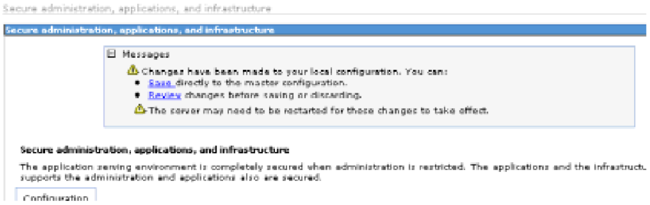
Related Items

- Manage repositories

Figure 28. The Integrated Solutions Console's Repositories in the realm screen

Click **OK**.

6. From the Integrated Solutions Console verification screen, click **Save**:



Secure administration, applications, and infrastructure

Messages

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

Secure administration, applications, and infrastructure

The application serving environment is completely secured when administration is restricted. The applications and the infrastructu supports the administration and applications also are secured.

Figure 29. The Integrated Solutions Console verification screen

7. This returns you to the initial Integrated Solutions Console sign-in screen:

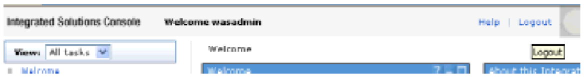


Figure 30. The Integrated Solutions Console's sign-in screen

Click **Logout**.

8. Restart the Tivoli Enterprise Portal Server.

(Optionally) test the LDAP lookup within TEPS/e: You can test the LDAP lookup within the TEPS/e console itself. If the lookup works correctly here, it will work within the Tivoli Enterprise Portal Server.

1. Enable the TEPS/e console again (see "Enable ISCLite (TEPS/e eWAS server administration)" on page 154), and then log back into it using your `wasadmin` userid and newly assigned password (see "Log into the TEPS/e administration console" on page 154).

   The initial TEPS/e screen is displayed:

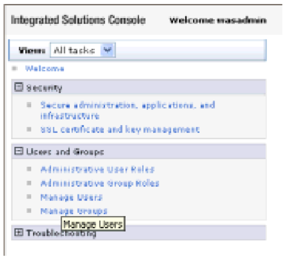

Integrated Solutions Console
Welcome wasadmin
View: All tasks
Welcome
Security
Secure administration, applications, and infrastructure
SSL certificate and key management
Users and Groups
Administrative User Roles
Administrative Group Roles
Manage Users
Manage Groups
Manage Users
Troubleshooting

Figure 31. The Integrated Solutions Console initial screen

2. Expand the list of **Users and Groups**; then select **Manage Users**.
3. Within the **Manage Users** pane, set **Search by** to **E-mail**, and specify your test userid (that is, email address) in the **Search for** field.
4. Click **Search**.

If the email address you specified is found, its characteristics are listed at the bottom of the **Manage Users** pane.

Define a test userid using portal server user administration:

1. Log into a Tivoli Enterprise Portal client using userid `sysadmin`. Note that `sysadmin` is still defined as a local portal server ID (in other words, it is not stored within or retrieved from your LDAP user registry).
2. Using Tivoli Enterprise Portal user administration (as explained in "LDAP user authentication through the portal server" on page 99), create a new user with any userid you wish.
3. In the Distinguished Name field, enter enough of that person's email address to make your specification unique; then click the **Find** button.

   The LDAP search is performed, and the full LDAP Distinguished Name is found for that email address.
4. Highlight that Distinguished Name, and click **OK**.
5. Complete that user's remaining userid fields, and then click **OK** to add it.

That person can now log into the Tivoli Enterprise Portal client using his or her email address, *longemailaddress@customer*.com, as the userid and his/her Active Directory password for the password. The user will be logged into his/her default primary workspace.

Chapter 6. Using Tivoli Enterprise Portal user authorization

Every portal work session begins with a successful logon and connection to the Tivoli Enterprise Portal. The logon user IDs and user groups are created and profiled through the Administer Users window.

Administer Users is a multi-tabbed two-paned window. The top frame has two tabs: **Users** and **User Groups**, that list the user IDs, distinguished names if the portal server is configured for authentication to an LDAP user registry, and the user groups that are stored on the portal server. The profile of the selected user or user group is reflected in the bottom frame:

Permissions has a list of the portal features in the Authorities box. On the right are the possible operations for the selected feature. A selected check box means the selected user or user group has permission to perform that operation; a ▫ indicator next to the check box means the permission was added to a user group the user belongs to.

Applications shows all the applications being monitored and that are available for assigning to the user or user group. One user or user group, for example, can be profiled to see only the OMEGAMON applications, another to see only Linux and Oracle, middleware, and another to see all applications.

Navigator Views shows all the Navigator views that are on the portal server and that are available for assigning to the user or user group. The user or user group can be restricted to seeing only a certain branch of a Navigator view rather than the entire hierarchy.

Member of, when the Users tab is selected, or **Members**, when the User Groups tab is selected, is a list of the groups the user belongs to or the user names in the group.

The User Administration function enables you to maintain user IDs and user groups on the portal server, and provides varying degrees of access to the features and views of your monitored environment to accommodate any combination of job roles, such as *operators* who respond to alerts and direct them to the appropriate person for handling and *administrators* who plan, design, customize, and manage the monitoring environment.

In some managed enterprises one person might assume all of these roles. In larger enterprises, the roles are often divided. You can choose to assign roles by individual user or by user type or both.

Tivoli Enterprise Portal user IDs are also required for users who access monitoring dashboards in IBM Dashboard Application Services Hub. How you manage dashboard users depends on the type of authorization configured in the portal server and whether the dashboard users will also use the Tivoli Enterprise Portal client. There are two types of authorization that can be configured for controlling access to monitored resources in IBM Dashboard Application Services Hub:

Role-based authorization policies

These policies are created using the tivcmd Command-Line Interface for Authorization Policy. They provide more granular authorization than Tivoli Enterprise Portal monitoring application assignments. Using role-based authorization policies, you can assign a user permission to view specific managed system groups or managed systems. When role-based authorization policies are enabled in the portal server, dashboard users need a Tivoli Enterprise Portal user ID but do not require any Tivoli Enterprise Portal permissions or monitoring application assignments unless they are also Tivoli Enterprise Portal client users. In this case, role-based authorization policies control what resources they can access in the monitoring dashboards, and Tivoli Enterprise Portal permissions and monitoring application assignments control what they can access in the Tivoli Enterprise Portal client.

Tivoli Enterprise Portal authorization

This is the default authorization mechanism for dashboard users. A dashboard user must have a Tivoli Enterprise Portal user ID and be assigned the permissions and monitoring applications to control their access to resources in monitoring dashboards. If a dashboard user is also a Tivoli Enterprise Portal client user then they are assigned a single set of permissions that control what monitored resources they can access in both applications.

Configuring the portal server and Dashboard Application Services Hub to share an LDAP user registry is the best practice approach for having a federated set of dashboard users and Tivoli Enterprise Portal client users. In this scenario, the dashboard users login to the dashboard hub with their LDAP username and you must map their LDAP distinguished name to a Tivoli Enterprise Portal user ID with the required permissions.

Tivoli Enterprise Portal user IDs are automatically created with no permissions if a dashboard user requests monitoring data and does not have a user ID mapped to their distinguished name. See “Notes on user administration” on page 176 for more details.

Administer Users

Your user ID and the user groups you are a member of are profiled with a set of permissions that determines which Tivoli Enterprise Portal features you are authorized to see and use, a list of monitored applications you are authorized to see, and a list of Navigator views (and the highest level within a view) you can access.

Clicking **Administer Users** opens the Administer Users window. This is a two-paned window with Users and User Groups tabs in the top frame, and several tabs in the bottom frame. This arrangement enables the administrator to manage user profiles by individual user, by user groups, or a combination of the two. You might create a user profile, then copy the profile for each additional user and change settings as needed (such as, for the Action feature, granting View permission to one user and granting Modify permission to a different user). Or you might create a user group with a particular profile and add users to the group. Then you can modify the permissions once for a group and apply to all members automatically.

When you modify the permissions or the list of monitored applications for a Tivoli Enterprise Portal user or user group, the authorization change does not take effect until the user logs out of the Tivoli Enterprise Portal client and then logs back in. If Tivoli Enterprise Portal permissions are being used to authorize monitored resources for dashboard users, the authorization changes also do not take effect until the user logs out of Dashboard Application Services Hub and logs back in.

Related tasks:

"Adding a user ID" on page 170
Create a user ID for all users that should be able to log onto the Tivoli Enterprise Portal Server using a portal client or the **`tacmd tepsLogin`** command. A user ID is also required for IBM Dashboard Application Services Hub users who request monitoring data. You can use the default user profile or copy the profile of an existing user.

"Viewing and editing a user ID" on page 171
After a user has been added to the **Users** list in the Administer Users window, you can check and edit the profile settings at any time.

Users and User Groups

The **Users** and **User Groups** tabs list the user IDs and the user groups that are stored on the portal server.

After you select a user or user group from one of the lists, you can click any of the tabs in the lower half of the window to see the what permissions have been granted and what has been assigned. User groups enable the administrator to authorize the same set of functional permissions, applications, and Navigator views to multiple users at one time. Management of user authorization can be done by groups as well as individually. A user can be associated with one or more user groups; authorization by group will be by inclusion and not exclusion (nested groups are supported). Authorization will also be by global authority and by association with managed system and managed system groups. This security is not dependent on external authorization.

Permissions

You can authorize the same set of functional permissions multiple users, user group or each user ID at one time.

The following features are enabled or disabled individually for each user ID or user group.

Action

☑ **View** allows the user to see and run a take action command from the available list of commands in the Take Action view and in the pop-up menu for the Navigator item.

☑ **Modify** allows the user to create and save Take Action commands. When enabled, **Edit Action** appears in the Navigator pop-up menu.

When issuing a take action command, you must be authorized on the relevant system for the requested command. For example, to issue a TSO command, your user ID must be both a valid TSO ID and a valid user ID on the portal server. The user ID must be typed with the same letter casing exactly as typed when logging on to the portal server (with the same letter casing).

Agent Management

☑ **Manage** allows the user to perform agent deployment throughout the managed network. This includes installing a monitored product, keeping the software revisions up-to-date, and removing an agent from the managed network. This permission also requires Action - Modify to be enabled.

☑ **Start/Stop** allows the user to start a monitoring agent or to stop it running.

Custom Navigator Views

☑ **Modify** allows the user to create new Navigator views, edit and delete them. With Modify cleared, the user will not see **Edit Navigator View** in the Navigator toolbar.

Event

☑ **Attach** allows the user to attach a file (such as detailed notes) to the situation event. This permission requires that the user also have the Acknowledge and View permissions.

☑ **Close** lets you close a pure event or an event that was open before a situation was stopped manually. When it is enabled, ☑ **Close Situation Event** appears in the pop-up menu of the situation event flyover list, event Navigator item, and situation event console view when the selected event is a pure event or the situation has been stopped.

☑ **View** enables you to see situation event indicators in the Navigator when situations become true.

☑ **Acknowledge** allows you to acknowledge a situation event. When this permission is enabled, **Acknowledge Event** appears in the pop-up menu of the situation event flyover list, event Navigator item, and situation event console view.

Feature

☑ **Enable** is dimmed because you cannot change it. The access to this feature is determined by your organization's IBM Tivoli Monitoring license.

History

☑ **Configure** allows the user to open the History Collection Configuration window, configure history files and data rolloff, and start and stop data collection for the different attribute groups. When this permission is enabled, **History Configuration** appears in the main toolbar.

Launch Application

☑ **Launch** allows the user to invoke any of the launch definitions available for the Navigator item, table view, chart view, or situation event console view. When this permission is enabled, **History Configuration** appears in the main toolbar.

☑ **View** allows the user to see the composition of the selected launch definition.

☑ **Modify** allows the user to create, edit and delete launch definitions.

Managed System Group

☑ **View** allows the user to access the Object group editor for viewing managed system groups. The user also needs Modify permission for the Object group editor tools to be available.

☑ **Modify** allows the user to open the Object group editor to create, edit, and delete managed system groups.

Policy

☑ **View** allows the user to open the Workflows window to see the policies and their definitions. With View permission, the **Workflow Editor** is available in the main toolbar and **Manage Policies** is available in the Navigator pop-up menu at the agent level.

☑ **Start/Stop** lets you start and stop policies. With this permission enabled, ▶ **Start Policy** and ● **Stop Policy** are available when you select a policy.

☑ **Modify** allows the user to open the Workflow editor to create and edit policies. With the Modify permission enabled, **New Policy** is available after the user selects a policy, as are the other editing tools: **Edit Workflow**, **Copy Policy**, and **Delete Policy**.

Query
☑ **View** allows the user to access the Query editor through the Properties editor and select a query for the selected table or chart. With the View permission enabled, the user can assign a query through the Query tab of the Properties editor.

☑ **Modify** allows the user to create, edit and delete queries in the Query editor. With the Modify permission enabled, **Query Editor** is available from the main toolbar, as are the query editing tools.

Situation
☑ **View** allows the user to see situations in the Situation editor, including any expression overrides, and in the Manage Situations at Managed System window. With the View permission enabled, **Situation Editor** is available in the main toolbar and in the Navigator item (except at the platform level) pop-up menu.

☑ **Modify** lets you create new situations and manage them. When the Modify permission has been granted, the situation editing tools and pop-up menu options are available in the Situation editor, as well as the **Override Formula** button in the Distribution tab for qualifying situations.

☑ **Start/Stop** lets you start or stop a situation and enable or disable a situation override. When this permission is enabled, **Start Situation** and **Stop Situation** are available in the situation event flyover list, situation event console view, Situation editor pop-up menu, and the Manage Situations at Managed System window; and **Enable Situation Overrides** and **Disable Situation Overrides** are available in the Situation editor pop-up menu.

Terminal Script
☑ **View** allows the user to run or stop running a terminal emulator script and to see scripts, but not to edit them. If View is disabled the user will be able only to run or stop a script.

☑ **Modify** allows the user to create or record new terminal emulator scripts, edit, and delete them.

User Administration
If you are viewing your own user ID, you will see that View and Modify are disabled; you cannot change your User Administration permissions.

☑ **Logon Permitted** enables log on to the portal server with this user ID. The administrator can clear the check box to deny a user access to the portal. This option works in conjunction with the KFW_AUTHORIZATION_MAX_INVALID_LOGIN (the default is 0, unlimited attempts are allowed) parameter in the Tivoli Enterprise Portal Server Environment Configuration file, *kfwenv*. When the value has been set and the invalid attempts have been exceeded, the check box is cleared automatically and the administrator must select the check box to reset the logon attempt count. See the *IBM Tivoli Monitoring Administrator's Guide* for details.□ **Modify** allows the editing of user IDs and removing them. When this permission is enabled, **Administer Users** is available in the main toolbar and the tools are available in the Administer Users window.

☑ **Author Mode Eligible** allows the user to enable or disable their Author Mode permission under **Workspace Administration** (see next authority), but not for any other user IDs.

□ **View** allows the user to open the Administer Users window and see their user profile.

☑ **Administration Mode Eligible** allows the user to enable or disable their Administration Mode permission under **Workspace Administration** (see next authority), but not for any other user IDs.

Workspace Administration

☑ **Workspace Author Mode** allows the user to create and edit workspaces, links, and terminal emulator scripts. If Workspace Author Mode is disabled, the user cannot make any of these changes but can continue monitoring and responding to alerts; the tools can still be seen, but they are disabled.

□ **Workspace Administration Mode** is available only for the SYSADMIN user ID and new IDs made from it in the Create Another User window. When administration mode is enabled, changes you make to workspaces affect all users who log on to the same portal server. When it is disabled, workspace changes you make are not shared with other users. Be sure to select ☑ **Do not allow modifications** in the Workspace Properties whenever you create or edit a workspace in administration mode. Otherwise, if a user edits that workspace, you no longer own the workspace and cannot override their changes.

WebSphere MQ Configuration Authorities

IBM Tivoli OMEGAMON XE for Messaging: WebSphere MQ Configuration installations will see this folder.

☑ **View** allows the user to see, but not change, your organization's WebSphere MQ configuration in the Navigator Configuration view.

☑ **Modify** allows the user to change your organization's WebSphere MQ configuration or to schedule updates in the Configuration view.

Storage Subsystem Authorities

IBM Tivoli OMEGAMON XE for Storage installations will see this folder. Select ☑ **View** to allow the user to see, but not change data. Select ☑ **Modify** to allow the user to change data.

Data Collection Configuration allows the user to view or modify the collection control interval for the DFSMSrmm Status workspace and Dataset Attributes System Summary workspace.

Dataset Groups Collection Interval allows the user to view or modify the control interval for the Dataset Group Summary workspace.

Define/Update Dataset Groups allows the user to view or modify the group definitions for the Dataset Group Summary workspace.

Applications

Your user ID is set so you can see some or all the application types being monitored. For example, one user might be able to see only mainframe applications, while another can see only middleware, and another sees all applications.

Allowed Applications

Shows the applications that you can access from Tivoli Enterprise Portal.

Available Applications

Shows the applications available for assignment to the selected user. If **<All Applications>** is on the **Allowed Applications** list, then no other entries can be added. You must move it back to **Available Applications** before you can select a subset of applications to assign.

Select the applications you want to add, or select **<All Applications>**, and ◀ move them to the **Allowed Applications** list. After selecting the first application, you can use Ctrl+click to select other applications or Shift+click to select all applications between the first selection and this one.

Navigator views

When a Navigator view is created, only the author is able to see the view, but it is available for the administrator to assign to users. An assigned Navigator view means the user can open it. For each assigned view, the user can be restricted to see only a certain branch rather than the entire hierarchy.

Assigned Views

Shows the Navigator views the user is able to see and access. The first Navigator view in this list is the default for the user; it displays automatically whenever the user logs on. You can select any views to which you do not want the user to have access, and click ➡ right arrow to move them to the **Available Views** list. Select the appropriate entries and click ◀ left arrow to move them to the **Assigned Views.** You can move a Navigator view to the top of the list to make it the default by clicking the ▲ up arrow.

Available Views

Shows the Navigator views not assigned to the user and available for assignment. Select the Navigator views you want to add and move them to the **Assigned Views** list by using the ◀ left arrow. After selecting the first view, you can use Ctrl+click to select other views or Shift+click to select all views between the first selection and this one.

Assigned Root

Shows the Navigator view chosen in Assigned Views, with the user's assigned Navigator root highlighted. The root is the top-most level of this Navigator view that the user can access. The user can access this item and all items below it, but no items parallel to or above it in the Navigator.

For example, you can assign UNIX Systems as the assigned root. The user sees the UNIX Systems workspaces and those below, but is not able to see the Enterprise workspaces or anything under Windows Systems.

Member Of and Members

When you select a user or user group from the list, the last tab on the bottom set of tabs reads either **Member Of** or **Members** (reflecting the selection of a User or User Group). Assignment of users to groups can be done in either tab.

Managing user IDs

Managing user IDs begins with planning the authorities to grant to users and whether they will belong to user groups.

The Administer Users window provides the tools for creating and maintaining user IDs, and adjusting permissions. This is also where user IDs are mapped to their unique identifier in the LDAP user registry if user authentication through the portal server has been configured.

Adding a user ID

Create a user ID for all users that should be able to log onto the Tivoli Enterprise Portal Server using a portal client or the `tacmd tepsLogin` command. A user ID is also required for IBM Dashboard Application Services Hub users who request monitoring data. You can use the default user profile or copy the profile of an existing user.

Before you begin

To use this function, your user ID must have Modify permission for User Administration.

Procedure

1. Click **Administer Users**.
2. Create a new user ID or create one from another:
   - To create a new user ID with the default user profile, click **Create New User**.
   - To create a new user ID from an existing one, select the profile that you want to use from the **Users** list and click **Create Another User**.
3. In the Create New User window, enter the user information:
   - **User ID**: The logon name. The name must use ASCII characters, can be up to 10 characters, and can contain no spaces. The name is limited to eight characters if user authentication is at the hub monitoring server and uses RACF (resource access control facility) security for z/OS.
   - **User Name**: The name of the user or job classification or both. This name can include spaces and be up to 32 characters. The user name is displayed in **Users** list.
   - **Distinguished Name:** The unique identifier in the Lightweight Directory Access Protocol (LDAP) user registry for the name given in the **User ID** field. Click **Find** to locate and insert the distinguished name, such as UID=FRIDA,O=DEFAULTWIMITMBASEDREALM
   - **User Description:** Optional description for the user. The text can include spaces and punctuation.
4. Click **OK** to close the window and see the new user ID arranged alphabetically in the **Users** list.
5. To change the **Permissions**, select a function from the **Authorities** tree and select or clear each option as appropriate for all functions with permissions that you want to change.
6. To assign access privileges to applications (managed system types), click the **Applications** tab, then select **<All Applications>** or the individual applications the user should see, and click ◂ to move them to the **Allowed Applications** list. After selecting the first application, you can use Ctrl+click to select other applications or Shift+click to select all applications between the first selection and this one.
7. To assign Navigator views, click the **Navigator Views** tab:
   a. Select a Navigator view (or more with Ctrl + click and Shift + click) from the **Available Views** and click ◂ to move it to the **Assigned Views**.

   b. Use ▲ to place the view that you want to be the default at the top of the list; use ▼ and ▲ to arrange the other Navigator views in the order that they should appear in the Navigator toolbar View ▼ list.

   c. For the selected Navigator view, change the **Assigned Root** as needed.

8. When you are finished creating the user profile, save your changes with **Apply** if you want to keep the Administer Users window open, or **OK** if you want to close it.

What to do next

The Tivoli Enterprise Portal client logon window has a field for entering a user ID and password. If you want the user ID and password to be authenticated, configure the monitoring server or portal server to authenticate users. See Chapter 5, "Enabling user authentication," on page 89 for details.

Related reference:

"Administer Users" on page 164
Your user ID and the user groups you are a member of are profiled with a set of permissions that determines which Tivoli Enterprise Portal features you are authorized to see and use, a list of monitored applications you are authorized to see, and a list of Navigator views (and the highest level within a view) you can access.

Viewing and editing a user ID

After a user has been added to the **Users** list in the Administer Users window, you can check and edit the profile settings at any time.

Before you begin

To use this function, your user ID must have Modify permission for User Administration.

About this task

Use the following steps to edit a user ID:

Procedure

1. Click 👤 **Administer Users**.
2. Do one of the following in the **Users** list:
   - Click inside the **Name** or **Description** field to edit either of them.
   - Double-click anywhere in a row to open the Modify User window for editing any of the fields.
   - Right-click the user profile you want to edit and click 👤 **Modify User**.
3. Edit the **User Name**, **Distinguished Name** or **User Description**, then click **OK**. Distinguished Name is required if user authentication is through the portal server to an LDAP user registry. You cannot change the one-word User ID other than to change the letter casing. To edit the one-word User ID, delete the user profile and create a new one.
   - If you have not yet added the DN, click **Find** to locate the name that matches the user ID.
     If your monitored environment was previously configured for authentication through the Tivoli Enterprise Monitoring Server and then reconfigured to authenticate through the Tivoli Enterprise Portal Server, you might see two

entries for the name. Select the one where o=defaultWIMFileBasedRealm and not O=DEFAULTWIMITMBASEDREALM.

4. To change the Permissions, select a function from the **Authorities** tree and select or clear each option as appropriate for all functions with permissions that you want to change. You can change your own user permissions except Create and Modify for User Administration
5. To assign access privileges to applications (managed system types), click the Applications tab, select any applications you want to remove from the **Allowed Applications** list and ▸ move them to the **Available Applications** list; select the applications you want to add from the **Available Applications** list (or select **<All Applications>**), and ◂ move them to the **Allowed Applications** list.

   After selecting the first application, you can use Ctrl+click to select other applications or Shift+click to select all applications between the first selection and this one.
6. To change any Navigator view assignments, click the Navigator Views tab, then add or remove Navigator views from the **Assigned Views** list, and select and ▴ move the one to be the default to the top of the list. For each Navigator view, change the **Assigned Root** as needed.
7. When you are finished editing the user profile, save your changes with **Apply** if you want to keep the Administer Users window open, or **OK** if you want to close it.

Results

The next time the user logs on, the permission changes will be in effect.

Related reference:

"Administer Users" on page 164
Your user ID and the user groups you are a member of are profiled with a set of permissions that determines which Tivoli Enterprise Portal features you are authorized to see and use, a list of monitored applications you are authorized to see, and a list of Navigator views (and the highest level within a view) you can access.

Removing a user ID

You can remove a user ID as needed.

About this task

To use this function, your user ID must have Modify permission for User Administration.

Use the following steps to remove a user ID:

Procedure

1. Click **Administer Users**.
2. Select the user ID that you want to delete. You can select additional user IDs with Ctrl+click, or with Shift+click to select all user IDs between the first selection and this one.
3. Click **Remove Users** to delete the selected user ID and profile from the list.
4. When a message asks you to confirm the user ID removal, click **Yes**. The user is permanently removed from the user ID list. If the user is currently signed on, this does not affect their work session, but they will not be able to log on again.

Note: You cannot remove your user ID (the one you logged on with) or the <Default User> ID.

Default user

The first user ID in the **Users** list is <Default User>.

To use this function, your user ID must have Modify permission for User Administration.

The Default User ID is used as the template ID for users created with **Create New User**. Edit this user ID if you want to change any of the default settings. The initial defaults enable all the functions listed under Tivoli Enterprise Portal Authorities except User Administration Create and Modify. Any changes you make to the <Default User> ID apply to users created from this point on; they do not affect any existing user ID settings.

Managing user groups

User groups enable the administrator to authorize the same set of functional permissions, applications, and Navigator views to multiple users at one time. Management of user authorization can be done by groups as well as individually.

A user can be associated with one or more user groups. If a permission is granted to a user directly through their user ID, they maintain that permission even if a user group they belong to does not grant that permission. The reverse is also true, so that if an individual user ID is not granted a permission but the group ID is, the user will have the permission through their membership in the user group. Thus, the user's permission set is collected from what is given to the individual user ID and to any and all user groups that they belong to.

Authorization will also be by global authority and by association with managed system and managed system groups. This security is not dependent on external authorization.

When the active top tab is **Users**, the last tab on the bottom set of tabs reads **Member Of**. When the active top tab is **User Groups**, you will also have a **Members** tab. Assignment of users to groups can be done in either of these lower tabs.

Click the group in the details view at the top, then go to the **Members** tab to see the list of users that belong to this group. likewise, to see the groups a user belongs to.

Viewing user group memberships

You can view both the groups a user ID belongs to, and the list of user IDs belonging to a user group.

About this task

To use this function, your user ID must have Modify permission for User Administration.

Procedure

1. Click **Administer Users**. The Administer Users window is divided into two, with **Users** and **User Groups** tabs at the top, and **Permissions**, **Applications**, **Navigator Views**, and **Member Of** below.
2. To see the groups a user belongs to, select a name from the **Users** list, then click the **Member Of** tab. The groups the user belongs to are listed in the **Assigned Members Of** list.
3. To see the user IDs assigned to a group, select a name from the **User Groups** list, then click the **Members** tab. The users belonging to the group are in the **Assigned Members** list.

Adding a user group

You can create a new user group from the beginning or you can copy a group with similar permissions and user assignments to what you want, then modify the copy.

Before you begin

To use this function, your user ID must have Modify permission for User Administration.

About this task

Complete these steps to add a user group:

Procedure

1. Click **Administer Users** to open the Administer Users window.
2. Click the **User Groups** tab.
3. Do one of the following:
   - To create a new user group, click **Create New Group**.
   - To copy an existing user group, select the group name from the list and click **Create Another Group**.
4. In the Create New Group or Create Another Group window, enter the following user information:
   a. **Group ID:** The group identifier. This name can be up to 10 characters and can contain no spaces. The name is limited to eight characters if the hub monitoring server uses RACF (resource access control facility) security for z/OS.
   b. **Group Name:** The name or job classification for the user group. This name can include spaces..
   c. **Group Description:** The text to describe the user group, such as their responsibilities. The description can include spaces and punctuation.
5. Click **OK** to close the window and see the new user group arranged alphabetically in the User Group list.
6. Add members to the group in the **Members** tab by selecting one or more user IDs in the **Available Members** list and clicking ◀ to move to the **Assigned Members** list.
7. To change the Permissions for the group, select a function from the **Authorities** tree and select or clear each option check box for all functions.
8. To assign access privileges to applications (managed system types) for the group, click the Applications tab, then select **<All Applications>** or the individual applications the user should see, and click ◀ to move them to the

Allowed Applications list. After selecting the first application, you can use Ctrl+click to select other applications or Shift+click to select all applications between the first selection and this one.

9. To assign Navigator views to the group, click the Navigator Views tab, then add or remove Navigator views from the **Assigned Views** list, and use ▲ to place the default view at the top of the list. For each Navigator view, change the **Assigned Root** as needed.
10. When you are finished creating the user group, save your changes with **Apply** to keep the Administer Users window open, or **OK** to close it.

Reviewing and editing a user group

After a user group has been added to the **User Groups** list in the Administer Users window, you can check and edit the profile settings at any time.

About this task

To use this function, your user ID must have Modify permission for User Administration.

Use the following steps to edit a user ID:

Procedure

1. Click **Administer Users** to open the Administer Users window.
2. Click the **User Groups** tab.
3. Right-click the user group to edit and click .
4. Edit the **Group Name** and **Group Description**, then click **OK**. You cannot change the one-word group ID. You must, instead, create another user group from this one and give it a new name, then delete this one.
5. To change the Permissions, select a function from the **Authorities** tree and select or clear each option as appropriate for all functions with permissions that should change.
6. To change the group access privileges to applications (managed system types), click the Applications tab, select any applications you want to remove from the **Allowed Applications** list and click ➡; select the applications you want to add from the **Available Applications** list (or select **<All Applications>**), and click ◀ . After selecting the first application, you can use Ctrl+click to select other applications or Shift+click to select all applications between the first selection and this one.
7. To change any Navigator view assignments for the group, click the Navigator Views tab, then add or remove Navigator views from the **Assigned Views** list, and use ▲ to place the one you want to be the default at the top of the list. For each Navigator view, change the **Assigned Root** as needed.
8. When you are finished editing the user group, save your changes with **Apply** to keep the Administer Users window open, or **OK** to close it. The user group changes are effective the next time each group member logs on.

   Note: You can change the permissions, except Create and Modify for User Administration, of any groups you are a member of.

Removing a user group

You can remove a user group.

About this task

To use this function, your user ID must have Modify permission for User Administration.

Use the following steps to remove a user ID:

Procedure

1. Click **Administer Users** to open the Administer Users window.
2. Click the **User Groups** tab.
3. Select the user group to delete from the list and click **Remove Selected Group**. You can select additional user IDs with Ctrl+click, or with Shift+click to select all user groups between the first selection and this one.
4. When a message asks you to confirm the user group removal, click **Yes**. The group is permanently removed from the user group list. Any members of this user group who receive permissions from the group will not be affected until they next log on to the portal server.

Notes on user administration

Read these notes to understand the user ID contribution to Tivoli Enterprise Portal functions and modes.

Workspace administration mode

Any changes you make to workspaces, links, and terminal host session scripts in the Tivoli Enterprise Portal are available only to your user ID. The exception is while Workspace Administration Mode is enabled.

Workspace administration mode enables you to customize and add workspaces, links, and terminal emulator scripts that are shared with all users connected to the same Tivoli Enterprise Portal. See Starting workspace administration mode.

SYSADMIN logon ID

The Tivoli Enterprise Portal requires your logon ID whenever you start a work session. Every ID must first have been registered on the portal server. You can log onto the portal server with **SYSADMIN** and register other user IDs through the Administer Users window. The initial user ID, **SYSADMIN**, has full access and complete administrator authority. The system administrator registers additional users and sets their access privileges and authority.

User ID and groups

Each user ID is stored at the Tivoli Enterprise Portal Server and contains:

- The user name
- Job description
- Permissions for viewing or modifying Tivoli Enterprise Portal functions
- Assigned Navigator views and which Navigator item in each view appears as the root (default is the first item)
- Access to specific monitoring applications
- The user groups the user belongs to and indicators to signify when a permission has been granted to the user by a user group

Each user group is also stored at the portal server and has the same contents as for individual user IDs. But, instead of a list of user groups, it has a list of the user IDs assigned to the group.

Default user

The first user ID in the list is **<Default User>** and is used as the template ID for users created with Create New User. Edit this user ID if you want to change any of the default settings. The initial defaults enable all the functions listed under Tivoli Enterprise Portal Authorities, except the Modify permission for **User Administration**. Any changes you make to <Default User> ID apply to users created from this point on; they will not affect any existing user ID settings.

Granting access to a user

You set the authority privileges for each user when you create their user IDs. Giving users access to operational areas and customization options takes planning. Consider the job responsibilities of each user and the company security requirements when specifying authority privileges.

Important: Anyone with permission to create custom queries obtains access to all of the ODBC data source names (DSNs) created at the Tivoli Enterprise Portal Server. Add database user IDs, to be used in the DSN, to your database software, making sure to restrict user access to only those tables, columns, and so on, allowed by your organization's security policies.

Automatic Tivoli Enterprise Portal user ID creation

The first time a new user accesses a monitoring dashboard in IBM Dashboard Application Services Hub, a Tivoli Enterprise Portal user ID is automatically created and mapped to the user's LDAP distinguished name if a Tivoli Enterprise Portal user ID does not already exist for the user. The Tivoli Enterprise Portal user ID is a randomly generated string. If you need to assign Tivoli Enterprise Portal permissions and monitoring applications to a dashboard user and their Tivoli Enterprise Portal user ID was automatically created, you can either assign the permissions to the randomly generated user ID or perform these steps:

1. Delete the Tivoli Enterprise Portal user ID that was automatically created.
2. Create a new user Tivoli Enterprise Portal user ID, map it to the LDAP distinguished name for the user, and then assign it permissions and monitoring applications.

Validating user access

The Tivoli Enterprise Portal Server verifies user IDs whenever users log on. If a job description changes and the user requires different access to the portal server, you must review and perhaps change the user's permissions.

The user ID for logging on to the portal server might include a password. You do not establish passwords in the portal. Instead, you must define a matching user ID with password to the network domain user accounts or to the operating system where the Tivoli Enterprise Monitoring Server resides:

- User Accounts on the Windows system
- Password file on the UNIX system
- RACF or ACF/2 host security system on the z/OS system

As well, the monitoring server must be configured to Validate User. When users log on to the portal server, the hub monitoring server makes a request to the domain or the operating system to validate the user ID and password.

If the monitoring server has been installed on a distributed system, you can check if it has been configured to Validate User:

1. Start the Manage Tivoli Enterprise Monitoring Services program:

   Windows Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services.

   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 (default is opt/IBM/ITM).
2. Right-click the Tivoli Enterprise Monitoring Server row for TEMS1 (hub) and select **Reconfigure**.
3. In the Tivoli Enterprise Monitoring Server Configuration window, observe the setting of the ☑ **Security: Validate User** check box.

   When this option is selected, the password is required whenever a user logs on to the portal server; when it is cleared, the user name is required to log on but no password is required.

Note: Be aware that passwords must follow the security requirements for your organization. If this includes periodic password changes, you might get a **Logon password has expired** message while attempting to log on to the portal server. Should this happen, you must change your system password before you can log on. For example, on Windows this means changing the password through the Administrative Tools User Accounts.

Launching into the portal from other applications

In addition to any security requirements for launching into the Tivoli Enterprise Portal (such as single sign-on requirements), the Tivoli Enterprise Portal user ID that receives control after a launch from an external application must be pre-authorized to access the target managed system and workspaces. The user ID also must be authorized to issue any required take action commands.

User ID for Take Action commands

When the Tivoli Enterprise Portal sends a Take Action command to a managed system, the user ID might or might not be checked for authority to perform the action. In the simplest case, the command is sent to the managed system and executed using the user ID under which the agent is running. TheTivoli Enterprise Portal user ID is sent along with the action command in these contexts:

- On-demand: user ID currently logged on
- Situation action: user ID of the last person to update the situation
- Workflow action: user ID of the last person to update the policy

However, the ID is ignored by the managed system unless told otherwise by a command prefix. These are command handlers implemented in the IBM Tivoli Monitoring products to control whether the Tivoli Enterprise Portal user ID should be validated before passing the command to the agent for execution.

Command prefix

When a command prefix is present in the Take Action, the agent passes the command to the application handler rather than executing the command. The syntax of the prefix and take action command is

productcode:CNPuserID:command and the agent routes it to the application for execution. The application is free to execute the command with whatever user ID is appropriate. In the case of OMEGAMON XE for WebSphere MQ on z/OS, the Tivoli Enterprise Portal user ID is used.

If the special prefix is missing, the agent executes the command with the user ID under which the agent is running.

Most monitoring products do not employ a command prefix. IBM Tivoli Monitoring for WebSphere MQ does and, in fact, prepends any on-demand Take Action commands with a hidden **MQ:CNPuserID:** prefix, although you cannot see it.

UNIX setuid command

In addition to the command prefix and security exit, UNIX offers another option: a setuid command, which causes the process to dynamically change its userid. Thus, the agent could be changed to set the ID to the value passed as a parameter, issue the command, then change the user ID back again after the command is issued.

Troubleshooting logon error messages

Logon prompts and progress messages are displayed in the Logon window status bar. If a user cannot log on, a message is displayed.

If a user cannot log on, one of the following messages is displayed:

Failed connection to Tivoli Enterprise Portal Server

1. On the system where the Tivoli Enterprise Portal Server is installed, click **Start** -> **Programs** -> **IBM Tivoli Monitoring** -> **Manage Tivoli Monitoring Services**.
2. Optional: Right-click the Tivoli Enterprise Portal Server entry and click **Change Startup**. In the window that opens, select ● **System Account** and ☑ **Allow Service to Interact with Desktop** and click **OK**.

   This opens a command line window when the Tivoli Enterprise Portal Server is started and displays the internal commands.
3. Ensure that the Tivoli Enterprise Portal Server is started:

   If it is started, recycle it.

   If it is stopped, start it.
4. If you are still unable to connect, review the following information. If it does not apply to your situation, contact IBM Software Support.

If you are running in browser mode and going across networks to reach the Tivoli Enterprise Portal Server, it is possible the host name cannot be resolved by the network system. If this is the case, doing the following should resolve the problem:

1. On the system where the Tivoli Enterprise Portal Server is installed, click **Start** -> **Programs** -> **IBM Tivoli Monitoring** -> **Manage Tivoli Monitoring Services**.
2. Right-click the Tivoli Enterprise Portal Browser service and click **Reconfigure.**
3. Change the host name to the IP address in two places:

   In the **Launch URL** field, change *hostname* in http://*hostname*:15200 to the IP address of the Tivoli Enterprise Portal Server. For example, `http://10.21.2.166:15200`.

In the **Host** field, change the host name to the IP address of the Tivoli Enterprise Portal Server.

4. Click **OK**.
5. Start Tivoli Enterprise Portal browser mode using the IP address instead of the host name.
6. If you are still unable to connect, contact IBM Software Support.

Logon password has expired

If the hub Tivoli Enterprise Monitoring Server is set to Validate Users, then passwords are required. Passwords must follow the security requirements of your organization. If this includes periodic password changes, you might get this message while attempting to log on to the portal server. Should this happen, you must change your system password before you can log on. For example, on Windows this means changing the password through the Administrative Tools User Accounts.

User authorization has failed -OR- Unable to process logon request

Tivoli Enterprise Portal uses the TEPS database to locally validate users. If your hub monitoring server is set for user validation (Windows default), the user ID is also validated at the monitoring server to verify the password.

The portal server did not validate the user credentials as entered. For the “Unable to process logon request” message, the portal server was able to validate the user credentials but did not complete the logon request. In either case, have the user try logging on again. If the message is displayed again, do the following:

1. On the system where the monitoring server is installed, ensure that the server is running in Manage Tivoli Monitoring Services.
2. If the monitoring server is running, ensure that the user ID has been defined in Tivoli Enterprise Portal: Click **Administer Users**, then find the ID in the **Users** list.
3. If the user has been defined, check if host level security was turned on for the hub monitoring server and that the user ID has been authorized to the host environment:

   In Manage Tivoli Monitoring Services, right-click **Tivoli Enterprise Monitoring Server**, and click **Reconfigure.** If host level security has been configured, the **Security: Validate User** box is selected.

   If the monitoring server has been configured to Validate User, the user ID for Tivoli Enterprise Portal must also be added to the network domain user accounts or to the operating system where the monitoring server is installed, including a password.

   If non-ASCII characters were included in the user ID, they are not saved with the user ID.
4. Try logging on to Tivoli Enterprise Portal with the user ID in question.
5. If you cannot log on to Tivoli Enterprise Portal and the monitoring server is running properly, the problem might be with the Tivoli Enterprise Portal Server. Try recycling the portal server. If the user is still unable to log on, contact IBM Software Support.

This message is also displayed after a retry period of several minutes (the default is 10 minutes and can be changed through Manage Tivoli Monitoring Services) where the status bar shows **Validating user credentials** continuously. This can be a symptom that the monitoring server is stopped.

Chapter 7. Using role-based authorization policies

The Tivoli Authorization Policy Server feature provides you with role-based access control capabilities to protect your monitoring resources from unauthorized access by dashboard users of IBM Dashboard Application Services Hub.

Using authorization policies provides the following capabilities:

- The ability to restrict access for dashboard users to specific managed system groups and to individual managed systems.
- The ability to assign role-based policies to users and user groups in a federated LDAP user registry to simplify policy management.
- A new command-line interface that is highly automatable.
- Central management of authorization policies for multiple IBM Tivoli Monitoring environments, also called domains.

Tivoli Enterprise Portal permissions are the default authorization method for controlling access to resources in monitoring dashboards. They are also the mechanism used to authorize Tivoli Enterprise Portal client users. However, authorization policies provide greater control over resource access. With authorization policies, you can grant a dashboard user permission to view data from specific managed system groups or managed systems as compared to Tivoli Enterprise Portal authorization which assigns view permission for monitoring applications (monitoring agents). In other words, with Tivoli Enterprise Portal authorization, a user is assigned permission to view all managed systems of a particular agent application type, for example all Windows OS agents.

If you want to use the role-based access control provided by authorization policies, you must install the Tivoli Authorization Policy Server and the tivcmd Command-Line Interface for Authorization Policy. The Authorization Policy Server is installed with IBM Dashboard Application Services Hub along with monitoring dashboard applications such as Infrastructure Management Dashboards for Servers or custom dashboards. The tivcmd CLI is installed on computers used by authorization policy administrators and provides the command-line interface for creating and working with authorization policies. It sends HTTP or HTTPS requests to the Authorization Policy Server which maintains the master policy store. For installation information, see “Installing and configuring the Tivoli Authorization Policy Server and tivcmd Command-Line Interface for Authorization Policy” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

After successful installation of these two packages, you can execute tivcmd CLI commands as required to create and work with roles, grant permissions, exclude permissions, revoke permissions, and assign users and user groups to a role. For a complete list of tivcmd CLI commands, see the *IBM Tivoli Monitoring Command Reference*.

Once the initial set of authorization policies have been created, you enable authorization policy checking in the Tivoli Enterprise Portal Server. The portal server periodically downloads the authorization policies from the Authorization Policy Server application. When a dashboard user requests monitoring data, IBM Dashboard Application Services Hub forwards the request to the dashboard data provider component of the portal server. The dashboard data provider uses the authorization policies to determine which monitored resources the user is allowed to access.

Because both the Dashboard Application Services Hub and the portal server must have knowledge of the dashboard user, a typical dashboard environment includes a federated user registry provided by an LDAP server and single sign-on. For detailed information on the set of tasks involved in setting up a dashboard environment that uses authorization policies, see “Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls” on page 34.

Authorization policy concepts

An authorization policy either grants or excludes permission to a *user* or *user group*, acting in one of more *roles*, to perform an *operation* on an type of *object*, for a *resource* which is scoped by its *resource type*.

The elements of an authorization policy are described below:

User
Who initiates the operation.

User group
A set of users who can initiate the operation.

Role
A collection of permissions that can be assigned to users or user groups.

Operation
An action such as `create`, `delete`, `modify`, `distribute`, or `view`.

Object type
A categorization of the object that the operation is performed on. For example, monitoring data (`attributegroup`), `event`, or `role`.

Resource
The entity that the operation is being performed against such as a specific managed system group or managed system.

Resource type
A categorization of the resource. Managed system groups (`managedsystemgroup`), managed systems (`managedsystem`), and sets of roles (`rolegroup`) are the predefined resource types.

To create an authorization policy, perform the following tasks:

1. Create managed system groups that you want to control access to. Managed system groups are created using the Tivoli Enterprise Portal client and **`tacmd createsystemlist`** command.

   These can be the same managed system groups that you also use to distribute situations and historical collections.
2. Create user groups in LDAP that contain users that perform a similar job function.
3. Create a role that represents a job function within your organization.

   For example, you can define a role called `Eastern region Windows administrators` to control the monitored resources that can be accessed by the Windows OS administrators at your eastern region data center.
4. Next you grant or exclude one or more permissions to the role.
   - A *grant permission* specifies the operation that can be performed on a type of object for one or more resources of a specified type.

     For example, you can grant permission to view monitoring data for the managed system group `EasternRegionWindowsComputers` where the operation is `view`, the object type is `attributegroup` (which represents monitoring data), the resource is `EasternRegionWindowsComputers`, and the resource type is `managedsystemgroup`.

- An *exclude permission* allows you to restrict access to one or more members of a managed system group. You should create an exclude permission if you do not want a role to have access to all members of a managed system group. For example, the `EasternRegionWindowsComputers` managed system group might contain two or three computers that you do not want your eastern region Windows administrators to have access to. In this case, you can grant `view` permission for the `EasternRegionWindowsComputers` managed system group and `exclude` permission for specific managed systems. An exclude permission prevents any operation from being performed on objects of a managed system.

5. The final step is to assign the role to one or more users or user groups. Only users or user groups that have been assigned to an authorization policy role are able to access monitored resources in a dashboard. The user names and user group names are defined in the LDAP user registry that is shared by IBM Dashboard Application Services Hub and the portal server.

You can also revoke permissions from a role if you later decide that you need to remove a grant or exclude permission from a role. Authorization policies are also used to control which users can create and work with roles.

The following table lists the supported resource types, their associated object types and operations, and the type of permission that can be assigned for resources of this type.

Table 17. Authorization policy resource types and their supported permissions and elements

Permission	Operation	Object type	Resource type	Description
grant	view	attributegroup	managedsystemgroup	Using this combination, you can grant permission to view monitoring data such as metrics or status for all managed systems in a managed system group.
grant	view	event	managedsystemgroup	Using this combination, you can grant permission to view situation events from all managed systems in a managed system group. **Note:** If you want to grant permission to view the monitoring data that triggered the situation event then you must grant permission to view monitoring data for the managed system group.
grant	view	attributegroup	managesystem	Using this combination, you can grant permission to view monitoring data such as metrics or status for a specific managed system.
grant	view	event	managedsystem	Using this combination, you can grant permission to view situation events from a specific managed system. **Note:** If you want to grant permission to view the monitoring data that triggered the situation event then you must grant permission to view monitoring data for the managed system group.
exclude			managedsystem	Using this combination, you can exclude permission to perform any operation for a specific managed system.
grant	create	role	rolegroup	Using this combination, you can grant permission to create roles or events for specific managed systems.
grant	delete	role	rolegroup	Using this combination, you can grant permission to delete roles.

Table 17. Authorization policy resource types and their supported permissions and elements (continued)

Permission	Operation	Object type	Resource type	Description
grant	distribute	role	rolegroup	Using this combination, you can grant permission to distribute policies from the Authorization Policy Server to the Tivoli Enterprise Portal Server.
grant	modify	role	rolegroup	Using this combination, you can grant permission to modify roles.
grant	view	role	rolegroup	Using this combination, you can grant permission to view roles and permissions that you are assigned. This permission can be used if you have users who should be able to view their permissions but not permissions for other users.
grant	viewall	role	rolegroup	Using this combination, you can grant permission to view all roles and permissions.

When you are granted permission to view attribute groups (monitoring data) or events for a managed system group, you are granted permission to view the group and you are also granted permission to view all of the group members, unless there is an exclude permission for a group member.

In a large deployment of IBM Tivoli Monitoring, you might have multiple monitoring domains. A monitoring *domain* is defined as a collection of IBM Tivoli Monitoring components such as portal servers, monitoring servers, monitoring agents, and a Tivoli Data Warehouse that are centered around a particular hub monitoring server. In this type of deployment, you might have some authorization policies that are common across your monitoring domains as well as authorization policies that are specific to a particular domain. When you create permissions, the tivcmd CLI allows you to specify if the authorization policy applies to all domains (the default behavior) or to specific domains.

A *role group* is a set of roles that are shared across all the IBM Tivoli Monitoring domains using a single Authorization Policy Server. The Authorization Policy Server supports only one role group named `default`. It is specified as the resource name when creating permissions that perform operations on roles.

For information about working with authorization policies in a multi-domain deployment, see "Working with multiple domains" on page 202.

Predefined roles and permissions

The Tivoli Authorization Policy Server provides predefined roles and permissions to help you get started. The predefined roles are also called *core roles*. These roles cannot be modified or deleted, but they can be copied to create new roles.

The following roles and permissions are predefined:

RoleAdministrator

The main security administrator role with the authority to manage all roles and policies.

Note: When the Authorization Policy Server is installed, the installation program prompts for an IBM Dashboard Application Services Hub administrative user ID and password. The installer assigns the user ID to the RoleAdministrator role. To allow other users to create and work with roles and assign permissions, you must install the tivcmd CLI and use it to login to the Authorization Policy Server with the credentials that were specified during installation. Then use the tivcmd commands to assign other users to the RoleAdminisrator role or a custom role. For more details, see "Creating and assigning administrator roles" on page 189.

Table 18. RoleAdministrator permissions

Operation	Object type	Resource type	Resource
view	attributegroup	managedsystemgroup	any
view	event	managedsystemgroup	any
view	attributegroup	managedsystem	any
view	event	managedsystem	any
create, delete, modify, view, viewall	role	rolegroup	default

PolicyDistributor

The role with permission to download authorization policies.

This role, or a custom role with the same permission, must be assigned to the user ID that is specified when authorization policies are enabled in the portal server. The portal server uses the specified user ID and other connection properties to periodically connect to the Authorization Policy Server and download the latest policies. When the Authorization Policy Server receives a request for authorization policies, it verifies that the user who sent the request has been granted permission to distribute policies.

Table 19. PolicyDistributor permissions

Operation	Object type	Resource type	Resource
distribute	role	rolegroup	default

LinuxOperator

A role that has attribute group and event viewing permissions for all Linux agents.

UNIXOperator

A role that has attribute group and event viewing permissions for all UNIX agents.

WindowsOperator

A role that has attribute group and event viewing permissions for all Windows agents.

Table 20. LinuxOperator, UNIXOperator, and WindowsOperator permissions

Role	Operation	Object type	Resource type	Resource
LinuxOperator	view	attributegroup and event	managedsystemgroup	*LINUX_SYSTEM
UNIXOperator	view	attributegroup and event	managedsystemgroup	*ALL_UNIX
WindowsOperator	view	attributegroup and event	managedsystemgroup	*NT_SYSTEM

VCenterOperator

A role that has access to all VMWARE Virtual Centers and ESX Servers.

Table 21. VCenterOperator permissions

Operation	Object type	Resource type	Resource
view	attributegroup	managedsystemgroup	*VMWARE_VI_AGENT *VMWARE_VI
view	event	managedsystemgroup	*VMWARE_VI_AGENT *VMWARE_VI

Preparing to enable authorization policies

Before enabling authorization policies, ensure you are prepared by reading and following the information provided in this topic.

When a Tivoli Enterprise Portal Server is installed, the authorization policy enforcement is disabled by default. There is an **Enable authorization policies** check box in the Tivoli Enterprise Portal Server configuration panels that controls this feature. If the box is unchecked, it means the dashboard data provider will not use authorization policies to control managed system group and managed system access. Instead it uses Tivoli Enterprise Portal authorization to control access to monitored resources in the dashboards.

Complete the following steps before enabling authorization policy enforcement:

1. Identify which users will need to administer authorization policies.

   When the Authorization Policy Server is installed, the installation program prompts for an IBM Dashboard Application Services Hub administrative user ID and password. The installer assigns the user ID to the predefined RoleAdministrator role. To allow other users to create and work with roles and assign permissions, you must install the tivcmd CLI and use it to login to the Authorization Policy Server with the credentials that were specified during installation. Then use the tivcmd commands to add other policy administrators to the predefined RoleAdministrator role or create an equivalent role with similar permissions and add them to that role. For detailed steps, see “Creating and assigning administrator roles” on page 189.

   For details on installing the tivcmd CLI, see “Installing and configuring the Tivoli Authorization Policy Server and tivcmd Command-Line Interface for Authorization Policy” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. Assign at least one user to a role, typically the predefined PolicyDistributor role, that has the permission to distribute policies.

   The same user must be specified in the Tivoli Enterprise Portal Server configuration panel when you are ready to reconfigure the portal server to enable authorization policies.

   You must assign a user permission to distribute policies so that the dashboard data provider has the authority to retrieve policy updates from the Authorization Policy Server. For detailed steps, see “Creating and assigning policy distributor roles” on page 190.
3. Create roles and permissions, or leverage predefined roles and permissions, for all existing managed system groups and managed systems that can be displayed in dashboard views, then add users or user groups to these roles. For detailed examples, see “Policy management examples” on page 191.

   Note: The set of predefined roles and permissions are intended as a convenience to cover some, and perhaps even most of your security needs. However, you might have additional managed system groups and managed systems in your environment that require their own policy definitions. The information in “Policy management scenarios” is intended to help you better understand how to utilize the tivcmd CLI to create these additional policies.

4. Determine how frequently the dashboard data provider should retrieve authorization policies from the Authorization Policy Server.

   To avoid time-consuming accesses across the network, the dashboard data provider retrieves its own local copy of the Authorization Policy Server's master policy store. The policy retrieval occurs once during startup of the dashboard data provider, and is then repeated at regular intervals, which defaults to 30 minutes. You can change the default through the Tivoli Enterprise Portal Server configuration panels when you enable authorization policies and can specify a value between 5 and 1440 minutes. Be aware that any policy changes made using the tivcmd CLI will not take effect at the dashboard data provider until its next successful policy store retrieval.

When you are ready to start using authorization policies, reconfigure the portal server to enable authorization policies and to specify the connection properties for the Authorization Policy Server. The connection properties include the user ID that has been assigned a role with permission to distribute policies.

For a list of tasks to perform before you enable authorization policies, see “Setting up a monitoring dashboard environment with single sign-on and with per user authorization controls” on page 34.

Policy management scenarios

In the Tivoli Authorization Policy Server, there is no universal permission that may be assigned to a role which gives a user access to everything. Roles must be given explicit access to specific managed system groups or managed systems. Also a dashboard user can only see resources in monitoring dashboards if they are assigned to a role with access to managed system groups or managed systems.

Best practices for creating authorization policies

Review the best practices for creating authorization policies in your environment.

- Grant permissions to managed system groups rather than to individual managed systems.

  This approach has the advantage that authorization policy updates are not required when managed systems are added to your environment. Instead you add a new managed system to a managed system group that has already been granted view permission, and that contains similar systems using the same set of situation definitions or historical collection definitions. For new managed systems, you only need to create a new authorization policy in these scenarios:

  – There is not an existing managed system group that the managed system can be added to. In this case, add a new managed system group that contains the new managed system. Then grant view permission for the managed system group to one or more roles.

  – Create an exclude permission for the managed system because it should not be accessed by users who have been granted view permission for managed system groups that contain the new managed system.

- Assign user groups to roles rather than assigning individual users to roles.

When you follow this approach, authorization policy updates are not required when there is a new dashboard user. Instead you add the new user to a user group in LDAP that has already been assigned to an authorization policy role matching the new user's job function.

Create a new authorization policy for a new user if there is not an existing user group that the user can be added to. In this case, add a new user group in LDAP, add the new user to the group, and assign the group to one or more authorization policy roles. If necessary, create a new role that matches the new user group's permission scope.

- Create a user group in LDAP for your authorization policy administrators and assign the user group to a role that has been granted permission to create, modify, delete, and view roles. The predefined RoleAdministrator role has these permissions. Also ensure that multiple users are members of the user group so that authorization updates can be made by more than one user.
- If you want your dashboard users to see monitoring data and situation events for resources, then create roles that grant permission to view monitoring data (attribute groups) and to view situation events for the resources that they can work with.

  Only grant permission to view events but not monitoring data and vice versa for those users who need more restrictive access.

  When you grant a user group (or user) permission to view attribute groups and events, grant view permission for both object types against the same resource type (either managed system group or managed system).

  - Exemplary best practice: Grant permission to view events and attribute groups for managed system group resources.

    ```
    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystemgroup --resources "West_Coast_Systems" --objecttype
    attributegroup --operations view

    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystemgroup --resources "West_Coast_Systems" --objecttype event
    --operations view
    ```
  - Best practice variation: Grant permission to view events and attribute groups to a managed system.

    ```
    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystem --resources "Primary:server1:NT" --objecttype
    attributegroup --operations view

    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystem --resources "Primary:server1:NT" --objecttype event
    --operations view
    ```
  - Example of a policy that does not meet the best practice criteria:

    ```
    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystemgroup --resources "West_Coast_Systems" --objecttype
    attributegroup --operations view

    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystem --resources "Primary:server1:NT" --objecttype event
    --operations view
    ```

    where, `Primary:server1:NT` is a member of the `West_Coast_Systems` managed system group
  - Example of a policy that does not meet the best practice criteria:

    ```
    tivcmd grant --rolename "West Coast Administrators" --resourcetype
    managedsystem --resources "Primary:server1:NT" --objecttype
    attributegroup --operations view
    ```

```
tivcmd grant --rolename "West Coast Administrators" --resourcetype
managedsystemgroup --resources "West_Coast_Systems" --objecttype event
--operations view
```

where, `Primary:server1:NT` is a member of the `West_Coast_Systems` managed system group

- Additional best practices and considerations specific to the Infrastructure Management Dashboards for Servers application:
  - If you want dashboard users to use the **Managed System Groups** page, they must be assigned a role that grants permission to view events or attribute groups or both for managed system groups. If a user is assigned to roles that only have view permission for one or more managed systems, they will not see any monitored resources on the **Managed System Groups** page.
  - If you want dashboard users to use the **Situation Events** page, they must be assigned a role that grants permission to view events for managed system groups (best practice) or individual managed systems. If you want the user to see the monitoring data that caused the situation event to be opened, the user must also be assigned to a role that grants permission to view attribute groups for managed system groups (best practice) or individual managed systems.

Creating and assigning administrator roles

When the Authorization Policy Server is installed, a Dashboard Application Services Hub administrative user is assigned to the predefined RoleAdministrators role. Typically, this is smadmin. You can add your own administrative users to the predefined RoleAdministrator role, or create your own custom roles with the same permissions.

A best practice is to create a user group in LDAP for your policy administrators and assign the user group to the roles that have permission to create and work with authorization policies. By taking this approach, you only update the group membership (and not the authorization policies) when you add or remove policy administrators.

About this task

Any roles that are used for role administration must have the following permission:

Role administration permission definition	
Parameter	**Value**
Operation	'create', 'delete', 'modify', 'view', 'viewall'
Object Type	'role'
Resource Type	'rolegroup'
Resource	'default'

Procedure

- To assign a user or user group the predefine RoleAdministrator role, use the following steps:

1. Define a user in LDAP, for example uid=JohnDoe,cn=itm,o=ibm or define a user group, for example, cn=Administrators,cn=itm,o=ibm. Then add the policy administrator user ID, such as uid=JohnDoe,cn=itm,o=ibm, to the group in LDAP.
2. Add the user group to the predefined RoleAdministrator role using the following command:

   ```
   tivcmd addtorole --rolename RoleAdministrator
   --groups gid=Administrators,cn=itm,o=ibm
   ```
3. Alternatively, add the user to the predefined RoleAdministrator role using the following command:

   ```
   tivcmd addtorole --rolename RoleAdministrator
   --users uid=JohnDoe,cn=itm,o=ibm
   ```

- To create a new role with the same permission as the RoleAdministrator role, use the following steps:
  1. Define a user in LDAP, for example uid=JohnDoe,cn=itm,o=ibm OR define a group in LDAP, for example cn=Administrators,cn=itm,o=ibm, and then add the policy administrator user IDs to the group in LDAP.
  2. Create a new role and add the permissions to create and work with authorization policies to the new role. Then assign the user or user group to the new role. The following example commands show that you can add users and groups to the custom role for policy administration:

     ```
     tivcmd createrole --rolename EastCoastAdministrators --description
     "East Coast users with permission to manage roles and policies"

     tivcmd grant --rolename EastCoastAdministrators --resourcetype rolegroup
     --resources default --objecttype role --operations create delete modify
     view viewall

     tivcmd addtorole --rolename EastCoastAdministrators
      --users uid=JohnDoe,cn=itm,o=ibm
      --groups cn=Administrators,cn=itm,o=ibm
     ```
  3. Alternatively, you can duplicate the existing RoleAdministrator role and assign the new role to the user group using the following commands:

     ```
     tivcmd copyrole --fromrolename RoleAdministrator --torolename
     EastCoastAdministrators --description "East Coast users allowed to
     administer roles and policies for this authorization policy server"
     --permissionsonly

     tivcmd addtorole --rolename EastCoastAdministrators
      --groups cn=Administrators,cn=itm,o=ibm
     ```

Creating and assigning policy distributor roles

When you setup a new dashboard environment, user IDs must be created in the LDAP user registry for each dashboard user and policy administrator. You also need a user ID that is granted permission to distribute policies. This user ID must be specified when enabling authorization policies in the portal server. The portal server includes that user ID in the requests that it sends to the Authorization Policy Server to download the latest authorization policies. The Authorization Policy Server verifies that the user has permission to retrieve the policies. IBM Tivoli Monitoring provides the predefined PolicyDistributor role that has this permission already. Administrators can create new roles with this permission, or use the predefined role.

About this task

Any roles that are used for policy distribution must have the following permission:

Policy distribution permission definition	
Parameter	**Value**
Operation	'distribute'
Object Type	'role'
Resource Type	'rolegroup'
Resource	'default'

Procedure

- To assign a user the predefined PolicyDistributor role, use the following steps:
  1. Define a user in LDAP, for example `uid=PolicyAdmin,cn=itm,o=ibm`.
  2. Add the user to the predefined PolicyDistributor role using the following command:

     ```
     tivcmd addtorole --rolename PolicyDistributor
       --users uid=PolicyAdmin,cn=itm,o=ibm
     ```
- To create a new role with the same permission as the PolicyDistributor role, use the following steps:
  1. Define a user in LDAP, for example `uid=PolicyAdmin,cn=itm,o=ibm`
  2. Create a new role with the policy distribute permission and assign it to the user using the following commands:

     ```
     tivcmd createrole --rolename EastCoastDistributor --description
     "East Coast user IDs for downloading policy"

     tivcmd grant --rolename EastCoastDistributor --resourcetype rolegroup
     --resources default --objecttype role --operations distribute

     tivcmd addtorole --rolename EastCoastDistributor
     --users uid=PolicyAdmin,cn=itm,o=ibm
     ```
  3. Alternatively, you can duplicate the existing PolicyDistributor role using the following commands:

     ```
     tivcmd copyrole --fromrolename PolicyDistributor --torolename
     EastCoastDistributor --description "East Coast user IDs to
     download policy" --permissionsonly

     tivcmd addtorole --rolename EastCoastDistributor
       --users uid=PolicyAdmin,cn=itm,o=ibm
     ```

Policy management examples

The objective of authorization policies is to give you granular control over your monitored resources. When you setup a new dashboard environment, the dashboard user IDs must be created in the LDAP user repositories. Best practice is to also setup LDAP groups that contain the set of users that are assigned to authorization policy roles. This makes policy management easier, rather than assigning each individual user ID to a role. Use the examples in this topic to help you get started with your policies.

You use the tivcmd Command-Line Interface for Authorization Policy commands to managed your policies. For detailed information about the commands, see the *IBM Tivoli Monitoring Command Reference*.

This example assumes there is an existing role named West Coast Administrators, and that you want to grant this role the ability to view all attribute group data and events for the managed system group called West_Coast_DataCenter_Systems and another managed system group called West_Coast_Regional_Systems and assign this role to the user group cn=westcoastadmins,cn=itm,o=ibm.

```
tivcmd grant --rolename "West Coast Administrators"
--resourcetype  managedsystemgroup --resources
"West_Coast_DataCenter_Systems"
--objecttype attributegroup --operations view

tivcmd grant --rolename "West Coast Administrators"
--resourcetype managedsystemgroup --resources
"West_Coast_DataCenter_Systems"
--objecttype event  --operations view

tivcmd grant --rolename "West Coast Administrators"
--resourcetype managedsystemgroup --resources "West_Coast_Regional_Systems"
--objecttype attributegroup --operations view

tivcmd grant --rolename "West Coast Administrators"
--resourcetype managedsystemgroup --resources "West_Coast_Regional_Systems"
--objecttype event  --operations view

tivcmd addtorole --rolename "West Coast Administrators"
--groups cn=westcoastadmins,cn=itm,o=ibm
```

This example assumes you want to prevent members of the user group cn=westcoastadmins,cn=itm,o=ibm from viewing attribute group data and events for the Primary:server1:NT managed system. In this scenario, Primary:server1:NT is a member of the West_Coast_DataCenter_Systems managed system group that the user group was granted permission to view in the previous example.

```
tivcmd exclude --rolename "West Coast Administrators" --resourcetype
managedsystem --resources Primary:server1:NT
```

This example assumes you want to remove the grant permissions to view attribute group data and events for managed system group West_Coast_DataCenter_Systems and the exclude permission for the Primary:server1:NT managed system from the West Coast Administrators role but leave the grant permissions for the West_Coast_Regional_Systems managed system group.

```
tivcmd revoke --rolename "West Coast Administrators" --resourcetype
managedsystemgroup --resources "West_Coast_DataCenter_Systems"
--objecttype attributegroup --operations view  --grantcommand

tivcmd revoke --rolename "West Coast Administrators" --resourcetype
managedsystemgroup --resources "West_Coast_DataCenter_Systems"
--objecttype event  --operations view --grantcommand

tivcmd revoke --rolename "West Coast Administrators" --resourcetype
managedsystem --resources Primary:server1:NT  --excludecommand
```

In this example, you are an IBM Tivoli Monitoring administrator who wants to control dashboard access to the managed systems belonging to three geographic regions: Eastern, Central, and Western. The monitoring server has managed system group definitions for EasternRegionSystems, CentralRegionSystems, and WesternRegionSystems, which contain managed systems for the respective geographic regions. You want access to the managed systems in all three regions, but want the operator named Annette to only have access to Western region systems. This example assumes the local LDAP user registry includes user groups called EasternRegionOperators, CentralRegionOperators, and WesternRegionOperators and that Annette is a member of the WesternRegionOperators group.

1. Login to the Authorization Policy Server:

   ```
   tivcmd login --username <user> --password <password>
   ```

2. Create three new roles, one for each geographic region:

   ```
   tivcmd createrole --rolename EasternRegionOperator --description
   "A role to govern access to data for Eastern Region Systems"

   tivcmd createrole --rolename CentralRegionOperator --description
   "A role to govern access to data for Central Region Systems"

   tivcmd createrole --rolename WesternRegionOperator --description
   "A role to govern access to data for Western Region Systems"
   ```

3. Confirm that the new roles were created:

   ```
   tivcmd listroles --rolename EasternRegionOperator --showdescription

   tivcmd listroles --rolename CentralRegionOperator --showdescription

   tivcmd listroles --rolename WesternRegionOperator --showdescription
   ```

4. Display the usage rules for **tivcmd grant** command:

   ```
   tivcmd grant -?
   ```

5. Issue grant commands allowing the EasternRegionOperator role to have view access to attribute data and events for EasternRegionSystems:

   ```
   tivcmd grant --rolename EasternRegionOperator --resourcetype managedsystemgroup
   --resources EasternRegionSystems --objecttype attributegroup --operations view

   tivcmd grant --rolename EasternRegionOperator --resourcetype managedsystemgroup
   --resources EasternRegionSystems --objecttype event --operations view
   ```

6. Confirm that the EasternRegionOperator role has the correct permissions:

   ```
   tivcmd listroles --rolename EasternRegionOperator --showpermissions
   ```

7. Repeat the commands to grant the other two roles the same permissions to their respective geographic regions:

   ```
   tivcmd grant --rolename CentralRegionOperator --resourcetype managedsystemgroup
   --resources CentralRegionSystems --objecttype attributegroup --operations view

   tivcmd grant --rolename CentralRegionOperator --resourcetype managedsystemgroup
   --resources CentralRegionSystems --objecttype event --operations view

   tivcmd grant --rolename WesternRegionOperator --resourcetype managedsystemgroup
   --resources WesternRegionSystems --objecttype attributegroup --operations view

   tivcmd grant --rolename WesternRegionOperator --resourcetype managedsystemgroup
   --resources WesternRegionSystems --objecttype event --operations view
   ```

8. Display the usage rules for **tivcmd addtorole** command:

   ```
   tivcmd addtorole -?
   ```

9. Associate each LDAP user group to its corresponding role:

   ```
   tivcmd addtorole --rolename EasternRegionOperator --groups
   cn=EasternRegionOperators,cn=itm,o=tivoli

   tivcmd addtorole --rolename CentralRegionOperator --groups
   cn=CentralRegionOperators,cn=itm,o=tivoli

   tivcmd addtorole --rolename WesternRegionOperator --groups
   cn=WesternRegionOperators,cn=itm,o=tivoli
   ```

10. Display the membership of each role to confirm that the user group associations were completed properly:

```
tivcmd listroles --rolename EasternRegionOperator --showmembership

tivcmd listroles --rolename CentralRegionOperator --showmembership

tivcmd listroles --rolename WesternRegionOperator --showmembership
```

11. Ensure that you have access to systems in all three regions. This can be accomplished by adding your user ID to each of the three new roles:

    ```
    tivcmd addtorole --rolename EasternRegionOperator --users
    uid=<userid>,cn=itm,o=tivoli

    tivcmd addtorole --rolename CentralRegionOperator --users
    uid=<userid>,cn=itm,o=tivoli

    tivcmd addtorole --rolename WesternRegionOperator --users
    uid=<userid>,cn=itm,o=tivoli
    ```

12. Add your user ID to the predefined PolicyDistributor role. This command ensures that your ID can be used by the dashboard data provider to download policy file store updates from the Authorization Policy Server:

    ```
    tivcmd addtorole --rolename PolicyDistributor
    --users uid=<userid>,cn=itm,o=tivoli
    ```

13. Display all of the roles that your user ID belongs to:

    ```
    tivcmd listroles --username uid=<userid>,cn=itm,o=tivoli
    ```

The security setup is now complete.

- Annette is only a member of the `WesternRegionOperators` user group.
- The `WesternRegionOperators` user group is only assigned to the `WesternRegionOperator` role.
- The `WesternRegionOperator` role is only granted access to the `WesternRegionSystems` managed system group.
- Annette can only view attribute data and events for managed systems belonging to `WesternRegionSystems`.

Enabling authorization policies in the portal server

After you have created the initial set of authorization policies and assigned a user to the role with permission to distribute policies, enable authorization policy enforcement in the dashboard data provider by configuring the Tivoli Enterprise Portal Server using Manage Tivoli Enterprise Monitoring Services or the command-line.

Procedure

- Using Manage Tivoli Enterprise Monitoring Services
  1. Start Manage Tivoli Enterprise Monitoring Services on the computer where the portal server is installed:

     Windows Click **Start → Programs →IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services**.

     Linux UNIX Where *install_dir* is the IBM Tivoli Monitoring installation directory, change to the `*install_dir*/bin` directory and run `./itmcmd manage [-h *install_dir*]`.
  2. Right-click **Tivoli Enterprise Portal Server:**

     Windows Click **Reconfigure**, and click **OK** to accept the existing configuration and go to the second TEP Server Configuration window.

Linux UNIX Click **Configure**. The Common Event Console Configuration window is displayed. Click **OK** to accept the current values. On the Configure Tivoli Enterprise Portal window, select the **Dashboard data provider** tab.

3. In the dashboard data provider area of the configuration window, verify the **Enable authorization policies** check box is selected. If it is not selected, then select it.
   a. When the dashboard data provider is enabled, you can specify a domain override value. This value is optional. It changes the default dashboard data provider ID and domain name for authorization policies from `itm.<`*`hub_monitoring_server_name`*`>` to `itm.<`*`domain_override_value`*`>`. The value may not exceed 124 characters. You should configure a domain override value for these scenarios:
      - The Hot Standby high availability feature is being used for the hub monitoring server. By configuring a domain override value, the dashboard data provider ID and domain name will not change when the portal server is configured to connect to the new acting hub monitoring server. If you do not configure a domain override value in this scenario, you must reconfigure the connection between the IBM Dashboard Application Services Hub and the dashboard data provider and update any domain-specific authorization policies when the portal server is configured to connect to the new acting hub monitoring server.
      - You have multiple hub monitoring servers that are using a common set of authorization policies for controlling dashboard access and you want to create some domain-specific authorization policies. You should specify a domain override value for this scenario if you want to use a more user-friendly domain name in your authorization policies than the default value of `itm.<`*`hub_monitoring_server_name`*`>`.

      If you modify the domain override after you have configured a connection in your Dashboard Application Services Hub to the dashboard data provider then you must delete the connection and re-add it. See "Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 for details on how to configure a dashboard data provider connection. Also, if you have created any domain-specific authorization policies using the default domain name, then you must delete the permissions that use the previous domain name and create new permissions that use the new domain name when you change the domain override value.
   b. The **Enable authorization policies** option is selected if you want to use authorization policies to control which managed systems and managed system groups a user can access in monitoring dashboards. Only enable authorization policies if you are setting up a dashboard environment with single sign-on, you plan to use authorization policies to control access to monitoring dashboards, and your administrators have already created the initial set of policies for dashboard user access.
4. In the Authorization Policy Server Configuration window specify the following information:

Table 22. Configuration information for the Authorization Policy Server

Field	**Description**
Hostname or IP Address	IP Address or fully qualified hostname of the IBM Dashboard Application Services Hub with the Authorization Policy Server. This parameter is required. **Note:** If your environment includes an HTTP server for load balancing across multiple Dashboard Application Services Hub servers, the Authorization Policy Server can only be installed with one of the dashboard servers because it does not support load balancing. Therefore, you must determine which computer system has Dashboard Application Services Hub and the Authorization Policy Server installed and specify the hostname or IP address of that system.
Protocol	Choose the protocol used to connect to the IBM Dashboard Application Services Hub with the Authorization Policy Server. The default value is HTTPS. This parameter is not required. **Note:** You should only select HTTPS if you have already configured TLS/SSL between the portal server and the Authorization Policy Server. See "Configuring TLS/SSL communication with the Authorization Policy Server" on page 216..
Port	Choose the port used to connect to the IBM Dashboard Application Services Hub with the Authorization Policy Server. The default value is 16311 for the HTTPS protocol and 16310 for the HTTP protocol. The valid port values are from 1 to 65535 inclusive. This parameter is not required.
Polling Interval	How often the local data store is updated from the Authorization Policy Server by the policy client running on the portal server. The default is 30 minutes. Valid values are from 5 to 1440 minutes inclusive. This parameter is not required.
Policy Store Expiration Interval	If the policy store cannot be updated from the Authorization Policy Server, this interval is the amount of time the local policy store will continue to be utilized from the last update. If the Authorization Policy Server cannot be accessed for the time interval specified by this parameter, all subsequent requests for dashboard data will fail with an authorization error until the Authorization Policy Server is available again. The default is 7 days and 0 hours. The value specified for hours must be in the range of 0-23 hours. If the expiration interval is set to 0 days and 0 hours, the policy store will never expire. This parameter is not required.
User ID	The name of the user that the portal server will use to access the IBM Dashboard Application Services Hub with Authorization Policy Server. This user must be added to the PolicyDistributor authorization policy core role or to a custom role that has been granted permission to perform the distribute operation for the role object type. This parameter is required.
Password	The password for the user. This parameter is required.
Confirm Password	Confirm the password by entering it again. This parameter is required.

Enter the required information for the Authorization Policy Server connection parameters in the fields provided and click **OK**.

5. You are prompted to reconfigure the warehouse connection information, answer **No**.
6. On Windows, after some processing of the configuration settings, the Common Event Console Configuration window is displayed. Sometimes this window does not open in the foreground and is hidden by other windows. If processing seems to be taking longer than expected, minimize other windows and look for the configuration window. When the Common Event Console Configuration window is displayed, click **OK**.
7. If you made configuration changes, ensure the portal server is restarted.

- Using the command-line

  If the Tivoli Enterprise Portal Server is on Linux or UNIX, you can modify the portal server configuration from the command-line and enable authorization policies if it is not already enabled.

  1. Log on to the computer where the Tivoli Enterprise Portal Server is installed.
  2. At the command-line, change to the `*install_dir*/bin` directory, where *install_dir* is the directory where you installed the product.
  3. Run the following command to configure the Tivoli Enterprise Portal Server: **`./itmcmd config -A cq`**.

     The message `Agent configuration started` is displayed, followed by a prompt: `Tivoli Enterprise Portal Server will be stopped during configuration. Do you want to continue? [1=Yes, 2=No] (Default is: 2).`
  4. Enter 1. The following prompt is displayed: `Edit "Common event console for IBM Tivoli Monitoring" settings? [ 1=Yes, 2=No ] (default is: 1).`
  5. Enter 2. The following prompt is displayed: `Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1).`
  6. Accept the default values for this prompt and the prompts that follow it until you see the prompt about configuring the dashboard data provider. If it is not enabled, select a value of 1 to enable it.
  7. Next you are asked if you want to specify a domain override value. Enter `1` for Yes and `2` for No.

     When the dashboard data provider is enabled, you can specify a domain override value. This value is optional. It changes the default dashboard data provider ID and domain name for authorization policies from `itm.<hub_monitoring_server_name>` to `itm.<domain_override_value>`. The value may not exceed 124 characters. You should configure a domain override value for these scenarios:

     - The Hot Standby high availability feature is being used for the hub monitoring server. By configuring a domain override value, the dashboard data provider ID and domain name will not change when the portal server is configured to connect to the new acting hub monitoring server. If you do not configure a domain override value in this scenario, you must reconfigure the connection between the IBM Dashboard Application Services Hub and the dashboard data provider and update any domain-specific authorization policies when the portal server is configured to connect to the new acting hub monitoring server.
     - You have multiple hub monitoring servers that are using a common set of authorization policies for controlling dashboard access and you want to create some domain-specific authorization policies. You should specify a domain override value for this scenario if you want to use a more

user-friendly domain name in your authorization policies than the default value of `itm.<hub_monitoring_server_name>`.

If you modify the domain override after you have configured a connection in your Dashboard Application Services Hub to the dashboard data provider then you must delete the connection and re-add it. See "Creating a connection to the IBM Tivoli Monitoring dashboard data provider" on page 58 for details on how to configure a dashboard data provider connection. Also, if you have created any domain-specific authorization policies using the default domain name, then you must delete the permissions that use the previous domain name and create new permissions that use the new domain name when you change the domain override value.

8. If the dashboard data provider is enabled, you are prompted whether you want to enable authorization policies. Use the information in Table 22 on page 196.

   Only enable authorization policies if you are setting up a dashboard environment with single sign-on, you plan to use authorization policies to control access to monitoring dashboards, and your administrators have already created the initial set of policies for dashboard user access.
9. After the command has completed the configuration, the following message is displayed: `Agent configuration completed` and you are asked if you want to restart the portal server. Select 1 to restart it.

Results

You have successfully enabled authorization policies on the portal server.

After you have recycled the Tivoli Enterprise Portal Server with the **Enable authorization policies** box checked, the dashboard data provider will start making authorization calls against its local policy store to allow or exclude managed system group and managed system access for dashboard users.

If authorized dashboard users do not see any monitored resources in the dashboards or they do not see the correct set of resources, see the *IBM Tivoli Monitoring Troubleshooting Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_troubleshoot.htm) for steps to diagnosis this issue.

Authorization policy auditing

The Authorization Policy Server generates audit messages when a user executes one of the following tivcmd commands that modify authorization policies: addtorole, copyrole, createrole, deleterole, exclude, grant, removefromrole, and revoke. An audit messages is also generated if a user attempts to execute a tivcmd command that they are not authorized to use.

For example, an audit message is generated if a user executes the **`tivcmd createrole`** command but they are not assigned to a role with permission to create roles.

The audit messages use the IBM Tivoli Monitoring audit record format and are written to audit log files on the computer where the Authorization Policy Server is installed. The default location is `<JazzSM_install_dir>/AuthPolicyServer/PolicyServer/audit`. During installation of the Authorization Policy Server, you can customize the location of the audit log file directory, the maximum size of the audit log files, and the maximum number of audit log files to keep at one time.

After installation, you can change these parameters by following the instructions in "Changing the Authorization Policy Server configuration properties after installation and configuration." Since the Authorization Policy Server is not associated with a monitoring agent, its audit messages cannot be viewed from the Tivoli Enterprise Portal or saved in the Tivoli Data Warehouse . Additionally, you cannot write situations against the audit messages. Instead you must display the audit log files if you want to view the audit messages.

Audit messages for authorization policy enforcement are generated by the dashboard data provider component of the Tivoli Enterprise Portal Server. The dashboard data provider generates an audit message if a user requests attribute group data or situation events for a managed system group or managed system that they do not have view permission for. Audit messages are also generated when authorization policies are downloaded from the Authorization Policy Server and when the policies cannot be retrieved. Since the dashboard data provider is a component of the portal server managed system, these audit messages can be viewed from the Tivoli Enterprise Portal client and saved to the Tivoli Data Warehouse.

For more details on auditing, including how to view audit messages for the portal server and the audit record format, see Chapter 9, "Audit logging," on page 247.

Changing the Authorization Policy Server configuration properties after installation and configuration

After the Tivoli Authorization Policy Server package is installed and configured, you can change the Authorization Policy Server configuration parameters for audit logging and policy distribution.

Before you begin

Audit properties
You can modify the properties that specify the location of the Authorization Policy Server audit log files, the maximum size of an audit log file, and the maximum number of audit log files to keep at one time.

Policy distribution properties
The Authorization Policy Server periodically compresses a file of the current set of authorization policies that is available for distribution. On a periodic interval, the dashboard data provider component of the portal server makes a request to the Authorization Policy Server for the latest compressed file of policies. If there is a new file, it is obtained and extracted and this set of policies becomes the current set of policies that are used by the dashboard data provider. You can modify the properties that specify the directory where the authorization policies are saved for distribution and how often the current authorization policies are copied to this directory.

About this task

You use the WebSphere Application Server administrator console of the Dashboard Application Services Hub where the Authorization Policy Server is installed to change the configuration properties of the policy server. After any change is made, you must restart the WebSphere Application Server for Dashboard Application Services Hub to pick up the property changes.

Perform the following steps to change the configuration properties for audit logging and policy distribution:

Procedure

1. Log in to the WebSphere Administrative Console of the Dashboard Application Services Hub where the Authorization Policy Server is installed.
   a. In a browser, open the Dashboard Application Services Hub console. By default, the URL is `https://hostname:16311/ibm/console`.

      If your environment was configured with a port number other than the default, enter that number instead. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.

   b. Enter a user name and password and click **Go**. The user name must be assigned to the Dashboard Application Services Hub `administrator` and `iscadmins` roles.
   c. Click the **Console Settings** icon and select **WebSphere Administrative Console**.
   d. Click **Launch WebSphere administrative console**.
2. Navigate to the page that contains the configuration properties for audit logging and policy distribution.
   a. Click **Resources** → **Resource Environment** → **Resource Environment entries**.
   b. On the page that opens, click the **AuthzResourceReference** link.
   c. On the page that opens, under **Additional Properties**, click **Custom properties**.
   d. A table is displayed with the following properties:

      AUDIT_COUNT
      The maximum number of audit log files to keep at one time. Default value is 5. Range is greater than 1 and less than 99999.

      AUDIT_FILE_SIZE
      The maximum size of each log file in megabytes.

      AUDIT_ROOT_DIRECTORY
      The directory into which the audit log files are stored.

      Default value is `<`*`JAZZSM_INSTALL_DIR`*`>\AuthPolicyServer\PolicyServer\audit`

      If you modify this value to point to a different directory, you must ensure that the directory exists and has the same permissions as the `<`*`JAZZSM_INSTALL_DIR`*`>\AuthPolicyServer\PolicyServer\xacml` directory.

      DIST_POLL_INTERVAL
      This property specifies how often the Authorization Policy Server updates the compressed file containing the authorization policies that is downloaded by the dashboard data provider.

      Default value is 5. Range is 1 - 1440 minutes.

      DIST_ROOT_DIRECTORY
      The directory into which the version of the policies for distribution is stored.

      Default value is `<`*`JAZZSM_INSTALL_DIR`*`>\AuthPolicyServer\PolicyServer\dist`

If you modify this value to point to a different directory, you must ensure that the directory exists and has the same permissions as the `<JAZZSM_INSTALL_DIR>\AuthPolicyServer\PolicyServer\xacml` directory.

SEED_ROOT_DIRECTORY
The policy store seed directory. You should not modify this property.

XACML_ROOT_DIRECTORY
The policy store root directory. You should not modify this property.

3. Modify the value of a property.
   a. Click a property name link in the Custom properties table, for example **AUDIT_COUNT**.
   b. On the page that opens, modify the **Value** field as required.
   c. Click **OK**.
   d. Repeat these steps for each property to be changed.
4. Save the changes.
   a. On the message box that opens after the first property change, click **Save**.
   b. Log out of the WebSphere administrative console.
   c. Recycle the Dashboard Application Services Hub's WebSphere Application Server.

Managing the authorization policy store

The Tivoli Authorization Policy Server stores the policies in multiple files on the file system. Review the details below to understand how to manage the policy store.

High availability
The Authorization Policy Server does not have built-in high availability mechanisms and does not support load balancing. Therefore, if you setup multiple Dashboard Application Services Hub severs for load balancing, you can only install the Authorization Policy Server with one of the Dashboard Application Services Hub servers. Also, when enabling authorization policies in the portal server in your IBM Tivoli Monitoring environment, you must configure the location of the single Dashboard Application Services Hub server where the authorization policy server package is installed and configured. The tivcmd CLI users must also specify the hostname and port number of the Dashboard Application Services Hub with the Authorization Policy Server instead of the hostname and port number of the load balancing HTTP server.

Because the portal server has its own copy of the authorization policies, it is able to enforce the policies even if the Authorization Policy Server is not available. You can configure the maximum amount of time the portal server can utilize its local policy store after the last update. If the Authorization Policy Server cannot be accessed for the time interval specified by this parameter, all subsequent requests for dashboard data will fail with an authorization error until the Authorization Policy Server is available again. The default value is 7 days.

Migration and backup
The Authorization Policy Server does not offer migration, backup, or export and import tools for the policy file store. Over time, you might

create many policy definitions. If the policy store became damaged or inadvertently deleted, recreating your policy definitions would not be easy.

Best practice is for you to implement periodic backups, which can be done with zip or tar utilities. The files comprising the policy store are maintained in the /xacml subdirectory under the directory where your Authorization Policy Server was installed. For example, assume you installed the Authorization Policy Server into the following directory on Windows: C:\Program Files\IBM\JazzSM\AuthPolicyServer. Zipping up all files in the C:\Program Files\IBM\JazzSM\AuthPolicyServer\PolicyServer\xacml directory effectively backs up the entire policy store. You can later use this zip file to help with migration, for example, from a test to production Authorization Policy Server. Unzipping the file on a new production system will create and populate the /xacml subdirectory with all of the policy roles and permissions you previously had defined on the test system. These unzipped files can be used as-is by the production Authorization Policy Server.

Working with multiple domains

You can use the Authorization Policy Server to manage policies for multiple hub monitoring server environments, also called domains.

A *domain* is defined as a collection of IBM Tivoli Monitoring components such as portal servers, monitoring servers, monitoring agents, and Warehouse Proxy agents that are centered around a particular hub monitoring server. Each domain has its own name space for situations, take actions, managed systems names, managed system groups, queries, workflow policies, and any other IBM Tivoli Monitoring object.

An authorization policy that is created without specifying a domain name (or with a domain name of 'any') is applied across all IBM Tivoli Monitoring domains that are being managed by the Authorization Policy Server. Domain names are the same as the dashboard data provider ID and are in the format itm.*hub_monitoring_server_name*. You can create a more user-friendly string to use in place of the hub monitoring server name when you configure the portal server and enable the dashboard data provider.

To list your domains use the **tivcmd listdomains** command. For example:

```
tivcmd listdomains
itm.HUB_DOMAIN1
itm.HUB_DOMAIN2
itm.HUB_DOMAIN3
```

The **tivcmd listdomains** command returns the list of dashboard data providers for which a connection is defined in the Dashboard Application Services Hub that the Authorization Policy Server is installed with. The command also returns any domain names that were specified when creating a permission. Typically, there is a single dashboard data provider connection defined per Dashboard Application Services Hub.

To determine the hub names for each of your domains, use one of the following methods:

- Run the **tacmd listsystems** command against each hub monitoring server to determine the hub monitoring server's managed system name. Monitoring servers use the EM product code.

- Alternatively, you can log into the Dashboard Application Services Hub for each domain and display the dashboard data provider connection information. In a typical environment, you might see that the provider ID has been set to `ITMSD` instead of the hub name. However, the connection name is the hub name if it was not customized when the connection was created.

Once you know your the name of the hub monitoring server then your domain name is `itm.`*`hub_monitoring_server_name`*, for example `itm.HUB_server1`.

Deployment scenarios

The deployment scenarios in this topic can help you make decisions in your environment.

Multiple domains with shared roles and authorization policies

This deployment scenario uses similar strategies for user management and setup as a single domain deployment, but with the added ability to target authorization policies for specific IBM Tivoli Monitoring domains.

This deployment scenario is useful if you want to share the same authorization policy administration infrastructure for a set of domains. It allows you to create a set of common authorization policies for all domains as well as policies that are specific to one or more domains. When you grant or exclude a permission for a role, you specify whether the policy applies to all domains or a specific domain. If you do not specify a domain name with the **`tivcmd grant`**, **`tivcmd exclude`**, or **`tivcmd revoke`** commands, then the policy applies to all domains. To create a domain specific policy use the `--domain` argument with these commands. For more information about the tivcmd CLI commands, see the *IBM Tivoli Monitoring Command Reference*.

Preparation for deployment

The following table describes what you need for this deployment:

Table 23. Multiple domains with shared roles and policies deployment requirements

Quantity	Component	Description
1 per domain **Note:** If load balancing is used, there can be multiple dashboard services hubs per domain.	Dashboard Application Services Hub	Dashboard applications such as Infrastructure Management Dashboards for Servers are also installed with each domain's dashboard services hub.
1	Tivoli Authorization Policy Server	The Authorization Policy Server can either be installed with Dashboard Application Services Hub for one of your domains or you can install an instance of Dashboard Application Services Hub that is just used for authorization policy administration and that does not have any dashboard applications installed.

Table 23. Multiple domains with shared roles and policies deployment requirements (continued)

Quantity	Component	Description
1 per domain	Hub Tivoli Enterprise Monitoring Server	If Hot Standby is being used, there can be two hub monitoring servers per domain. Each hub monitoring server can have multiple remote monitoring servers connected to it. The monitoring agents are connected to the monitoring servers.
1 per domain	Tivoli Enterprise Portal Server	Each portal server is configured to retrieve authorization policies from the same Authorization Policy Server since there is one policy server that is being shared by all domains.
1 or more	LDAP user registry	Configure all of the portal servers and each Dashboard Application Services Hub to use the same set of LDAP user registries in order to share authorization policies.

Multiple domains with independent authorization policies

If you plan to not have any common authorization policies across your domains and you also plan to have a separate set of authorization policy administrators per domain, install an Authorization Policy Server with the Dashboard Application Services Hub in each domain and then manage the authorization policies independently.

In this scenario, do not use the `--domain` argument because each Authorization Policy Server is only managing policies for a single domain.

Creating policies for specific IBM Tivoli Monitoring domains

In a multi-domain deployment, best practice is to create policies for users through roles that differentiates access by the domain.

Any policies that are used for specific IBM Tivoli Monitoring domains must have the following permission:

Managed System Group or Managed System permission definition	
Parameter	**Value**
Domain	'any' or a specific domain name **Note:** Specifying a value of 'any' or omitting the domain parameter, indicates the permission applies to all domains.
Operation	'view'
Object Type	'attributegroup', 'event'
Resource Type	'managedsystemgroup', 'managedsystem'
Resource	*managed_system_name* or *managed_system_group_name*

Example: Creating common policies and domain specific policies

This example demonstrates how to grant a user access to all the UNIX OS agents from one particular domain but not those of another domain. The *ALL_UNIX managed system group is an created automatically and managed by each hub monitoring server. In addition, an administrator is granted access to all UNIX OS systems across all domains.

The following managed system groups are used in this example:

Managed System Groups		
Type	**Domain**	**Name**
managedsystemgroup	itm.eastcoast	*ALL_UNIX
managedsystemgroup	itm.westcoast	*ALL_UNIX

The following roles are used in this example:

- EastCoastOperators
- WestCoastOperators
- SuperAdministrator

1. Define a user in LDAP, for example `uid=John,cn=itm,o=ibm` and define a group in LDAP, for example `cn=EastCoastMachineUsers,cn=itm,o=ibm`, and then add the user ID to the group in LDAP.
2. Define a user in LDAP, for example `uid=Jane,cn=itm,o=ibm` and define a group in LDAP, for example `cn=WestCoastMachineUsers,cn=itm,o=ibm`, and then add the user ID to the group in LDAP.
3. Define a user in LDAP, for example `uid=Joe,cn=itm,o=ibm` and define a group in LDAP, for example `cn=SuperAdministratorUsers,cn=itm,o=ibm`, and then add the user ID to the group in LDAP.
4. Create new roles:

```
tivcmd createrole --rolename EastCoastOperators --description
"East Coast users with permission to access the east coast machine
ITM Domain"

tivcmd createrole --rolename WestCoastOperators --description
"West Coast users with permission to access the west  coast machines
for the itm.westcoast domain"

tivcmd createrole --rolename SuperAdministrator --description
"Users with permission to access machines from all domains"
```

5. Grant attribute group and event access to the *ALL_UNIX managed system group for the itm.eastcoast domain to the EastCoastOperators role:

```
tivcmd grant --rolename EastCoastOperators --resourcetype
managedsystemgroup --resources "*ALL_UNIX"
--objecttype attributegroup --operations view --domain itm.eastcoast

tivcmd grant --rolename EastCoastOperators --resourcetype
managedsystemgroup --resources "*ALL_UNIX"
--objecttype event --operations view --domain itm.eastcoast
```

6. Grant attribute group and event access to the *ALL_UNIX managed system group for the itm.westcoast domain to the WestCoastOperators role:

```
tivcmd grant --rolename WestCoastOperators --resourcetype
managedsystemgroup --resources "*ALL_UNIX" --objecttype
attributegroup --operations view --domain itm.westcoast
```

```
tivcmd grant --rolename WestCoastOperators --resourcetype
managedsystemgroup --resources "*ALL_UNIX" --objecttype event
--operations view --domain itm.westcoast
```

7. Grant attribute group and event access to the *ALL_UNIX managed system group for all domains to the SuperAdministrator role:

   ```
   tivcmd grant --rolename SuperAdministrator --resourcetype
   managedsystemgroup --resources "*ALL_UNIX"
   --objecttype attributegroup --operations view  --domain any

   tivcmd grant --rolename SuperAdministrator --resourcetype
   managedsystemgroup --resources "*ALL_UNIX"
   --objecttype event --operations view --domain any
   ```

8. Assign the user groups to the new roles:

   ```
   tivcmd addtorole --rolename EastCoastOperators --groups
   cn=EastCoastMachineUsers,cn=itm,o=ibm

   tivcmd addtorole --rolename WestCoastOperators --groups
   cn=WestCoastMachineUsers,cn=itm,o=ibm

   tivcmd addtorole --rolename SuperAdministrator --groups
   cn=SuperAdministrator,cn=itm,o=ibm
   ```

When a user in the EastCoastOperators group accesses the Server Dashboards page in the Dashboard Application Services Hub for the itm.eastcoast domain, they see the *ALL_UNIX managed system group and its members for this domain. If the same user logs into the Dashboard Application Services Hub for the itm.westcoast domain, they will not see the *ALL_UNIX managed system group.

When a user in the WestCoastOperators group accesses the Server Dashboards page in the Dashboard Application Services Hub for the itm.westcoast domain, they see the *ALL_UNIX managed system group and its members for this domain. If the same user logs into the Dashboard Application Services Hub for the itm.eastcoast domain, they will not see the *ALL_UNIX managed system group.

When a user in the SuperAdministrator group accesses the Server Dashboards page in the Dashboard Application Services Hub for either domain, they see the *ALL_UNIX managed system group and its members for domain connected to the dashboard server.

Example: Creating authorization policies for common managed system groups

If you have the same managed system group names in multiple domains and you want dashboard users to view data from those managed system groups for all domains, create a role and grant permissions as shown in the example commands below:

```
tivcmd createrole --rolename  WindowsDataCenterOperators

tivcmd grant --rolename WindowsDataCenterOperators  --operations view
--objecttype attributegroup  --resources DataCenterServers
--resourcetype managedsystemgroup

tivcmd grant --rolename WindowsDataCenterOperators  --operations view
--objecttype event   --resources DataCenterServers
--resourcetype managedsystemgroup
```

Because the `--domain` argument is not specified on the grant command example above, the authorization policy applies to all domains. As a result, any user or user group assigned to the WindowsAdministrators role can view data from the DataCenterServers managed system group in all domains.

Example: Using a common role to manage domain-specific resources

If you do not have the same managed system group names in multiple domains, but you have users or user groups that perform the same role for multiple domains, you can create a common role with domain-specific permissions as shown in the example commands below:

```
tivcmd createrole --rolename  LinuxRegionalOperators

tivcmd grant --rolename LinuxRegionalOperators  --operations view
--objecttype attributegroup  --resources SeattleServers
--resourcetype managedsystemgroup --domain itm.HUB_west

tivcmd grant --rolename LinuxRegionalOperators  --operations view
--objecttype event   --resources SeattleServers
--resourcetype managedsystemgroup  --domain itm.HUB_west

tivcmd grant --rolename LinuxRegionalOperators  --operations view
--objecttype attributegroup  --resources BostonServers
--resourcetype managedsystemgroup --domain itm.HUB_east

tivcmd grant --rolename LinuxRegionalOperators  --operations view
--objecttype event   --resources AustinServers
--resourcetype managedsystemgroup  --domain itm.HUB_east
```

In this case, any user or user group assigned to the LinuxRegionalOperators role can view data from the managed system group SeattleServers when they are logged into the Dashboard Application Services Hub for the itm.HUB_west domain and can view the data from the managed system group BostonServers when they are logged into the Dashboard Application Services Hub for the itm.HUB_east domain.

Example: Creating a domain specific authorization policy

For those roles that are not common across domains, you can create a role that only has permissions for a single domain as shown in the example commands below:

```
tivcmd createrole --rolename  ChicagoDataCenterOperators

tivcmd grant --rolename ChicagoDataCenterOperators   --operations view
--objecttype attributegroup  --resources ChicagoServers
--resourcetype managedsystemgroup --domain itm.HUB_midwest

tivcmd grant --rolename ChicagoDataCenterOperators   --operations view
--objecttype event   --resources ChicagoServers
--resourcetype managedsystemgroup  --domain itm.HUB_midwest
```

In this scenario, a user or user group assigned to the ChicagoDataCenterOperators role can only view data from a managed system group in a single domain.

Chapter 8. Securing communications

To secure communication between Tivoli Enterprise Monitoring Agents, Tivoli Enterprise Monitoring Servers, and the Tivoli Enterprise Portal Server, use SPIPE as the protocol when you configure communications between the portal server and the hub monitoring server, between hub and remote monitoring servers, and between monitoring agents and monitoring servers.

Two additional protocols are used to secure communication between Tivoli Enterprise Portal clients and the portal server:

- Secure Hypertext Transport Protocol (HTTPS) to retrieve files and Interoperable Object Reference (IOR)
- Internet Inter-ORB Protocol (IIOP) to secure the communications between the portal server and client

Note: By default, both protocols are used. However, you can configure a portal client to use just HTTPS to communicate with the portal server.

HTTPS can also be used to secure communication between these components:

- Dashboard Application Services Hub and the IBM Tivoli Monitoring dashboard data provider
- tacmd Command-Line Interface and the hub Tivoli Enterprise Monitoring Server
- tivcmd Command-Line Interface for Authorization Policy and the Dashboard Application Services Hub where the Tivoli Authorization Policy Server is installed
- The Open Services Lifecycle Collaboration Performance Monitoring service provider and Registry Services, Security Services, and OSLC clients
- Tivoli Integrated Portal and the portal server's IBM Tivoli Monitoring charting web service

In addition, these types of secure communication are also supported:

- Using TLS/SSL to secure communication between the hub monitoring server and an LDAP server
- Using TLS/SSL to secure communication between the portal server and an LDAP server
- Using TLS/SSL to secure communication between a monitoring agent and the IBM IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF Version 12 or later

In addition to choosing a protocol such as IP.SPIPE or HTTPS that supports secure communications, you set up TLS/SSL asymmetric encryption through the use of public-private key files, which involves performing the following tasks:

- Working with a key database
- Requesting a new public-private key pair if you do not want to use the self-signed certificate shipped with the product
- Adding a certificate authority signer certificate and signed digital certificate to your key database if you do not want to use the self-signed certificates shipped with the product
- Adding the signer certificates for the applications that IBM Tivoli Monitoring components send requests to

- Enabling components to perform certificate authentication

Note: Requesting new certificates is best practice, but you can also use the self-signed certificates shipped with the product in a test environment to become familiar with the procedures for setting up secure communications.

IBM Tivoli Monitoring provides two applications that are used to work with keys and certificate stores when setting up secure communications:

- The Global Security Toolkit (GSKit) program is installed with IBM Tivoli Monitoring components on distributed platforms. It includes the iKeyman utility and a command-line interface for working with certificates and keys.
- The Tivoli Enterprise Portal Server extended services (TEPS/e) administration console (also called ISCLite) is used with the portal server to secure communications for the services running in TEPS/e.

A default self-signed certificate and key are provided when you install IBM Tivoli Monitoring. If you prefer to use a certificate authority signed certificate, use the GSKit utilities to create a certificate request, and then create a key database and import the certificates. A stash file provides the key database password for unattended operation. When GSKit is installed with an IBM Tivoli Monitoring component, the key file names are specified using the following environment variables:

- KDEBE_KEYRING_FILE=C:\IBM\ITM\keyfiles\keyfile.kdb
- KDEBE_KEYRING_STASH=C:\IBM\ITM\keyfiles\keyfile.sth
- KDEBE_KEY_LABEL=IBM_Tivoli_Monitoring_Certificate

If the keyring file, stash file, or label used for the new certificate in the key store is changed, you must complete the following steps:

1. Update all configuration files with the respective environment variable. For example, with the environment variable `KDEBE_KEY_LABEL=`*`Custom_Certificate_Label_Name`*, you would update the following files:

   On Linux and UNIX, update the agent configuration files (.ini) files directly.

   On Windows, update the variables using Manage Tivoli Enterprise Monitoring Services or update the agent environment files (*ENV) files directly.
2. Update the Tivoli Enterprise Monitoring Server `ms_<`*`hub_monitoring_server`*`>.config` file with the same variables but with the values in single quotes ('). For example, `KDEBE_KEY_LABEL='`*`Custom_Certificate_Label_Name`*`'`.
3. Restart each component.

Work with the administrators of the other products that IBM Tivoli Monitoring communicates with to setup secure communications. If you are using any of the Jazz for Service Management components (Dashboard Application Services Hub, Registry Services, or Security Services) with IBM Tivoli Monitoring, use the WebSphere Application Server administration console to work with their trust and certificate stores.

The following table lists the communication flows that can be secured and where to find information on how to secure the interaction.

Note: Unless otherwise stated, the tasks below are used to setup TLS/SSL and server certificate authentication. When server certificate authentication is used, the client (the source of the request) authenticates the certificate it receives from the server (the target of the request).

Table 24. Tasks to secure communication

Task to secure communication	Where to find information
Use TLS/SSL between the Tivoli Enterprise Portal clients and the portal server.	See "Using SSL between the portal server and the client" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
Use IP.SPIPE with certificate validation to secure communication for these interactions: • hub and remote monitoring server communication • hub monitoring server and portal server communication • monitoring server and monitoring agent communication Use HTTPS with certificate validation to secure communications for these interactions: • tacmd CLI or SOAP client to hub monitoring server communication • requests to the monitoring server, portal server, and monitoring agent service console	See the ITM Certificate Authentication Configuration Guide for ITM V6.2.2 and later in the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring).
Use TLS/SSL between the hub monitoring server and a LDAP server.	"Configuring TLS/SSL communication between the hub monitoring server and the LDAP server" on page 212
Use TLS/SSL between the portal server and a LDAP server.	"Configuring TLS/SSL communication between the portal server and the LDAP server" on page 119
Use TLS/SSL when the IBM Dashboard Application Services Hub sends requests to the IBM Tivoli Monitoring dashboard data provider.	"Configuring TLS/SSL communication between Dashboard Application Services Hub and the dashboard data provider" on page 213
Use TLS/SSL when the dashboard data provider sends requests to retrieve authorization policies from the Authorization Policy Server.	"Configuring TLS/SSL communication with the Authorization Policy Server" on page 216
Use TLS/SSL when the tivcmd Command-Line Interface for Authorization Policy sends requests to the Authorization Policy Server.	"Configuring TLS/SSL communication with the Authorization Policy Server" on page 216
Use TLS/SSL for sending private situation events from monitoring agents to the IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF. For this interaction, client certificate authentication is configured so that the probe uses certificates to authenticate the monitoring agents (the clients).	"Sending private situation events by using TLS/SSL communication" on page 393

Table 24. Tasks to secure communication (continued)

Task to secure communication	Where to find information
Use TLS/SSL when Tivoli Business Service Manager or Tivoli Integrated Portal send HTTPS requests to the portal server's charting web service.	"Tivoli Business Service Manager and Tivoli Enterprise Portal Server integration over SSL" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
Enable the Federal Information Processing Standard (FIPS) for IBM Tivoli Monitoring components.	"Enabling FIPS for IBM Tivoli Monitoring" on page 228
After updating the IBM Tivoli Monitoring certificate, import the TEPS/e certificates into the portal server keyfile database to ensure the portal server web server plug-in and TEPS/e can continue to communicate securely.	"Importing the TEPS/e certificates into the portal server keyfile database" on page 240
If you want to use HTTPS in a dashboard environment that is using an HTTP server to load balance multiple portal servers, you must configure TLS/SSL communication from the HTTP server to the portal server.	"Configuring TLS/SSL communication between the load balancing HTTP Server and each portal server's local HTTP server" on page 223
If you want to use HTTPS in a dashboard environment that is using an HTTP server to load balance multiple portal servers, you must configure TLS/SSL communication from the Dashboard Application Services Hub to the HTTP server.	"Configuring TLS/SSL communication between Dashboard Application Services Hub and an HTTP server used for load balancing multiple portal servers" on page 226

Configuring TLS/SSL communication between the hub monitoring server and the LDAP server

You can configure TLS/SSL communication from the hub monitoring server to an LDAP server to secure requests to authenticate users and groups.

After setting up the LDAP server for TLS/SSL and obtaining its public signer certificate, use the hub monitoring server's GSKit iKeyman utility or command line interface to set up a new key database of type `CMS` and a stash file containing the password for the key database. Then import the LDAP server's public signer certificate into the new key database and specify a label name for the certificate. See "Using the GSKit command-line interface to work with key databases and certificates" on page 241 and "Using the GSKit iKeyman utility to work with key databases and certificates" on page 242 for information on using GSKit.

Then reconfigure the hub monitoring server to enable LDAP TLS/SSL communication. When reconfiguring the hub monitoring server, you must provide the location of the key database (also called the LDAP key store file), the stash file containing the key database (also called the LDAP key store stash), the label name for the public signer certificate, and the password of the key database. Also check with the LDAP server administrator to determine if you should modify the LDAP port value since the secured port number is typically port `636`.

Note:

LDAP TLS/SSL requires some actions by an LDAP administrator that are not covered by the Tivoli Monitoring documentation. The following topics in the IBM Security Systems Information Center include information about setting up LDAP servers for TLS/SSL:

- Configuring Microsoft Active Directory for SSL access
- Configuring the Tivoli Directory Server client for SSL access
- Configuring Oracle Java System Directory Server for SSL access

Configuring TLS/SSL communication between Dashboard Application Services Hub and the dashboard data provider

If you want to use HTTPS, you can configure TLS/SSL communication from Dashboard Application Services Hub to the dashboard data provider in the portal server.

The Dashboard Application Services Hub communicates with the IBM Tivoli Monitoring dashboard data provider using either Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS). HTTPS is intended to run on top of Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL). These layers provide encryption using key exchanges.

Roadmap

In order to use HTTPS and its security encryption features, complete the following tasks in the roadmap.

Table 25. Roadmap for setting up TLS/SSL for the dashboard data provider

Step	Description and information provided
1	You have two options for obtaining the public-private key pair used by the portal server: • Use the default self-signed certificates installed with IBM Tivoli Monitoring. If you choose this option, proceed to step 2. OR • Use a digital certificate that has been signed by a certificate authority. In this case, you must create a certificate request and send it to the certificate authority for signing. Once the digital certificate has been signed, you add the certificate authority signer's certificate to the portal server's trust stores, and then add the new digital signature to the portal server's key stores. For more information, see “Using third party certificate authority signed certificates for the portal server.”
2	At each Dashboard Application Services Hub with a connection configured to the portal server's dashboard data provider, add the public signer certificate used by the portal server to the Dashboard Application Services Hub WebSphere trust store. Follow the steps in “Configuring TLS/SSL communication for the Dashboard Application Services Hub server” on page 215.

Using third party certificate authority signed certificates for the portal server

You can use third party certificates to configure TLS/SSL for the dashboard data provider by adding the signer certificate and private digital certificate to the key database managed by GSKit, and to the trust and key stores used by TEPS/e.

Before you begin

Obtain the certificate authority's signer certificate.

Ensure the TEPS/e administration console is enabled. For detailed steps, including information on how to log on, see "Starting the TEPS/e administration console" on page 114.

Procedure

1. Use either the TEPS/e administration console or the GSKit command-line interface to create a private certificate request to be signed by the certificate authority. The following instructions explain how to perform this step using the TEPS/e administration console.
   a. Log on to the TEPS/e administration console.
   b. Select **Security → SSL certificate and key management**.
   c. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultKeyStore** link.
   d. In the Additional Properties area, click the **Personal certificate requests** link and in the page that is displayed, click **New**.
   e. In the page that is displayed specify the following information:
      - Set **File name** to the location to store the private certificate request. For example, `C:\dashboardcerts\TEPSCertRequest.arm`.
      - Set the **Key label** to the desired label for the certificate. For example, `TEPS Certificate.`
      - Set the **Key size** to `2048`.
      - Leave the **Signature algorithm** as `SHA1withRSA`.
      - Set the **Common name** to a unique name for the TEPS/e computer. Typically, this is a hostname.
      - Set **Organization** to a meaningful value. Typically, this is a company name.
      - Set **Organization unit** to a meaningful name. For example, `TEPS`.
      - Set **Country or region** to desired value. For example, `US`.
   f. Click **OK**, then **Save**.
2. Send the certificate request generated above to the certificate authority to request a new digital certificate. The certificate authority can take two to three weeks to generate the new digital certificate.
3. After the certificate authority returns your new digital certificate, save it to a location on the computer where the portal server and TEPS/e are installed. For example, `C:\dashboardcerts\TEPSSignedCert.arm`.
4. Use the GSKit command-line interface to create a new key database of type `CMS` and save the key database's password to a stash file. Then import the certificate authority's signer certificate and the new digital certificate into the new key database. This key database is used by the portal server's embedded HTTP server.
5. You must also add the certificate authority public signer certificate into the TEPS/e trust store using the TEPS/e administration console.
   a. Log on to the TEPS/e administration console.
   b. Select **Security → SSL certificate and key management**.
   c. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.

d. In the Additional Properties area, click the **Signer certificates** link and in the page that is displayed, click **Add**.

e. In the page that is displayed specify the following information:

   - Set **Alias** to the desired label for the certificate. For example, `TEPS Signer Certificate`.
   - Set **File name** to the location of the extracted certificate authority signer certificate. For example, `C:\dashboardcerts\CASignerCert.arm`.
   - Leave the **Data type** as **Base64-encoded ASCII data**.

f. Click **OK**, then **Save**.

6. Receive the signed digital certificate into the TEPS/e key store using the TEPS/e administration console.

   a. Log on to the TEPS/e administration console.

   b. Select **Security → SSL certificate and key management**.

   c. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultKeyStore** link.

   d. In the Additional Properties area, click the **Personal certificates** link and in the page that is displayed, click **Receive from a certificate authority**.

   e. In the page that is displayed specify the following information:

      - Set **File name** to the location of the signed digital certificate. For example, `C:\dashboardcerts\TEPSSignedCert.arm`.
      - Leave the **Data type** as **Base64-encoded ASCII data**.

   f. Click **OK**, then **Save**.

7. Set the new private certificate as the default server certificate for TEPS/e.

   a. Log on to the TEPS/e administration console.

   b. Select **Security → SSL certificate and key management**.

   c. In the Related Items area, click the **SSL configurations** link and in the table click the **NodeDefaultSSLSettings** link.

   d. In the page that is displayed, click **Default server certificate alias** and choose the signed TEPS/e certificate. For example, `TEPS Certificate`.

   e. Click **OK**, then **Save**.

   f. Select **Security → SSL certificate and key management** again.

   g. Click on the **Manage endpoint security configurations** link.

   h. Click on the node name link under **Inbound → thecellname → nodes**.

   i. Click **Certificate alias in key store** and choose the signed TEPS/e certificate. For example, `TEPS Certificate`.

   j. Click **OK**, then **Save**.

Configuring TLS/SSL communication for the Dashboard Application Services Hub server

Add the public signer certificate used by the portal server to the Dashboard Application Services Hub WebSphere trust store to configure TLS/SSL.

Note: If you requested a new digital certificate for the portal server, wait until the certificate has been received before performing this procedure.

Procedure

1. Log into the Dashboard Application Services Hub WebSphere Administrative Console.

   a. Enter the following URL in your Internet Explorer or Firefox browser: `https://hostname:16311/ibm/console`.

      If your environment was configured with a port number other than the default, enter that number instead. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.
   b. Enter the Dashboard Application Services Hub administrative user ID and password then click **Go**.

      The user ID must be assigned the `administrator` and `iscadmins` roles.
   c. In the Console Settings area click on **WebSphere Administrative Console** and then click the **Launch WebSphere administrative console** button.
2. Select **Security → SSL certificate and key management**.
3. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
4. In the Additional Properties area, click the **Signer certificates** link and in the page that is displayed, click **Retrieve from port**.
5. Enter the hostname of the portal server.
6. Enter port `15201`.
7. Enter an alias name, for example `ITM-TEPS`.
8. Click **Retrieve signer information**.
9. Click **OK**, then **Save**.

Results

The certificates are now setup for communication between the Dashboard Application Services Hub server and to the portal server and its dashboard data provider.

Return to the roadmap instructions in Chapter 3, “Preparing your dashboard environment,” on page 29 for information on how to reconfigure the dashboard data provider connection to use HTTPS instead of HTTP.

Configuring TLS/SSL communication with the Authorization Policy Server

If you want to use HTTPS, you can configure TLS/SSL communication with the Tivoli Authorization Policy Server.

There are two IBM Tivoli Monitoring components which communicate with the Authorization Policy Server using either Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS):

- The tivcmd Command-Line Interface for Authorization Policy sends HTTP/HTTPS requests to the Authorization Policy Server to process CLI commands.
- The Tivoli Enterprise Portal Server sends HTTP/HTTPS requests to the Authorization Policy Server to obtain the latest policy store.

HTTPS is intended to run on top of Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL). These layers provide encryption using key exchanges.

Roadmap

In order to use HTTPS and its security encryption features, complete the following tasks in the roadmap.

Note: The following instructions assume that the portal server and the tivcmd CLI send requests directly to the IBM Dashboard Application Services Hub application server, and not to a HTTP server that might be used in conjunction with the dashboard hub. If you are using a HTTP server with IBM Dashboard Application Services Hub, then you must also update the certificates that the HTTP server uses.

Table 26. Roadmap for setting up TLS/SSL for the Authorization Policy Server

Step	Description and information provided
1	Using the WebSphere Application Server administrative console for the Dashboard Application Services Hub where the Authorization Policy Server is installed, you can choose one of the following options to obtain a public-private key pair: • "Using the WebSphere generated certificates to configure TLS/SSL for the Authorization Policy Server" During installation, the WebSphere Application Server generates a public signer certificate and a default private signed certificate. These certificates can be used if desired. • "Using third party certificates to configure TLS/SSL for the Authorization Policy Server" on page 218 Add the third party's signer certificate to the WebSphere Application Server trust store. A certificate request is created at the WebSphere Application Server and forwarded to the certificate authority for signing. Once signed, it is added to the WebSphere Application Server key store. The private signed certificate must be set as the default certificate.
2	At each tivcmd Command-Line Interface for Authorization Policy installation: 1. Create a new clean key database. 2. Add the public signer certificate used by the Authorization Policy Server to the new key database. 3. Set an environment variable to enable validation of the server certificate. By default, HTTPS used between the tivcmd CLI and the Authorization Policy Server does not exchange certificates or use security encryption. This environment variable must be set to make this happen. Follow the steps in "Configuring the tivcmd CLI for TLS/SSL" on page 220.
3	At each portal server configured to communicate with the Authorization Policy Server, add the public signer certificate used by the Authorization Policy Server to the TEPS/e trust store. Follow the steps in "Configuring TLS/SSL communication between the portal server and the Authorization Policy Server" on page 222.
4	Use the `-s` argument for the **`tivcmd login`** command to indicate that the HTTPS protocol is used when sending requests to the Authorization Policy Server. If the tivcmd CLI environment variable `ITM_AUTHENTICATE_SERVER_CERTIFICATE` is set to `Y`, then the tivcmd CLI will validate the certificate of the Authorization Policy Server before accepting the connection. The public key of the Authorization Policy Server must be imported into the client keystore.

Using the WebSphere generated certificates to configure TLS/SSL for the Authorization Policy Server

During the installation of the WebSphere Application Server used by the Authorization Policy Server and Dashboard Application Services Hub, a public signer certificate and a default private signed certificate are generated. You can use these certificates for TLS/SSL communication by extracting the public signer certificate.

Procedure

1. Log into the WebSphere Administrative Console for the Authorization Policy Server and Dashboard Application Services Hub.
   a. Enter the following URL in your Internet Explorer or Firefox browser: `https://hostname:16311/ibm/console`.

      If your environment was configured with a port number other than the default, enter that number instead. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.
   b. Enter the Dashboard Application Services Hub administrative user ID and password then click **Go**.

      The user ID must be assigned the `administrator` and `iscadmins` roles.
   c. In the Console Settings area click on **WebSphere Administrative Console** and then click the **Launch WebSphere administrative console** button.
2. Select **Security → SSL certificate and key management**.
3. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
4. In the Additional Properties area, click the **Signer certificates** link and in the table that is displayed, select the `root` entry check box.
5. Click **Extract** and in the page that is displayed, in the **File name** field, enter a certificate file name. For example, `C:\policyauthcerts\PolicyAuthServerSignerCert.arm`.
6. From the **Data type** list select the **Base64-encoded ASCII data** option and click **OK**.

What to do next

The extracted public signer certificate can now be distributed to the portal server and tivcmd Command-Line Interface for Authorization Policy computers for importing.

Using third party certificates to configure TLS/SSL for the Authorization Policy Server

You can use third party certificates to configure TLS/SSL for the Authorization Policy Server.

Before you begin

Many of the steps below require you to be logged in to the WebSphere Administrative Console for the Authorization Policy Server and Dashboard Application Services Hub. Use the following steps to log into the console:

1. Enter the following URL in your Internet Explorer or Firefox browser: `https://hostname:16311/ibm/console`.

   If your environment was configured with a port number other than the default, enter that number instead. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.

2. Enter the Dashboard Application Services Hub administrative user ID and password then click **Go**.

   The user ID must be assigned the `administrator` and `iscadmins` roles.
3. In the Console Settings area click on **WebSphere Administrative Console** and then click the **Launch WebSphere administrative console** button.

Procedure

- Add the certificate authority public signer certificate to the WebSphere Application Server trust store.
  1. Select **Security → SSL certificate and key management**.
  2. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
  3. In the Additional Properties area, click the **Signer certificates** link and in the page that is displayed, click **Add**.
  4. In the page that is displayed specify the following information:
     - Set **Alias** to the desired label for the certificate. For example, `Authorization Policy Server Signer Certificate`.
     - Set **File name** to the location of the certificate authority signer certificate. For example, `C:\policyauthcerts\CASignerCert.arm`.
     - Leave the **Data type** as **Base64-encoded ASCII data**.
  5. Click **OK**, then **Save**.

  The certificate authority public signer certificate can now be distributed to the portal server and tivcmd CLI computers for importing.
- Create a private certificate request to be signed by the certificate authority.
  1. Select **Security → SSL certificate and key management**.
  2. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultKeyStore** link.
  3. In the Additional Properties area, click the **Personal certificate requests** link and in the page that is displayed, click **New**.
  4. In the page that is displayed specify the following information:
     - Set **File name** to the location to store the private certificate request. For example, `C:\policyauthcerts\PolicyAuthServerCertRequest.arm`.
     - Set the **Key label** to the desired label for the certificate. For example, `Authorization Policy Server Certificate.`
     - Leave the **Signature algorithm** as **SHA1withRSA**.
     - Set the **Key size** to 2048.
     - Set the **Common name** to a unique name for the Authorization Policy Server. Typically, this is a computer name.
     - Set **Organization** to a meaningful value. Typically, this is a company name.
     - Set **Organization unit** to a meaningful name. For example, `PolicyAuth`.
     - Set **Country or region** to desired value. For example, `US`.
  5. Click **OK**, then **Save**.

  Send the certificate request generated above to the certificate authority to request a new digital certificate. The certificate authority can take two to three weeks to generate the new digital certificate.

  After the certificate authority returns your new digital certificate, save it to a location on the Authorization Policy Server computer. For example, `C:\policyauthcerts\PolicyAuthServerSignedCert.arm`.

- Receive the signed digital certificate using the WebSphere Administrative Console.
  1. Select **Security → SSL certificate and key management**.
  2. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultKeyStore** link.
  3. In the Additional Properties area, click the **Personal certificates** link and in the page that is displayed, click **Receive from a certificate authority**.
  4. In the page that is displayed specify the following information:
     - Set **File name** to the location of the signed digital certificate. For example, `C:\policyauthcerts\PolicyAuthServerSignedCert.arm`.
     - Leave the **Data type** as **Base64-encoded ASCII data**.
  5. Click **OK**, then **Save**.
- Set the new private certificate as the default server certificate.
  1. Select **Security → SSL certificate and key management**.
  2. In the Related Items area, click the **SSL configurations** link and in the table click the **NodeDefaultSSLSettings** link.
  3. In the page that is displayed, click **Default server certificate alias** and choose the signed Authorization Policy Server certificate. For example, `Authorization Policy Server Certificate`.
  4. Click **OK**, then **Save**.
  5. Select **Security → SSL certificate and key management** again.
  6. Click on the **Manage endpoint security configurations** link.
  7. Click on the node name link under **Inbound → thecellname → nodes**.
  8. Click **Certificate alias in key store** and choose the signed Authorization Policy Server certificate. For example, `Authorization Policy Server Certificate`.
  9. Click **OK**, then **Save**.

Configuring the tivcmd CLI for TLS/SSL

In order to use TLS/SSL with the Authorization Policy Server you must prepare the tivcmd Command-Line Interface, create a new key database, add the public signer certificate used by the Authorization Policy Server to the new key database, and then modify the tivcmd CLI environment variable file.

Note: If you requested a digital certificate for the Authorization Policy Server, wait until the certificate has been received before performing this procedure.

Before you begin

The following instructions for managing certificates on the tivcmd CLI computers use the GSKit command line tool that is installed with the tivcmd CLI component. These instructions should be followed on each computer that the tivcmd CLI is installed on.

See "Using the GSKit command-line interface to work with key databases and certificates" on page 241 for terms that are used in this procedure. Most terms are based upon the directory to which the tivcmd CLI is installed.

Procedure

1. Set the path to invoke the GSKit command line tool using the following commands:

- Windows 64-bit:

  ```
  set PATH=<gskithome>\lib64;%PATH%
  cd <gskithome>\bin
  ```

- Windows 32-bit:

  ```
  set PATH=<gskithome>\lib;%PATH%
  cd <gskithome>\bin
  ```

- Linux and UNIX 32-bit:

  ```
  export LD_LIBRARY_PATH=<gskithome>/lib:$LD_LIBRARY_PATH
  cd <gskithome>/bin
  ```

- Linux and UNIX 64-bit:

  ```
  export LD_LIBRARY_PATH=<gskithome>/lib64:$LD_LIBRARY_PATH
  cd <gskithome>/bin
  ```

2. Save the existing tivcmd CLI key database.

   In order to recover issues, best practice is to save the installed version of the tivcmd CLI key database on each tivcmd CLI computer.

   Copy the following files with extensions `.crl`, `.kdb`, `.rdb`, and `.sth`, to another location:

   - Windows: `<keydbdir>\<oldkeydbname>.*`
   - Linux and UNIX: `<keydbdir>/<oldkeydbname>.*`

3. Create a new tivcmd CLI key database.

   a. Create a new database and remove all the extraneous public signer certificates with the following command:

      ```
      <gskittoolcmd> -keydb -create -db <newkeydb>
      -pw <newkeydbpw> -expire 3650 -stash -fips
      ```

   b. Verify the database is empty with the following command:

      ```
       <gskittoolcmd> -cert -list -db <newkeydb>
      -pw <newkeydbpw> -fips
      ```

      If the database is not empty, use the delete command to remove any remaining certificates.

4. Add the public signer certificate to the new tivcmd CLI key database.

   This step assumes that the public signer certificate has been placed in a location on the tivcmd CLI computer. For example, `C:\policyauthcerts\PolicyAuthSignerCert.arm` or `C:\policyauthcerts\CASignerCert.arm`. This location is referenced in this step as *<policyauthsignercert>*.

   Add the public signer certificate to the new tivcmd CLI key database using the following command:

   ```
   <gskittoolcmd> -cert -add -db <newkeydb> -pw <newkeydbpw>
   -label "Authorization Policy Signer Certificate" -trust enable
   -format ascii -file <policyauthsignercert> -fips
   ```

5. Enable TLS/SSL certificate exchange at each tivcmd CLI computer.

   At each tivcmd CLI computer, use the following steps to enable TLS/SSL certificate exchange using the public signed certificate.

   a. Delete the current key database. Remove the `<oldkeydbname>.*` files in the `<keydbdir>` directory.

   b. Rename all new key database files. For example, `<newkeydbname>.*` to `<oldkeydbname>.*` in the `<keydbdir>` directory.

   c. Set the environment variable to enable authentication of the Authorization Policy Server certificate.

- Windows: Edit the tivcmd CLI environment file <*authclidir*>`\KDQ\bin\KDQENV` by adding the variable `ITM_AUTHENTICATE_SERVER_CERTIFICATE=Y` after the `KDEBE_KEY_LABEL` variable.
- Linux and UNIX: Edit the tivcmd CLI environment file <*authclidir*>`/bin/tivcmd` by adding the variable `export ITM_AUTHENTICATE_SERVER_CERTIFICATE=Y` after the `KDEBE_KEY_LABEL` variable.

Configuring TLS/SSL communication between the portal server and the Authorization Policy Server

Add the public signer certificate used by the Tivoli Authorization Policy Server to the portal server's TEPS/e trust store to configure TLS/SSL.

Note: If you requested a new digital certificate for Authorization Policy Server, wait until the certificate has been received before performing this procedure.

Before you begin

Ensure the TEPS/e administration console is enabled. For detailed steps, including information on how to log on, see “Starting the TEPS/e administration console” on page 114.

About this task

This step assumes that the public signer certificate is located on the portal server computer. For example, `C:\policyauthcerts\PolicyAuthSignerCert.arm` or `C:\policyauthcerts\CASignerCert.arm`. This location is referenced in this procedure as <*policyauthsignercert*>.

Procedure

1. Log on to the TEPS/e administration console.
2. Select **Security → SSL certificate and key management**.
3. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
4. In the Additional Properties area, click the **Signer certificates** link and in the page that is displayed, click **Add**.
5. In the page that is displayed specify the following information:
   - Set **Alias** to the desired label for the certificate. For example, `Authorization Policy Server Signer Certificate`.
   - Set **File name** to the location of the public signer certificate. For example, <*policyauthsignercert*>.
   - Leave the **Data type** as **Base64-encoded ASCII data**.
6. Click **OK**, then **Save**.
7. Reconfigure the portal server to use HTTPS instead of HTTP for the Authorization Policy Server connection. For details on reconfiguring the Authorization Policy Server connection parameters, see “Enabling authorization policies in the portal server” on page 194.

Results

HTTPS is now used between the Authorization Policy Server and the portal server.

Configuring TLS/SSL communication between the load balancing HTTP Server and each portal server's local HTTP server

If you want to use HTTPS in a dashboard environment that is using a HTTP server to load balance multiple Tivoli Enterprise Portal Servers, you must configure TLS/SSL communication from the load balancing HTTP server to each portal server's local HTTP server.

Roadmap

Use the following roadmap to help you get started:

Table 27. Roadmap for configuring TLS/SSL communication between the load balancing HTTP Server and each portal server's local HTTP server

Step	Description	Where to find information
1	You have two options for obtaining the public-private key pair used by the portal server's HTTP server: • Use the default self-signed certificates installed with IBM Tivoli Monitoring. If you choose this option, proceed to step 2. OR • Use a digital certificate that has been signed by a certificate authority. In this case, you must create a certificate request for each portal server and send it to the certificate authority for signing. Once the digital certificate has been signed, you add the certificate authority signer's certificate to the portal server's trust stores, and then add the new digital signature to the portal server's key stores.	"Using third party certificate authority signed certificates for the portal server" on page 213
2	For each portal server's HTTP server, extract the public signer certificate from the trust store.	"Extracting the portal server's local HTTP server public signer certificate"
3	Add the public signer certificate for each portal server's HTTP server into the load balancing HTTP server's trust store.	If you are using the IBM HTTP Server, follow the steps in "Configuring TLS/SSL communication for the IBM HTTP Server used for load balancing multiple portal servers" on page 224.

Extracting the portal server's local HTTP server public signer certificate

Extract the portal server's local HTTP server public signer certificate from its trust store using the iKeyman graphical interface.

About this task

On the computer system where the portal server is installed, extract the portal server's HTTP server certificate from the `keyfile.kdb` file.

Alternatively, you can use the gskcmd command-line interface to complete this task. For detailed information on using the gskcmd command-line interface, see "Using the GSKit command-line interface to work with key databases and certificates" on page 241.

Procedure

1. Start the key management utility (iKeyman) using one of these methods:
   - Windows Click **Start** > **Programs** > **IBM HTTP Server V8.0 for Tivoli Enterprise Portal Server** > **Start Key Management Utility**.
   - Linux UNIX From the command-line run **`<install_dir>/<interp>/iu/ihs/HTTPServer/bin/ikeyman`** or change to the `<install_dir>/<interp>/iu/ihs/HTTPServer/bin` directory and type `ikeyman`. If you start IKEYMAN to create a new key database file, the utility stores the file in the directory where you start IKEYMAN.
2. Navigate to the following directory and select the `keyfile.kdb` to open the file.

   Windows *`install_dir`*`\keyfiles\`

   Linux UNIX *`install_dir`*`/keyfiles`
3. Enter the password to open the file. The default password is `IBM61TIV`.
4. From the **Key database content** drop down list, select **Signer Certificates**.
5. Select the trusted root certificate used to sign the IBM HTTP server SSL certificate. By default it is the root certificate. If you requested a new signer certificate from a certificate authority then it is the name that you specified when you added the signer certificate to the key database.

   Tip: If you are not sure which signer certificate is the trusted root certificate, select a certificate and click **View / Edit**. In the **Key information** dialog window, check if the **Set the certificate as trusted root option** is selected.
6. Click **Extract**.
7. Leave the **Data Type** as **Base64-encoded ASCII data** and then save the certificate as *`hostname`*`ITMcert.arm`.
8. Click **OK** to export the certificate to the file name and directory specified in the previous steps.

Configuring TLS/SSL communication for the IBM HTTP Server used for load balancing multiple portal servers

To configure TLS/SSL communication for a load balancing environment, add the public signer certificate for each Tivoli Enterprise Portal Server's local HTTP server to the load balancing IBM HTTP Server trust store using the iKeyman graphical interface.

Before you begin

Copy the public signer certificate for each portal server's local HTTP server to the computer system where the load balancing HTTP Server is installed.

About this task

Alternatively, you can use the gskcmd command-line interface to complete this task. For detailed information on using the gskcmd command-line interface, see "Managing keys with the gskcmd command line interface (Distributed systems)" in the WebSphere Application Server Information Center.

Procedure

1. On the computer system where the load balancing HTTP Server is installed, import a public signer certificate into the key database using the iKeyman graphical interface.

   Start the key management utility (iKeyman) using one of these methods:

   - Windows Click **Start** > **Programs** > **IBM HTTP Server V8.5** > **Start Key Management Utility**.
   - Linux UNIX From the command-line run **`<install_dir>/bin/ikeyman`** or change to the `<install_dir>/bin` directory and type `ikeyman`, where `<install_dir>` is the directory where the WebSphere Plugin is installed (such as `/opt/IBM/WebSphere/Plugins`). If you start IKEYMAN to create a new key database file, the utility stores the file in the directory where you start IKEYMAN.
2. Click **Key Database File** from the main UI, then click **Open**.
3. Specify the location of the CMS key database file `plugin-key.kdb` that is specified in the HTTP server `plugin-cfg.xml` file.

   By default the file is `plugin-key.kdb`.

   When using HTTP server V8.5, the following default locations apply:

   Windows `C:\Program Files\IBM\WebSphere\Plugins_1\config\webserver1`

   Linux UNIX `/opt/IBM/Websphere/Plugins/config/webserver1`
4. Provide the password for the key database and click **OK**. The default password is `WebAS`.
5. From the **Key database content** drop down list, select **Signer Certificates**.
6. For each portal server's local HTTP server signer certificate:
   a. Click **Add**.
   b. Select the signer certificate and click **OK**.
   c. Insert a unique label, such as *`hostname`* `ITM certificate`, and click **OK**.
7. Click **Save** to update the file with your changes.

   If you do not have permission to overwrite the `plugin-key.kdb` file, you can save it as a different name, like `plugin-key2.kdb` and later update your `httpd.conf` and `plugin-cfg.xml` files accordingly.
8. Enable SSL for your HTTP server by updating the `httpd.conf` file. The default location of the `httpd.conf` file is:

   Windows `C:\Program Files\IBM\HTTPServer\conf`

   Linux UNIX `/opt/IBM/HTTPServer/conf`

   a. Open the `httpd.conf` file and search for the text string "Example SSL configuration" to view an example configuration.
   b. Add the following block to your `httpd.conf` file, changing the directory paths to Windows paths if your HTTP server is on Windows:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
 Listen 443
 <VirtualHost *:443>
 SSLEnable
 SSLProtocolDisable SSLv2
 ErrorLog "/opt/IBM/HTTPServer/logs/sslerror.log"
 TransferLog "/opt/IBM/HTTPServer/logs/sslaccess.log"
 KeyFile "/opt/IBM/WebSphere/Plugins/config/webserver1/plugin-key.kdb"
 SSLStashfile "/opt/IBM/WebSphere/Plugins/config/webserver1/plugin-key.sth"
 </VirtualHost>
 SSLDisable
```

   Note: The `plugin-key.kdb` file is the one that you saved in step 7.

9. Restart the load balancing HTTP Server to implement your changes. For more information, see "Starting and stopping IBM HTTP Server" in the WebSphere Application Server Information Center

Configuring TLS/SSL communication between Dashboard Application Services Hub and an HTTP server used for load balancing multiple portal servers

If you want to use HTTPS in a dashboard environment that is using a HTTP server to load balance multiple Tivoli Enterprise Portal Servers, you must configure TLS/SSL communication from the Dashboard Application Services Hub to the HTTP server.

Roadmap

In order to use HTTPS and its security encryption features, complete the following tasks in the roadmap.

Table 28. Roadmap to configure TLS/SSL communication between Dashboard Application Services Hub and a HTTP server used for load balancing multiple Tivoli Enterprise Portal Server

Step	Description	Where to find information
1	Determine if you want to use the self-signed certificate that came with the HTTP server or request a certificate that is signed by a certificate authority.	If you do not want to use a self-signed certificate, see your HTTP server documentation to determine how to request a certificate. If you are using the IBM HTTP Server, see "Securing IBM HTTP Server" in the WebSphere Application Server Information Center. **Note:** If you create a new key database to contain a certificate from a certificate authority, you must add the portal server signer certificates into the new database. For more information, see steps 2 and 3 of the roadmap in "Configuring TLS/SSL communication between the load balancing HTTP Server and each portal server's local HTTP server" on page 223.
2	Export the HTTP server's public signer certificate.	See your HTTP server documentation to determine how to perform this step. If you are using the IBM HTTP Server, see "Exporting the IBM HTTP Server signer certificate."
3	Add the HTTP public signer certificate to Dashboard Application Services Hub.	"Adding the HTTP Server certificate to the Dashboard Application Services Hub trust store" on page 228

Exporting the IBM HTTP Server signer certificate

On the computer system where the load balancing IBM HTTP Server is installed, export the public signer certificate from the key database using the iKeyman graphical interface.

About this task

Alternatively, you can use the gskcmd command-line interface to complete this task. For detailed information on using the gskcmd command-line interface, see "Managing keys with the gskcmd command line interface (Distributed systems)" in the WebSphere Application Server Information Center.

Procedure

1. On the computer system where the load balancing IBM HTTP Server is installed, export the public signer certificate from the key database using the iKeyman graphical interface.

   Start the key management utility (iKeyman) using one of these methods:

   - Windows Click **Start** > **Programs** > **IBM HTTP Server V8.5** > **Start Key Management Utility**.
   - Linux UNIX From the command-line run **`<install_dir>/bin/ikeyman`** or change to the `<install_dir>/bin` directory and type `ikeyman`, where `<install_dir>` is the directory where the WebSphere Plugin is installed (such as `/opt/IBM/WebSphere/Plugins`). If you start IKEYMAN to create a new key database file, the utility stores the file in the directory where you start IKEYMAN.
2. Click **Key Database File** from the main UI, then click **Open**.
3. Specify the location of the CMS key database file `plugin-key.kdb` that is specified in the HTTP server `plugin-cfg.xml` file.

   By default the file is `plugin-key.kdb`.

   When using HTTP server V8.5, the following default locations apply:

   Windows `C:\Program Files\IBM\WebSphere\Plugins_1\config\webserver1`

   Linux UNIX `/opt/IBM/Websphere/Plugins/config/webserver1`
4. Provide the password for the key database and click **OK**. The default password is `WebAS`.
5. From the **Key database content** drop down list, select **Signer Certificates**.
6. Select the trusted root certificate used to sign the IBM HTTP server SSL certificate.

   Tip: If you are not sure which signer certificate is the trusted root certificate, select a certificate and click **View / Edit**. In the **Key information** dialog window, check if the **Set the certificate as trusted root option** is selected.
7. Click **Extract**.
8. In the **Extract Certificate to a File** dialog box, set the following fields:

   Data type: Base64-encoded ASCII data.

   Certificate file name: Accept the default of `cert.arm` or specify a different name.

   Location: Type the drive and directory where you want to store the file or use **Browse** to select a drive and directory.
9. Click **OK** to export the public signer certificate to the file name and directory specified above.

Adding the HTTP Server certificate to the Dashboard Application Services Hub trust store

Add the public signer certificate used by the HTTP Server that is load balancing multiple portal servers to the Dashboard Application Services Hub WebSphere trust store to configure TLS/SSL communication.

Before you begin

Copy the public signer certificate that was exported from the load balancing HTTP server to the computer system where Dashboard Application Services Hub is installed.

Procedure

1. Log into the Dashboard Application Services Hub WebSphere Administrative Console.
   a. Enter the following URL in your Internet Explorer or Firefox browser: `https://hostname:16311/ibm/console`.

      If your environment was configured with a port number other than the default, enter that number instead. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.
   b. Enter the Dashboard Application Services Hub administrative user ID and password then click **Go**.

      The user ID must be assigned the `administrator` and `iscadmins` roles.
   c. In the Console Settings area click on **WebSphere Administrative Console** and then click the **Launch WebSphere administrative console** button.
2. Select **Security → SSL certificate and key management**.
3. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
4. In the Additional Properties area, click the **Signer certificates** link and in the page that is displayed, click **Add**.
5. Enter a unique alias name. For example, `TEPS_LOAD_BALANCE_SERVER`.
6. Enter the file location of the HTTP server public signer certificate.
7. Ensure the **Data type** selection matches the format of the certificate that you exported from the HTTP server.
8. Click **OK**, then **Save**.

Results

The certificates are now setup for communication between the Dashboard Application Services Hub server and HTTP server used to load balance multiple portal servers.

Enabling FIPS for IBM Tivoli Monitoring

You must configure IBM Tivoli Monitoring components to enable the Federal Information Processing Standard (FIPS).

Procedure

Complete configuration on the following components:

- Monitoring automation server

- Monitoring server and monitoring agent
- Portal client
- Portal server
- tacmd command-line interface
- tivcmd command-line interface
- Warehouse database

Note:

Best Practice is to reconfigure any components after editing environment variables to ensure any changes are implemented.

If you are using Jazz for Service Management with IBM Tivoli Monitoring, see the *Jazz for Service Management Installation Guide* in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html) for information on how to enable FIPS for its components.

Monitoring automation server

1. Edit the Tivoli Enterprise Monitoring Automation Server environment file:

   Windows Edit the `KASENV` file.

   Linux UNIX Edit the `as.ini` file on the computer where the automation server is installed.

   Change or add the following environment variable:

   ```
   KDEBE_FIPS_MODE_ENABLED=YES
   ```
2. Restart the automation server to implement your changes.

Monitoring server and monitoring agent configuration:

Note: You can use the following instructions to also configure the Warehouse Proxy Agent and the Summarization and Pruning Agent.

1. Edit the following environment files:

   Windows Edit the `KBBENV` file and the `K`*`XX`*`ENV` file for each monitoring agent (where *`XX`* is your 2 letter product code).

   Linux UNIX Edit the `ms.ini` on the monitoring server, and `*.ini` for each monitoring agent.

   Change or add the following environment variable:

   ```
   KDEBE_FIPS_MODE_ENABLED=YES
   ```

   If using autonomous agents, you must add the `KDEBE_FIPS_MODE_ENABLED` variable to your custom environment file.

   For the Linux and UNIX OS agents, the `KDEBE_FIPS_MODE_ENABLED` variable has additional possible values. For more information, see the *IBM Tivoli Monitoring Linux OS Agent Installation and Configuration Guide* and *IBM Tivoli Monitoring UNIX OS Agent Installation and Configuration Guide*.
2. Restart the monitoring server and each monitoring agent you edited to implement your changes.

Portal client configuration:

1. For desktop clients, browser clients, and WebStart clients, configure the clients to communicate using HTTPS protocol. Follow the instructions in “Configuring HTTP communication between the portal client and server” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. For desktop clients, browser clients, and WebStart clients, edit the associated configuration file using the *same method* as described in “Configuring HTTP communication between the portal client and server” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
   - For desktop clients, your edits modify the `cnp.bat` file.
   - For browser clients, your edits modify the `applet.html` file.
   - For WebStart clients, your edits modify the `tep.jnlpt` file.

   Add the following variables to each of the configuration files:

   ```
   com.ibm.TEPS.FIPSMODE set to true
   tep.sslcontext.protocol set to TLS
   https.protocols set to TLS
   ```
3. Edit the *install_dir*/CNP/cnp.bat file directly or through **Manage Tivoli Enterprise Monitoring Services > Desktop Client > Advanced > Edit ENV**.

   Modify the `_CMD` line to include the following definition:

   ```
   https.protocols set to TLS
   ```
4. Restart each portal client to implement your changes.

Portal server configuration:

1. Edit the Tivoli Enterprise Portal Server environment file on the computer where the portal server is installed.

   Windows Edit the `KFWENV` file.

   Linux UNIX Edit the `cq.ini` file.

   Change or add the following environment variables:

   ```
   KDEBE_FIPS_MODE_ENABLED=YES
   KFW_FIPS_ENFORCED=YES
   ```
2. Restart the portal server to implement your changes.
3. Enable the TEPS/e administration console. For more information, see “Starting the TEPS/e administration console” on page 114.
4. Enable FIPS 140-2 mode by following the directions in the “Configuring WebSphere Application Server for SP800-131 standard strict mode” topic in the WebSphere Application Server V8.5 Information Center http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp.

tacmd command-line interface configuration:

Windows

1. Edit the <*ITM_dir*>\BIN\KUIENV file.

   Change or add the following environment variables:

   ```
   TEPS_FIPS_MODE=YES
   KDEBE_FIPS_MODE_ENABLED=YES
   ```

Linux UNIX

1. Edit the <*ITM_dir*>/bin/ tacmd shell script.

Change or add the following environment variables:

```
export TEPS_FIPS_MODE=YES
export KDEBE_FIPS_MODE_ENABLED=YES
```

tivcmd command-line interface configuration:

Windows

1. Edit the <*tivcmd_install_dir*>\BIN\KDQENV file.

   Change or add the following environment variables:

   ```
   KDEBE_FIPS_MODE_ENABLED=YES
   ```

Linux UNIX

1. Edit the <*tivcmd_install_dir*>/bin/tivcmd shell script.

   Change or add the following environment variables:

   ```
   export KDEBE_FIPS_MODE_ENABLED=YES
   ```

Warehouse database configuration:

Warehouse database configuration is specific to your installation and outside the scope of this configuration. You must configure your ODBC client to access the database server using TLS/SSL. Configuration links for running the database in FIPS 140-2 mode are listed below.

- MSSQL 2005

  Refer to the following Microsoft knowledge base article for details on configuring Microsoft SQL Server to run in FIPS 140-2 mode. For more information, see http://support.microsoft.com/kb/920995.
- DB2® v9.1 Fix Pack 2+

  For DB2 9.1 Fix Pack 2 and higher TLS/SSL connections are always in FIPS 140-2 mode. Refer to the following IBM support document for further details on configuring the TLS/SSL ODBC connection. For more information, see http://www-01.ibm.com/support/docview.wss?uid=swg21249656.
- Oracle

  Refer to the following support document for configuring Oracle 10g (9.0.4) or later in FIPS 140-2 mode. For more information, see http://download.oracle.com/docs/cd/B14099_19/core.1012/b13999/fips.htm.

Results

You are now running a FIPS 140-2 Level 1 compliant configuration.

What to do next

When in FIPS 140-2 mode, Tivoli Management Services components and Tivoli Enterprise Monitoring Agents use one or more of these FIPS 140-2 approved cryptographic providers: IBMJCEFIPS (certificate 497), IBMJSSEFIPS (certificate 409), and IBM Crypto for C (ICC (certificate 775) for cryptography. The certificates are listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm.

All IP.SPIPE connections and TLS/SSL-enabled LDAP connections utilize TLS 1.0 or higher protocol. TLS/SSL must be enabled between the Tivoli Enterprise Portal client and the Tivoli Enterprise Portal Server, and is described in the “Using SSL between the portal server and the client” topic in the *IBM Tivoli Monitoring Installation and Setup Guide*. Failure to enable TLS/SSL might expose credentials.

Enable IP.SPIPE between all IBM Tivoli Monitoring components to preserve integrity and confidentiality of data using FIPS 140-2 compliant cryptography. Certificates used in IP.SPIPE communication require NIST and FIPS prescribed cryptographic strength. For detailed information on how to replace cryptographic certificates, see the various topics in Chapter 8, “Securing communications,” on page 209. If your environment uses the provided GSKit utilities, the `-fips` flag must be included in all operations. Refer to your local security administrator or to the NIST website for more details on FIPS 140-2 compliance.

Related reference:

http://www-01.ibm.com/software/sysmgmt/products/support/IBMTivoliMonitoring.html
Search IBM Tivoli Monitoring Support for guidelines on configuring components for FIPS 140-2 compliance.

http://csrc.nist.gov/
The Computer Security Division of the National Institute of Standards and Technology has publications on FIPS 140-2 compliance.

Enabling SP800-131a for IBM Tivoli Monitoring

The National Institute of Standards and Technology (NIST) Special Publications (SP) 800-131a standard strengthens algorithms and increases the key lengths to improve security. To enable SP800-131a you must configure IBM Tivoli Monitoring components individually.

Before you begin

- Update your IBM Tivoli Monitoring infrastructure components to Version 6.3 Fix Pack 2 or higher before you enable SP800-131a. Your operating system agents and Tivoli Enterprise Management Agent framework for other agents must be at Version 6.3 or higher. Infrastructure components that are configured to use the SP800-131a mode can only interact with the following components:
  - IBM Tivoli Monitoring Version 6.3 Fix Pack 2 or higher monitoring agents or infrastructure components.
  - IBM Tivoli Monitoring Version 6.3 or higher monitoring agents or infrastructure components that are configured for FIPS 140-2 mode.

  Tivoli Monitoring agents and infrastructure components prior to Version 6.3 are not able to communicate with Tivoli Monitoring infrastructure components configured in SP800-131a mode.
- Ensure that you have strong certificates for each of your computers. Then distribute the certificates to each of the agent keyfile directories, one per IBM Tivoli Monitoring installation. Certificates used for TLS/SSL communication must have a minimum RSA key length of 2048, or be Elliptical Curve (EC) certificates with a minimum key length of 256 bits. For more information, see Chapter 8, “Securing communications,” on page 209.
- Perform this task during a maintenance period.

About this task

SP800-131a mode for IBM Tivoli Monitoring has the following properties:

Note: In IBM Tivoli Monitoring Version 6.3 Fix Pack 2, the generated self-signed certificates comply to the standards mentioned in this section by default.

- All communication is over TLSv1.2 protocol.
- All certificates for communication are RSA with 2048 bit keys signed, with at least SHA-256 bit digital signatures or Elliptic curve cryptography certificates. Use all RSA certificates or all Elliptic curve certificates.
- Any SNMP connections must conform to either SNMP V1, V2 or V3 using authentication with SHA-1 only.
  Note: SNMP V3 data privacy is not SP800-131 compliant.
- SSH connections must use certificates that are 2048-bit in size.
- Elliptic Curve Certificates can be used by any IBM Tivoli Monitoring component, but using an Elliptic Curve Certificate implies the exclusive acceptance of only TLSv1.2 for communication, because TLSv1.2 is the only protocol that supports Elliptic Curve Certificates. Using Elliptic Curve Certificates at the monitoring server allows only monitoring agents at IBM Tivoli Monitoring Version 6.3 Fix Pack 2 or higher to connect.
- All services and autonomous agents that interface using HTTPS and IP.SPIPE communication ports, must use the TLSv1.2 protocol. The Microsoft Internet Explorer 8.0 or higher browser supports TLSv1.2.
- Many application agents are 32-bit agents installed on a 64-bit operating system. 32-bit application agents must be upgraded using the **`tacmd updateFramework`** command to update their framework to Version 6.3 or higher. Updating the framework allows the agent to communicate with an SP800-131a compliant Tivoli Enterprise Monitoring Server. For more information on the **`tacmd updateFramework`** command, see the *IBM Tivoli Monitoring Command Reference*.
- You can optionally enable TLSv1.0 in the Tivoli Enterprise Portal Server to access the online help. If TLSv1.0 is disabled in the IBM HTTP Server, the Tivoli Enterprise Portal functions as normal but will not provide online help. All data and management is performed over TLSv1.2. Online help and other text content is transmitted over TLSv1.0. You can disable TLSv1.0 if you do not need the online help and dialog help. If TLSv1.0 is disabled, the error `Secure Connection Failed: ssl_error_no_cypher_overlap` might appear in certain workspaces that display help information. You can continue to create objects and access workspaces over TLSv1.2.
- When the monitoring server is configured in SP800-131a mode, IBM Tivoli Monitoring Version 6.3 Fix Pack 2 monitoring agents, the portal server, and the tacmd command line can still communicate with the monitoring server without being explicitly reconfigured in SP800-131a mode.
- The **`tacmd tepslogin`** and other tacmd commands directed at the portal server must communicate over TLSv1.0 on port 15001. Additionally, tacmd commands directed at the monitoring server must communicate over TLSv1.2 with SP800 restrictions. Ensure that you enable TLS/SSL communication for the portal client connections at the portal server.
- Situations that use the Linux OS agent and UNIX OS agent File Information attribute group must ensure they are using SHA-1, SHA-256, SHA-384, or SHA-512 in SP800-131a mode.

Procedure

Complete configuration on the following components in the order listed if applicable:

- Monitoring server and monitoring agent
- z/OS AT-TLS
- Portal server
- Portal client
- tacmd command-line interface
- tivcmd command-line interface
- Authorization Policy Server

Note:

Best Practice is to reconfigure any components after editing environment variables to ensure any changes are implemented.

Monitoring server and monitoring agent configuration:

Note: You can use the following instructions to also configure the Warehouse Proxy Agent and the Summarization and Pruning Agent.

1. Edit the following environment files:

   Windows In the Manage Tivoli Enterprise Monitoring Services window, right-click the component and click **Advanced** → **Edit Variables**. Alternatively, you can edit the KBBENV file and the K*XX*ENV file for each monitoring agent (where *XX* is your 2 letter product code) directly.

   Linux UNIX Edit the ms.ini on the monitoring server, and *.ini for each monitoring agent.

   Change or add the following environment variable:

   ```
   KDEBE_FIPS_MODE_ENABLED=SP800-131a
   ```

   If using autonomous agents, you must add the KDEBE_FIPS_MODE_ENABLED variable to your custom environment file.
2. Restart the monitoring server and each monitoring agent you edited to implement your changes.

z/OS AT-TLS configuration:

In z/OS environments, configuring SP800-131a for IP.SPIPE connections requires configuring the Application Transparent Transport Layer Security (AT-TLS) policy. TLSv1.2 protocol is available with z/OS 2.1. TLSv1.2 is also available with z/OS 1.13, but you must apply the following APARs to your system:

- CommServer toleration APAR PM62905
- AT-TLS toleration for System SSL APAR OA39422

If a secure protocol (SPIPE or HTTPS) is used between monitoring agents and monitoring servers on z/OS, AT-TLS must be configured and running. To configure AT-TLS, an authorized system programmer must create a policy for AT-TLS. A security administrator (RACF or ACF2) must grant permission to the policy that defines the authentication certificate that is used in the TLS protocol.

If z/OS components communicate with monitoring servers on a distributed operating system, or a distributed component, such as the Tivoli Enterprise Portal Server, communicates with a z/OS hub monitoring server, the AT-TLS policy must match the policy that is created for the distributed component by GSKIT.

To configure SP800-131a complete the following tasks:

1. ICSF must be enabled and started on the monitoring server and agent-only runtime environments (RTE). For more information, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/ztemsconfig/ztemsconfig.htm).
2. Configure your z/OS monitoring server and monitoring agents to use IP.SPIPE communications. For more information, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/ztemsconfig/ztemsconfig.htm).
3. Configure an AT-TLS policy to restrict to TLSv1.2.

   See the *Communications Server IP Configuration Guide* in the Communication Server Information Center for further reference.

   The following is an example of an AT-TLS policy for SP800-131a:

```
TTLSGroupAction              group_action0
{
  TTLSEnabled                ON
}
TTLSEnvironmentAction        environment_action0
{
  TTLSKeyRingParms
 {
    Keyring                  /etc/itm/at-tls/keyring.db
    keyringPw                itm
    keyringStashFile         /etc/itm/at-tls/keyring.sth
 }
  HandshakeRole              Client
  TTLSEnvironmentAdvancedParms
  {
   SSLv2 Off
   SSLv3 Off
   TLSv1 Off
   TLSv1.1 Off
   TLSv1.2 On
   FIPS140 On
   CertificateLabel IBM_Tivoli_Monitoring_Encryption_Key
  }
  TTLSSignatureParmsRef
  {
   ## TLS_SIGALG_SHA224_WITH_RSA
    SignaturePair 0301
   ## TLS_SIGALG_SHA224_WITH_ECDSA
    SignaturePair 0303
   ## TLS_SIGALG_SHA256_WITH_RSA
    SignaturePair 0401
   ## TLS_SIGALG_SHA256_WITH_ECDSA
    SignaturePair 0403
   ## TLS_SIGALG_SHA384_WITH_RSA
    SignaturePair 0501
   ## TLS_SIGALG_SHA384_WITH_ECDSA
    SignaturePair 0503
   ## TLS_SIGALG_SHA512_WITH_RSA
    SignaturePair 0601
   ## TLS_SIGALG_SHA512_WITH_ECDSA
    SignaturePair 0603
  }
}
```

Portal server configuration:

1. Enable TLS/SSL for all Tivoli Enterprise Portal clients. For detailed steps, see "Using SSL between the portal server and the client" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. Edit the Tivoli Enterprise Portal Server environment file on the computer where the portal server is installed.

   Windows In the Manage Tivoli Enterprise Monitoring Services window, right-click the component and click **Advanced** → **Edit Variables**. Alternatively, you can edit the `KFWENV` file directly.

   Linux UNIX Edit the `cq.ini` file.

   Change or add the following environment variables:

   ```
   KDEBE_FIPS_MODE_ENABLED=SP800-131a
   KFW_FIPS_ENFORCED=YES
   ```
3. Restart the portal server to implement your changes.
4. Enable SP800-131 Transistion mode in the TEPS/e administration console.
   a. Follow the instructions in "Starting the TEPS/e administration console" on page 114.

   b. Click **Security** → **SSL certificate and key management** → **Manage FIPS**.

   c. If you have not imported compliant certificates, select **Convert Certificates**. When converting the certificates, select **Algorithm Strict** and then select **SHA256WithRSA** or another algorithm. If you select an ECDSA Certificate algorithm, then all browsers and clients (including the Dashboard Application Services Hub servers) connecting to the WebSphere Server must support TLSv1.2 and Elliptic Curve Certificates.

      You must accept the new certificate using the WebSphere command line utilities. Run one of the following commands, and then when prompted, accept the certificate:

      Windows **`updateTEPSEPass.bat wasadmin <password>`**

      Linux UNIX **`updateTEPSEPass.sh wasadmin <password>`**

   d. Select **Enable SP800-131** , **Transistion**, and **Update SSL configurations to require TLSv1.2**.

      Note: Once you apply and then save your changes to the master file for WebSphere, you might need to reconnect to the WebSphere console if you are logged out, since the new certificate and algorithms take effect immediately.

   e. Update the `ssl.client.props` file to allow administration of WebSphere.

      Windows `*install_dir*\CNPSJ\profiles\ITMProfile\properties\ssl.client.props`

      Linux UNIX `*install_dir*/*arch*/iw/profiles/ITMProfile/properties/ssl.client.props`

      Once the server is configured for SP800-131 transition mode, the `ssl.client.props` file must be modified so that the administrative client can communicate with the WebSphere server running in SP800-131 mode. They are not able to make a TLSv1.2 connection to the server without the change. Edit the `ssl.client.props` file by completing the following steps:

      1) Modify `com.ibm.security.useFIPS` to be set to `true`.

      2) Add `com.ibm.websphere.security.FIPSLevel=SP800-131` directly beneath the `useFips` property.

   3) Change the com.ibm.ssl.protocol property to TLSv1.2. **Note**: The com.ibm.ssl.protocol property is further down in the file than the first two properties.

   For further instruction, see “Transitioning WebSphere Application Server to the SP800-131 security standard” in the WebSphere Application Server V8.5 Information Center http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp.

5. Synchronize certificates between the IBM HTTP Server and the portal server. Import the new WebSphere certificates into the IBM Tivoli Monitoring key repository. For detailed steps, see “Importing the TEPS/e certificates into the portal server keyfile database” on page 240.
6. Update the IBM HTTP Server acceptable protocols. On the computer where the portal server is installed edit the httpd.conf file to update the virtualhost for port 15201 to include:

   Note: In the following examples, the SSLAttributeSet information is entered on one line.

   Windows *install_dir*\IHS\CONF

```
<VirtualHost *:15201>
   DocumentRoot "<ITM_HOME>/CNB"
   SSLEnable
   SSLProtocolDisable SSLv2
   SSLProtocolDisable SSLv3
   SSLProtocolEnable TLSv10
   SSLProtocolDisable TLSv11
   SSLProtocolEnable TLSv12
   SSLFIPSEnable
   SSLAttributeSet 245 "GSK_TLS_SIGALG_RSA_WITH_SHA224,
GSK_TLS_SIGALG_RSA_WITH_SHA256,GSK_TLS_SIGALG_RSA_WITH_SHA384,
GSK_TLS_SIGALG_RSA_WITH_SHA512,GSK_TLS_SIGALG_ECDSA_WITH_SHA224,
GSK_TLS_SIGALG_ECDSA_WITH_SHA256,GSK_TLS_SIGALG_ECDSA_WITH_SHA384,
GSK_TLS_SIGALG_ECDSA_WITH_SHA512" BUFF

  SSLServerCert IBM_Tivoli_Monitoring_Certificate
   ErrorLog "<ITH_HOME>/IHS/logs/sslerror.log"
   TransferLog "<ITM_HOME>/IHS/logs/sslaccess.log"
   KeyFile "<ITM_HOME>/keyfiles/keyfile.kdb"
   SSLStashfile "<ITM_HOME>/keyfiles/keyfile.sth"
</VirtualHost>
```

   Linux UNIX *install_dir*/*arch*/iu/ihs/HTTPServer/conf

```
<VirtualHost *:15201>
   DocumentRoot "<ITM_HOME>/<arch>/cw/"
   SSLEnable
   SSLProtocolDisable SSLv2
   SSLProtocolDisable SSLv3
   SSLProtocolEnable TLSv10
   SSLProtocolDisable TLSv11
   SSLProtocolEnable TLSv12
   SSLFIPSEnable
   SSLAttributeSet 245 "GSK_TLS_SIGALG_RSA_WITH_SHA224,
GSK_TLS_SIGALG_RSA_WITH_SHA256,GSK_TLS_SIGALG_RSA_WITH_SHA384,
GSK_TLS_SIGALG_RSA_WITH_SHA512,GSK_TLS_SIGALG_ECDSA_WITH_SHA224,
GSK_TLS_SIGALG_ECDSA_WITH_SHA256,GSK_TLS_SIGALG_ECDSA_WITH_SHA384,
GSK_TLS_SIGALG_ECDSA_WITH_SHA512" BUFF

  SSLServerCert IBM_Tivoli_Monitoring_Certificate
   ErrorLog "<ITM_HOME>/<arch>/iu/ihs/HTTPServer/logs/sslerror.log"
   TransferLog "<ITM_HOME>/<arch>/iu/ihs/HTTPServer/logs/sslaccess.log"
   KeyFile "<ITM_HOME>/keyfiles/keyfile.kdb"
   SSLStashfile "<ITM_HOME>/keyfiles/keyfile.sth"
</VirtualHost>
```

7. Update the HTTP plugin.

Windows *install_dir*\IHSPlugins\config\ITMWebServer\plugin-cfg.xml

Linux UNIX *install_dir*/*arch*/iu/ihs/Plugins/config/ITMWebServer/plugin-cfg.xmlAdd or change the following properties as attributes on the Config XML tag:

FIPSEnable set to "true"

StrictSecurity set to "true"

8. Restart the portal server to implement your changes.

Portal client configuration:

1. For desktop clients, browser clients, and WebStart clients, configure the clients to communicate using HTTPS protocol. Follow the instructions in “Configuring HTTP communication between the portal client and server” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. For desktop clients, browser clients, and WebStart clients, edit the associated configuration file using the *same method* as described in “Configuring HTTP communication between the portal client and server” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
   - For desktop clients, your edits modify the cnp.bat file.
   - For browser clients, your edits modify the applet.html file.
   - For WebStart clients, your edits modify the tep.jnlpt file.

   Add the following variables to each of the configuration files:

   ```
   com.ibm.TEPS.FIPSMODE set to true
   tep.sslcontext.protocol set to TLSv1.2
   https.protocols set to TLSv1.2
   com.ibm.ssl.protocol set to TLSv1.2
   ```
3. For browser client users, you must enable TLSv1.2 in the Java Control Panel. Open the Java Control Panel for the Java that is being used in the browser client using **Advanced** → **Advanced Security Settings** and set **Use TLS 1.2**.
4. For desktop client users, edit the *install_dir*/CNP/cnp.bat file directly or through **Manage Tivoli Enterprise Monitoring Services > Desktop Client > Advanced > Edit ENV**.

   Modify the _CMD line to include the following definition:

   ```
   https.protocols set to TLSv1.2
   com.ibm.ssl.protocol set to TLSv1.2
   ```
5. Restart each portal client to implement your changes.

tacmd command-line interface configuration:

Windows

1. Edit the <*ITM_dir*>\BIN\KUIENV file.

   Change or add the following environment variables:

   ```
   TEPS_FIPS_MODE=YES
   KDEBE_FIPS_MODE_ENABLED=SP800-131a
   ```

Linux UNIX

1. Edit the <*ITM_dir*>/bin/ tacmd shell script.

   Change or add the following environment variables:

```
export TEPS_FIPS_MODE=YES
export KDEBE_FIPS_MODE_ENABLED=SP800-131a
```

tivcmd command-line interface:

Windows

1. Edit the <*tivcmd_install_dir*>\BIN\KDQENV file.

   Change or add the following environment variables:

   ```
   KDEBE_FIPS_MODE_ENABLED=SP800-131a
   ```

Linux UNIX

1. Edit the <*tivcmd_install_dir*>/bin/tivcmd shell script.

   Change or add the following environment variables:

   ```
   export KDEBE_FIPS_MODE_ENABLED=SP800-131a
   ```

Authorization Policy Server:

1. Ensure that your Dashboard Application Services Hub WebSphere is at version 8.0.0.6 or 8.5.0.2 or higher to be SP800-131a compliant.
2. Log in to the WebSphere Administrative Console of the Dashboard Application Services Hub where the Authorization Policy Server is installed.
3. Configure WebSphere with SP800-131 Transitional mode by following the instructions in the topic "Transitioning WebSphere Application Server to the SP800-131 security standard" in the WebSphere Application Server V8.5 Information Center http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp.

Results

You are now running an SP800-131a compliant configuration.

What to do next

Application agents might initiate their own communications for data collection. Those remote servers must be configured to be SP800-131a compliant to ensure the agent's communication is SP800-131a compliant.

When in SP800-131a mode, Tivoli Management Services components and Tivoli Enterprise Monitoring Agents use one or more of these SP800-131a approved cryptographic providers: IBMJCEFIPS (certificate 497), IBMJSSEFIPS (certificate 409), and IBM Crypto for C (ICC certificate 775) for cryptography. The certificates are listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm.

All IP.SPIPE connections and TLS/SSL-enabled LDAP connections utilize TLSv1.2. TLS/SSL must be enabled between the Tivoli Enterprise Portal client and the Tivoli Enterprise Portal Server, as described in the "Using SSL between the portal server and the client" topic in the *IBM Tivoli Monitoring Installation and Setup Guide*. Failure to enable TLS/SSL might expose credentials.

Enable IP.SPIPE between all IBM Tivoli Monitoring components to preserve integrity and confidentiality of data using SP800-131a compliant cryptography. Certificates used in IP.SPIPE communication require NIST and FIPS prescribed cryptographic strength. For detailed information on how to replace cryptographic certificates, see the various topics in Chapter 8, "Securing communications," on page 209. If your environment uses the provided GSKit utilities, the `-fips` flag must be included in all operations. Refer to your local security administrator or to the NIST website for more details on SP800-131a compliance. Information on how to generate certificates using GSKit is also provided on IBM Service Management Connect (http://www.ibm.com/developerworks/servicemanagement).

Importing the TEPS/e certificates into the portal server keyfile database

If you have created a custom keyfile database on the computer where the portal server is installed and it contains a new self-signed or CA-signed IBM Tivoli Monitoring certificate, you must also import the certificates used by TEPS/e into the new keyfile database. This allows TEPS/e and the Tivoli Enterprise Portal Server web server plug-in to communicate internally over a secure connection.

About this task

Use the following procedure to manually import the TEPS/e certificates into the IBM Tivoli Monitoring keyfile database for the portal sever.

Procedure

1. Open a command prompt (Windows) or shell (AIX or Linux).
2. Set the *JAVA_HOME* variable as described in "Setting the JRE for GSKit and starting Key Manager" on page 242, but do *not* start the GSKit Key Manager.
3. Go to the `lib` (32-bit) or `lib64` (64-bit) directory under the GSKit home directory. You can use either 32-bit of 64-bit.

   Windows *install_dir*/GSK8 or *install_dir*/GSK8_x64

   Linux AIX *install_dir*\arch\gs\lib or *install_dir*\arch\gs\lib64

   The GSKit binary is called `gsk8capicmd` (32-bit) or `gsk8capicmd_64` (64-bit).
4. Execute the following commands:
   - Windows
     ```
     ..\bin\gsk8capicmd[_64] -cert -import -db ..\..\CNPSJ\profiles\
     ITMProfile\config\cells\ITMCell\nodes\ITMNode\trust.p12 -pw WebAS
     -type pkcs12 -label root -target ..\..\keyfiles\keyfile.kdb -target_pw
     IBM61TIV -target_type cms -new_label root.ihs
     ..\bin\gsk8capicmd[_64] -cert -import -db ..\..\CNPSJ\profiles\
     ITMProfile\config\cells\ITMCell\nodes\ITMNode\key.p12 -pw WebAS  -type
     pkcs12 -label default -target ..\..\keyfiles\keyfile.kdb -target_pw
     IBM61TIV -target_type cms -new_label default.ihs
     ```
   - Linux AIX
     ```
      ../bin/gsk8capicmd[_64] -cert -import -db ../../../arch/iw/profiles/
     ITMProfile/config/cells/ITMCell/nodes/ITMNode/trust.p12 -pw WebAS
     -type pkcs12 -label root -target ../../../keyfiles/keyfile.kdb
     -target_pw IBM61TIV -target_type cms -new_label root.ihs
     ../bin/gsk8capicmd[_64] -cert -import -db ../../../arch/iw/profiles/
     ITMProfile/config/cells/ITMCell/nodes/ITMNode/key.p12 -pw WebAS -type
     pkcs12 -label default -target ../../../keyfiles/keyfile.kdb -target_pw
     IBM61TIV -target_type cms -new_label default.ihs
     ```

     The password for the source or target key store might be different if you created your own keystores.
5. Restart the portal server.

Using the GSKit command-line interface to work with key databases and certificates

The GSKit command line tool installed with each IBM Tivoli Monitoring component on a distributed platform and is used to manage key files and certificates.

For information about the GSKit command-line interface, see the *IBM Global Security Kit GSKCapiCmd V8.0 User's Guide*.

Before you begin

The following table lists the terms that are used in procedures involving the GSKit. Most terms are based upon the directory in which the IBM Tivoli Monitoring component and GSKit are installed:

<authclidir>
The directory into which the IBM Tivoli Monitoring component is installed. For example:

- `c:\IBM\ITM` or `/opt/IBM/ITM` for the monitoring server, automation server, portal server, tacmd CLI, and agents
- `c:\IBM\TivoliMonitoring` or `/opt/IBM/TivoliMonitoring` for the tivcmd CLI

<interp>
The machine specific interp. For example, sol296, li6263, or aix536.

<gskithome>
The directory into which the GSKit is installed.

Windows 32-bit: `<`*`itmcompdir`*`>\GSK8`.

Windows 64-bit: `<`*`itmcompdir`*`>\GSK8_64`.

Linux and UNIX 32-bit: `<`*`itmcompdir`*`>/<`*`interp`*`>/gs`

Linux and UNIX 64-bit: `<`*`itmcompdir`*`>/<`*`interp`*`>/gs`

<gskittoolcmd>
The actual GSKit CLI command syntax.

Windows 32-bit: `<`*`gskithome`*`>\bin\gsk8capicmd.exe`

Windows 64-bit: `<`*`gskithome`*`>\bin\gsk8capicmd_64.exe`

Linux and UNIX 32-bit: `./<`*`gskithome`*`>/bin/gsk8capicmd.exe`

Linux and UNIX 64-bit: `./<`*`gskithome`*`>/bin/gsk8capicmd_64.exe`

<keydbdir>
The directory into which the default key database is stored.

Windows: `<`*`itmcompdir`*`>\keyfiles`

Linux and UNIX: `<`*`itmcompdir`*`>/keyfiles`

<oldkeydbname>
The base name of the key database installed with the IBM Tivoli Monitoring component. This base key database name is keyfile. The four files associated with this key database are: `keyfile.crl`, `keyfile.kdb`, `keyfile.rdb`, and `keyfile.sth`.

<oldkeydb>
The name of the key database installed with the IBM Tivoli Monitoring component. This key database name is `<`*`oldkeydbname`*`>.kdb`.

<oldkeydbpw>
The installed key database password. The default is IBM61TIV.

<newkeydbname>
The base name of the new key database. Any name other than keyfile can be chosen. For example, `itmcompkeyfile`.

<newkeydb>
The name of the new key database. This key database name is `<`*`newkeydbname`*`>.kdb`.

<newkeydbpw>
The password associated with the new key database. Any valid password can be chosen.

Setting the path to invoke the GSKit command-line tool

In order to run the GSKit command line tool, the GSKit tool `lib` directory must be included in the system path.

Windows **32-bit**

```
set PATH=<gskithome>\lib;%PATH%
cd <gskithome>\bin
```

Windows **64-bit**

```
set PATH=<gskithome>\lib64;%PATH%
cd <gskithome>\bin
```

Linux UNIX **32-bit**

```
export LD_LIBRARY_PATH=<gskithome>/lib:$LD_LIBRARY_PATH
cd <gskithome>/bin
```

Linux UNIX **64-bit**

```
export LD_LIBRARY_PATH=<gskithome>/lib64:$LD_LIBRARY_PATH
cd <gskithome>/bin
```

Using the GSKit iKeyman utility to work with key databases and certificates

A default self-signed certificate and key are provided when you install IBM Tivoli Monitoring. If you prefer to use a certificate authority signed certificate, use the iKeyman utilities to create a certificate request and then create a key database and import the certificates into the database.

Note: The iKeyman utility is available on the distributed computers where the monitoring server, portal servers, portal client desktop client, and tacmd CLI are installed.

For information about the iKeyman graphical user interface and its command-line interface, see the *IBM Developer Kit and Runtime Environment iKeyman V8.0 User's Guide*.

Setting the JRE for GSKit and starting Key Manager

You must set the path to the Java Runtime Environment before starting GSKit.

Otherwise, you might get an error similar to `Failed to parse JAVA_HOME setting`.

Procedure

- Windows
  1. From the command prompt, run this script to get the IBM Java location:

     install_dir`\InstallITM\GetJavaHome.bat`
  2. Set the JAVA_HOME variable to point to the IBM Java location.
  3. Get the GSKit location by running this script:

     install_dir`\InstallITM\GetGSKitHome.bat`
  4. Execute the following command:

     `$JAVA_HOME\jre\bin\ikeyman.exe` *[properties]*

     where, *[properties]* can be zero or more system properties.
- AIX Linux Solaris
  1. From the console, run this script to get the IBM Java location:

     install_dir`/bin/CandleGetJavaHome.sh`

2. Export variable JAVA_HOME to point to the IBM Java path.
3. Execute the following command:

   ```
   $JAVA_HOME\jre\bin\ikeyman.exe [properties]
   ```

   where *[properties]* can be zero or more system properties.

- HP-UX
  1. From the console, run this script to get the IBM Java location:

     ```
     install_dir/bin/CandleGetJavaHome.sh
     ```
  2. Export variable JAVA_HOME to point to the IBM Java path. For 64-bit, the gsk7ikm has to be 64-bit Java.
  3. Check the path for a local GSKit by looking in this file:

     ```
     install_dir/config/gsKit.config
     ```

     `GskitInstallDir` points to a 32–bit GSKit and `GskitInstallDir_64` points to a 64–bit GSKit.
  4. Start IBM Key Management (through the graphical utility that requires X Windows System) by running the following command:

     ```
     GskitInstallDir/bin/gsk7ikm_32
     ```

Creating a new key database

Create a new key database using the iKeyman utility.

About this task

Use the following steps to create a new key database:

1. If you have not already done so, start iKeyman.
2. Click **Key Database File** → **New**.
3. Select **CMS** in the **Key database type** field.
4. Type `keyfile.kdb` in the **File Name** field.
5. Type the following location in the **Location field**: `<`*`itm_installdir`*`>/keyfiles`.
6. Click **OK**. The Password Prompt window is displayed.
7. Enter a password in the **Password** field, and confirm it again in the **Confirm Password** field. Click **OK**.
8. A confirmation window is displayed. Click **OK**.

The IBM Key Management window is displayed. This window reflects the new CMS key database file and your signer digital certificates.

Creating a new public-private key pair and certificate request

Create a new public-private key pair and certificate request in iKeyman.

About this task

Use the following steps to create a new public-private key pair and certificate request:

Procedure

1. If you have not already done so, start iKeyman.
2. Click **Key Database File** → **Open**.
3. Select the keyfile.kdb key database and click **Open**.
4. Type the password for the key database and click **OK**.

5. Select **Personal Certificate Requests** from the pull-down list and click **New**.
6. Click **New**.
7. Type `IBM_Tivoli_Monitoring_Certificate` in the **Key Label** field.
8. Type a **Common Name** and **Organization**, and select a **Country**. For the remaining fields, either accept the default values, or type or select new values.
9. At the bottom of the window, type a name for the file.
10. Click **OK**. A confirmation window is displayed, verifying that you have created a request for a new digital certificate.
11. Click **OK**.

Results

The IBM Key Management window is displayed.

What to do next

Send the file to a CA to request a new digital certificate, or cut and paste the request into the request forms on the CA's web site.

Using a temporary self-signed certificate

It can take between two and three weeks to receive a CA-signed digital certificate. If you want to use a digital certificate other than the one provided with IBM Tivoli Monitoring and you have not yet received the CA-signed digital certificate, you can create a self-signed certificate on the portal server. A self-signed digital certificate is not as secure as a CA-signed certificate; this is strictly a temporary measure until the CA-signed certificate arrives.

About this task

To create and use a self-signed certificate, complete the following procedure:

Procedure

1. Create a CA key database.
2. Create the self-signed certificate.
3. Export the self-signed certificate.
4. Receive the self-signed certificate into the key databases on the portal server.

What to do next

When you receive the CA-signed certificate, you must delete the self-signed certificate.

Receiving the CA-signed certificate

About this task

After the CA returns your new digital certificate, save it on the computer where the portal server is running. Repeat for the client. If the CA returns the certificate as part of an e-mail message, copy and paste it from the e-mail into a text file.

Complete the following procedure to receive the digital certificate from the CA into key database on each computer:

Procedure

1. If you have not already done so, start iKeyman.
2. Click **Key Database File → Open**.
3. Select the keyfile.kdb database and click **Open**.
4. Type the password for the database and click **OK**.
5. Select **Personal Certificates** from the pull-down list.
6. Click **Receive**.
7. Click **Data type** and select the data type of the new digital certificate, such as **Base64-encoded ASCII data**.
8. Type `keyfile.sth` for the **Certificate file name** and `<`*`itm_installdir`*`>/keyfiles` as the **Location** for the new digital certificate.
9. Click **OK**.
10. Type `IBM_Tivoli_Monitoring_Certificate` for the new digital certificate and click **OK**.

Saving the password to a stash file

Because many of the IBM Tivoli Monitoring components work without user intervention, you must save the key database password to a stash file on your computer. Save this password so that product components can use TLS/SS without requiring any intervention from you.

About this task

Complete the following procedure to save the password to a stash file:

Procedure

1. If you have not already done so, start iKeyman.
2. Select **Key Database File → Stash File**. An information window is displayed telling you that the password was saved to a stash file.
3. Click **OK**.

Chapter 9. Audit logging

By using the auditing capability, you can capture significant events occurring in your IBM Tivoli Monitoring environment. You can also record these events in permanent storage for later retrieval and analysis. Each audit record fully describes some event that has changed the state of your IBM Tivoli Monitoring system.

These auditing and logging records can be stored in the Tivoli Data Warehouse. Standard reports are provided via the Tivoli Common Reporting feature.

The auditing facility covers the self-describing agents (including their auto-refresh feature), actions of the Warehouse Proxy Agent, EIF-SSL connections, automated Take Action commands, and the integration of IBM Tivoli Monitoring with Tivoli Application Dependency Discovery Manager.

Supported platforms include Windows, Linux, UNIX, IBM i, and z/OS systems.

Audit records are stored in two places:

Collected ITM Audit attribute data accessible from the portal client

In the Managed System Status workspace you can right-click your monitoring components and select Audit Log to view component-specific collected audit log information. You can then create situations against the ITM Audit table to monitor audited events and collect audit data historically in the Tivoli Data Warehouse.

When examining audit information look for Results with non-zero values. A value of 0 indicates success. Creating situations that monitor for records that have non-zero value Results can help filter out general information messages.

The *Tivoli Enterprise Portal User's Guide* contains more information about the ITM Audit attribute group and workspace. For information about the Audit Log workspace and how to enable historical collection for the ITM Audit attribute group, see Managed System Status workspace. For attribute definitions, see ITM Audit attributes.

Locally stored XML formatted log file

The log file can be used by a third-party product to parse and evaluate the audit information. Use the provided SAPM DTD to assist you with third-party products. The DTD is provided on the IBM Tivoli Monitoring Tools DVD in the `XML` directory; see the `SAPMAudit.dtd` file.

Log files are stored in the `auditlogs` directory under the `<install_dir>` directory. Each agent process has its own log file and is formatted in XML. See the following log files names:

Windows Linux UNIX

For single-instance: *<UserID>*`.`*<hostname>*`_`*<pc>*`_audit.log`

For multi-instance: *<UserID>*`.`*<hostname>*`_`*<pc>*`_`*<instance>*`_audit.log`

IBM i

`/QIBM/ProdData/IBM/ITM/support`

z/OS
Collect log from SMF Facility.

When enabled, ITM Audit records are stored in the Systems Management Facility–format (SMF) type-112 records, coded in UTF8, and are included in a common repository (SYS1.MANn datasets) with all other z/OS event data. For more information, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/ztemsconfig/ztemsconfig.htm).

Audit trace levels

Auditing events have three different trace levels: Minimum, Basic, and Detail. Every event is assigned a trace level. You might want to increase or decrease the trace level to collect additional data.

Minimum: Major state changes to the product

Basic: Any actions that modify objects or cause an access failure

Detail: Any action that causes a successful or failed access control

Event record types

A record type is associated with each audit event to indicate the nature of the audit record. The event record types are categorized in the following table:

Full event name	Short name (displayed in logs)	Description
Authorization Checking	CHECKING	Events related to checking whether a user has permission to perform a particular operation or event.
Authentication Validation	VALIDATE	Events related to authenticating the identify of the user or entity.
Contextual Event	CONTEXT	Any other event that might occur contextually within an application.
Object Maintenance	OBJMAINT	Events related to changing an object such as updating, deleting, creating, or moving any IBM Tivoli Monitoring object or table.
System Administration	SYSADMIN	Events related to program startup and shutdown, audit and authorization system changes, configuration changes, table creation, and data synchronization configuration.
Security Maintenance	SECMAINT	Events related to granting or revoking privileges.

Audit log XML elements mapped to the ITM Audit attribute group

The audit log XML contains elements that coordinate to ITM Audit attributes.

For detailed XML syntax information, see the `SAPMAudit.dtd` on the IBM Tivoli Monitoring Tools DVD.

For descriptions of the attributes see the ITM Audit attributes.

Use the following table for your reference:

Note:

- Some cells are intentionally blank to represent the audit log XML structure. Empty ITM Audit attribute cells indicate that a coordinating attribute has not been created yet for the coordinating XML element.
- XML elements and XML attributes in parenthesis ( ) indicate that the item is not implemented by IBM Tivoli Monitoring.
- *Extra Attributes means the XML element or attribute is inserted as a Name=Value pair in the Extra Attribute column in the Audit Log table.

Logical group	XML element(s)	XML attribute	ITM Audit attribute
AuditEvt	AuditEvt	Domain	Domain
		Level	Trace Level
		Type	Event Record Type
		Ver	Audit Record Version
Who	AuthID		Authorization ID
		(Repository)	
	RunAs		RunAs
		(Repository)	
	UserID		User ID
		(Repository)	
	Entity		Entity
		Type	Entity Type
What	Op	(CDMID)	
		Name	Operation Name
		Type	Operation Type
		OpObjType	Operation Object Type
	Msg	Text	Message
		RBKey	Resource Bundle Key
	Param		Extra Attributes*
		Order	Extra Attributes*
	Result		Result

Logical group	XML element(s)		XML attribute	ITM Audit attribute
When	Corr			Correlator
	Seq			Sequence
	EvtTS		MS	Timestamp (MS)
			ITM	Timestamp
			(UTC)	
	(LogTS)			
OnWhat	Obj		Type	Object Type
			(Ver)	Object Version
			Name	Object Name
			(CDMID)	Extra Attributes*
			Path	Object Path
	(SecMaint)	Grantee	Type	
		SecPolicy		
		Constraint		
	(SecPolicy)			Security Policy Name
	(Grantee)			Grantee
			Type	Grantee Type
	(PriAuthEvt)			Privilege Or Authority Event
			Type	Privilege Or Authority Type
			AuthID	Assumable Auth ID
			Repository	
	(AuthVal)			
	(AuthCheck)			Extra Attributes*
	(AuthPlugin)		Type	Authorization Plugin Type
			Server	Extra Attributes*

Logical group	XML element(s)		XML attribute	ITM Audit attribute
Where	Origin	Node	Name	Origin Name
			Type	Origin Type
			AddrType	Origin Protocol
			Addr	Origin Address
			Host	Origin Hostname
			Port	Origin Port
			SYSID	Origin
	App		Code	Application Code
			Ver	Application Version
			Comp	Application Component
	SvcPt			Service Point
WhereFrom	Source	Node	Name	Source Name
			Type	Source Type
			AddrType	Source Protocol
			Addr	Source Address
			Host	Source Hostname
			Port	Source Port
			SYSID	Source
	Relay	Node	Name	
			Type	
			AddrType	
			Addr	
			Host	
			Port	
			SYSID	
WhereTo	Target	Node	Name	Target Name
			Type	Target Type
			AddrType	Target Protocol
			Addr	Target Address
			Host	Target Hostname
			Port	Target Port
			SYSID	Target

Audit log XML example

The following sample audit record was generated during start-up and indicates that self-describing agent services on a particular monitoring server is disabled.

```
<AuditEvt Domain="" Type="SYSADMIN" Level="Minumum" Ver="1">
 <Who>
  <UserID/>
  <AuthID>SYSTEM</AuthID>
 </Who>
 <What>
  <Op Name="Self-Describing Agent Status"
   OpObjType="ibm-prod-tivoli-itm:SelfDescribingAgentInstall" Type="Disable"/>
  <Msg Text="Self-Descrbing Agent Feature disabled at the local TEMS."
   RBKey="KFASD010"/>
  <Result>0</Result>
 </What>
 <When>
  <EvtTS MS="1307723083106" ITM="1110610162443106"/>
  <Seq>1</Seq>
 </When>
 <OnWhat>
  <Obj Type="ibm-prod-tivoli-itm:SelfDescribingAgentInstall" Name="SDA Services"/>
 </OnWat>
 <Where>
  <Origin>
   <Node Name=Tivoli Enterprise Monitoring Server" Type="SERVER" AddrType="IPv4"
    Addr="10.1.1.1" SYSID="HUB_NC051039"/>
  </Origin>
  <App Code="KMS" Ver="06.23.00" Comp="KFA"/>
  <SvcPt>system.nc051039_ms</SvcPt>
 </Where>
 <WhereFrom>
  <Source>
   <Node Name="Tivoli Enterprise Monitoring Server" Type"SERVER" SYSID="HUB_NC051039"
    Addr="10.1.1.1" AddrType="IPv4"/>
  </Source>
 </WhereFrom>
 <WhereTo>
  <Target>
   <Node Name="Tivoli Enterprise Monitoring Server" Type="SERVER" AddrType="IPv4"
    Addr="10.1.1.1" SYSID="HUB_NC051039"/>
  </Target>
 </WhereTo>
</AuditEvt>
```

In this example the following questions can be answered:

Question	Tag(s)	Value	Interpretation
Who	UserID	Empty	The empty UserID tag indicates that this event was generated by an unknown UserID or an autonomous process performing an action that was not initiated directly by a user.
	AuthID	SYSTEM	Indicates the ID that this event was authorized under.
What	Op	Self-Describing Agent Status	The “Self-Describing Agent Status” operation was successfully completed (Result 0) with the explanatory message indicating that the self-describing agent feature has been disabled.
	Msg	Self-Describing Feature disabled at the local TEMS.	
	Result	0	
	Type	Disable	Indicates that this particular operation is of the generic “disable” type. Operations are typically self-explanatory, but they are all classified into a generic event model type (GEM), as specified by the Tivoli Security and Information Event Manager.

Question	Tag(s)	Value	Interpretation
When	ITM	1110610162443106	The time that the event was generated (not logged) in Coordinated Universal Time (UTC) format (CYYMMDDhhmmssms). This date reads: June 10, 2011 at 04:24:43 106 ms.
OnWhat	Name	SDA Services	The object name is the affected code, component, of other contextually relevant identifier that receives the operation. In this example, the object “SDA Services” received the operation “Self-Describing Agent Status” which successfully completed (with a result of 0) on the object “SDA Services”.
Where	SYSID	HUB_NC051039	This is where the event was logged. The application KMS on Managed System ID HUB_NC051039 (IP 10.1.1.1) logged this event. This system identifies itself as the Tivoli Enterprise Monitoring Server.
	Addr	10.1.1.1	
	Name	Tivoli Enterprise Monitoring Server	
	App	KMS	
WhereFrom	SYSID	HUB_NC051039	This event was initiated on MSN HUB_NC051039 (IP 10.1.1.1). This system identifies itself as the Tivoli Enterprise Monitoring Server.
	Addr	10.1.1.1	
	Name	Tivoli Enterprise Monitoring Server	
WhereTo	SYSID	HUB_NC051039	The event is targeted at MSN HUB_NC051039 (IP 10.1.1.1). This target system is identified as the Tivoli Enterprise Monitoring Server.
	Addr	10.1.1.1	
	Name	Tivoli Enterprise Monitoring Server	

Audit environment variables

Environment variables can be modified to control the audit capability.

Environment variables

The following environment variables are defined for configuring the Auditing Facility:

Environment variable	Description	Acceptable input	Default if not defined
AUDIT_FILE	Used to disable the creation of an xml audit.log. Audit events are still created in the ITM Audit table.	• Disabled • Enabled **Note:** Not Supported is a state that is not accepted as input but may be returned on platforms that do not create audit logs, such as z/OS systems.	Enabled

Environment variable	Description	Acceptable input	Default if not defined
AUDIT_LOG_DIR _PATH	The path to the directory where the audit log files are kept. Not available on z/OS systems. **Note:** If you change the output directory, the pdcollect tool cannot collect the associated audit log.	Provide an operating system specific path.	*<install_dir>*/ auditlogs
AUDIT_LOG_FILE _LIMIT_MB	The maximum file size of the log file in megabytes (2^20 bytes).	1 - MAXINT-1	9
AUDIT_LOG_FILE _NAME	The log file name. Not available on z/OS systems.	Provide the log file name.	*<UserID>*.*<hostname>* _*<pc>*_audit.log
AUDIT_LOG_MAX _FILES_COUNT	Maximum number of log files for rollover. This variable applies only to distributed platforms. Not available on z/OS systems.	1...MAXINT-1 (on distributed)	5
AUDIT_MAX_HIST	Maximum number of records kept in short-term memory for direct queries.	1...MAXINT-1	100
AUDIT_TRACE	The trace level to pass messages. Message trace levels (from low to high) are Minimum, Basic, Detail. Higher levels trace all lower levels.	• Minimum • Basic • Detail • Disabled	Basic
ITM_DOMAIN	An optional 128-character identifier on distributed systems and 32-character identifier on z/OS systems that you can use to associate with these records. Best suited for commonly identifying agents that are associated with each other. You might use this variable for sorting records by a particular customer. Case is preserved.	Provide an alphanumeric string that can contain the plus (+), minus (-), semicolon (;), and colon (:) characters.	No domain is provided.

Modifying the Audit environment variables

You can configure the tracing environment variable for the Auditing Facility by using the following procedures. You can modify any of the environment variables previously mentioned.

Windows

Use Manage Tivoli Enterprise Monitoring Services (Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services) to edit environment files. Right-click the component you want to modify and click **Advanced** → **Edit ENV File**. You must recycle the component to implement the changes.

Linux UNIX

1. Change to the `<`*`install_dir`*`>/config` directory and open the coordinating file:

   For the monitoring server: `<`*`hostname`*`>_ms_<`*`tems_name`*`>.config`

   For the portal server and single-instance agents: `<`*`pc`*`>.ini`

   For multi-instance agents: `<`*`pc`*`>_<`*`instance`*`>.config`
2. On a new line, add the environment variable followed by the value. For example, `AUDIT_TRACE=BASIC`
3. Save and close the file.
4. Recycle the component to have your changes take effect.

z/OS

See *Configuring the Tivoli Enterprise Monitoring Server on z/OS* for more information.

Take Action and command execution audit logging

If you have IBM Tivoli Monitoring V6.3 or later, audit records are generated for Take Action and **`tacmd executecommand`** execution. Take Action execution includes Take Actions initiated from the Tivoli Enterprise Portal, running the **`tacmd executeaction`** command, situation Take Action commands, and workflow policy Take Action commands. The identity of the user who initiated the Take Action is passed to the monitoring agent using a secure session token.

The session token leverages the common IBM Tivoli Monitoring encryption key and synchronization of time between the IBM Tivoli Monitoring servers and monitoring agents. If the encryption key is not synchronized, then any commands are rejected as invalid due to validation errors with the identity. If system times between the portal server (for Tivoli Enterprise Portal users) or the hub monitoring server (for tacmd command users) is more than 25 minutes out of sync with that of the target monitoring agent according to Universal Coordinated Time (UTC), then the command is rejected as unauthorized due to a permission time out.

Situation Take Action execution and workflow policy Take Action execution records the identity of the user who last modified the situation or workflow policy.

The audit messages are available in the audit log at the monitoring agent or through the Tivoli Enterprise Portal as historical data or real-time queries of the audit log.

The TEMS Security Compatibility Mode allows IBM Tivoli Monitoring server components that are at a version before V6.3 to execute commands or Take Actions for monitoring agents with Tivoli Enterprise Monitoring Agent Framework V6.3 or later. If TEMS Security Compatibility mode is not enabled and you have a portal server or monitoring server at version before V6.3, then Take Actions or **tacmd executecommand** commands might be rejected as unauthorized and audited. When TEMS Security Compatibility Mode is enabled, the identity of the original user might not be available in the audit records. Best practice is to upgrade your infrastructure to IBM Tivoli Monitoring V6.3 or later and to disable TEMS Compatibility mode for maximum security and assurance that the identity of the Take Actions and **tacmd executecommand** executions are properly audited.

You can also use AAGP policies to control which users can execute a TakeAction or **tacmd executecommand** against a managed system. For more details, see"Access Authorization Group Profile" on page 398.

Chapter 10. Situation event integration with Tivoli Enterprise Console

If your monitoring environment includes the Tivoli Enterprise Console event server and situation event forwarding has been configured on the hub Tivoli Enterprise Monitoring Server , you can forward situation events generated by Tivoli Enterprise Monitoring Agents to the event server.

The *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm) provides the instructions to enable situation event forwarding: configuring the event server to receive the events, installing the event synchronization component on the event server, enabling situation forwarding on the hub monitoring server, and defining a default Event Integration Facility (EIF) destination.

Default mapping of situation events to IBM Tivoli Enterprise Console events

This section provides information about attribute mapping of situation events to IBM Tivoli Enterprise Console events. You can use this mapping information when you forward a situation event to the IBM Tivoli Enterprise Console and you want to write correlation rules in the IBM Tivoli Enterprise Console.

The situation event forwarder generates a IBM Tivoli Enterprise Console event with an event class based on the attribute group associated with the situation. When the situation event is forwarded to the event server the associated generated event class inherits event class attribute definitions (either directly or indirectly) from the parent: *Omegamon_Base* class. Because IBM Tivoli Enterprise Console uses hierarchical event classes, use the Omegamon_Base parent class when you want to write a rule for all situation events that you forward to the event server.

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;
```

In specialized cases where a situation event is mapped into an existing IBM Tivoli Enterprise Console event class and the event hierarchy cannot be modified (Omegamon_Base cannot be added to the hierarchy) it is important that the slots from Omegamon_Base be included in the existing event class or in a class somewhere in the hierarchy. This mechanism is not preferred because it does not allow a rule to recognize the presence of Omegamon_Base in the event hierarchy.

As part of the generic mapping for these situations, the IBM Tivoli Monitoring event forwarder assigns associated values for attributed defined in the event class attributes when forwarding an event to the Tivoli Enterprise Console event server. In addition to these event class attributes, values are assigned to the following attributes inherited from the EVENT class, if available: source, hostname, fqhostname, origin, sub_origin, adapter_host, origin, severity, and message attributes that are inherited from the base EVENT class.

Table 29. IBM Tivoli Enterprise Console event class attributes

Event class attributes	Values and meaning
adapter_host	Base EVENT class attribute. Same as hostname (see below). This is application-specific data related to the event, if any.
appl_label	Reserved for future use.
cms_hostname	TCP/IP host name of the Tivoli Enterprise Monitoring Server that forwards the event.
cms_port	The monitoring server port on which the web service is listening.
fqhostname	Base EVENT class attribute that contains the fully qualified hostname, if available.
hostname	Base EVENT class attribute that contains the TCP/IP hostname of the managed system where the event originates, if available.
integration_type	Indicator to help IBM Tivoli Enterprise Console performance. • N for a new event, the first time the event is raised • U for update event, subsequent event status changes
master_reset_flag	Master reset indicator set for master reset events. Value is NULL for all other events: • R for Tivoli Enterprise Monitoring Server recycle master_reset • S for hotstandby master_reset
msg	Base EVENT class attribute that contains the situation name and formula.
origin	Base EVENT class attribute contained in the TCP/IP address of the managed system where the event originates, if available. The address is in dotted-decimal format.
severity	Base EVENT class attribute that contains the resolved severity.
situation_displayitem	Display item of associated situation, if available.
situation_eventdata	Raw situation event data starting from the second event data row, if any. Event data attributes are in key-value pair format. 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	Displayed name of the associated situation.
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 29. IBM Tivoli Enterprise Console event class attributes (continued)

Event class attributes	Values and meaning
situation_status	Current status of the situation event.
situation_time	Timestamp of the situation event.
situation_type	Situation event type**S** for sampled event; **P** for pure event.
situation_thrunode	Reserved for future use.
source	Base EVENT class attribute that contains **ITM**
sub_origin	Base EVENT class attribute. This is the same as the managed system name for the associated situation_displayitem, if any.
sub_source	Base EVENT class attribute that contains the origin managed system name for the associated situation.

Expanding a generic event message situation description

The message slot gives you a descriptive way of looking at an event in the IBM Tivoli Enterprise Console.

The situation name alone does not provide detailed event identification where there are large numbers of like-events from various sources. Rather, the situation name in the message slot sent from the hub monitoring server to the event server is expanded to include the following event attributes:

Situation-Name [(formula) ON Managed-System-Name ON DISPLAY-ITEM (threshold Name-Value pairs)]

where:

Situation-Name
The name 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_Critical_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)]
```

Generic mapping for agent specific slots

Generic mapping identifies the target event class based on information out of a situation that is triggered and forwarded to the event server.

The event class name of the IBM Tivoli Enterprise Console event is derived from the attribute group associated with the situation. It is a combination of **ITM_** plus the attribute group name associated with the situation. For example, a situation using the NT_Process attribute group will generate a IBM Tivoli Enterprise Console event with class *ITM_NT_Process*.

Note: Some agents have very long attribute group names, which might cause the generated event class name to exceed the limit imposed by the event server. In these cases, the event class name will be a combination of **ITM_** plus the table name of the attribute group.

Additional event slot values are populated with situation attribute values from the situation event data. The slot names are the attribute names after special character processing.

For example, a situation using the Process_CPU attribute causes generation of a slot process_cpu in the IBM Tivoli Enterprise Console event. In case the attribute name conflicts with the slot names in IBM Tivoli Enterprise Console EVENT class or Omegamon_Base class, the *applname* associated with the attribute group, for example: *knt_*, is pre-pended to the attribute name to form the slot name.

For complex situations, the situation definition can involve more than one attribute group. In this case, the IBM Tivoli Enterprise Console event class used is derived from the first attribute group encountered in the situation event data of the triggering situation. The exception is when the first attribute group found is Local_Time or Universal_Time; then it is passed over and the next different attribute group, if any, will be used.

For example, if a situation is written for the NT_Process and NT_System attribute groups, NT_Process being the first attribute group, the IBM Tivoli Enterprise Console event class *ITM_NT_Process* is used. Additional event slots are generated based on the attributes of the attribute group selected.

Table 30. Special characters for attribute groups and names in IBM Tivoli Enterprise Console 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
R/3	r3
/ forward slash	_per_
\ backward slash	_ (underscore)
<space>	_ (underscore)
(open parenthesis) close parenthesis	_ (underscore)
< open pointed bracket > close pointed bracket	_ (underscore)

Note: After special character processing, the leading and trailing underscore in the final event class or slot name, if any, will be removed.

Assigning severity for Tivoli Enterprise Console events

The severity of a Tivoli Enterprise Console event associated with a situation is assigned automatically from the situation name or you can set a severity in the Tivoli Enterprise Portal Situation editor.

The severity of a Tivoli Enterprise Console event associated with a situation can be directly specified under the EIF tab of the Situation editor. If no Tivoli Enterprise Console severity is specified for a situation, the event forwarder attempts to derive a severity from the suffix of the situation name using the following rule:

Table 31. Situation name suffix mapping to Tivoli Enterprise Console event severity

Situation name suffix	Assigned IBM Tivoli Enterprise Console severity
Warn or _Warning	WARNING
Cri, _Crit, _Critical	CRITICAL
none of the above	UNKNOWN

Localizing message slots

Edit the KMS_OMTEC_GLOBALIZATION_LOC variable to enable globalization of the EIF event message slots that get mapped to alert summaries by the Tivoli Enterprise Console event server.

About this task

Some products ship with event mapping files and language bundles. The message slots for these defined IBM Tivoli Enterprise Console events are globalized. The language selection is done through a Tivoli Enterprise Monitoring Server environment variable called KMS_OMTEC_GLOBALIZATION_LOC.

By default, this variable is set to American English and the message slots are filled with the American English messages. Edit the variable to enable one of the language packs that are installed in your environment.

Procedure

1. On the computer where the Hub Tivoli Enterprise Monitoring Server is installed, open the KBBENV file:
   - Windows Start Manage Tivoli Monitoring Services, right-click **Tivoli Enterprise Monitoring Server**, and click **Advanced → Edit ENV file**.
   - Linux UNIX In a text editor, open the `<install_dir>/config/<tems_name>_ms_<address>.cfg` file, where *<tems_name>* is the value supplied during the monitoring server configuration, and *<address>* is the IP address or fully qualified name of the computer.
2. Locate (or add) the KMS_OMTEC_GLOBALIZATION_LOC environment variable and enter the desired language and country code, where xx is the language and 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 (such as pt_BR for Brazilian Portuguese or zh_CN for Simplified Chinese).
   ```
   KMS_OMTEC_GLOBALIZATION_LOC=xx_XX
   ```
3. Save and close the monitoring server environment file.

Situation event statuses and IBM Tivoli Enterprise Console event generation

This topic describes the meaning of the situation event statuses and the setting of the common slots in the generated IBM Tivoli Enterprise Console event.

situation is true

integration_type: N, the first time the situation is true; U, all subsequent times

situation_status: Y

situation_name: Name of the situation

situation_display_item: Value of the attribute that was selected as the display item in the situation definition, if any.

master_reset_flag: None

situation reset (no longer true)

integration_type: U

situation_status: N

situation_name: Name of the situation

situation_display_item: Value of attribute selected as display item in the situation definition, if any.

master_reset_flag: None

acknowledge

integration_type: U

situation_status: A

situation_name: Name of the situation

situation_display_item: Value of the attribute that was selected as the display item in the situation definition, if any.

master_reset_flag: None

situation start

integration_type: None

situation_status: S

situation_name: Name of the situation

situation_display_item: None

master_reset_flag: None

No IBM Tivoli Enterprise Console event is forwarded.

situation stop

integration_type: U

situation_status: P

situation_name: Name of the situation

situation_display_item: None

master_reset_flag: None

All opened situation events that originated from this Tivoli Enterprise Monitoring Server will be closed on the event server.

situation startup error

integration_type: None

situation_status: X

situation_name: Name of the situation

situation_display_item: None

master_reset_flag: None

No IBM Tivoli Enterprise Console event is forwarded.

acknowledge expired

integration_type: U

situation_status: F

situation_name: Name of the situation

situation_display_item: Value of the attribute that was selected as the display item in the situation definition, if any.

master_reset_flag: None

Expiration that was specified in the acknowledge has expired.

resurface

integration_type: U

situation_status: E

situation_name: Name of situation

situation_display_item: Value of the attribute that was selected as the display item in the situation definition, if any.

master_reset_flag: None

The acknowledgement was removed before it had expired and the situation is still true.

hub start

integration_type: None

situation_status: N

situation_name: “**'

situation_display_item: None

master_reset_flag: R

After the hub monitoring servers started, a master reset event is sent with situation_status=N. Master reset causes the event server to close all opened situation events from the hub monitoring server (`cms_hostname` value).

hub restart

integration_type: None

situation_status: N

situation_name: “**'

situation_display_item: None

master_reset_flag: R

After the hub monitoring server is started, a master reset event is sent with situation_status=N. Master reset causes the event server to close all opened situation events from the hub monitoring server (`cms_hostname` value).

hub Standby failover

integration_type: None

situation_status: N

situation_name: “**”

situation_display_item: None

master_reset_flag: S

After the hub monitoring server switch takes place, a hot standby master reset event is sent with situation_status=N. Master reset causes the event server to close all opened situation events from the hub monitoring server. The name of the old primary hub is in the `situation_origin` slot.

Note: The `integration_type` value is solely used by the IBM Tivoli Enterprise Console synchronization rule to improve its performance. It has no other meaning related with the event.

Synchronizing situation events

The event synchronization component, the Event Integration Facility or EIF, sends updates to situation events that are forwarded to a Tivoli Enterprise Console event server back to the Tivoli Enterprise 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.

Checking the IBM Tivoli Enterprise Console event cache

The event server rules event cache must be large enough to contain the volume of events expected at any given time.

To check the rules cache size for a running event server, run the following the IBM Tivoli Enterprise Console command:

```
wlsesvrcfg -c
```

To set this rules cache size, run the IBM Tivoli Enterprise Console command:

```
wsetesvrcfg -c number_of_events
```

Note: For more information regarding these two commands, see the "Event server commands" in the *Tivoli Enterprise Console Command and Task Reference*.

If the rules event cache become full, the IBM Tivoli Enterprise Console rules engine generates a TEC_Notice event, `Rule Cache full: forced cleaning`, indicating that 5 percent of the events from the cache were removed. Events are removed in order by age, with the oldest events removed first allowing newer events to be processed.

When the hub monitoring server forwards a status update for a situation event previously forwarded to the Tivoli Enterprise Console event server, if the original situation event is deleted from the rules event cache, then a TEC_ITM_OM_Situation_Sync_Error event is generated to indicate that the monitoring server and the event server are out of synchronization.

When using any IBM Tivoli Enterprise Console viewer to acknowledge or close any situation event, if the situation event has been deleted from the rules event cache, the status change is not processed by the IBM Tivoli Enterprise Console rules engine. Also, the situation event update is not forwarded to the originating Tivoli Enterprise Monitoring Server. This behavior results from theIBM Tivoli Enterprise Console rules engine not processing any event status changes for any event not contained in the rules event cache. In this case, the event status change is updated only in the IBM Tivoli Enterprise Console database.

Both situations can be remedied by performing a IBM Tivoli Enterprise Console server configuration parameters analysis and performance analysis to determine the optimal configuration parameter settings and desired performance requirements. Refer to “Rule engine concepts”, in the *IBM IBM Tivoli Enterprise Console Rule Developer’s Guide* for more information.

Changing the configuration of the event synchronization 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.

About this task

You can run this command by using one of the following options:

Procedure

- Manually modify the configuration file for event synchronization (named `situpdate.conf` by default and located in the and located in the `/etc/TME/TEC/OM_TEC` directory on operating systems such as UNIX, and the %SystemDrive%\Program Files\TME\TEC\OM_TEC\etc directory on Windows), and then run the following command:

  ```
  sitconfig.sh update <config_filename>
  ```

- Run the **sitconfig.sh** command directly, specifying only those settings that you want to change. See *IBM Tivoli Monitoring: Command Reference* for the full syntax of this command.

What to do next

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process from the `$BINDIR/TME/TEC/OM_TEC/bin` directory with the **stopSUF** and **startSUF** commands.

Defining additional monitoring servers for the event synchronization on the event server

For each monitoring server that is forwarding situation events to the event server, you must have the required server information defined so that the Situation Update Forwarder process forwards situation event updates to the originating monitoring server.

About this task

Run the following command to add new monitoring server information:

```
sitconfsvruser.sh add serverid=server userid=user password=password
```

where:

serverid=*server*
The fully qualified host name of the monitoring server.

userid=*user*
The user ID to access the computer where the monitoring server is running.

password=*password*
The password to access the computer.

Repeat this command for each monitoring server that you want to add.

What to do next

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process from the `$BINDIR/TME/TEC/OM_TEC/bin` directory with the **stopSUF** and **startSUF** commands (.cmd file extension on Windows; .sh on operating systems such as UNIX).

Closing sampled events

When a situation event from a sampled situation is forwarded to the IBM Tivoli Enterprise Console event server and that event is subsequently closed in the event server, the behavior of event synchronization is to send a request to the Tivoli Enterprise Monitoring Server to acknowledge the situation with a specified timeout. The reason for this is because closing events from sampled situations causes problems with the situation's ability to fire after the close in IBM Tivoli Monitoring.

About this task

If the acknowledgment of the situation expires and the situation is still true, then a new situation event is opened in the IBM Tivoli Enterprise Console. If the situation becomes false, then it resets itself in IBM Tivoli Monitoring and the event remains closed in the IBM Tivoli Enterprise Console.

The default acknowledgment expiration time is 59 minutes. This can be changed in the situation timeouts configuration file on the event server (`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 IBM Tivoli Enterprise Console rule using the `sitconfig.sh` refresh command in `$BINDIR/TME/TEC/OM_TEC/bin`.

Changing rule set parameters for the omegamon.rls rule set file

The omegamon.rls rule set file has parameters that you can edit, according to your environment, to tune performance or to set your own customized values. Using these parameters, you can write and customize IBM Tivoli Enterprise Console rules. During installation, you can choose the location of the rule base. Otherwise, you can use the wrb -lscurrb -path to find the current rule base.

Here are some reasons why you might want to change the behavior of the rule:

- For the omegamon.rls file, *omegamon_admin* is the name of the rule set but you can name your rule set after your administrator's name or some other value.
- Similarly, the *sit_ack_expired_def_action* rule set name is set to REJECT by default. This setting means that whenever a situation event acknowledgement expires in the Tivoli Enterprise Portal and the event becomes OPEN in the portal, the IBM Tivoli Enterprise Console event server rejects this action and re-acknowledges the event in the portal. You have the option of accepting the change that was initiated by the portal and changing the status in the IBM Tivoli Enterprise Console instead.

The following user-configurable parameters are available:

omegamon_admin

Use this identifier when a rule defined in this rule set closes an event. This identifier is used to differentiate close operations that were originated automatically rather than initiated by the console operator.

omsync_timeout
This attribute sets the period in seconds that you must wait to distinguish between the synchronization of single or multiple events. The default timeout is 3 seconds.

omsync_maxentries
This attribute sets the maximum number of events allowed per batch. Default batch size is 100 events.

Warning: Setting this value less than 20 events might cause contentions within the IBM Tivoli Enterprise Console task process, causing poor performance of events that are synchronized back to the Tivoli Enterprise Monitoring Server.

sit_resurface_def_action
This attribute determines the default action of the rules in case a situation update event arrives from Tivoli Enterprise Monitoring Server to resurface or reopen an event that has already been acknowledged. The two possible values are ACCEPT and REJECT. The default is ACCEPT.

sit_ack_expired_def_action
This attribute determines the default action of the rules in case a situation update event arrives from the Tivoli Enterprise Monitoring Server to reopen an event that has already been acknowledged. This happens when a situation's acknowledgement in the monitoring server expires and the situation event is reopened. The two possible values are ACCEPT and REJECT. The default is REJECT.

sf_check_timer
This attribute specifies the interval at which the state of the situation update forwarder is checked. It reads events from the cache files and send them to the Tivoli Enterprise Monitoring Server using Web Services. The default is 10 minutes.

After modifying any configuration parameters and saving omegamon.rls, you must recompile and reload the rule base and recycle the event server. To recompile the rule base, enter the following command, where `Rulebase_Name` is the name of the actively loaded rule base containing the omegamon.rls rule set:

```
wrb -comprules Rulebase_Name
```

To reload the rule base, issue the following command:

```
wrb -loadrb Rulebase_Name
```

To stop the Event server, issue the following command:

```
wstopesvr
```

To restart the Event server issue the following command:

```
wstartesvr
```

For more information regarding the **wrb, wstopesvr, and wstartesvr** commands, see the *Command and Task Reference* at the Tivoli Enterprise Console Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.itec.doc_3.9/welcome_nd.html).

Tuning considerations

Integration parameters supporting actions at the IBM Tivoli Enterprise Console event console that are reflected at the Tivoli Enterprise Portal event console provide good response times with a reasonable system resource investment.

The tuning parameters to consider include:

- `omsync_timeout` in the `omegamon.rls` with a default of 3 seconds.
- `PollingInterval` in event synchronization with a default of 3 seconds.
- IBM Tivoli Enterprise Console event console refresh interval with a default 60 seconds
- Tivoli Enterprise Portal event console refresh interval

Note: Shorter intervals result in the consumption of more system resources.

The delivery time of situation changes from the IBM Tivoli Enterprise Console event console to the Tivoli Enterprise Portal event console results from the `omsync_timeout` and `PollingInterval` settings working in parallel. To improve the response time, you can reduce these settings down to a minimum of 1 second. .

You can adjust the refresh interval for both consoles:

- For the IBM Tivoli Enterprise Console, change the allowable range using the IBM Tivoli Enterprise Console event console – Configuration. In the subsequent Event View displays, adjust the preferences.
- For the Tivoli Enterprise Portal event console, click **View > Refresh Every** to access the refresh intervals.

Using the Rules Check utility

The Rules Check utility provides you with the ability to assess the impact on an existing set of rules whenever the designs of BAROC (Basic Recorder of Objects in C) event classes are changed. This utility allows you to verify which rules might have been impacted by these event class definition changes.

There are two important sets of files that are used and required by the Rules Check utility to check the possible impacts of event classes design changes to the rules:

- BAROC Event Classes Definition files:

  IBM Tivoli Enterprise Console class definitions are hierarchical in nature with inheritances. One class can inherit from another class, and all attributes from the parent class are available in the child class. The EVENT class is the base IBM Tivoli Enterprise Console class. The other classes usually derive from the IBM Tivoli Enterprise Console EVENT class.

  In IBM Tivoli Enterprise Console, the BAROC Event Class Definition files (*.baroc files) are located in the actively loaded rule base's TEC_CLASSES subdirectory. They provide the event class definitions used by the IBM Tivoli Enterprise Console Server. Although the tool is closely integrated with IBM Tivoli Enterprise Console and uses the active rule base's TEC_CLASSES subdirectory by default input, the tool is not dependent on this subdirectory, and accepts as alternative input any other directory that contains the correct BAROC files and to which the user has read privileges.
- Rules files:

  The IBM Tivoli Enterprise Console product rule language also supports the inheritance nature of the IBM Tivoli Enterprise Console class definitions. When a predicate in the IBM Tivoli Enterprise Console rule is looking for a particular class, all classes that inherit from that particular class also satisfy the rule predicate.

In IBM Tivoli Enterprise Console, the rule set files (*.rls files) are located in the actively loaded rule base's TEC_RULES subdirectory. They provide the rule sets and are deployed to the IBM Tivoli Enterprise Console Server. Although the tool is closely integrated with IBM Tivoli Enterprise Console and uses the active rule base's TEC_RULES subdirectory by default input, the tool is not dependent on this subdirectory. The tool accepts as an alternative input any other directory that contains the correct rule sets and to which the user has read privileges.

The Rules Check utility is included with IBM Tivoli Monitoring. This utility is installed in the $BINDIR/TME/TEC/OM_TEC/bin directory as part of the IBM Tivoli Enterprise Console Event Synchronization installation. It does not require any specific directory configuration if the required privileges for access to the input and output files are granted.

To run the Rules Check command you must have:

- Read access to the *.rls and *.baroc files that are used as inputs.
- Write access to the output that is used to store the results of the check.
- IBM Tivoli Enterprise Console administrator authority.
- When no `–cd` and `–rd` options are specified, the user issuing the command must have the proper TME authorization, and verify the level of wrb subcommands that are required.

To run the Rules Check utility and see sample output, refer to the *Command Reference*.

Editing the Event Integration Facility configuration

Edit the **Tivoli Event Integration Facility** EIF file to customize the configuration such as to specify up to five failover EIF servers or to adjust the size of the event cache.

Before you begin

After 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 configuration file specifies the default EIF receiver of forwarded situation events.

See the *Tivoli Event Integration Facility Reference* for more details on the parameters and values.

If you are enabling EIF after your environment has been installed and configured, you must enable EIF through Manage Tivoli Enterprise Monitoring Services or with the CLI **`itmcmd config -S`** and then recycle the monitoring server and portal server.

See the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm) for instructions on configuring the monitoring server to enable the Tivoli Event Integration Facility.

About this task

Take these steps to edit the EIF configuration file:

Procedure

1. Open the `om_tec.config` file:
   - Windows In the Manage Tivoli Monitoring Services window, right-click Tivoli Enterprise Monitoring Server and click **Advanced** → **Edit EIF Configuration**.
   - Linux UNIX Open `*install_dir*/tables/*host name*/TECLIB/om_tec.config` in a text editor.
2. Edit any of the event server configuration parameters for the event integration facility.

Option	Description
ServerLocation=	This is the *host name* or *ip address* of the event server. To provide event failover, you can indicate up to five default event servers, separating each with a comma. When the default event server is unavailable, the situation event goes to the next server in the list. Value: **tec_server_addr**
ServerPort=	The event server listening port, which is 5529 by default. Specify `0` if the event server uses the port mapper. If you specified multiple server locations, add the corresponding port numbers here, separating each with a comma. Value: **[port:0]**
EventMaxSize=	Maximum number of characters allowed in the event. This number is disabled by default. To enable it, remove the # (pound symbol) at the beginning of the line. Value: **4096**
RetryInterval=	The number of times to retry connection with the event server before returning an error. Value: **5**
getport_total_timeout_usec=	The number of seconds to continue attempting to connect to the event server port before timing out. The default is 14 hours. Value: **50500**
NO_UTF8_CONVERSION=	Events are already in UTF8 format; no conversion is needed. This parameter must be set to `YES`. Value: **YES**
ConnectionMode=	The connection mode. Value: **co**
BufferEvents=	Whether the EIF buffers the event. This must be set to YES. Value: **YES**

Option	Description
BufEvtMaxSize=	Maximum size of the event cache. The default is initially 4096 KB and you can change it here. Value: **4096**
BufEvtPath=	Path of the event cache file. The default is ./TECLIB/om_tec.cache. Value: **./TECLIB/om_tec.cache**
FilterMode=	Enable event filtering. This is set to OUT by default. Value: **OUT**
TcpTimeout=	Use this parameter for connection-oriented mode that allows the agent to timeout the connect call with the primary server if it is not available, and then failover to the secondary server. For example, use this parameter when a firewall blocks ICMP Ping calls. The value is in seconds. Value: **75** **Restriction:** This cannot be used with parameters PingTimeout and NumberOfPingCalls.
PingTimeout=	The maximum timeout for the ping call to try to access the destination server. If the PingTimeout is not specified, a ping call is not executed by the EIF before calling the socket connect call. This parameter can be used with connection-less or connection-oriented connection types. This parameter must be used with NumberOfPingCalls. The value is in seconds. Value: **75** **Restriction:** This cannot be used with parameter TcpTimeout.
NumberOfPingCalls=	The number of times the ping function should be invoked before determining the destination server is available. Due to some TCP/IP configurations, the very first ping call after the destination server is unplugged can return successfully. This parameter can be used with connection-less or connection-oriented connection types. This parameter must be used with PingTimeout. **Restriction:** This cannot be used with parameter TcpTimeout.
Filter:	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. Value: **Class=ITM_Generic; master_reset_flag='';**

3. When you are finished editing `om_tec.config`, save the file.
4. You must restart the monitoring server or, alternatively, you can use the **refreshTECinfo** command to complete the updates without having to restart

the monitoring server. To use this command, log into the command-line interface with **tacmd login** , then run **tacmd refreshTECinfo -t eif** to complete the EIF configuration.

Results

The monitoring server uses the edited EIF configuration to forward event to the receiver.

What to do next

If this is the first time that you have configured the EIF forwarding after a Tivoli Management Services upgrade, you also must recycle the Tivoli Enterprise Portal Server and users must restart the Tivoli Enterprise Portal. Otherwise, the EIF tab will be missing from the Situation editor.

An alternative method for editing the EIF configuration is provided through the Command Line Interface **tacmd createEventDest**. See the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) for a description.

Related reference:

Tivoli Event Integration Facility Reference
Tivoli Enterprise Console Information Center for more information about the parameters and values.

Tivoli Monitoring Installation and Setup Guide
Configure the monitoring server to enable the Tivoli Event Integration Facility

Tivoli Monitoring Command Reference
Complete EIF configuration updates with tacmd refreshTECinfo; make updates with tacmd createEventDest

Specifying EIF forwarding for a situation event

When the Tivoli Enterprise Monitoring Server has been configured for the **Tivoli Event Integration Facility**, situation events can be forwarded to the event receiver. Use the Tivoli Enterprise Portal Situation editor to set the destination event receiver for individual situations.

Before you begin

One of the Tivoli Enterprise Monitoring Server configuration options is **Tivoli Event Integration Facility**. When this option is enabled, the default EIF receiver is specified in the event server Location and Port Number window that opens (and described in the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm) and *Tivoli Event Integration Facility Reference*). Thereafter, all situation events are forwarded to the EIF receiver by default, using the severity derived from the situation name or the ⊗ Critical severity if none can be derived.

You can override this default for individual situations through the EIF tab of the Situation editor in the Tivoli Enterprise Portal.

Up to eight event destinations can be specified for a forwarded situation event. The event destination association can be done on the EIF tab of the Situation editor. The event destinations must be predefined with the **tacmd createEventDest** command. Changes to the list of event destinations do not take effect until either the **tacmd refreshTECinfo** command is issued or the hub monitoring server is recycled. Additionally, if this is the first time that you have configured the EIF forwarding after a Tivoli Management Services upgrade, you also must recycle the Tivoli Enterprise Portal Server and users must restart the Tivoli Enterprise Portal to see the EIF tab in the Situation editor.

Alternate event destinations that were specified in the tecserver.txt file from earlier releases will be defined as valid event destinations automatically as part of the tecserver.txt file migration.

If multiple default event destinations are specified (in other words, multiple event destination servers have Default set to 'Y'), they all must be selected in the Tivoli Enterprise Portal for events to be forwarded to all defined default destinations.

About this task

Complete these steps to specify the destination EIF receiver and severity for forwarded events:

Procedure

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 to forward.
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 that is used for the situation at this Navigator item.
6. To assign other EIF receivers instead of or in addition to the <Default EIF Receiver>, use one of the following steps:
   - 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 defined through Manage Tivoli Monitoring Services or with the **tacmd createEventDest** command. See the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).
7. Save the situation definition with your changes by clicking **Apply**, to keep the Situation editor open, or **OK**, to close the Situation editor.

Related reference:

Tivoli Event Integration Facility Reference
Tivoli Enterprise Console Information Center for more information about the parameters and values.

Tivoli Monitoring Installation and Setup Guide
Configure the monitoring server to enable the Tivoli Event Integration Facility

Tivoli Monitoring Command Reference
Complete EIF configuration updates with tacmd refreshTECinfo; make updates with tacmd createEventDest

Customizing the event message

From the Situation editor **EIF** tab, you can create map definitions for situation events sent to the EIF receiver. 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 events and events forwarded to the Tivoli Enterprise Console event server.

When the Base Slot name is `msg`, the Literal value column is used for the message template. The message template consists of fix message text and variable substitution references, or *symbols*. The symbol can refer to common or event slot data or a special reference to the situation formula. Common slots are those that are included in all forwarded events, such as situation_name; event slots are those specific to the situation. The following syntax rules apply when setting event slots:

- For an event slot, use the fully qualified attribute name ($Attribute_Table.Attribute_Name$)
- For a common slot, use the variable name that is not fully qualified (no . periods) unless it is the situation symbol
- For a situation formula, use $formula$

These characters are not supported: < less than, > greater than, " quotation mark, ' single quotation mark, and & ampersand. This column is available only if no value is selected in the Mapped attribute column. See the Tivoli Enterprise Portal online help or the *Tivoli Enterprise Portal User's Guide* for more information.

For the `msg` slot, typical users specify a Literal value, not a Mapped attribute value. If a value is specified in the Mapped attribute column for the `msg` slot, the following occurs:

- If **Map all attributes** is not selected, then the Mapped attribute for the `msg` will not be present in the event, and will be ignored.
- If **Map all attributes** is selected, then the Tivoli Enterprise Console event will only have the default message template and not the Mapped attribute specified in the `msg` slot.

Updating the XML used by the MCS Attribute Service

The default XML file used by the Multiple Console Support (MCS) Attribute Service includes only the event classes defined in the BAROC files within the TECLIB branch of the hub monitoring server installation. Generate a new XML file for EIF Slot Customization whenever a new type of agent is added into the Tivoli Management Services infrastructure or when a new event class has been added into the Tivoli Enterprise Console event server.

Before you begin

If an event class specified for a rule is not found within the current event class definition set and you continue building the rule with the current definition set, any unrecognized event classes will be removed from the rule.

The EIF Event 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 for 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.

To generate a new MCS XML file, install the Tivoli Enterprise Console Event Definition Generator (TEDGEN) utility supplied on the IBM Tivoli Monitoring **Tools** DVD. Install the TEDGEN utility on a distributed computer where the hub monitoring server or portal server is installed, or where the Tivoli Enterprise Console is installed. The computer on which you generate the MCS XML file must have the necessary BAROC files.

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 the new agent, be sure all the BAROC definitions for the OS agents and new agent are loaded at the IBM Tivoli Enterprise Console event server or are all in the same directory on the hub monitoring server or portal server before running the TEDGEN utility to generate the MCS XML file.

To install and configure the TEDGEN tool, perform these steps.

1. Install the TEDGEN utility from the IBM Tivoli Monitoring **Tools** DVD on either the hub monitoring server, portal server or where the Tivoli Enterprise Console is installed. The utility 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. Linux UNIX If you installed the TEDGEN utility on a portal server on Linux or UNIX system, also perform these additional configuration steps:
   a. Create the `*install_dir*/tables/cicatrsq/TECLIB` directory if it does not 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.

About this task

These steps assume you have installed the TEDGEN utility on the computer where you want to run the tool: at the Tivoli Enterprise Console event server, the hub monitoring server or portal server.

You must also install the application support for the agents whose EIF events are customized on the hub monitoring server or portal server where you want to run the utility. If you want to run the TEDGEN utility on the Tivoli Enterprise Console, you must also load the agent's BAROC files on the computer where the Tivoli Enterprise Console is installed.

After installing the utility and application support, then run the TEDGEN command to create a new XML file for EIF Slot Customization.

Note:

- If you have installed the TEDGEN utility on a portal server installed on Linux or UNIX, the BAROC files will not exist unless the **Install TEMS support for remote seeding** option was selected during the agent's application support installation. See Installing application support files on a computer with no monitoring server in the *IBM Tivoli Monitoring Installation and Setup Guide*. This action places the BAROC files on the portal server under the `install_dir/tables/ cicatrsq/TECLIB` directory where *install_dir* is the directory where IBM Tivoli Monitoring is installed.
- If you installed the TEDGEN utility on 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 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.

Procedure

1. Complete one of the following steps to run the TEDGEN command:
   - On the computer where the hub monitoring server or portal server is located, issue these commands:

     Windows

     ```
     set CANDLE_HOME=install_dir
     ```

     ```
     cd TEDGEN_Install_dir\scripts
     ```

     ```
     tedgen -itmDir install_dir\{CMS|CNPS}
     \TECLIB -id server_id -xmlPath output_xml_file_path
     ```

     where, *install_dir* is the directory where IBM Tivoli Monitoring is installed, and *TEDGEN_Install_dir* is the directory where the TEDGEN utility is installed.

     Linux UNIX

     ```
     export CANDLEHOME=install_dir
     ```

     ```
     cd TEDGEN_Install_dir/scripts
     ```

     ```
     tedgen -itmDir
     install_dir/tables/{tems_name|cicatrsq}/
     TECLIB -id server_id -xmlPath output_xml_file_path
     ```

     where, *install_dir* is the directory where IBM Tivoli Monitoring is installed, and *TEDGEN_Install_dir* is the directory where the TEDGEN utility is installed.

     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
     ```

   - On the computer where the Tivoli Enterprise Console event server is located, install the TEDGEN utility from the **Tools** DVD that comes with the event server installation media. Then create a new XML file:

   a. Issue the **wrb -imprbclass** command to import the BAROC file that is installed with a newly added agent, and OS agents if they are not already installed:

      ```
      wrb -imprbclass class_file [ -encoding encoding ]
      [-before class_file | -after class_file] [-force] rule_base
      ```

   b. Issue the wrb -loadrb command to reload the rulebase:

      ```
      wrb -loadrb rule_base
      ```

   c. Stop and restart the event server by running these commands:

      ```
      wstopesvr
      wstartesvr
      ```

   d. Issue the TEDGEN command to generate the XML file:

      ```
      tedgen [ -bcDir baroc_classes_directory | -rbName rule_base_name ]
      -id server_id -xmlPath output_xml_file_path
      ```

   Example

   In the following example, the XML file named **tec.xml** is generated from the current rulebase on the Tivoli Enterprise Console event server named **mytec**.

   ```
   tedgen -id mytec -xmlPath tec.xml
   ```

2. Copy the newly generated XML file to the computer where the Tivoli Enterprise Portal Server is installed.
3. Edit the portal server environment file to specify the path to the XML file:
   a. Windows In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server** and click **Advanced → Edit ENV File** to open the `kfwenv` file in the text editor.
   Linux UNIX Open *install_dir*`/config/cq.ini` in a text editor.
   b. Locate the KFW_MCS_XML_FILES environment variable and type = (equal sign) followed by the path to the MCS XML file.
   c. Save and close the environment file.
   d. Windows Restart the portal server.
   Linux UNIX Reconfigure the portal server and then restart it.

Using the NetView console through the IBM Tivoli Enterprise Console event viewer

You can launch the IBM Tivoli NetView® Java console from the IBM Tivoli Enterprise Console views, navigating from an event row to the associated network topology and diagnostics. The selected event must contain a valid host name or IP address to support the topology display of the node associated with the event. Otherwise, the standard topology view is displayed without a specific node selected.

About this task

IBM Tivoli Enterprise Console rules automatically synchronize the events forwarded by Tivoli NetView to the IBM Tivoli Enterprise Console server. The event status updates are reflected on the system where you launch the Netview event console.

Ensure that you have netview.rls and netview BAROC files in the actively loaded rule base. For details, see the *Rule Set Reference* at the Tivoli Enterprise Console Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.itec.doc_3.9/welcome_nd.html).

If you want to use the NetView console through the IBM Tivoli Enterprise Console view in the Tivoli Enterprise Portal, you must configure the *NVWC_HOME* variable in the shell script that launches the Tivoli Enterprise Portal client, to point to the installation directory of NetView Web Console.

To set the *NVWC_HOME* variable, complete the following procedure:

Procedure

- Windows *install_dir*\cnp\cnp.bat
- Linux or UNIX *install_dir*/bin/cnp.sh

What to do next

The NetView Web Console must be installed on the computer where the Tivoli Enterprise Portal client is running to launch the NetView console from IBM Tivoli Enterprise Console view.

See the IBM Tivoli Enterprise Console product documentation for more detailed information about using the NetView console.

Chapter 11. Situation event integration with Tivoli Netcool/OMNIbus

Use the Tivoli Event Integration Facility (EIF) interface to forward enterprise 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 server.

Updates to those events are also sent to OMNIbus. When an OMNIbus user acknowledges, closes, or reopens a forwarded event, OMNIbus sends those changes to back to the monitoring server that forwarded them.

Situation events from Tivoli Enterprise Monitoring Agents that are sent as SNMP alerts to the Netcool/OMNIbus SNMP Probe can also be used to integrate with Netcool/OMNIbus.

See "Setting up event forwarding to Tivoli Netcool/OMNIbus" in the *IBM Tivoli Monitoring Installation and Setup Guide* for instructions to enable situation event forwarding: configuring the OMNIbus server for program execution from scripts, updating the OMNIbus db schema, configuring the EIF probe, enabling situation forwarding on the hub monitoring server, and defining a default event integration facility (EIF) destination.

Chapter 12. Configuring connectors for the common event console

The *common event console* is a Tivoli Enterprise Portal view that provides a single, integrated display of events from multiple event systems. In one table, the common event console presents events from the event systems, and users can sort, filter, and perform actions on these events. The following event systems are supported:

- IBM Tivoli Monitoring
- IBM Tivoli Enterprise Console
- IBM Tivoli Netcool/OMNIbus

A *common event connector* (frequently called a *connector*) is software that enables the integrated display of events from multiple event systems in the common event console. A connector retrieves event data from an event system and sends user-initiated actions to be run in that event system. For example, if you perform an action on a Tivoli Enterprise Console or Netcool/OMNIbus event in the common event console, the associated common event console connector sends that action to the originating event system (Tivoli Enterprise Console or Netcool/OMNIbus) for execution. To have the events from a specific event system displayed in the common event console, you must configure a connector for that event system and set a variable in the Tivoli Enterprise Portal Server environment file.

Common Event Console Configuration window

Use the Common Event Console Configuration window to configure a common event console connector for each of your event system instances. Because the connector for the IBM Tivoli Monitoring product is pre-configured when you install the product, the common event console includes situation events by default. However, to have IBM Tivoli Enterprise Console or IBM Tivoli Netcool/OMNIbus events included in the common event console, you must configure a connector for each of these event systems after you install the IBM Tivoli Monitoring product. This configuration includes specifying which event systems are used to obtain events for display in the common event console. You might also want to change some of the configuration values for the IBM Tivoli Monitoring connector.

About this task

To configure connectors, open the Common Event Console Configuration window by performing the following steps on the computer where the Tivoli Enterprise Portal Server is installed and complete the following procedure:

Procedure

- Windows
  1. Select **Start** → **Programs** → **IBM Tivoli Monitoring** → Manage Tivoli Enterprise Monitoring Services.
  2. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server**.
  3. In the menu, click **Reconfigure**.

4. In the first configuration window, click **OK**.
5. In the second configuration window, click **OK**.
6. Click **No** in answer to the question “Do you want to reconfigure the warehouse connection information for the Tivoli Enterprise Portal Server?”

- Linux or UNIX
  1. At the command line, change directory (cd) to `*install_dir*/bin` and enter `./itmcmd manage`.
  2. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Server**.
  3. In the pop-up menu, click **Configure**.

Results

The portal server stops and, after a moment, the Common Event Console Configuration window opens with the following tabs:
- ITM Connector
- TEC Connector
- OMNIbus Connector
- Names of Extra Columns

ITM Connector tab

Click the **ITM Connector** tab to view or change the information for the IBM Tivoli Monitoring connector. Because the Tivoli Monitoring event system has a single hub Tivoli Enterprise Monitoring Server, you configure only one IBM Tivoli Monitoring connector.

The following information defines the IBM Tivoli Monitoring connector:

Enable this connector
: You can choose `Yes` or `No`. A value of `Yes` means that IBM Tivoli Monitoring events are available in the common event console.

Connector name
: The name that is to be displayed in the common event console for this connector.

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.

View closed events
: You can choose `Yes` or `No`. A value of `Yes` means that closed events for this connector are available in the common event console.

TEC Connector tab

Click the **TEC Connector** tab to view or change the information for an IBM Tivoli Enterprise Console connector. To have the events from a Tivoli Enterprise Console server displayed in the common event console, you must configure an IBM Tivoli Enterprise Console connector.

To configure a connector, click **New**. The resulting TEC Connector page contains the following information that defines an IBM Tivoli Enterprise Console connector:

Connector name
: The name that is to be displayed in the common event console for this connector.

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.

Computer name of event system

The computer name of the event system that is associated with this connector.

Port number of event system

The object dispatcher (oserv) port number, typically 94. This is the port that the connector uses to retrieve events from the Tivoli Enterprise Console event system.

This is not the port used to connect to the Tivoli Enterprise Console event server (5529 by default).

User name for accessing event system

The user name that is used when accessing the event system that is associated with this connector.

Password

The password that is associated with the user name.

Event group that defines events for common event console

The Tivoli Enterprise Console event group that defines which events are available in the common event console.

If you do not specify an event group, all Tivoli Enterprise Console events are available in the common event console.

If you want to restrict events further, you can also define a clause in the **SQL WHERE clause that restricts events for common event console** field.

SQL WHERE clause that restricts events for common event console

This clause can be applied only to the part of an event that is built from the Tivoli Enterprise Console base attribute table. For example, `status <> 30` causes all events with a status that is not equal to 30 to be available in the common event console.

If you do not define a clause, all Tivoli Enterprise Console events are available in the common event console, unless they are excluded by an event group that you specified in the **Event group that defines events for common event console** field.

View closed events

You can choose `Yes` or `No`. A value of `Yes` means that closed events for this connector are available in the common event console.

Time interval (in minutes) for polling event system

The number of minutes between each poll of the event system for new or changed events.

Time interval (in minutes) for synchronizing events

The number of minutes between each poll of the event system to determine which events have been deleted.

Time interval (in seconds) between reconnection attempts

The number of seconds of delay between reconnection attempts when the connector loses its connection to the event system.

Number of reconnection attempts

The maximum number of consecutive reconnection attempts to make if the connector loses its connection to the event system.

If this value is set to -1 and the connector loses its connection, the connector attempts to reconnect indefinitely.

Information for extra table columns

The common event console includes five extra table columns that you can customize. In the remaining fields on this TEC Connector page, you can define the Tivoli Enterprise Console attribute type and attribute name that identify the attribute that is to be mapped to each of these customizable columns.

For the attribute type, you can choose one of the following values:

- `Base`, which means that the attribute is from the Tivoli Enterprise Console base attribute table.
- `Extended`, which means that the attribute is from the Tivoli Enterprise Console extended attribute table.

OMNIbus Connector tab

Click the **OMNIbus Connector** tab to view or change the information for an IBM Tivoli Netcool/OMNIbus connector. To have the events from a Tivoli Netcool/OMNIbus ObjectServer displayed in the common event console, you must configure an IBM Tivoli Netcool/OMNIbus connector.

To configure a connector, click **New**. The resulting OMNIbus Connector page contains the following information that defines an IBM Tivoli Netcool/OMNIbus connector:

Connector name

The name that is to be displayed in the common event console for this connector.

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.

Computer name of event system

The computer name of the event system that is associated with this connector.

Port number of event system

The ObjectServer port number (usually 4100), which this connector uses to retrieve events from the Tivoli Netcool/OMNIbus event system.

User name for accessing event system

The user name that is used when accessing the event system that is associated with this connector.

Password

The password that is associated with the user name.

SQL WHERE clause that restricts events for common event console

This clause can be applied only to the part of an event that is built from the Tivoli Netcool/OMNIbus `alerts.status` table. For example, `Severity <> 0` causes all events with a severity that is not equal to 0 to be available in the common event console.

If you do not define a clause, all Tivoli Netcool/OMNIbus events are available in the common event console.

View cleared events
You can choose `Yes` or `No`. A value of `Yes` means that cleared events for this connector are available in the common event console.

Time interval (in minutes) for polling event system
The number of minutes between each poll of the event system for new or changed events.

The Tivoli Netcool/OMNIbus ObjectServer automatically sends new or changed events to the common event console as they become available. Therefore, the primary purpose of this checking is to ensure that the server and the connection to the server are functioning properly.

Time interval (in seconds) between reconnection attempts
The number of seconds of delay between reconnection attempts when the connector loses its connection to the event system.

Number of reconnection attempts
The maximum number of consecutive reconnection attempts to make if the connector loses its connection to the event system.

If this value is set to `0` and the connector loses its connection, the connector remains inoperable indefinitely.

If this value is set to `-1` and the connector loses its connection, the connector attempts to reconnect indefinitely.

Information for extra table columns
The common event console includes five extra table columns that you can customize. In the remaining fields on this page, you can define the Tivoli Netcool/OMNIbus field type and field name that identify the field that is to be mapped to each of these customizable columns.

For the field type, you can choose one of the following values:

- `alerts.status`, which means that the field contains data from the `alerts.status` table in the Tivoli Netcool/OMNIbus ObjectServer.
- `alerts.details`, which means that the field contains data from the `alerts.details` table in the Tivoli Netcool/OMNIbus ObjectServer.
- `Extended`, which means that the field contains extended attributes from a Tivoli Enterprise Console event that has been forwarded to the Tivoli Netcool/OMNIbus event system.

Names of Extra Columns tab

The common event console includes five extra table columns that you can customize. By default, the following names are used for these columns:

- Extra Column 1
- Extra Column 2
- Extra Column 3
- Extra Column 4
- Extra Column 5

Click the **Names of Extra Columns** tab to view or change the names of these columns.

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.

Purpose of extra table columns

The common event console displays only a basic set of information from the Tivoli Enterprise Console base attribute table and the Tivoli Netcool/OMNIbus `alerts.status` and `alerts.details` tables.

If, for example, you want to see an additional attribute named "origin" from a Tivoli Enterprise Console event, you can perform the following steps:

1. In the **Attribute type for extra column 1** field on the TEC Connector page, choose the attribute type, for example, `base`.
2. In the **Attribute name for extra column 1** field on the TEC Connector page, enter the attribute name, for example, `origin`.
3. In the **Name of extra column 1** field on the Names of Extra Columns page, enter the name that you want to use for the column that you have customized. For example, you might enter `Origin`.

In the "Origin" column for each row that is a Tivoli Enterprise Console event, the common event console displays the value of the origin attribute.

TEC Connector tab: defining information for extra table columns

In the following fields on the TEC Connector page, you define the information that is to be mapped to the customizable columns:

- Attribute type for extra column 1
- Attribute name for extra column 1
- Attribute type for extra column 2
- Attribute name for extra column 2
- Attribute type for extra column 3
- Attribute name for extra column 3
- Attribute type for extra column 4
- Attribute name for extra column 4
- Attribute type for extra column 5
- Attribute name for extra column 5

OMNIbus Connector tab: defining information for extra table columns

In the following fields on the OMNIbus Connector page, you define the information that is to be mapped to the customizable columns:

- Field type for extra column 1
- Field name for extra column 1
- Field type for extra column 2
- Field name for extra column 2
- Field type for extra column 3
- Field name for extra column 3
- Field type for extra column 4
- Field name for extra column 4
- Field type for extra column 5
- Field name for extra column 5

Best practices for using event synchronization

In your environment, if Tivoli Monitoring events are forwarded to the Tivoli Enterprise Console or Tivoli Netcool/OMNIbus event system for the purpose of event synchronization, configure the common event connectors to retrieve only one copy of the same event to avoid having duplicate event information in the common event console.

Follow these best practices to restrict the common event console to include only the Tivoli Enterprise Console or Tivoli Netcool/OMNIbus events that do not originate as Tivoli Monitoring events:

When Tivoli Monitoring events are forwarded to Tivoli Enterprise Console event system

1. On the Tivoli Enterprise Console server, create an event group that defines only the Tivoli Enterprise Console events that do not originate as Tivoli Monitoring events and is named, for example, `All_but_ITM`.
2. When you configure a TEC Connector, type `All_but_ITM` in the **Event group that defines events for common event console** field.
3. When you configure the ITM Connector, click **Yes** in the **Enable this connector** field.

When Tivoli Monitoring events are forwarded to Tivoli Netcool/OMNIbus event system

1. When you configure an OMNIbus Connector, type `ITMStatus = ''` in the **SQL WHERE clause that restricts events for common event console** field, where `''` is two single quotation marks with no space between them. This clause restricts the Tivoli Netcool/OMNIbus events in the common event console to only those that do not originate as Tivoli Monitoring events.
2. When you configure the ITM Connector, click **Yes** in the **Enable this connector** field.

The resulting configuration causes the common event console to retrieve Tivoli Monitoring events directly from the Tivoli Monitoring event system rather than the Tivoli Enterprise Console or Tivoli Netcool/OMNIbus event system, which prevents you from having duplicate event information in the common event console.

Troubleshooting problems with connection to Tivoli Enterprise Console server on Linux systems

The following information can be used to troubleshoot problems with connection to Tivoli Enterprise Console server on a Linux system.

Problem

The Tivoli Enterprise Console connector cannot connect to the Tivoli Enterprise Console server. Therefore, Tivoli Enterprise Console events are not available in the common event console.

Explanation

The /etc/hosts file on the computer where the Tivoli Enterprise Portal server is installed must include the local host with the correct IP address. The following line shows approximately what the default Linux configuration is:

```
127.0.0.1 my_hostname localhost
```

The default Linux configuration causes the connection request to be sent to the Tivoli Enterprise Console server with the `127.0.0.1` address, which is not the correct IP address of the computer where the Tivoli Enterprise Portal server is installed. For the Tivoli Enterprise Portal server to connect, it must be able to do a reverse lookup.

Solution

Ensure that the /etc/hosts file includes the local host with the correct IP address. The following two lines show approximately what the correct Linux configuration is, where *xxx.xxx.xxx.xxx* is the IP address of the computer where the Tivoli Enterprise Portal server is installed:

```
127.0.0.1 localhost
xxx.xxx.xxx.xxx my_hostname
```

Chapter 13. Maintaining monitoring agents

Maintenance of Tivoli Enterprise Monitoring Agents involves such tasks as upgrading to the latest release, editing environment variables to change their behavior, and controlling their display in the Tivoli Enterprise Portal Navigator Physical view.

The methods available for agent maintenance depend on the size and configuration of your managed network, the type of task, and your preferences.

Agent tasks in the Tivoli Enterprise Portal

The Navigator Physical view in the Tivoli Enterprise Portal displays the managed systems in your monitored network. From the Navigator menu, you can remotely deploy and manage Tivoli Enterprise Monitoring Agents that run on distributed operating systems and that connect to a Tivoli Enterprise Monitoring Server that runs on a distributed operating system.

Before you can remotely install and configure agents, each target computer must have an operating system (OS) agent installed. Monitoring agents that do not support the remote agent deployment feature do not show the **Add Managed System**, **Configure**, and **Remove** options in the Navigator pop-up menu. The types of managed systems that you can add to a computer depend on what agent bundles are in the *agent depot* on the monitoring server that the OS agent is connected to.

Adding an agent through the Tivoli Enterprise Portal

Use the Tivoli Enterprise Portal client to add individual managed systems to the monitored network.

Before you begin

The types of agents that you can remotely install on a computer depend on what agent bundles are in the agent depot on the monitoring server to which the OS agent is connected. The “Deploying monitoring agents across your environment” topics in the *IBM Tivoli Monitoring Installation and Setup Guide* describe how establish an agent depot on the monitoring server and an OS agent on each computer where agents will be deployed.

After the OS agents have been installed, the Navigator Physical view adds an item for each online managed system.

To use this feature, your user ID must have **Manage** permission for **Agent Management**.

About this task

Follow these instructions to install and configure managed systems through the Tivoli Enterprise Portal:

Procedure

1. In the Navigator physical view, right-click the system-level Navigator item for the computer where you want to install the monitoring agent. In this example, the computers named ORANGE, PEAR, CABBAGE, and ONION are available.

   - Enterprise
     - Linux Systems
       - ORANGE
       - PEAR
     - Windows Systems
       - CABBAGE
       - ONION
2. Click **Add Managed System** to open the Select a Monitoring Agent window. The agents shown in this list are those available for the operating system on which this computer runs. The two-digit version number is followed by a two-digit release number and a modification number of up to five digits.
3. Highlight the type of the monitoring agent to install and click **OK**. For some agent types, the new managed system operation is queued and the transaction ID is displayed. For other agent types, a wizard is provided for you to configure the agent on this system.
4. Complete the fields to configure the agent, clicking **Next** and **Back** to move among the pages.
5. On the **Agent** page, establish the operating system user ID under which the agent will run on the managed system. *Windows*: Either accept the default to start the managed system with your user ID (you can also select the check box to **Allow service to interact with desktop** to enable remote control) or select **Use this account** and fill in the user name and password under which the agent will run.

   Non-Windows: Enter the **Username** under which the agent will run and the **Group name**.
6. Click **Finish** to complete the managed system configuration. If any of the information provided is invalid, you will receive an error message and be returned to the configuration window. Check your entries and edit as appropriate to configure correctly. Installation and setup begins and might take several minutes to complete depending on your Tivoli monitoring configuration, the location of the managed system, and the type of monitoring agent.
7. After the managed system has been added to the enterprise, click **Apply Pending Updates** in the Navigator view toolbar. The new managed system (such as Universal Database) is displayed below the system Navigator item.

Configuring an agent through the Tivoli Enterprise Portal

The Tivoli Enterprise Portal client offers a convenient feature for configuring individual managed systems. This method of configuring agents does not apply to the OS agents because they are already configured and running.

Before you begin

To use this feature, your user ID must have Manage permission for Agent Management.

About this task

To configure your monitoring agents, complete the following steps.

Procedure

1. Right-click the Navigator item for the agent to configure or upgrade.
2. Click **Configure** to open the Configure Managed System window.
3. Edit the fields to configure the agent, clicking **Next** and **Back** to move among the pages. Any pages besides **Agent** are specific to the agent type.
   - Performance Analyzer, Summarization and Pruning Agent, and Warehouse Proxy: See “Tivoli Data Warehouse solutions” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
   - Non-base agents: See your product's installation guide in the IBM Tivoli Monitoring Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/welcome.htm) or on IBM Tivoli Documentation Central (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central).
4. On the **Agent** page, establish the user ID that will be used to maintain the agent:

   Windows:
   Accept the default ● **Use local system account** to use your Tivoli Enterprise Portal user ID. You can also select □ **Allow service to interact with desktop** to enable remote control. Or select ○ **Use this account** and fill in the user name and password under which the agent will be controlled.

   Non-Windows:
   Enter the **Username** under which the agent will run and the **Group name**.
5. Click **Finish** to complete the managed system configuration. If any of the information provided is invalid, you will receive an error message and be returned to the configuration window. Check your entries and edit as appropriate to configure correctly.

Starting, stopping, and recycling an agent through the Tivoli Enterprise Portal

You can start an offline managed system, or recycle or stop it through the Tivoli Enterprise Portal.

Before you begin

To use this feature, your user ID must have **Manage** permission for **Agent Management**.

About this task

All deployment commands are passed through the operating system agent that is installed at the target computer. If an operating system agent is not installed, you cannot start or stop the deployed agent.

Procedure

- To start a monitoring agent from the Tivoli Enterprise Portal:

1. In the Navigator Physical view, right-click the Navigator item of the offline agent.
2. Click **Start**. The request to start the monitoring agent is sent to the monitoring server to which it is connected. Depending on your monitoring configuration, it might take a few moments before the agent starts running and to see the Navigator item enabled. If the monitoring agent does not start and you get an error message, the computer might be unavailable.

- To stop a monitoring agent from the Tivoli Enterprise Portal:
  1. In the Navigator physical view, right-click the agent to stop.
  2. Click **Stop**. The agent goes offline and the Navigator item is dimmed. The agent does not come online until you start it manually or, if it is set to start automatically, after you restart the monitoring server to which it is connected.
- To recycle a monitoring agent from the Tivoli Enterprise Portal:
  1. In the Navigator physical view, right-click the agent to stop.
  2. Click **Restart** to stop, then start the monitoring agent. This might take a short time depending on the network traffic.

Updating an agent through the Tivoli Enterprise Portal

When a new version of a distributed monitoring agent is released, you can apply the new version locally or remotely to one managed system at a time, or to many simultaneously. Use the Configure Managed System window in the Tivoli Enterprise Portal client to apply the update.

Before you begin

This capability does not apply to the OS monitoring agents, z/OS-based agents, or any products that do not support the remote agent deployment feature. The agents to be updated must also have been originally installed using remote agent deployment. The types of managed systems that you can add to a computer depend on what agent bundles are in the agent depot on the monitoring server to which the OS agent is connected. See the "Deploying monitoring agents across your environment" topics in the *IBM Tivoli Monitoring Installation and Setup Guide* for more information.

Before starting the update, you must install application support on the Tivoli Enterprise Portal Server for any agent that you are going to deploy with the procedure that follows.

Note: Agent application support updates are automatic and this procedure is not necessary if your monitoring agent is running on an IBM Tivoli Monitoring Version 6.2.3 or later infrastructure, unless the self-describing capability is disabled.

About this task

Complete these steps to apply a patch for a monitoring agent through the portal client:

Procedure

1. Right-click the Navigator item for the agent that you want to upgrade.
2. Click **Configure** to open the Configure Managed System window.
3. Click the **Agent** tab.

4. Compare the installed version of the monitoring agent with any available product updates, then highlight the row of the agent to update and click **Install Updates**.

Results

Installation of the updates begins and might take several minutes to complete. The list that displays reflects the contents of the deployment depot. If **Install Updates** is disabled, one or more of the following conditions exist:

- The depot entry does not match the product type.
- The **VVRR** fields for the agent and the depot entry are *not* the same, where `VV` is the version number and `RR` is the revision number. For example, an entry of **`0610`** prevents you from applying a fix pack intended for a version 6.2 agent.
- The depot entry is at an older version than the agent.
- The host version field of the depot entry does not contain the host platform for the agent.
- The prereq field of the depot entry does not contain an agent of the same type as the agent itself. For example, if 6.1 UD (DB2 monitoring) is the selected agent, the prereq field in the depot entry must contain a deployment bundle notation such as ud:061000000, which is one way to denote a patch deployment bundle.

Removing an agent through the Tivoli Enterprise Portal

You can also uninstall 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.

Before you begin

To use this feature, your user ID must have **Manage** permission for **Agent Management**.

About this task

Complete the following steps to remove and optionally uninstall an agent:

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.

Procedure

1. Right-click the Navigator item for the agent you want to remove.
2. Click **Remove**.
3. Click **Yes** when you are asked to confirm the removal of the agent. If you are removing an agent that has subagents, another message will ask if you want them all removed.
4. 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 your system.

Updating an agent through the command-line interface

Updating agents involves stopping any that are running, applying the changes, and restarting them. After determining the specifics about monitoring agents that you want to update, including the type and version, run the **tacmd updateAgent** command from the command-line interface. If a version is not specified, the agent is updated to the latest version.

About this task

Complete the following steps at a command-line interface. For reference information about this command and related commands, see the *IBM Tivoli Monitoring Command Reference*.

Note: Use only tacmd commands that are included with Tivoli products 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 might void your warranty.

Procedure

1. Use the **tacmd login** command to log into a Tivoli Enterprise Monitoring Server.

   ```
    tacmd login {-s|-–server} {[{https|http}://]HOST[:PORT]}
   [{-u|--username} USERNAME]
   [{-p|--password}  PASSWORD]
   [{-t|--timeout} TIMEOUT] [-t TIMEOUT]
   ```

   a. For example, to log in to the system *ms.austin.ibm.com* with the user name *Admin* and the password *log1n*, run the following command:

      ```
      tacmd login -s ms.austin.ibm.com -u Admin -p log1n
      ```

2. After logging in, use the **tacmd updateAgent** command to install an agent update to a specified node.

   ```
   tacmd updateAgent {-t|--type} TYPE {-n|--node} MANAGED-OS
    [{-v|--version} VERSION] [{-f|--force}]
   ```

   a. For example, the following command updates a UNIX agent (type *UX*) on *itmserver*:

      ```
      tacmd updateagent -t UX -n itmserver:KUX -v 6111
      ```

Clearing the Deployment Status table

Each time you issue an IBM Tivoli Monitoring **tacmd** command or use the Tivoli Enterprise Portal navigator to remotely manage a Tivoli Enterprise Monitoring Agent, information about the transaction is preserved in the Tivoli Enterprise Monitoring Server Deployment Status table. To make it easier to manage the contents of this table, especially in large environments, you can schedule the periodic removal of completed transactions from the table.

About this task

Enable this feature to review completed deployment transactions at opportune times and reduce the amount of monitoring server overhead by maintaining a modest table size. To schedule periodic clearing of completed transactions from the Deployment Status table, you need to specify how often you want the clearing to occur.

This feature is controlled by the monitoring server `CLEARDEPLOYSTATUSFREQ=`*`X`* environment variable, where *`X`* is the number of hours between the automated clearing of the table. If *`X`* is zero (0) or if the environment variable is not specified, automatic clearing is disabled. Valid values include 0 - 720.

You can enable the feature in two ways:

- Add the environment variable to the monitoring server configuration file directly, so that the server enables automated clearing at startup.
- Set the environment variable on an already running monitoring server by using the IBM Tivoli Monitoring Service Console.

When automated clearing is enabled, the monitoring server automatically finds completed deployment transactions, removes them from the Deployment Status table, and then records the information about the deleted transaction in a log file for examination by the user at a later time. The automated clearing runs at the hourly interval that you specify when setting the environment variable.

Procedure

- Modify the monitoring server environment file
  1. Open the environment file on the computer where the monitoring server is installed:
     – Windows Use Manage Tivoli Enterprise Monitoring Services (**Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**). Right-click Tivoli Enterprise Monitoring Server and click **Advanced** → **Edit ENV File**.
     – Linux UNIX Change to the *`install_dir`*`/config` directory and open the `<`*`hostname`*`>_ms_<`*`temsname`*`>.config` and `ms.ini` files in a text editor.
  2. Add the environment variable, specifying your hourly interval. For example: `CLEARDEPLOYSTATUSFREQ=1`.
  3. Save the file.
  4. Recycle the monitoring server to implement the changes.
- Modify the monitoring server environment file by using the IBM Tivoli Monitoring Service Console

  See "Using the IBM Tivoli Monitoring Service Console" in the *IBM Tivoli Monitoring Troubleshooting Guide* for detailed information.
  1. Open a web browser to `http://`*`hostname`*`:1920`, where *`hostname`* is the host name or IP address of the system where the monitoring server is running. The utility then displays with information about the components that are currently running on this system.
  2. Select the **ms** link to modify the environment variable.
  3. Enter your user ID and passwords.
  4. Enter the **`BSS1 SET CLEARDEPLOYSTATUSFREQ=1`** command, where 1 is your hourly interval.

Results

The log file `cleardeploystatus.log` in the monitoring server logs subdirectory (*`install_dir`*`/logs` for Linux and UNIX or *`install_dir`*`\logs` for Windows) contains a text line for each Deployment Status table transaction that is cleared. Each time the monitoring server is started, it logs `---Clear Deploy Status Log---` in the log file.

Each transaction that is cleared from the table has the following information written to the log file:

- Transaction ID: Global transaction identifier of the transaction that completed.
- Submitted: Timestamp that the transaction was initially submitted for processing.
- Command: The deployment command processed.
- Status: The completion status (SUCCESS or FAILURE).
- Retries: The number of times the transaction was tried before it completed.
- Monitoring server name: The name of the monitoring server responsible for processing the transaction.
- Target host name: The managed system name or managed node identifier where the command was completed.
- Platform: The reported platform architecture of the OS agent executing on the target.
- Product: The product code of the agent for which the transaction was processed.
- Version: The version of the product for which the transaction was attempted.
- Completion message: If the status returned is a failure, an explanation of the reason for the failure.

What to do next

You can change the location of the log file where the monitoring server records the transactions which are cleared from the Deployment Status table, by adding the environment variable `CLEARLOG` to the monitoring server configuration file specifying a fully qualified path name on the local system, or any fully qualified path name on a mounted file system.

Using a mounted file system is useful when there is a monitoring server and a server backup. By using a mounted file system as the destination, the logs for both systems can be set to the same fully qualified path name to accommodate failover conditions.

Note: The acting hub monitoring server performs the automated cleansing of the Deployment Status table for the entire enterprise. If you have a backup monitoring server for the hub, then also set the environment variable on the backup to the same value as specified on the hub so that the clearing process occurs at the same time in the event of a hub failover.

Changing the monitoring server an agent connects to

A monitored environment with multiple Tivoli Enterprise Monitoring Servers can have all or some of the agents connect to remote monitoring servers. You can change the monitoring server an agent connects to by reconfiguring it.

About this task

Use one of these options to reassign a monitoring agent to a different monitoring server:

Procedure

- Use the Manage Tivoli Enterprise Monitoring Services application on the computer system where the agent is installed. Right-click the monitoring agent, and click **Reconfigure**. Click OK in the first Agent Advanced Configuration window, then enter the **Hostname** or **IP Address** of the monitoring server you want to connect to. If the port you are using is different from the default 1918, enter the **Port** number.

If the agent is installed on Linux or UNIX you can also use the `itmcmd agent -A <product_code>` command to reconfigure.

- Use the `tacmd setAgentConnection` command to remotely reconfigure the agent. See the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) for detailed information on this command.
- On IBM Tivoli Monitoring V6.3 or later monitoring servers, use the IBM Tivoli Monitoring login daemon solution that is available on IBM Service Management Connect. The login daemon is a tool which handles monitoring agents coming online, then calls the following customer-provided scripts:

  `Select TEMS`
  This script returns an indication of what primary and secondary monitoring servers an agent should be reconfigured to connect to, if the agent connects to the wrong monitoring server.

  `After Login`
  This script can be used to perform setup or configuration of an agent after it has connected to its designated monitoring server. For example, this script can be used to add an agent to the managed system groups or scan the system where the agent is installed and remotely deploy additional agents as needed.

  For more information on the login daemon solution and whether or not it can be used in your environment, see TEMS login policies for agents or go directly to the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring) and search for "login policies". **Note**: This solution is only available through IBM Service Management Connect.

Self-describing monitoring agents

IBM Tivoli Monitoring V6.2.3 or later monitoring agents that are enabled for self-description have all the required application support files to update the Tivoli Management Services servers. You do not have to perform manual support installation steps and recycle each individual server component that supports the agent.

Self-describing monitoring agents apply version updates to other components automatically as the agent connects, without the need to recycle the Tivoli Management Services servers: hub monitoring server, remote monitoring server, and portal server. This automatic product support installation feature helps to eliminate errors that might occur from the inconsistent installation of application data on the IBM Tivoli Monitoring server. To use the self-describing capability your Tivoli Management Services must be at V6.2.3 or later.

If a monitoring agent supports the self-describing capability, application support is installed on the agent system. If the monitoring agent includes the IBM Tivoli Monitoring V6.2.3 or later agent framework or is installed on a system where the V6.2.3 or later agent framework is already installed, the monitoring servers and portal servers retrieve the application support files from the agent after agent startup and automatically apply the support. The infrastructure servers retrieve only the application support from a self-describing agent if they have not already applied the application support or if they have an earlier version of the application support.

The self-describing update occurs only once for each specific agent version. The update files are stored at the agent so that when the agent connects to a monitoring server, the monitoring server is automatically informed of the available application support update. If the self-describing function is enabled, the application support is retrieved from the agent, and the monitoring server support is updated.

Note: If you are using the Tivoli Enterprise Portal desktop client, you must use the monitoring agent's installation images to install the Tivoli Enterprise Portal application support on each of the desktop clients.

If you installed your monitoring server on Linux or UNIX, the application support files for all base monitoring agents and other supported agents were automatically installed on the monitoring server. This process is different from installing the monitoring server on Windows, in that the Linux or UNIX installation always automatically installs the application support files for monitoring agents. Check the monitoring server and portal server to ensure that self-describing agent products are installed as expected.

The Tivoli Enterprise Monitoring Automation Server component, provided in IBM Tivoli Monitoring V6.3, contains the Open Services Lifecycle Collaboration Performance Monitoring (OSLC-PM) service provider. The Performance Monitoring server provider does not support the dynamic refresh of application support files. This means that the Tivoli Enterprise Monitoring Automation Server must be recycled when new application support is added to a system, regardless of whether the support was added through a self-describing agent, a regular installation, or a manual copy. However, if the monitoring agent does not provide OSLC support, then the automation server does not need to be recycled. Typically, agents before IBM Tivoli Monitoring V6.3 do not provide OSLC support. See the agent documentation to determine whether an agent provides OSLC support.

Roadmap

Use the following roadmap to help you configure, enable, and use the self-describing feature. Because this roadmap is meant to provide a comprehensive overview, when applicable, links are provided to relevant sections in other Tivoli Monitoring guides.

Step	Description and information provided
1	Two types of installation scenarios are available: a pristine installation and an upgrade installation. See “Configuring self-describing agent seeding” in the *IBM Tivoli Monitoring Installation and Setup Guide*. **Additional information:** Seeding information is also provided in “Self-describing auto refresh and seeding” on page 308. For information about the **editSdaOptions** command, see the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

Step	Description and information provided
2	Configure self-describing at the hub monitoring server. See “Enabling self-describing agent capability at the hub monitoring server” in the *IBM Tivoli Monitoring Installation and Setup Guide*, for more information about this initial setup. This step includes setting each hub monitoring server environment variable `KMS_SDA=Y`. **Additional information:** For future reference and administrator tasks after initial setup, see “Enabling or disabling the self-describing capability at the monitoring server” on page 310.
3	**Note**: If you are not running IBM Tivoli Monitoring V6.3, skip this step. In IBM Tivoli Monitoring V6.3, by default, the hub monitoring server blocks all self-describing agent installations (even if you turned on self-describing at the hub monitoring server with setting `KMS_SDA=Y` from step 2) until you issue one of the following commands: • **`tacmd addSdaInstallOptions`** to specify the products and versions that the self-describing agent facility is allowed to install. OR • **`tacmd editSdaInstallOptions -t DEFAULT -i ON`** to allow installations for all products and versions without any blocking. (This setting is essentially the default self-describing agent behavior in V6.2.3 and V6.2.3 FP1.) This feature provides more control over what products and versions are installed on your monitoring server and portal server by the automatic self-describing agent process. After modifying your install options, you can use the **`tacmd listSdaInstallOptions`** command to display the current installation configurations for the hub monitoring server. For detailed information and **`tacmd`** commands, see “Managing your self-describing agent installations” and “Dynamically controlling the hub monitoring server self-describing agent capability” in the *IBM Tivoli Monitoring Installation and Setup Guide*. **Additional information:** For full syntax information, see the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm). For future reference you can update or modify these installation options after initial setup, see “Dynamically updating the self-describing installation options” on page 306.
4	Use the **`tacmd listappinstallrecs`** command to monitor the application support installation records. Use the **`tacmd listSdaStatus`** command to monitor the self-describing enablement and suspend status for all monitoring servers in your environment. See “Self-describing agent installation” on page 302. **Additional information:** For information about installation records that can be tried again and terminal installation records, see “Self-describing agent installation errors” on page 304. For full syntax information, see the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

Step	Description and information provided
5	After a self-describing application update is completed, you can see the following new agent data in the portal client: • Historical Configuration is updated with any new attributes • Workspaces are updated • New or updated situations, policies, and take actions (new situations are distributed and auto-started by using user-configurable seeding options) • Queries are updated • Help server files are updated Application support is not automatically applied to the Tivoli Enterprise Portal clients. An indicator is displayed on these clients to tell the user that they need to recycle their client to apply new or modified application support. • If you are using the Tivoli Enterprise Portal desktop client, you must use the monitoring agent's installation images to install the Tivoli Enterprise Portal application support on each of the desktop clients. • If you are using the Tivoli Enterprise Portal browser client or Java WebStart client and you have the required user permissions to view the updated application, after the indicator is displayed, you can choose to close and restart these clients to see the new updates. **Additional information:** For more information about support indicators, see Responding to an application support event in the *Tivoli Enterprise Portal User's Guide*.
6	To begin preliminary diagnostics tests and troubleshoot any problems, see "Monitoring agent troubleshooting" in the *IBM Tivoli Monitoring Troubleshooting Guide*. **Additional information:** For detailed message information, see the *IBM Tivoli Monitoring Messages* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_messages.htm).

Self-describing event flow at the monitoring server

The automated event flow for a self-describing agent differs if the agent is connected to a hub monitoring server or a remote monitoring server.

Self-describing agent connected to the hub monitoring server

The following steps outline the event flow of a self-describing agent connecting to the hub monitoring server:

1. The self-describing agent manager of the hub monitoring server determines whether the version of the application support for the product is already installed on the hub monitoring server. If the application support version for the product is not already installed, the hub monitoring server retrieves the support files from the agent.
2. The hub monitoring server begins the self-describing agent product installation and dynamic refresh of the monitoring server internal product definition structures. The completion status of the installation of this hub monitoring server self-describing agent is recorded in the following locations:
   - The local Tivoli Enterprise Monitoring Server application properties table.
   - Tivoli Enterprise Monitoring Server audit log facility.
   - Tivoli Enterprise Monitoring Server MSG2 log facility.
   - Tivoli Enterprise Monitoring Server RAS1 log.

- On distributed Tivoli Enterprise Monitoring Server platforms, the self-describing agent installation program log files are `installsdsupport_*.trc` and `installsdsupport_*.log`. On Windows computers, the logs are in the *install_dir*`\logs` directory. On Linux and UNIX computers, the logs are in the *install_dir*`/logs` directory.

3. After the hub monitoring server product installation completes successfully, any Tivoli Enterprise Portal Server process that is running and connected to the hub monitoring server is notified of the new or updated product support.

   In an environment enabled for Hot Standby (FTO), self-describing configuration data is replicated to the standby hub monitoring server. The standby hub monitoring server does not support direct connections from agents for a remote monitoring server, which means self-describing installations cannot initiate directly to the standby hub. Self-describing installations are initiated at the standby hub by the acting hub, after the acting hub completes the self-describing installation. The self-describing installation process at the standby hub retrieves the product support files from the acting hub. For more information about an FTO environment, see the *IBM Tivoli Monitoring High Availability Guide for Distributed Systems*.
4. The Tivoli Enterprise Portal Server performs the same basic steps as performed by the hub monitoring server. It determines if the version of the product is already installed on the Tivoli Enterprise Portal Server. However, it retrieves the product support files directly from the hub monitoring server and not from the connected agent.
5. The Tivoli Enterprise Portal Server begins the self-describing agent product installation and dynamic refresh of the portal server internal product definition structures.
6. The Tivoli Enterprise Portal Server notifies any running Tivoli Enterprise Portal Browser Client and Tivoli Enterprise Portal Desktop Client that new or updated product support is available. The completion status of this Tivoli Enterprise Portal Server self-describing agent product installation is recorded in the following locations:
   - Tivoli Enterprise Portal Server audit log facility.
   - Tivoli Enterprise Portal Server RAS1 log.
   - The Tivoli Enterprise Portal Server self-describing agent installation program log files `installsdsupport_*.trc` and `installsdsupport_*.log`. On Windows computers, the logs are in the *install_dir*`\logs` directory. On Linux/UNIX computers, the logs are in the *install_dir*`/logs` directory.

Self-describing agent connected to the remote monitoring server

The following steps outline the event flow of a self-describing agent connecting to the remote monitoring server:

1. The self-describing agent manager of the remote monitoring server ensures that the product was first installed on the hub monitoring server.
2. If the product was not first installed on the hub monitoring server, the remote monitoring server tells the hub monitoring server to install the product first.
3. After the hub monitoring server installation is complete, the remote monitoring server determines if the product is installed locally on this monitoring server.
4. The remote monitoring server product installation occurs in the same way as described for the hub monitoring server.
5. If the hub monitoring server product installation fails for any reason, the remote monitoring server does not install the product.

The monitoring server does not allow all other failed self-describing agent installation requests to be tried again until the existing monitoring server error condition is corrected and the failed self-describing agent product installation records are removed from the monitoring server application properties table. For more information, see "Self-describing agent installation errors" on page 304.

Note:

The availability of the self-describing agent feature on a hub monitoring server can influence the availability of the feature on a remote monitoring server. An error message is displayed when a self-describing agent error on a hub monitoring server causes the self-describing agent feature to be disabled on the remote monitoring server connected to that hub. For more information about the error messages, see the *IBM Tivoli Monitoring Troubleshooting Guide*.

After the self-describing agent error is fixed on the hub monitoring server, the remote monitoring server detects, at the next reconnection to the hub, that the self-describing agent feature is available on the hub. As a result, it re-enables the self-describing agent locally on the remote monitoring server.

Changes detected by the monitoring server at startup

The Tivoli Enterprise Monitoring Server detects the following information during startup:

- Products installed manually, for example applications that were not installed by using the self-describing capability.
- Manual updates (catalog and version file changes) to installed products, for example user-initiated product changes that happen outside of a self-describing agent installation.
- Failed self-describing agent installations.

The Tivoli Enterprise Monitoring Server makes adjustments and corrections based upon the changes detected in installed products. Only products that have a valid monitoring server version file (VER file) are detected during startup. This process happens automatically when the self-describing agent installation manager is enabled on the monitoring server (**KMS_SDA=Y**). This function helps to maintain an accurate inventory of installed products and versions. If the self-describing agent installation manager is not enabled, this function does not run.

Self-describing agent installation

The local monitoring server application properties table (TAPPLPROPS) stores the results of the monitoring server self-describing agent product installations. This table includes the current detected version of manually installed monitoring server products.

This table also stores self-describing information such as installation options, seed distribution, and suspend records. For more information about the TAPPLPROPS attributes, see the Application Property Installation attributes in the *Tivoli Enterprise Portal User's Guide* for more information.

You can use two different **tacmd** commands to check the self-describing status of your monitoring servers:

- **tacmd listSdaStatus** for operational status
- **tacmd listappinstallrecs** for product installation status

Using the `tacmd listSdaStatus` command

In IBM Tivoli Monitoring V6.3 or later, the **`tacmd listSdaStatus`** command displays the self-describing enablement status for a monitoring server. You can display the self-describing enablement status for a list of monitoring servers or for all monitoring servers, which is the default action. The operational state of the self-describing agent facility (suspended or active) for the hub monitoring server is also provided if the command is issued to an IBM Tivoli Monitoring V6.3 or later hub monitoring server. The self-describing operational state (suspended or active) of the hub determines the self-describing activity for all the monitoring servers that are attached to that hub monitoring server.

The self-describing enablement status is indicated by the values of `HUB/RTEMS name`, `STATE`, and `STATUS`.

In this example, a message states that the self-describing function is currently suspended. HUB_A and RTEMS_1 are both enabled for self-description. STATUS code 0 indicates self-describing is enabled. RTEMS_2 is not enabled for self-description and STATUS code 16 (`SDA disabled because KMS_SDA=N`) is provided as an explanation.

```
tacmd listSdaStatus

KUILSS203I: SDA functions are suspended.

HUB/RTEMS    STATE   STATUS

HUB_A        ON      0

RTEMS_1      ON      0

RTEMS_2      OFF     16
```

For more information, examples, and descriptions of the **`tacmd listSdaStatus`** command `STATUS` codes, see the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

Using the `tacmd listappinstallrecs` command

Use the **`tacmd listappinstallrecs`** command to monitor the application support installation records.

The command returns the application support installation records and displays the current self-describing agent product installation status for all monitoring servers in the environment. Remember that this command is not available when the monitoring server is not running. If the monitoring server and the portal server are both started, you can also see self-describing agent information and settings in the Audit Logs workspace. See Chapter 9, "Audit logging," on page 247.

The **`tacmd listappinstallrecs`** command displays the current self-describing agent product installation status for all monitoring servers in the environment.

```
HUB/RTEMS   PRODUCT     VERSION     GRPID       ID          IDVER       SEEDSTATE   STATE       STATUS

HUB_LZ      A4          06300000    5655        TMS         06300000                            0

HUB_LZ      HD          06300000    5655        TMS         06300000    Y           IC          0

HUB_LZ      HD          06300000    5655        TPW         06300000                IC          0
HUB_LZ      LZ          06230000    5655        TPS         06230000                IC          0
HUB_LZ      NT          06230000    5655        TMS         06230000                            0
HUB_LZ      TM          06230000    5655        TMS         06230000                            0
HUB_LZ      11          06230000    5655        TMS         06230000                ME          1005
RTEMS_LZ    A4          06230000    5655        TMS         06230000                            0
RTEMS_LZ    LZ          06230000    5655        TMS         06230000    Y           IC          0
RTEMS_LZ    R6          06230000    5655        TMS         06230000                            0
RTEMS_LZ    11          06230000    5655        TMS         06230000    Y           ME          1014
```

The HUB/RTEMS column lists the node name of the monitoring server where the record was collected. Rows with IC in the STATE column stand for self-describing agent Install Complete. You can also see in this example that the hub monitoring server and the remote monitoring server both have an appinstallrecs entry for monitoring server support (TMS), but the hub monitoring server has two additional appinstallrecs entries for portal server support (TPS) and portal browser client support (TPW) packages. These additional packages are installed only on the hub monitoring server so they are available to the portal server.

If the STATE column is blank, application support was manually installed. A MANUALINST column shows Y if you use the -d (detail) option. The self-describing agent appinstallrecs states are as follows:

IR Install Request: New install request on the work queue.

IM Install Metadata: Retrieve the install request from the work queue and begin the metadata install and auto-refresh work.

MC Metadata Complete: Metadata install and auto-refresh has completed.

IC Install Complete: SDA application support has completed successfully.

ME Metadata Error: Metadata Installation Error. STATUS code provides additional information.

Tip: IR, IM, and MC appinstallrecords all indicate that normal self-describing agent installation is in progress. For information about application support installation records with a STATE value of ME, see "Self-describing agent installation errors."

For detailed information about the **tacmd listappinstallrecs** command STATUS codes, see the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

In IBM Tivoli Monitoring V6.3 or later, the **listappinstallrecs -t TEMS** option is no longer supported. Use the **listSdaStatus** command instead.

Self-describing agent installation errors

A failure code is reported in the **STATUS** column of the **listappinstallrecs** output. Some errors can be tried again, others are terminal.

Self-describing agent installation errors that can be tried again

The Tivoli Enterprise Monitoring Agents self-describing agent service decides what types of previously failed self-describing agent registration or installation requests are tried again. Only the following types of failures or error codes returned from the Tivoli Enterprise Monitoring Server, on behalf of a self-describing agent registration request to the monitoring server, are tried again:

1006	Duplicate SDA Installation Request
1009	HUB Not There
1017	Temporary Installation Error
1021	Server Timed Out

Self-describing agent installation errors that can be tried again are registration or installation requests that started, but have not yet modified any of the Tivoli Enterprise Monitoring Server files or internal structures.

Terminal self-describing agent installation errors

For any error records with a **STATE** value of ME, the installation is not tried again.

In this example, the installation 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** value of ME indicates that a self-describing agent metadata installation error 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 deleteappinstallrecs** command to clear the self-describing agent error record after you resolve the problem. For more information, see the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

To determine whether a self-describing agent product installation failed with a terminal error condition, run the **tacmd listappinstallrecs** command by using the **-e** option to display error records only. For any error records with a **STATE** value of ME, the installation is not tried again.

Take the following steps to try the self-describing agent installation again:

1. To avoid the same failure from occurring again, you must first correct the condition that caused the installation to fail. In addition, the 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. This command removes the blocking self-describing agent product installation record. See the *IBM Tivoli Monitoring Command Reference* for more information.
3. When each monitoring server failed product installation record is cleared, the monitoring server self-describing agent facility immediately notifies any running self-describing agent that can provide this level of product support, to try the product installation again. For example, if the previous installation 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 tries the installation again.

4. Run the **tacmd listappinstallrecs -t <pc>** again for product *pc* to determine the current installation state.

If the self-describing agent product installation failed at the primary hub monitoring server, the standby hub does not attempt to install the support. Correct the reason for the failure at the primary hub, then use the **deleteappinstallrecs** command to remove the error entry from the primary hub. Clearing the error ensures that the self-describing agent product installation is tried again.

If the self-describing agent product installation fails only at the standby hub, correct the reason for the failure at the standby hub. While logged on the primary hub, you can then use the **deleteappinstallrecs** command with the **-n <standby_TEMS_name>** option to remove the error entry at the standby monitoring server to allow the self-describing agent installation to try again. Deleting the application installation record from the standby hub does not automatically trigger the self-describing agent product installation to try again the way it does when the record is deleted from the primary monitoring server or a remote monitoring server. There is no problem if the standby hub is temporarily missing the agent product support.

- If the standby hub is recycled, any missing product support is discovered, and the self-describing agent product installation takes place at that time.
- If the standby hub is promoted to the acting hub, self-describing agent product installation takes place when the first agent of that type connects to the promoted hub.
- You can force the self-describing agent installation to try again immediately at the standby hub, by repeating the self-describing agent installation at the primary hub. After clearing the error application installation record at the standby monitoring server, use the **deleteappinstallrecs -a** command to delete the non-error state record from the primary monitoring server. This command repeats the self-describing agent installation at the primary monitoring server, triggering the standby monitoring server installation when it completes successfully.

Dynamically updating the self-describing installation options

You can specify what self-describing enabled products and versions are allowed to be installed on your monitoring server and portal server by the automatic self-describing agent process while your hub monitoring server is running.

About this task

The hub monitoring server must be started. The following configuration commands can be issued regardless of whether your self-describing capability is enabled or disabled (KMS_SDA=Y|N). The configuration settings you specify are stored and then implemented once self-describing is turned on with variable KMS_SDA=Y. For more information, see “Dynamically controlling the hub monitoring server self-describing agent capability” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Procedure

1. Suspend the self-describing capability (without recycling the hub monitoring server) by using the **tacmd suspendSda** command. This command ensures that no self-describing installations take place while you are updating the options.
2. Run one of the following commands:

tacmd addSdaInstallOptions
Specify the products and versions that can be installed by the self-describing agent facility.

tacmd editSdaInstallOptions
Modify an existing product version configuration, or change the default self-describing agent installation behavior.

tacmd deleteSdaInstallOptions
Remove a specific version configuration, or remove all versions for a product.

tacmd listSdaInstallOptions
Display the existing self-describing agent installation configurations from the hub monitoring server.

See the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) for syntax and examples.

3. Resume the self-describing capability (without recycling the hub monitoring server) by using the **tacmd resumeSda** command.

Results

Your installations take place only on the new products of versions you specified once self-description is enabled by setting the variable KMS_SDA=Y.

Suspending the self-describing capability

You can suspend and resume the self-describing capability for your environment by using the **tacmd suspendSda** and **tacmd resumeSda** commands without having to recycle the monitoring server. You can suspend the self-describing capability for various reasons such as, when modifying the self-describing installation options or when going into maintenance mode.

Before you begin

The **tacmd suspendSda** and **tacmd resumeSda** commands dynamically change the self-describing behavior only when the hub monitoring server is configured with KMS_SDA=Y. Issuing the **tacmd suspendSda** and **tacmd resumeSda** commands will not override the KMS_SDA=N setting. If the **tacmd resumeSda** command is issued when KMS_SDA=N, the command is ignored until KMS_SDA=Y.

Issuing either the **tacmd suspendSda** or **tacmd resumeSda** commands updates the TAPPLPROPS table. This setting is saved regardless of the KMS_SDA setting. Best practice is to set the hub monitoring server environment variable to KMS_SDA=Y, then run the **tacmd suspendSda** or **tacmd resumeSda** commands.

See the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) for complete syntax information.

Procedure

- To suspend the self-describing capability, issue the **tacmd suspendSda** command.
- To resume the self-describing capability, issue the **tacmd resumeSda** command.

What to do next

You can run the **tacmd listSdaStatus** to view the suspend state of your monitoring servers. See "Self-describing agent installation" on page 302.

Self-describing auto refresh and seeding

The self-describing capability enables your monitoring infrastructure to be automatically refreshed and seeded with the most current version of application support.

Auto refresh is what allows continuous and uninterrupted updates to your monitoring infrastructure. After a successful auto refresh, *seeding* updates your environment with the most current product definitions.

Auto refresh

Each product provides application support. Before IBM Tivoli Monitoring V6.2.3 or later, the application support was installed at a monitoring server, portal server, and portal client, and a recycle of each component was necessary for the new application support to be activated. For IBM Tivoli Monitoring V6.2.3, auto refresh enables the monitoring server and portal server to provide dynamic application refresh after a self-describing agent installation event.

The non-intrusive auto refresh operation occurs when a new self-describing agent initiates an application installation through the IBM Tivoli Monitoring infrastructure. The processing is triggered the first time a self-describing agent connects to a monitoring server and the application support is not already present at the remote monitoring server or hub monitoring server.

The application support is automatically installed first at the hub monitoring server, followed by the remote monitoring server the agent is currently connected to (which retrieves the support from the agent), and then at the portal server (which retrieves the support from the hub monitoring server). If an agent switches to a different remote monitoring server where the support is missing, the support is dynamically updated at the new hosting remote monitoring server.

Auto-refresh occurs at a monitoring server immediately following the metadata deployment. Metadata deployment transfers and stores attribute files, catalog files, EIF mapping files, out-of-the-box product definition files (seeding files), and version files at the monitoring server. After the files are successfully deployed, the monitoring server internal caches are updated and the new metadata is immediately available for the monitoring server components to use for monitoring. Auto-refresh also occurs at the portal server, updating all necessary files.

If the portal server database is restarted while the portal server is still running, then the portal server must be restarted for the support to finish updating the portal server. In order for auto refresh on the portal server to work correctly, the portal server database must be running.

Auto refresh guarantees that you have continuous access to the product's metadata to support monitoring activities during a self-describing agent refresh, by using existing metadata and when the metadata auto-refresh completes, the new metadata becomes available. Some internal monitoring server components provide notification when you have new metadata, for example, to move `pending wait` situations to `started`.

Seeding

Included with a product's application support are out-of-the-box monitoring definitions that include where the definitions will run. The application of a distribution or "where" is commonly termed *distribution*. The storing of the products monitoring and distribution definitions in the monitoring server is termed *seeding*. Seeding happens automatically as part of the self-describing agent product installation so that default monitoring definitions are enabled. You can alter or disable this behavior through the CLI, but you should only do so if you do not want all of the product provided monitoring definitions. For more information, see “Configuring self-describing agent seeding” in the *IBM Tivoli Monitoring Installation and Setup Guide*, and "tacmd editSdaOptions" in the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

The **`tacmd listappinstallrecs -d`** command returns the application support installation records from the monitoring server application properties table, including the values contained in the SEEDSTATE column. See the *IBM Tivoli Monitoring Command Reference* for more information. The following values in the SEEDSTATE column reflect the seed state of self-describing agent installed products:

`<blank>` The default value. This value indicates that seeding has not yet run or does not apply to this product. The SEEDSTATE state applies only to self-describing agent records that have an ID of TMS.

`I` Product seeding in progress.

`Y` Product seeded.

`N` Product not seeded (SQL file not found).

`E` Seeding Error.

During the monitoring server startup, messages are produced in the `TEMS MSG2` log and audit log facility when changes or self-describing agent installation errors are detected in installed products. The monitoring server application support installation records are updated as needed to reflect the changes indicated by the MSG2 messages or audit log. The `TEMS RAS1` log contains trace messages that show the success or failure of application support changes. Inspect the MSG2 messages and audit log messages in the *IBM Tivoli Monitoring Troubleshooting Guide* and complete the following actions:

- Verify that detected changes in installed products were expected. If the changes were unexpected or incorrect, take the required action to install the correct product versions, through a self-describing agent installation or manual installation.
- If any errors are noted in a self-describing agent installed product, clean up the application support installation records, if required. You can do this by deleting the records for the failed installations by using the **`tacmd deleteappinstallrecs`** command, and reinitiate the self-describing agent installations, if required. See the *IBM Tivoli Monitoring Command Reference* for commands that you can use to perform these tasks.

Note: The self-describing agent feature does not automate the installation of agent language packs. Language pack installation procedures for self-describing agents are the same as the procedures for standard agents. For more information, see “Installing language packs” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Enabling or disabling the self-describing capability at the monitoring server

You can enable or disable the self-described agent capability for a specific monitoring server through an environment configuration variable at the Tivoli Enterprise Monitoring Server.

Disable the self-describing feature any time you do not want automatic update and propagation of self-described agent application metadata to occur for a specific monitoring server. Then, the application support for the agent must be manually installed and activated at each monitoring server and portal server. By default the self-describing capability for remote monitoring servers is turned on, and for hub monitoring servers it is turned off.

About this task

Best practice is to control the self-describing agent capability from the hub monitoring server, since enabling or disabling at the hub monitoring server affects all remote monitoring servers and agents that connect to it.

Enabling or disabling the self-describing capability at a specific remote monitoring server affects all agents that connect to that server only.

Use the following steps to temporarily stop self-describing at a single monitoring server by editing the target monitoring server environment variable.

Procedure

- Windows
  1. On the computer where the monitoring server is installed, in the Manage Tivoli Enterprise Monitoring Services application, right-click **Tivoli Enterprise Monitoring Server** and select **Advanced→ Edit ENV file**.
  2. Edit the existing environment variable: `KMS_SDA=Y | N`
- Linux UNIX
  1. On the computer where the monitoring server is installed, change to the `<`*`install_dir`*`>/config/` directory.
  2. Open the `<`*`tems_hostname`*`>_ms_<`*`tems_name`*`>.config` file.
  3. Edit the existing environment variable: `KMS_SDA=Y | N`
- z/OS See `KMS_SDA` in the *IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS: Common Planning and Configuration Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/zcommonconfig/zcommonconfig.htm).

What to do next

Restart the monitoring server.

You can then review the agent support versions in the following ways:

- Review the agent and monitoring server operations logs to determine if the agent is operating in the standard or the self-described agent mode. The Managed System Status workspace has a link to the operations log for each monitoring agent.
- The **`tacmd listappinstallrecs`** command returns the application support installation records and the **`tacmd listSdaStatus`** command displays the current self-describing agent operational status for all monitoring servers in the

environment. Remember that these commands are not available when the monitoring server is not running. See "Self-describing agent installation" on page 302.

Enabling or disabling the self-describing capability at the agent

You can enable or disable the self-described agent capability through a monitoring agent environment configuration variable.

Disable the self-describing feature any time you do not want automatic update and propagation of self-described agent application metadata for one or more individual agents. Then, the application support for the agent must be manually installed and activated at the monitoring server and portal server. By default the self-describing capability for agents is turned on.

Before you begin

Best practice is to control the self-describing agent capability from the hub monitoring server. For more information, see "Enabling or disabling the self-describing capability at the monitoring server" on page 310.

Procedure

- Windows
  1. On the computer where the monitoring agent is installed, in the Manage Tivoli Enterprise Monitoring Services application, right-click the agent and select **Advanced→ Edit ENV file**.
  2. Edit the existing environment variable: `TEMA_SDA=Y | N`
- Linux UNIX
  1. On the computer where the monitoring agent is installed, change to the `<`*`install_dir`*`>/config/` directory.
  2. Open the coordinating file:
     For single-instance agents: `<`*`pc`*`>.ini`
     For multi-instance agents: `<`*`pc`*`>_<`*`instance`*`>.ini` file
     Where *pc* is the two-character product code.
  3. Edit the existing environment variable: `TEMA_SDA=Y | N`
- IBM i
  1. Open `/qautotmp/kmsparm.kbbenv`.
  2. Edit the existing environment variable: `TEMA_SDA=Y | N`
- z/OS See `TEMA_SDA` in the *IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS: Common Planning and Configuration Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/zcommonconfig/zcommonconfig.htm).

What to do next

Restart the monitoring server.

You can review the agent support versions in the following ways:

- Review the agent and monitoring server operations logs to determine if the agent is operating in the standard or the self-described agent mode. The Managed System Status workspace has a link to the operations log for each monitoring agent.

- The **tacmd listappinstallrecs** command returns the application support installation records and the **tacmd listSdaStatus** command displays the current self-describing agent operational status for all monitoring servers in the environment. Remember that these commands are not available when the monitoring server is not running. See "Self-describing agent installation" on page 302.

Determining if agents are enabled for self-description

You can tell if an agent is enabled for self-description prior to installation or after installation.

Before you begin

The IBM Tivoli Monitoring V6.2.3 or later base operating system agents are enabled for self-description, however not all agents are enabled for self-description.

The Tivoli Performance Analyzer is not enabled for self-description. For a complete list of agents enabled for self-description, see the IBM Tivoli Monitoring agents enabled for self-description (SDA) topic in the IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring).

About this task

Use the following steps to determine which agents are enabled for self-description.

Procedure

- To tell prior to installation:

  Look in the installation image for the K<*PC*>MSMAN.txt self-describing manifest file.

  – Windows The file is located in the WINDOWS directory. For agents that run on Windows Itanium, the file is located in the WIA64 directory.

  – Linux UNIX The file is located in the unix directory.

  – For agents created using Agent Builder, the file is located in the K<*PC*>/support directory.

- To tell after installation:

  – Windows Run the **kincinfo -e** command. Look in the SDA STATUS column to view if the agent is enabled for self-description.

  In this example, you can see the Performance Analyzer Agent has Disabled listed in the SDA STATUS column, meaning the agent is *not* enabled for self-description.

```
kincinfo -e

**************** Thursday, July 14, 2011 1:29:36 PM ****************

User: Administrator                          Group: NA

Host Name: ICVW3A03                          Installer: Ver: 062300000

CandleHome : C:\IBM\d1191a\ITM

Installitm : C:\IBM\d1191a\ITM\InstallITM

*********************************************************************

...Application support propagation

PC     PRODUCT DESC                          PLAT       VER            BUILD          SDA STATUS
```

```
NT     Monitoring Agent for Windows          WINNT      06.23.00.00     11871           Enabled
       OS
R2     Agentless Monitoring for Windows      WINNT      06.23.00.00     201107051647    Enabled
       Operating Sy
R3     Agentless Monitoring for AIX          WINNT      06.23.00.00     201107051650    Enabled
       Operating System
R4     Agentless Monitoring for Linux        WINNT      06.23.00.00     201107051653    Enabled
       Operating Syst
R5     Agentless Monitoring for HP-UX        WINNT      06.23.00.00     201107051655    Enabled
       Operating Syst
R6     Agentless Monitoring for Solaris      WINNT      06.23.00.00     201107051658    Enabled
       Operating Sy
PA     Tivoli Performance Analzyer           WINNT      06.23.00.00     d1184a          Disabled
```

– Linux UNIX Run the ***<install_dir>*/bin/cinfo -e** command. Look in the SDA STATUS column to view if the agent is enabled for self-description.

In this example, you can see the Tivoli Performance Analyzer has Disabled listed in the SDA STATUS column, meaning the agent is *not* enabled for self-description.

```
<inst_dir>/bin/cinfo -e
*********** Thu Jul 14 13:19:41 EDT 2011 ******************
User: root                                Groups: root bin daemon sys adm disk wheel
Host name : icvr5d06                      Installer Lvl:06.23.00.00
CandleHome: /data/tmv623-d1191a-201107110121/ITM
Version Format: VV.RM.FF.II (V: Version; R: Release; M: Modification; F: Fix; I: Interim Fix)
*************************************************************
...Application support propagation
PC     PRODUCT DESC                          PLAT       VER             BUILD           SDA STATUS
80     Monitoring Agent for Self Describing  lx8266     03.00.00.00     201106281135    Enabled
       Agent
hd     Warehouse Proxy                       lx8266     06.23.00.00     d1191a          Enabled
lz     Monitoring Agent for Linux OS         lx8266     06.23.00.00     11871           Enabled

pa     Tivoli Performance Analyzer           lx8266     06.23.00.00     11891           Disabled
r2     Agentless Monitoring for Windows      lx8266     06.23.00.00     201107051647    Enabled
       Operating Systems
r3     Agentless Monitoring for AIX          lx8266     06.23.00.00     201107051650    Enabled
       Operating Systems
r4     Agentless Monitoring for Linux        lx8266     06.23.00.00     201107051653    Enabled
       Operating Systems
r5     Agentless Monitoring for HP-UX        lx8266     06.23.00.00     201107051655    Enabled
       Operating Systems
r6     Agentless Monitoring for Solaris      lx8266     06.23.00.00     201107051658    Enabled
       Operating Systems
sy     Summarization and Pruning Agent       lx8266     06.23.00.00     d1177a          Enabled
ul     Monitoring Agent for UNIX Logs        lx8266     06.23.00.00     11751           Enabled
pa     Tivoli Performance Analyzer           lx8266     06.23.00.00     d1184a          Disabled
```

Environment variables that control the self-describing capability

Environment variables turn the self-describing capability on or off at either the monitoring server, portal server, or the agent.

Purpose

The environment variables control the main functionality of the self-describing capability, and are the only parameters typical users modify. All other self-describing environment variables are modified only at the direction of IBM support.

Attention: This topic is meant only to provide reference to the main self-describing environment variables. For a complete list of self-describing environment variables and complete details, see "Environment variables" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Parameters

Do not specify YES or NO; instead, always specify Y or N.

Monitoring server parameters

`KMS_SDA`=`Y | N`
N disables the self-describing agent capability at the monitoring server, whereas Y enables it. Disabling self-describing at the hub monitoring server disables *all* self-describing capability.

The default value is N for hub monitoring servers.

The default value is Y for remote monitoring servers.

`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. The customer must create and set the correct permissions for any custom or alternate directory specified. The directory is not created by the monitoring server. This parameter must be set for the self-describing capability to be enabled; normally it is set during component installation.

`TEMS_JAVA_BINPATH`
This variable locates the Java installation path within the z/OS USS environment. It can be dynamically superseded by a local configuration file each time the z/OS engine generates its USS shell interface. For more information, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/ztemsconfig/ztemsconfig.htm).

Portal server parameters

`TEPS_SDA`=`Y | N`
N disables the self-describing agent capability at the portal server, whereas Y enables it.

The default value is Y.

TEPS_MANIFEST_PATH=*file_loc*
: Set by default to the location where the portal server writes and stores retrieved product support JAR files. Normally this parameter is set during component installation.

Agent parameters

TEMA_SDA=Y | N
: N disables the self-describing agent capability at the agent, whereas Y enables it. A value of N blocks the 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 monitoring server for other products.

  The default value is Y.

Chapter 14. Agent Management Services

Use the Agent Management Services to monitor the availability of agents and respond automatically (such as with a restart) if the agent becomes unhealthy or exits unexpectedly. By using these services, you can see improved agent availability ratings.

Features of the Tivoli Agent Management Services

The Agent Management Services relies only on attributes that are common to all agents (such as file system installation location, file system log file location, and executable name) and APIs that are common to operating systems (such as enumerating the list of running processes). Using this information, the Agent Management Services improves agent availability and provides a simple, unified interface for the view and control of the agents' availability.

You can bring an agent under Agent Management Services management without making any changes to the agent. As additional agents are added to a system, they can easily be brought under Agent Management Services management.

Agent Management Services is a strategic approach to Tivoli Monitoring agent management that provides these features:

- Ability to monitor the availability of other agents and respond automatically to abnormalities according to user policy.
- An automated method through policy settings and a manual method through Tivoli Enterprise Portal take action commands to start, stop, *manage*, and *unmanage* an agent manually.
- Agent management workspaces with views of the information being collected by the Agent Management Services. The agent management workspaces are provided for the base and most distributed Tivoli Enterprise Monitoring Agents.

Component relationships

The Agent Management Services uses three interfaces to communicate with other components in the OS agent process.

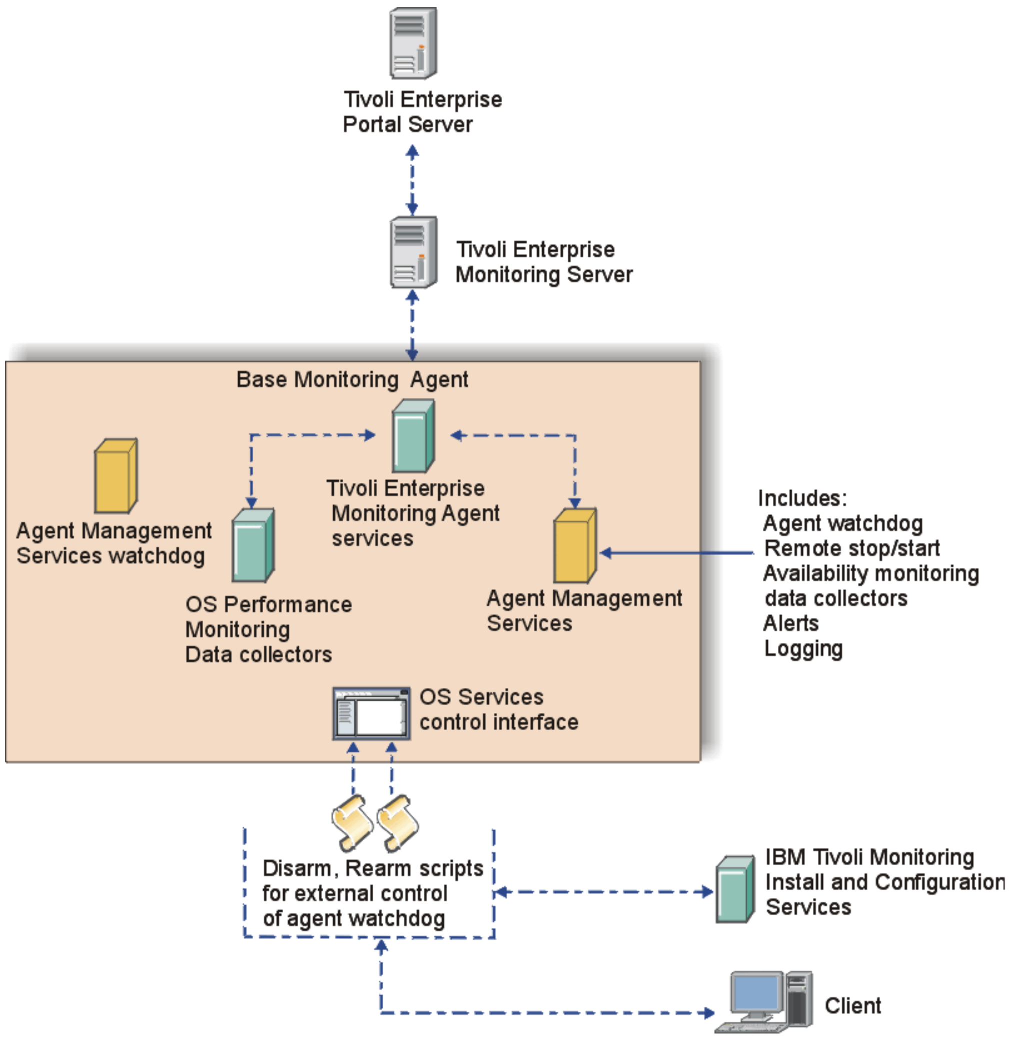

Tivoli Enterprise Portal Server
Tivoli Enterprise Monitoring Server
Base Monitoring Agent
Agent Management Services watchdog
Tivoli Enterprise Monitoring Agent services
OS Performance Monitoring Data collectors
Agent Management Services
Includes:
Agent watchdog
Remote stop/start
Availability monitoring data collectors
Alerts
Logging
OS Services control interface
Disarm, Rearm scripts for external control of agent watchdog
IBM Tivoli Monitoring Install and Configuration Services
Client

Figure 32. Interactions of Agent Management Services components with IBM Tivoli Monitoring components

Component descriptions

Agent Management Services includes two components: Agent watchdog and Agent Management Services watchdog:

Agent watchdog
The agent watchdog performs specific availability monitoring actions against an agent based on the policy in the agent's *common agent package* (CAP) file. This component runs inside the OS agent process as a logical component. Other than the OS agent itself, the agent watchdog watches any monitoring agent that has an XML file in the `CAP` directory of the OS agent installation.

Agent Management Services watchdog
Who watches the watchdog? That is the job of the *Agent Management Services watchdog*, also known as the *Proxy Agent Services Watchdog*. You can check the status of this watchdog in the **Agents' Runtime Status** view in the Tivoli Enterprise Portal. It provides similar monitoring as the agent watchdog within the OS agent, but it is used only to watch the OS agent. The agent watchdog within the OS agent provides additional capability including the monitoring of other agents, responding to Tivoli Enterprise Portal Desktop queries and handling the Take Actions using the

communication facility of the OS Agent. The Agent Management Services watchdog is included as a stand-alone executable file with the OS agents and runs as process **kcawd** on Linux and operating systems such as UNIX, and as process **kcawd.exe** on Windows.

Tivoli Enterprise Portal user interface

The Tivoli Enterprise Portal is the user interface for the Agent Management Services services, with predefined take action commands for manually starting or stopping management of an agent by the Agent Management Services, and for starting or stopping an agent when it is being managed by the Agent Management Services. These take action commands are available from the Agent Management Services workspace pop-up menus and can be referenced in situations for reflex automation.

Note: You can also continue use the familiar methods for starting and stopping an agent, such as through Manage Tivoli Enterprise Monitoring Services and through the Tivoli Enterprise Portal navigator pop-up menu.

Tivoli Agent Management Services installation and configuration

The Agent Management Services is installed automatically with the Linux OS agent, UNIX OS agent, or Windows OS agent, depending on the host platform. Application support files for these agents are also installed on the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

With IBM Tivoli Monitoring V6.2.3 Fix Pack 1 or later OS agents are now monitored through sockets in addition to the process status and `cinfo` check. The socket monitored is a PIPE internally used by the Proxy Agent Services for external requests. You can choose to restart the OS agent if it does not respond to the socket in the consecutive number of tries specified by `KCA_MAX_RETRIES_ON_PIPE` environment variable. Once the variable has been changed in the agent ENV file, you must restart the agent for the changes to be implemented. By default the variable is not defined, meaning that the OS agent is never restarted. You must use a value for this variable that is greater than 5.

Common agent package file

The monitoring behavior of the Agent Management Services towards a particular agent is governed by settings in an XML-based policy file, referred to as a *common agent package* (CAP) file. Every agent that can be managed by the Agent Management Services installs a CAP file named, where *pc* is the product code, k*pc*_default.xml into a directory defined by the **KCA_CAP_DIR** environment variable in the OS monitoring agent configuration file for the relevant platform. Agents that run natively on 64-bit Windows put their CAP files in the 64-bit Tivoli Monitoring Agent directory; all others go in the 32-bit directory:

Windows *install_dir*\TMAITM6[_x64]\CAP

Linux UNIX *install_dir*/config/CAP.

On the zLinux platform, Agent Management Services has the following behaviors:

- V6.2.3 Fix Pack 1 (with a clean installation): Enabled by default
- V6.2.3 Fix Pack 1 (for upgrades): Enabled or disabled, depending on the state found before the upgrade
- V6.2.2 Fix Pack 2 or later: Disabled at installation or upon upgrade, regardless of whether it was active or inactive prior to the upgrade

To enable the Agent Management Services after an upgrade, set the `KCA_CAP_DIR` environment variable to an existing directory containing the CAP files.

On Intel Linux and other supported platforms, the Agent Management Services is enabled by default.

CAP file customizable elements

The CAP file installed by the agent is configured to be read-only and should not be directly modified. If you want to customize the policy settings of this file, create a copy of the file and name it with the convention k*pc*.xml. Watchdog automatically detects when a new CAP file is added or a CAP file is removed from the directory. Changes to an existing CAP file require the OS agent to be restarted in order to pick up the changes. An update to a non-OS agent CAP file can be done without needing to restart the OS agent. To complete this action remove the CAP file from the directory and then after Watchdog has discovered the file has been removed, copy the updated file back into the directory. An update to the OS agent CAP file requires the OS Agent to be restarted.

You can have one CAP file govern multi-instance monitoring agents or create a separate CAP file for each instance.

The elements defined in the CAP file can be viewed in the Tivoli Enterprise Portal “Agent's Management definitions” view of the Agent Management Services workspace. **Note:** Attributes not defined in the CAP file are not displayed in “Agent's Management definitions” view.

The order of the elements is important. Review `kwgcap.xsd` for a formal definition of the CAP file schema.

<checkFrequency>
The length of time between availability checks by Agent Management Services of the managed agent. If system load is heavy, consider increasing the checkFrequency interval along with the KCA_CMD_TIMEOUT agent environment variable setting.

Enter the frequency value in multiples of 5 seconds, up to a maximum of 3600 seconds (1 hour). Default: **120**.

<cpuThreshold>
The maximum average percent of CPU time that the agent process can consume over a time interval equal to “checkFrequency” seconds before being deemed unhealthy and then restarted by Agent Management Services.

Enter the threshold percentage as a positive integer from 1 to 100.

<memoryThreshold>
Maximum average amount of working set memory that the agent process can consume over a time interval equal to “checkFrequency” seconds before being deemed unhealthy and then restarted by Agent Management Services.

Enter the threshold value followed by the unit of measurement: KB, MB, or GB. Example: 50 MB.

<managerType>
The entity that performs availability monitoring of the agent.

Enter an enumerated value: NotManaged or ProxyAgentServices. Default: **NotManaged**.

<maxRestarts>
The number of times per day an abnormally stopped or unhealthy agent should be restarted. Agents that do not need to be kept running can have a value of 0.

Enter a positive integer. Default: **4**.

<subagent id>
Edit this value *only* if you are creating an instance-specific CAP file for a particular agent. For example, if you want to create a CAP file specifically for a set of DB2 agent instances where the kud_default.xml file has a subagent id="kudagent", set it to something like <subagent id="kud_instance">. The <agentName> value for both the agent's original CAP file and its instance-specific CAP files should match.

Enter a string value for the ID.

<instance>
Use this element to provide specific instance names that the target CAP file policies apply to. It must follow the <agentName> element in the CAP file. For example, to specify that an instance of a CAP file should apply to two specific instances of the Tivoli Monitoring DB2 agent, named test1 and test2, enter this information:

```
<subagent id="kud_instance">
<agentName>ITCAM Agent for DB2</agentName>
<instance>
   <name>test1</name>
   <name>test2</name>
</instance>
```

Enter a string value for the instance name within a <name> </name> tagging pair.

Database and messaging monitoring agents on Linux and UNIX

The database and messaging agents are typically started as non-root users. For the Agent Management Services to support this behavior, you can specify that an agent start as a particular user in the start script of the CAP file.

The Agent Management Services rely on the same file that the autoscript files rely on, **kcirunas.cfg**, to get configuration information about which user an agent should *RunAs*. This information is used when the Agent Management Services starts the agent to ensure that it runs as the correct user. In an environment where agents are remotely deployed, use the ***hostname*_kdyrunas.cfg** file. The file is also checked for *RunAs* information.

If you want to enable this support in an older CAP file, update the CAP file as illustrated in the following example of the Tivoli Log File Agent (lo) on Linux (lz):

```
<startScript>
  <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/agentInstanceCommand.sh
  lo START ##INSTANCE##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>

<startScript>
```

```
  <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/agentInstanceCommand.sh
  lo START ##INSTANCE## ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>
```

To enable this support in an older CAP file, update the stop script:

```
<stopScript>
  <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/agentInstanceCommand.sh
  lo STOP ##INSTANCE## ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</stopScript>
```

For single instance agents, such as the Linux OS Agent on Linux, use the following syntax:

```
<startScript>
  <command>su -c "$CANDLEHOME/bin/itmcmd agent start lz" -
  ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>

<stopScript>
  <command>su -c "$CANDLEHOME/bin/itmcmd agent stop lz" -
  ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</stopScript>
```

For single instance agents such as the UNIX Agent, use this syntax. It is identical to the syntax on Linux except that - ##USER## is placed after su rather than at the end:

```
<startScript>
  <command>su - ##USER## -c "$CANDLEHOME/bin/itmcmd agent start ux"
  </command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>

<stopScript>
  <command>su - ##USER## -c "$CANDLEHOME/bin/itmcmd agent stop ux"
 </command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</stopScript>
```

Related reference:

Linux or UNIX installation considerations: Autostart scripts
Agent RunAs configuration is in kcirunas.cfg.

Monitoring the availability of agents

Agent Management Services responds to a stopped or reconfigured agent by restarting it. The Agent Management Services determines that the agent is stopped based on its type, the command specified in the <availabilityStatusScript> element of the CAP file, or both.

For agents of type *Console*, the Agent Management Services determines if the agent is stopped by querying the operating system for the running application using the value from the `<agentPath>` element of the CAP file.

For agents of type *WinService*, the determination is done by querying the Windows service control manager for the status of the service defined in the `<serviceName>` element of the CAP file.

For agents and instances of type *ITM_Windows* and *ITM_UNIX*, the Agent Management Services determines if the agent is stopped by using the command specified in the `<availabilityStatusScript>` element of the CAP file, which is either kinconfg, or a script that calls cinfo.

If the operating system does not show the process in its list of running processes, Agent Management Services knows the process is down and will attempt to restart it using the command or script defined in the `<startScript>` element of the common agent package file. If there is no CAP file, the operating system is checked.

Managed agents that are configured but not started will be automatically started by the watchdog within 30 minutes of being configured. Managed agents whose configured instances are started by the user will be discovered immediately and appear in the Agents' Availability Status view.

If the number of connection attempts to the monitoring server exceeds CTIRA_MAX_RECONNECT_TRIES (default setting is 0), the agent attempts to shut down. If the Agent Management Services Watchdog is running, it immediately restarts the agent. If you want the agent to shut down when CTIRA_MAX_RECONNECT_TRIES is exceeded, this Watchdog process must be disabled. Use the AMS Stop Management action to disable this watchdog process.

Managing the agent manually

From the *Agent Management Services* workspace for the agent, you can run these Take Action commands to start, stop, manage, and unmanage agents.

The action taken will persist until you use the opposing action or start or stop an agent with another method (Tivoli Enterprise Portal, Manage Tivoli Monitoring Services, or at the command line). In the *Agents Management Status* table view, right-click the row of the agent whose status you want to change, then select the action:

AMS Recycle Agent Instance
Use this action to stop and and restart a particular instance of the monitoring agent.

AMS Reset Agent Restart Count
Use this agent to return to 0 the count of agent attempts to restart.

AMS Start Agent
Use this action to start an agent that is under the management of the IBM Tivoli Monitoring Agent Management Services. For a multi-instance agent, use **AMS Start Agent Instance**.

AMS Stop Agent
Use this action to stop an agent that is under the management of the IBM Tivoli Monitoring Agent Management Services.

AMS Start Agent Instance
Use this action to start a particular instance of the monitoring agent.

AMS Start Management
Use this action to put an agent under the management of the IBM Tivoli Monitoring Agent Management Services. This is useful when the agent was taken offline intentionally and you are ready to resume running the agent and having it managed.

AMS Stop Management
Use this action to remove an agent from management by the IBM Tivoli Monitoring Agent Management Services. This is useful when you want to take an agent offline and not have it restarted automatically.

For example, to start managing the Windows OS Agent (displayed in the Agent Management Services workspace, Agent Management Status view as *Unmanaged*), right-click the row and click **Take Action > Select**. Select the AMS Start Management action from the list of possible actions. The command reads, `NT:AMS_Start_Manage "Windows OS Agent"`. Click **OK** to start managing the agent. After you click **Refresh**, the Universal Agent status changes to *Managed*.

For further information on each command and Take Action commands in general, see the *Tivoli Enterprise Portal User's Guide* and the user's guide for the specific agent.

Related reference:

Take action commands
How to use and define take action commands in the Tivoli Enterprise Portal

Chapter 15. Agent-based services

A Tivoli Enterprise Monitoring Agent can run independently of the Tivoli Enterprise Monitoring Server. You can configure different levels of autonomy based on the functionality that the monitoring agent should have, resource constraints, and how much dependency the agent should have on the monitoring server. Any monitoring agent with an infrastructure of IBM Tivoli Monitoring V6.2.2 FP2 or later can be configured to run autonomously.

Monitoring agents start independently of their monitoring server and they collect data, run situations, and register events when they are disconnected from the monitoring server. This is the default behavior, which can be adjusted for greater or less autonomy.

If you configure an Agent Builder or OS agent to be autonomous it becomes a Tivoli System Monitor Agent. System monitor agents are installed and configured to have no dependency on nor any connection to a monitoring server. System monitor agents are like any other monitoring agent except that any processing that can be done only through the monitoring server is not available. As well, a system monitor agent must not be installed on the same system as a Tivoli Management Services component or an enterprise monitoring agent.

You can configure specialized XML files to define and run situations locally, to collect and save historical data locally, and to emit Simple Network Management Protocol (SNMP) alerts or Event Integration Facility (EIF) events or both to a corresponding receiver without connection to a monitoring server. These specialized XML files are available for both enterprise monitoring agents and system monitor agents.

Autonomous capabilities

In addition to the built-in autonomous capability of Tivoli Enterprise Monitoring Agents and Tivoli System Monitor Agents, you can configure special XML files that require no connection to a Tivoli Enterprise Monitoring Server. With these XML files you can define and run situations locally, emit situation events as SNMP alerts or EIF events to a receiver, collect and save historical data locally, and use Centralized Configuration to distribute XML file updates to selected monitoring agents.

Tivoli Enterprise Monitoring Agent

Tivoli Enterprise Monitoring Agents are configured for autonomous operation by default: The agent starts and continues to run with or without connection to its monitoring server. With no connection to the monitoring server, the agent can continue running situations autonomously; when the agent connects to the monitoring server, it uploads situation events that took place while it was disconnected. This incurs use of additional disk space at the agent.

Some situations might not be able to be evaluated completely on the agent alone and are unable to run when there is no connection to the monitoring server. For example, situations using a group function such as COUNT or AVG in the formula must be evaluated at the monitoring server. Even if the agent or the host system is restarted, the events are persistently preserved and uploaded on reconnect. This happens automatically on all agents that use the Tivoli Enterprise Monitoring Agent V6.2.2 or later framework. No configuration changes are required.

Autonomous mode was introduced in V6.2.1 as a configurable agent parameter: IRA_AUTONOMOUS_MODE. Starting with V6.2.2, this parameter is enabled (set to **Y**) by default. If you do not want autonomous behavior enabled for an agent, you can disable it by setting the parameter to **N**. Regardless of the setting for this parameter, historical data collection always runs autonomously and reflex automation for a situation is carried out when the situation becomes true. See “Environment variables for autonomous behavior” on page 328.

z/OS OMEGAMON XE for z/OS and OMEGAMON XE for Storage agents must run connected because they are configured in the Tivoli Enterprise Monitoring Server in the local RTE: If the monitoring server becomes unavailable, so too do these agents. Running connected to the monitoring server does not prevent them from supporting autonomous capabilities such as emitting SNMP traps and private situations. (The OMEGAMON XE on z/OS agent can be configured to run standalone, and therefore without a monitoring server connection, but that means that no plex data is available for alerts and situations.) OMEGAMON XE for IMS™ currently does not support any of the autonomous capabilities.

Tivoli System Monitor Agent

The Tivoli System Monitor Agent is installed on a computer that has no Tivoli Management Services components or Tivoli Enterprise Monitoring Agents installed other than agents built with Tivoli Monitoring Agent Builder V6.2.2 or later.

The Tivoli System Monitor Agent agent is an OS agent that never connects to a monitoring server. The autonomous version of the agent uses the same agent code that is installed for a full OS agent, but Java is not used for the installation process and the configuration user interface is not provided. The resulting installation is faster and has a small installed footprint. Local XML configuration files for defining such functions as private situations and SNMP alerts are processed during agent startup.

Private situations

Enterprise monitoring agents and system monitor agents can use locally defined situations to operate fully autonomously. These locally defined *private situations* are created in a private situation definition XML file. Private situations events come directly from the monitoring agent. You must place a private situation configuration file in the agent installation and restart the agent to enable this function. If you want to send an SNMP alert or EIF event when a private situation event is opened, then the SNMP trap configuration file or EIF event configuration file must also be in the agent installation.

Private situations on an enterprise monitoring agent have no interaction with or reporting of any kind to the monitoring server. Private situations and enterprise situations can run concurrently.

Important: Be aware that all situations, whether private or enterprise, must have unique names. Otherwise, actions invoked upon one situation are applied to the other situation with the same name. You can use the CLI **tacmd listSit** command to get a list of the enterprise situations on the hub monitoring server.

See “Private situations” on page 333.

SNMP alerts and EIF events

Prior to IBM Tivoli Monitoring V.6.2.2, situation events for an enterprise monitoring agent could be forwarded by the Tivoli Enterprise Monitoring Server to an EIF (Event Integration Facility) receiver. IBM Tivoli Monitoring V.6.2.2 or later enables you to configure SNMP alerts to be sent for situation events to an SNMP receiver directly from the agent without first passing the event through the monitoring server. Likewise, with IBM Tivoli Monitoring V.6.2.2 Fix Pack 1 or later, you can create an EIF event configuration file for emitting private situation events to an EIF receiver.

These methods of sending events to OMNIbus can coexist and your monitored environment can be configured for any combination thereof:

- Forward enterprise situation events through the monitoring server to receivers such as the IBM Tivoli Enterprise Console event server and Netcool/OMNIbus Probe for Tivoli EIF. (See Situation event integration with Tivoli Enterprise Console and Situation event integration with Tivoli Netcool/OMNIbus.)
- Send SNMP alerts for enterprise situation events, private situation events, or both to receivers such as the Netcool/OMNIbus SNMP Probe.
- Emit private situation events directly to an EIF receiver as defined in an EIF event configuration file.

Enterprise situations: You can create a trap configuration XML file that enables an agent to emit SNMP alerts directly to the event receiver with no routing through the monitoring server. The agent must connect to the monitoring server at least once to receive enterprise situation definitions. The user needs to place an SNMP trap configuration file in the agent installation and restart the agent to enable this function.

Private situations: Enterprise monitoring agents and system monitor agents can also send SNMP alerts for private situations directly to a receiver such as the Netcool/OMNIbus SNMP Probe or emit EIF events for private situations to an EIF receiver such as the IBM Tivoli Enterprise Console event server or the Netcool/OMNIbus Probe for Tivoli EIF.

Important: If you are forwarding enterprise situation events to the Netcool/OMNIbus Probe for Tivoli EIF and emitting SNMP alerts for enterprise situation events to the Netcool/OMNIbus SNMP Probe, there is a difference in the EIF forwarded situation event and the SNMP alert formats and the data contained by each. Be aware that an event for a situation that is sent to both probes connected to the same Netcool/OMNIbus ObjectServer will not be detected as the same event by OMNIbus deduplication. This results in duplicate entries for the same event within the ObjectServer that will be treated individually. Normally this is not desirable and might be difficult to manage.

See “SNMP alerts” on page 363 and “EIF events” on page 379.

Private history

Just as you can create private situations for the agents installed locally, you can configure private history for collecting short-term historical data in the same private situation configuration file using the HISTORY element. The resulting private history binary files can be viewed through the Agent Service Interface. If you have the Tivoli Data Warehouse configured, you can have the short-term data rolled off to a historical database at intervals.

The HISTORY element includes an attribute for setting the number of hours to keep historical data on the computer before it is trimmed or rolled off to the data warehouse. Although the default value is to retain historical data for 24 hours, there is no limit to the number of hours you can keep locally other than the practical limitations of the computer's storage. If you do not have the EXPORT parameter configured, you can use the provided file conversion programs, such as krarloff, to move data out of the historical files to text files.

The WAREHOUSE element specifies the location of the Warehouse Proxy agent to which historical data is exported.

See "Private history" on page 356.

Enterprise situation overrides

You can configure situation overrides for the locally installed enterprise monitoring agent by using a *`pc_thresholds.xml`* (where *pc* is the two-character product code) configuration file. And you can manage the overrides at the agent manually or with Centralized Configuration. Updated situation thresholds take effect after you restart the monitoring agent. The agent sends threshold overrides to a local file and maintains active situation thresholds over agent restarts.

You can apply a schedule by weekdays, days of the month, and start and stop time of a day. The enterprise monitoring agent maintains dynamic situation threshold overrides audit trails by writing active situation threshold records to the agent operation log, which you can add to a workspace in the Tivoli Enterprise Portal to review situation thresholds in effect.

See "Enterprise situation override XML specification" on page 358.

Agent Service Interface

The IBM Tivoli Monitoring Service Index utility provides links to the Agent Service Interface for each monitoring agent installed locally. After logging into the operating system, you can select one of these reports: agent information, situation, history, or queries.

Additionally, you can make a service interface request directly such as to initiate an immediate configuration download or to recycle a situation.

See "Agent Service Interface" on page 397.

Centralized Configuration

Use Centralized Configuration to maintain monitoring agent configuration XML files at a central location that are pulled from the *central configuration server* at intervals (default is every 60 minutes) or on demand. Agents participating in Centralized Configuration each have their own *configuration load list* XML file that tells where to connect to get the latest updates in the specified configuration files.

A computer that one or more monitoring agents connect to for configuration updates is called a *central configuration server*. A computer with one or more monitoring agents that download configuration updates is called a central configuration client.

See Centralized Configuration.

Environment variables for autonomous behavior

Use the environment file that is provided with the agent framework services to control the autonomous behavior of the Tivoli System Monitor Agent or of the Tivoli Enterprise Monitoring Agent when it is disconnected from the Tivoli Enterprise Monitoring Server.

The *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm) provides instructions for installing and configuring the Tivoli System Monitor Agent. Additionally, the *IBM Tivoli Monitoring Installation and Setup Guide* includes detailed information on environment variables that control autonomous behavior in the "Environment variables" section.

Situation limitations

The types of formula functions that can be used in situations are limited. Private situations can be defined for Tivoli Enterprise Monitoring Agents and Tivoli System Monitor Agents.

To use private situations effectively, you need to understand how they are different from enterprise situations.

Table 32. Availability of situation formula functions for enterprise situations whose events are emitted from the hub monitoring server and private situations whose events are emitted from the monitoring agent

	Situation functions that can be evaluated at the monitoring server	**Situation functions that can be evaluated at the monitoring agent**	
Formula function	**Supported in enterprise situations**	**Supported in enterprise situations**	**Supported in private situations**
Cell functions			
CHANGE	available	available	not available
DATE	available	available	not available
MISSING	available	available	available
PCTCHANGE	available	available	not available
SCAN	available	available	not available[1]
STR	available	available	not available[1]
TIME	available	not available	not available
VALUE	available	available	available
IN	available	available	not available
REGEX	not available	not available	available
Group functions can be applied to multiple row attribute groups and to those configured for historical data collection. Table and chart views require that a time range be set to show a span of data samplings.			
AVG	available	not available	not available
COUNT	available	not available	not available
MAX	available	not available	not available
MIN	available	not available	not available
SUM	available	not available	not available
Situation characteristics			
Embedded, including correlated situations	available	not available	not available
Multiple attribute groups	available	not available	not available

Table 32. Availability of situation formula functions for enterprise situations whose events are emitted from the hub monitoring server and private situations whose events are emitted from the monitoring agent (continued)

	Situation functions that can be evaluated at the monitoring server	**Situation functions that can be evaluated at the monitoring agent**	
Formula function	**Supported in enterprise situations**	**Supported in enterprise situations**	**Supported in private situations**
Persistence enabled	available	available	available[2]
Display item selected	available	not available	available[3]
Uses duper process	available	not available	not available
Distribution to managed system group	available	available	not available
Combination AND and OR logical connectors	available	available	not available[4]

[1] Use the REGEX function instead.

[2] Situation persistence is not evaluated at the agent when the situation is configured to emit a SNMP trap. It is supported when the situation is configured to emit an EIF event. SNMP traps can be emitted in two modes: RC (Rising Continuous) whereby a trap is emitted every time the situation is true; HY (Hysteresis) whereby a trap is emitted the first time the situation is true and a clearing trap is emitted when the situation is no longer true. As well, persistence can be enabled at the trap destination by implementing a persistence rule.

[3] If the situation is configured to emit an EIF event, a separate EIF event is sent for each row with a unique display item value. However, when situations are configured to emit SNMP events situations that include a display item (available for multiple row attribute groups) they are limited to sending one SNMP alert for the first row that evaluates to true; no alerts can be sent for any subsequent rows that evaluate to true.

[4] Private situations are limited to all AND or all OR connectors.

Restriction: Private situations with the MISSING function distributed to subnodes must have a list of subnodes in the DISTRIBUTION tag. Private situations with the MISSING function distributed to agents do not require this.

SNMP alerts from enterprise monitoring agents with subnodes

Monitoring agents that use subnodes, such as subnode agents created with Agent Builder, Monitoring for Energy Management, and Agentless Monitors, can emit an SNMP alert for only one subnode per agent instance where the situation evaluates to true; and for no other subnodes where the same situation evaluates to true.

For each agent instance, data samples are collected in one attribute group table. These metrics are filtered by subnode when displayed in the Tivoli Enterprise Portal, but situations running on multiple subnodes for an agent instance are actually evaluating on a single table. If a situation becomes true on one subnode, an SNMP alert defined for that situation is emitted, but no SNMP alerts are emitted for that situation on any other subnodes because no further rows are processed in the table.

Here are some alternatives to emitting SNMP alerts for agents with subnodes:

- Forward events from the monitoring server to an EIF receiver.
- When you configure the agent, define only one subnode for an agent instance.
- Define a separate situation for each subnode and distribute that situation to only a single subnode. In the following example,

  situation **KAB_Log_Message** is distributed to **ALL LOG** subnodes in the **AB agent**

  situation **KAB_Log1only_Message** is distributed to only the **AB:uxlog1:LOG** subnode

  situation **KAB_Log2only_Message** is distributed to only the **AB:uxlog2:LOG** subnode

  Instance 1 of the AB log monitoring agent is monitoring three logs (there is a subnode for each log file): **uxlog1**, **uxlog2**, and **uxlog3**:

  If a message appears in the file monitored by subnode **uxlog1**, these situations become true: **KAB_Log_Message** and **KAB_Log1only_Message**.

  If a message appears in the file monitored by subnode **uxlog3**, this situation becomes true: **KAB_Log_Message**

  The private situation configuration file for the agent:

```
<PRIVATECONFIGURATION>
  <PRIVATESIT>
   <SITUATION>KAB_Log_Message</SITUATION>
    <CRITERIA><![CDATA[ *IF *VALUE KAB_LOGFILE.Message *NE "" ]]>
   </CRITERIA>
   <INTERVAL>000000</INTERVAL>
   <DISTRIBUTION>AB:uxlog:LOG</DISTRIBUTION>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>KAB_Log1only_Message</SITUATION>
    <CRITERIA><![CDATA[ *IF *VALUE KAB_LOGFILE.Message *NE "" ]]>
   </CRITERIA>
   <INTERVAL>000000</INTERVAL>
   <DISTRIBUTION>AB:uxlog1:LOG</DISTRIBUTION>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>KAB_Log2only_Message</SITUATION>
    <CRITERIA><![CDATA[ *IF *VALUE KAB_LOGFILE.Message *NE "" ]]>
   </CRITERIA>
   <INTERVAL>000000</INTERVAL>
   <DISTRIBUTION>AB:uxlog2:LOG</DISTRIBUTION>
 </PRIVATESIT>
</PRIVATECONFIGURATION>
```

UTF-8 encoded XML files

Unicode Transformation Format, 8-bit encoding form is designed for ease of use with existing ASCII-based systems and enables use of all the characters in the Unicode standard. When composing a local configuration XML file in a language that goes beyond the ASCII character set, such as letters with diacritics and double-byte character sets, use an editor that supports saving the file in UTF-8 encoding.

Windows Linux UNIX

ASCII characters use one byte and comprise the first 128 characters. You can write the XML file in any text editor. For non-ASCII characters, such as characters with diacritics and Kanji characters, an editor that can save the file as UTF-8 is required.

z/OS

Because UTF-8 is not easily displayed or edited on z/OS, the XML can be encoded in UTF-8 or using the agent's code page. The code page is set in the agent environment file with the environment variable LANG, such as LANG=en_US.IBM-1047. The environment file can be found in *&hilev.&rte.*RKANPARU, with member name K*PC*ENV (where *PC* is the two-character product code). The LANG variable should match your terminal emulator if you are using the emulator to edit the file. The default code page for FTP is IBM-1047 if you are editing the file on Windows, Linux, or UNIX and then uploading the file as ASCII text to the host.

See "Configuring hub and remote monitoring servers on z/OS" in *Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

IBM i

(EIF event destination configuration is not supported, nor are SNMPv3 informs.) Because UTF-8 is not easily displayed or edited on IBM i, the XML can be encoded in UTF-8 or the agent's code page. The code page is set in the agent environment file with the environment variable LANG, such as LANG=/QSYS.LIB/EN_US.LOCALE. It is best to set the LANG environment variable before starting **qsh**, the Qshell interpreter. Some utilities do not work correctly if the locale is not valid for the Coded Character Set ID and language ID of the job.

Configuring Agent Management Services on Tivoli System Monitor Agent

Configure the Agent Management Services for Tivoli System Monitor Agents if you want to use the services to monitor and control agent availability.

Before you begin

Agent Management Services is configured differently in a system monitor agent environment:

- System monitor agents are managed by Agent Management Services by default. You suspend management by using the `disarmWatchdog` command, which disables the Agent Management Services watchdog for the system monitor agent and any agents created with Tivoli Monitoring Agent Builder on the same system. You resume management by the Agent Management Services by using the `rearmWatchdog` command, which enables the watchdog for the autonomous agents that are managed by the Agent Management Services. These commands are described in the agent user's guide.
- Agent Builder agents that are installed in a system monitor agent environment are not managed by the Agent Management Services watchdog by default. You can change whether the agent is managed by the watchdog.

About this task

After installing an Agent Builder agent in a system monitor agent environment, take these steps to start or stop Agent Management Services management.

Procedure

1. While the watchdog process is running, move the common agent package (CAP) file named `k`*pc*`_default.xml` (where *pc* is the two-character product code) out of the CAP directory to a temporary location. The file is located in the KCA_CAP_DIR directory.

Windows `install_dir\TMAITM6[_x64]\CAP\`
Linux UNIX `install_dir/config/CAP`
Removing the file from the CAP directory renders the agent invisible to the Agent Management Services.

2. Modify all instances of <managerType> in the CAP file to enable or disable management:
   - <managerType>**ProxyAgentServices**</managerType> to enable management.
   - <managerType>**NotManaged**</managerType> to disable management.

   A best practice is to rename the modified file to **k*pc*.xml** (where *pc* is the two-character product code). All CAP files located in the KCA_CAP_DIR are processed by Agent Management Services. If two or more CAP files share the same “subagent ID” value, they are processed in sorted order. For example, kca.xml is used before kca_default.xml. Also, renaming the CAP file to k*pc*.xml ensures that your changes do not get overwritten during a future upgrade.
3. Save the updated file.
4. While the watchdog process (kcawd) is running, move or copy the updated CAP file back to KCA_CAP_DIR.

Results

The updated Agent Management Services settings are processed after the CAP file is placed in KCA_CAP_DIR.

Private situations

Define private situations for monitoring criteria and the resulting events that are pertinent to your local agent environment or to an event receiver and not relevant to the Tivoli Enterprise Monitoring environment. Private situations can be defined for Tivoli Enterprise Monitoring Agents and Tivoli System Monitor Agents.

Private situation operation

Private situations are created in an XML formatted file that does not interact with the Tivoli Enterprise Monitoring Server. To use private situations effectively, you need to understand how they are different from enterprise situations.

Tivoli Management Services agent framework

Built into the agent framework of the Tivoli Management Services infrastructure is the ability to create situations that run locally and trigger events on the computer where you have either a Tivoli Enterprise Monitoring Agent or Tivoli System Monitor Agent installed.

Enterprise situations and private situations

Enterprise situations are created with the Tivoli Enterprise Portal Situation editor or with the CLI **`tacmd createSit`** command. Enterprise situations send events to the monitoring server and can forward events to an Event Integration Facility receiver such as a IBM Tivoli Enterprise Console event server or Netcool/OMNIbus Probe for Tivoli EIF when the hub monitoring server has been configured to forward events. Enterprise situation events can also be sent as SNMP alerts to a receiver such as the Netcool/OMNIbus SNMP Probe

Private situations are created in a local private situation configuration XML file for the agent. Eligible situation definitions that were exported from the monitored enterprise can also be added to the file to create situations. The events generated by private situations can remain local to your workstation or be sent as SNMP alerts to a receiver such as the Netcool/OMNIbus SNMP Probe. The private situation configuration file resides in the agent `localconfig/`*`pc`* directory, one file per agent, and it contains all the private situation definitions for the agent.

Creating private situations

This example of a private situation configuration XML file for the Windows OS agent has two situations defined. You can create situations in the file by entering them manually.

You can also create situations in this file by exporting existing enterprise situations from the monitoring server, using the CLI **`tacmd bulkExportSit`** and then copying the exported situations that are eligible for use as private situations from their XML file to the agent Private Situation configuration file. The last situation (named Disk_Queue) in the example came from an exported situation XML file.

```
<PRIVATECONFIGURATION>
  <PRIVATESIT>
   <SITUATION>NT_Missing_Scheduler_pr</SITUATION>
    <CRITERIA>
     <![CDATA[ *MISSING NT_Process.Process_Name *EQ ("schedule")]]>
    </CRITERIA>
    <INTERVAL>001000</INTERVAL>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>NT_Paging_File_Critical_pr</SITUATION>
    <CRITERIA>
     <![CDATA[ *VALUE NT_Paging_File.%_Usage *GE 80 ]]>
    </CRITERIA>
    <INTERVAL>001500</INTERVAL>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>Disk_Queue</SITUATION>
    <PDT><![CDATA[ *IF *VALUE NT_Physical_Disk.Avg_Disk_Queue_Length
    *GE 0.004 ]]></PDT>
    <REEV_TIME>003000</REEV_TIME>
  </PRIVATESIT>
</PRIVATECONFIGURATION>
```

The CRITERIA element contains the formula:

- ***VALUE** or ***MISSING** function name. **Value of expression** and **Check for Missing Items** are the only formula functions available for use in private situations.
- **attribute_group.attribute_name** as they are written in these places:
  - **name** element of the agent .atr file, located in the <*install_dir*>/TMAITM6/ATTRLIB/*pc* directory
  - **<PDT>** element of the <*situation_name*>.xml file output generated by the `tacmd bulkExportSit` CLI command
  - **<PREDICATE>** element of the Situation Summary report that is generated through the Agent Service Interface
  - **Display** column in the attribute definitions portion of the Queries report that is generated through the Agent Service Interface
- ***EQ**, ***LT**, ***GT**, ***NE**, ***LE**, or ***GE** Boolean operator.
- **Threshold** for the *VALUE function or comma-separated list for the *MISSING function.

- Multiple expressions can be connected by Boolean AND or OR logic, but not both, and only one attribute group can be used in the formula. Up to nine expressions connected by AND are supported; and up to ten expressions connected by OR are supported.
- The XML coding is case-insensitive with this exception: text attribute values must match the data sample. For example, missing process `notepad` is invalid if it is spelled `NOTEPAD`.

Activation

When the agent is initialized, an XML parser examines and validates the private situation definitions. All XML parsing error messages are recorded in the agent operations log. (See the *IBM Tivoli Monitoring Troubleshooting Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_troubleshoot.htm).)

Private situations continue to run until the agent is shut down.

The events that are opened when a situation becomes true can be sent as SNMPv1/v2 traps or SNMPv3 informs when an SNMP trap configuration file is created and a receiver such as the Netcool/OMNIbus SNMP Probe has been configured to receive them; or as EIF events when an EIF event configuration file is created and a receiver such as the IBM Tivoli Enterprise Console event server or Netcool/OMNIbus Probe for Tivoli EIF is configured to receive them. As well, the Agent Service Interface provides a summary report of situation activity.

You create a private situation file named *`pc`*`_situations.xml` and save it to the *`install_dir`*`/localconfig/`*`pc`* (where *pc* is the product code). If you prefer to name the file differently or use a different path, the IRA_PRIVATE_SITUATION_CONFIG and IRA_LOCALCONFIG_DIR agent environment variables are provided for you to change the file name and path.

Distribute private situations locally or remotely

To edit or delete a private situation, make the changes in the private configuration XML file where it was defined, then redistribute the situation locally or remotely.

Local distribution

After editing the private configuration file and saving it, you can restart the agent to reload the private situation definitions.

Alternatively, you can log on to the Agent Service Interface and enter private situation requests to start, stop or recycle individual private situations. See “Starting the Agent Service Interface” on page 398 and “Agent Service Interface request - Private situation control” on page 417.

Remote distribution

Use a configuration load list to specify the private configuration file for the monitoring agent to pull from the central configuration repository and activate. See “Centralized Configuration” on page 424.

Summary

Private situations are agent monitoring requests defined by a local administrator with criteria that is pertinent to the local agent environment. This is a summary of private situation characteristics:

- Created at the agent locally through a simple editor.

- Emit results and events with agent SNMP traps.
- Run from the time the agent starts until it stops regardless of monitoring server connectivity.
- Multiple expressions in a formula must have logic connectors that are uniformly conjunctive AND or disjunctive OR; a mix of the two connectors in a formula is not supported.
- Support up to nine expressions in the situation formula when connected by Boolean AND logic and up to ten expressions when connected by Boolean OR logic.
- All enterprise situation threshold operators are supported: equal (EQ), not equal (NE), greater than (GT), less than (LT), greater than or equal (GE), and less than or equal (LE).
- Include support for the reflex automation action command.
- Include support for the VALUE and MISSING formula functions only; include no support for group functions or other cell functions.
- Wild card characters are not supported in private situations or situation overrides.
- One attribute group in a situation. The use of two different attribute groups is not supported.
- Run concurrently with enterprise situations when the agent is connected to the monitoring server.
- Can run on a Tivoli Enterprise Monitoring Agent, whether connected or autonomous, or a Tivoli System Monitor Agent.
- Remain unknown to the IBM Tivoli Monitoring centrally managed infrastructure. Tivoli Enterprise Monitoring Server and other IBM Tivoli Monitoring component are unaware of their existence, including their monitoring data and events. Therefore, private situations do not participate in event caching or persistence across agent restarts while the agent is disconnected from its monitoring server.
- As a best practice, enterprise and private situations must have unique situation names.

Private situation XML specification

Use the elements from the private situation XML specification to create private situations for an agent on your computer.

Default private situation path and file name

Windows *install_dir*\localconfig*pc**pc*_situations.xml

Linux UNIX *install_dir*/localconfig/*pc*/*pc*_situations.xml

z/OS *PC*SICNFG in the RKANDATV data set

If you prefer to name the file differently or use a different path, use the IRA_PRIVATE_SITUATION_CONFIG and IRA_LOCALCONFIG_DIR agent environment variables to change the file name and path. For detailed information about the environment variables, see the "Environment variables" topic in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Elements

XML tags are case-insensitive. All other parameters are case-sensitive. For example, you can enter <PRIVATESIT>, <PrivateSit>, or <privatesit>.

Note: You can use the `SITS.xsd` schema file, located in the `samples/PrivateConfigSamples` directory on the Agent DVD image, to validate private situation XML files. However, the schema file requires the XML tag names to be in uppercase.

<PRIVATECONFIGURATION>

PRIVATECONFIGURATION is the root element identifying this XML as an agent private situation configuration document.

```
<PRIVATECONFIGURATION REFRESH="N">
<PRIVATESIT>
<SITUATION NAME="Check_Process_CPU_Usage"  INTERVAL="000500" />
<CRITERIA>
<![CDATA[ *VALUE NT_Process.%_Processor_Time *GE 65 *AND
  *VALUE NT_Process.Priority_Base *NE 0 *AND
  *VALUE NT_Process.Process_Name *NE _Total]]>
</CRITERIA>
<CMD><![CDATA[netstat >.\logs\netstat.dat]]></CMD>
<AUTOSOPT When="N" Frequency="N" />
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

REFRESH=

This parameter is typically ignored. This parameter is only applicable when updating private situation configuration through the Agent Service Interface or when downloading from the Centralized Configuration server. This parameter is ignored when the agent processes the private situation configuration file directly from the `localconfig` directory at agent start up.

Specify "Y" or "Yes" to run only the newly inputted private situations. When specifying Y, all of the existing private situations are stopped and deleted, then only the newly inputted private situations are processed and started. The monitoring agent's operation log provides a list of the old private situations that were stopped and also lists the processed private situations that were started.

Specify "N" or "No" to run all existing private situations.

Only Y, Yes, N, or No values are valid. If you use other values, the entire current private situation configuration XML specifications are rejected.

For more information see the Activate= parameter in "Configuration load list XML specification" on page 429.

<PRIVATESIT>

Enclose each situation definition in PRIVATESIT begin and end tags.

<SITUATION>

Within each set of PRIVATESIT begin and end tags, add a set of SITUATION begin and end tags. Within each set of SITUATION begin and end tags, is the complete situation definition. Define the situation with these attributes:

NAME=

The situation name, which must begin with a letter and can be up to 31 letters, numbers and _ underscores, such as "Missing_Process_Helper_Harmless". Be aware that all situations, whether private or enterprise, must have unique names. Otherwise, actions invoked upon one situation are applied to the other situation with the same name.

INTERVAL=

Unless this is a pure-event situation, specify the sampling interval in HHMMSS format. Default: **001500** (15 minutes). Alternatively, use the <INTERVAL> element.

CRITERIA=

The situation formula. Alternatively, use the <CRITERIA> element.

```
<SITUATION NAME="High_CPU_Usage" INTERVAL="000500"
CRITERIA="*VALUE NT_Process.%_Processor_Time *GE 65
*AND *VALUE NT_Process.Priority_Base *NE 0
*AND *VALUE NT_Process.Process_Name *NE _Total" />
```

DELETE=

Optional. Specify Y to delete the situation specified in the NAME= attribute. Use this attribute to dynamically remove a private situation without recycling the agent or deleting the local private situation XML file. If you specify a private situation name that is not defined or has already been deleted, no action is taken. You can specify multiple delete statements.

To dynamically refresh a private situation, include a delete statement *before* the new add statement. The order of specifications within a private situation XML file affect the final operational private situation definitions and configuration. The order in which the XML files are read is defined by the Centralized Configuration load list and does not follow alphabetical order.

This example deletes a single private situation:

```
<privateconfiguration>
<privatesit>
  <situation name="Check_Process_Name" delete="Y" />
</privatesit>
</privateconfiguration>
```

This example deletes multiple private situations:

```
<privateconfiguration>
<privatesit>
  <situation name="Check_Process_Name" delete="Y" />
</privatesit>
<privatesit>
  <situation name="Check_DiskSpace_Low" delete="Y" />
</privatesit>
</privateconfiguration>
```

This example refreshes a private situation by first deleting the situation and then adding the situation:

```
<privateconfiguration>
<privatesit>
  <situation name="Check_Process_CPU_Usage" delete="y" >
  </situation>
</privatesit>
<privatesit>
  <situation name="Check_Process_CPU_Usage" interval="000500" >
  </situation>
    <criteria>
    <![CDATA[*VALUE NT_Process_64.%_Processor_Time *GE 10 *AND
             *VALUE NT_Process_64.Priority_Base *NE 1 *AND
             *VALUE NT_Process_64.Process_Name *NE _Total]]>
      </criteria>
</privatesit>
</privateconfiguration>
```

<INTERVAL>

Specifies the situation sample interval in HHMMSS format. A value of 000000 (six zeroes) indicates a pure-event situation. For sampled-event situations, the minimum interval is 000030 (30 seconds) and the maximum is 235959 (23 hours, 59 minutes, and 59 seconds). Default: **001500** (15 minutes). This element is required if the INTERVAL attribute is not specified in the SITUATION element.

<CRITERIA>

The situation criteria is specified within this element and the `<![CDATA[ ]]>` element. Each expression has three parts, starting with *VALUE or *MISSING, followed by **attribute-table-name.attribute-name**, the logical operator (such as *EQ), and the attribute threshold value or, for the MISSING function, a comma-separated list of names. It is acceptable, but not required to begin the formula with *IF, as is done in enterprise situation formula syntax.

For the attribute, use the detailed attribute name in the format of attribute-table- name dot attribute-name. The product attribute file defines the agent product attribute tables and associated attributes, for example, **knt.atr** or **kux.atr** files residing in the ATTRLIB directory for a distributed agent installation.

Another way to find attribute names is by querying the table through the Agent Service Interface. Open ASI > Queries and select a table name. ASI returns a complete table schema including the table display name and display names for all table columns.

The Operator defines logical operation of filter value and data. The supported operators are: *EQ for equal, *NE for not equal, *GE for greater than or equal to, *LE for less than or equal to, *LT for less than, and *GT for greater than. Within the <CRITERIA> element, the command is enclosed in Character Data tags to exclude it from XML parsing. This example shows a formula that triggers an alert when the available disk space is 35% or below:

```
<CRITERIA> <![CDATA[*VALUE NT_Logical_Disk.%_Free *LE 35]]> </CRITERIA>
```

For multiple expressions, use the *AND or *OR connector. All connectors in the formula must be the same, either all *AND or all *OR. Mixing logical *AND and *OR connectors is not supported. You can have up to nine *AND connectors or up to 10 *OR connectors in a formula, .

In a formula with multiple expressions, there can be no more than one *MISSING expression, it must be the last expression in the formula, and only *AND connectors can be used. (See the *Tivoli Enterprise Portal User's Guide* for a description of 🔍 **Check for Missing Items**.)

Wildcards are not supported. For example, *VALUE NT_Process.Process_Name *EQ S* to find all processes that start with "S" is invalid in a private situation. Likewise, wildcards in a *MISSING list are invalid, such as NT_Process.Process_Name *EQ ('DB2*') to find all processes beginning with DB2.

Examples:

```
<CRITERIA>
<![CDATA[ *VALUE NT_Process.%_Processor_Time *GE 65 *AND
  *VALUE NT_Process.Priority_Base *NE 0 *AND
  *VALUE NT_Process.Process_Name *NE _Total]]>
</CRITERIA>
```

```
<CRITERIA>
<![CDATA[ *MISSING NT_Process.Process_Name *EQ ('schedule','notepad')]]>
</CRITERIA>
```

```
<CRITERIA>
<![CDATA[ *VALUE Linux_Process.State *NE Running *AND
*MISSING Linux_Process.Process_Command_Name *EQ ('MyHelp','myhelpw')]]>
</CRITERIA>
```

Enumerated attributes have a predefined set of values. You can specify either the enumeration symbol or the name. For example, both of these expressions with a process execution state of `Stopped` (`T`) are valid. If an SNMP alert is sent or an action taken, the symbol is used rather than the name:

```
<CRITERIA><![CDATA[ *VALUE Process.Execution_State *EQ Stopped]]></CRITERIA>
<CRITERIA><![CDATA[ *VALUE Process.Execution_State *EQ T]]></CRITERIA>
```

If the private situation uses any scaled attributes, their values must be normalized for proper evaluation. A scaled attribute value is used to specify how many positions to shift the decimal point to the left. For example, 55.255 is a valid value for an attribute that displays with a scale of 3. To normalize it, you would shift the decimal point right by three places to be 55255.

SCAL (Scale)	Integer comparison value (example used is 5000)
Not defined (0)	5000
1	Seen as 500 or 500.0 but represents 5000
2	Seen as 50 or 50.00 but represents 5000
3	Seen as 5 or 5.000 but represents 5000

The attribute description topics for your product should specify whether the value is scaled. For distributed agents, you can also review the attribute file for `scal` in the attribute definition. For example, khd.atr for the Warehouse Proxy agent has a work queue insertion rate attribute with `scal 2`. Location of k*pc*.atr files: Windows *<install_dir>*`\TMAITM6\ATTRLIB`; Linux UNIX *<install_dir>*`/platform/<pc>/tables/ATTRLIB`, where `platform` is the operating system and `pc` is the product code.

This example shows a hexadecimal integer as the comparison value:

```
<CRITERIA><![CDATA[ *VALUE Disk.Mount_Point_U *EQ '/opt' *AND
*VALUE Disk.Space_Used_64 *GT 0x80000000 ]]></CRITERIA>
```

The <CRITERIA> element is required if CRITERIA is not specified in the <SITUATION> element.

***REGEX**

IBM Tivoli Monitoring frequently requires text scan and pattern matching upon event and sample data, such as name, address, message, and log record. You can add the Regular Expression predicate filter function to enhance agent monitoring capability and applicability. The *REGEX predicate function specification syntax is defined as:
`*REGEX` *`attribute_name operator "Regular Expression"`*
where,

`attribute_name` is the select filter fully-qualified attribute name.

operator is the logical operation of the filter value and data. The supported operators are: `*EQ` (meaning value equal) and `*NE` (meaning value not equal).

`" "` are regular expression delimiters. You can also use other characters for delimiters such as `@ @` if the `"` character is needed in the expression. For example: `*REGEX` *attribute_name operator @Regular Expression@*

Regular Expression specifies the desired regular expression definition.

The following example shows *REGEX predicate function usage. In this example, the company ABC wants to inspect user input for P.O. Box information, in order to raise an event because ABC cannot ship to post boxes.

```
<privateconfiguration>
<privatesit>
<SITUATION NAME="Check_Valid_Address" INTERVAL="000030" >
 </SITUATION>
<criteria>
<![CDATA[*REGEX ABCCUSTOMER_PROFILE00.Address
  *EQ "(?:Post (?:Office )?|P[. ]?O\.? )?[Bb]ox\b"]]>
</criteria>
<DISTRIBUTION>ICVR5A05:ABC00</DISTRIBUTION>
</privatesit>
</privateconfiguration>
```

The *REGEX predicate function can be used in combination with additional predicate functions as show in the following example.

```
<privateconfiguration>
<privatesit>
<Situation Name="Check_Valid_Address" Interval="000030" />
<criteria>
<![CDATA[*VALUE ABCCUSTOMER_PROFILE00.Weight  *GE 5 *AND
  *REGEX ABCCUSTOMER_PROFILE00.Address
  *EQ $(?:Post (?:Office )?|P[. ]?O\.? )?Box\b$ ]]>
</criteria>
<Distribution>ICVR5A05:ABC00</Distribution>
</privatesit>
</privateconfiguration>
```

As a general rule, *REGEX filters on application column data in data row storage buffer. Therefore, the begin input (^ or \A) or end of line ($ or \Z) anchor do not apply and are unnecessary when matching at the beginning or end of column data.

Usage notes:

- Integer attributes are not supported in the regular expression predicate.
- Enumerated attributes are supported in the regular expression predicate, however, the actual column attribute value must be used in constructing the regular expression itself. The enumerate value substitution in regular expressions that requires automated modification of the expression itself, is unsupported.

  For example, the attribute Day_Of_Week in the table Local_Time have the enumerated character string values of Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, but have the actual column value of 00, 01, 02, 03, 04, 05, 06. The actual column values must be used in the regular expression construction.

- A regular expression containing attributes of unsupported data types such as, integer, integer-64, integer enumeration, and hexadecimal string, are flagged as an error and the private situation definition is rejected.

Regular expression implementation varies slightly according to the supporting engine. The private situation *REGEX predicate filter function leverages the ICU Regular Expression engine and its specifications. See the *ICU User Guide* for more information.

<CMD>

Optional. Defines the action command or script to invoke when the situation criteria are true. Within the <CMD> element, the command is enclosed in Character Data tags to exclude it from parsing. This example shows a system command that displays the timestamp in a message box at the agent when the situation becomes true. Without the CDATA tagging, the & ampersand and {} brackets would be considered an error by the XML parser.

```
<CMD>
<![CDATA[ net send &{Local_Time.Timestamp}  ]]>
</CMD>
```

tags.

<AUTOSOPT>

This is required if an action <CMD> is specified. It defines the action command execution options, WHEN (X), FREQUENCY (Y), WHERE (Z). The default is NNN:

WHEN= Optional. "Y" to run the command for each item; or "N" to run the command on only the first row of data returned that meets the situation criteria. If the attribute group returns multiple rows of data and more than one row satisfies the condition, you can choose to issue the command on only the first row that meets the criteria or once for each row that meets the criteria. Default: **"N"**.

FREQUENCY= Optional. "Y" to issue the command every time the situation evaluates to true; or "N" to issue the command when the situation is true, but not again until the situation evaluates to false, followed by another true evaluation. Default: **"N"**.

WHERE= "N" to run the command at the agent. Default: **"N"** Because there is only one possible setting for “where”, you do not need to include it in the AUTOSOPT element.

```
<AUTOSOPT When="Y" Frequency="Y" />
```

<DISTRIBUTION>

Required for products with subnodes (subagents). Specifies a managed system name or a list of managed system names separated by a semi-colon (;). There is no default value.

If you are using the <HISTORY> tag, nest the <DISTRIBUTION> tag within <HISTORY>.

<LSTDATE>

Optional. Situation last updated timestamp. If it is unspecified then the current data time is automatically generated. The format is CYYMMDDHHMMSSmmm (as in 1100715074501000 for July 15, 2010 at 07:45:01) where:

C = Century (1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

<LSTUSRPRF>

Optional. This is the ID of the user who last updated this situation definition. If it is unspecified then the current logon user ID is used. Example:

```
<LSTUSRPRF>SYSADMIN</LSTUSRPRF>
```

<LSTRELEASE>

Optional. Specifies the situation version. Example:

```
<LSTRELEASE>V622</LSTRELEASE>
```

<SITINFO>

Optional. Defines the situation qualifiers for EIF events. Enclose in the `<![CDATA[   ]]>` element. Alternatively, it defines qualifiers for EIF events using parameters. Multiple qualifiers are delimited by a semicolon (;).

ATOM= Optional. For multiple-row attribute groups. This is the catalog COLUMN name to use as the display item, which causes an event to be generated for each subset of rows with the same display item value.

COUNT= Optional. This is called "situation persistence" in the Tivoli Enterprise Portal. Specify the number of intervals that the situation must remain true before an event is opened. Valid values are positive numbers greater than 0.

SEV= Optional. The severity to assign to the EIF event: Fatal, Critical, Warning, Minor, Harmless, Informational, or Unknown.

TFWD=[Y|N] Optional. **Y** is the default. If you want to send only SNMP alerts and no EIF events, set this attribute to **N**.

TDST= Optional. Specify one or more EIF receiver destinations to send the event to. You can enter up to five valid destination server IDs, each separated by a comma (,). Valid destinations are defined in the `pc_eventdest.xml` file. If no TDST parameter is specified, the EIF event is sent to all default event destinations defined (destination entries with a default="Y" setting) in the event destination configuration file.

Examples:

```
<SITINFO><![CDATA[SEV=Fatal;~;]]></SITINFO>
<SITINFO><![CDATA[SEV=Critical;TFWD=Y;TDST=1,3;]]></SITINFO>
<SITINFO><![CDATA[SEV=FATAL;COUNT=2;ATOM=NTEVTLOG.COMPUTER;]]></SITINFO>
```

<HISTORY>

Optional. Use the history element to specify each attribute group that you want to collect historical data for. The agent does not support multiple <HISTORY> specifications for the same TABLE.

TABLE= Required. This parameter specifies the application attribute group name.

EXPORT= Optional. This parameter specifies the interval in minutes for exporting historical data to the Tivoli Data Warehouse. The minimum export interval is 1 minute and the maximum is 1440 (24 hours). Valid export intervals are 1, 5, 15, 30, and values divisible by 60; an interval greater than 60 could be 120, 180, 240, and so on, up to 1440. The

export interval must also be divisible by the INTERVAL parameter value. If you enter an invalid value, no historical data is collected nor exported for the specified attribute group. Default: **none**.
Note: If used in conjunction with the USE=A parameter, the following export integers are valid: 1, 2, 3, 4, 5, 6, 10, 12, and 15.

INTERVAL= Optional. This parameter specifies the historical data collection interval in minutes. The minimum collection interval is 1 minute and the maximum is 1440 (24 hours). Valid intervals are values that divide evenly into 60 or are divisible by 60: an interval below 60 could be 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, and 30; an interval greater than 60 could be 120, 180, 240, and so on, up to 1440. If you enter an invalid value, no history is collected for the specified attribute group. Default:**"15"**.

INTERVALUNIT= Optional. This parameter specifies the INTERVAL parameter time unit. The valid definitions are M (minutes) or S (seconds). When S is specified, the INTERVAL parameter value is interpreted as seconds and the valid value range is between 1 and 60. This parameter must be divisible by 60. In addition, S is only allowed in conjunction with parameter USE=A, otherwise, the history specification is rejected. Default: **"M"**.

RETAIN= Optional. Retain defines the short-term history data retention period in hours. The default is 24 hours and the minimum retention period is one hour. There is no limit other than that imposed by storage space on the computer. After the retention limit has been reached, the oldest data samples are deleted as new samples arrive. Default: **"24"**. Specify "0" to delete the short-term history file after the data has been exported to a comma-separated (CSV) file that is used for analytic usage.

USE= Optional. The parameter defines special history data disposition. It is only valid when defined in conjunction with the EXPORT parameter, otherwise it is ignored. Currently, the only valid definition is A (analytic). Default: **none**.

Attention: In private situations use only external names (same as display names). Private situations using internal names fail.
Examples:

The Windows OS agent collects NT_System table data every 15 minutes and maintains 96 data rows (four times per hour for 24 hours) in the history file.

```
<HISTORY TABLE="NT_System" />
```

The UNIX OS agent collects System table data every 5 minutes and maintains 3 days of short-term history.

```
<HISTORY TABLE="System" Interval="5" RETAIN="72"  />
```

The Windows OS agent collects NT_Logical_Disk table data every minute.

```
<HISTORY TABLE="NT_Logical_Disk" INTERVAL="1"  />
```

The <DISTRIBUTION> element is required for subnode environments. For example, the Universal agent collects TS2TCPIOQ00 table data every 10 minutes and maintains 1 day of short-term history on the subnode named SYSGTCPIOQ:TS200.

```
<HISTORY TABLE="TS2TCPIOQ00" INTERVAL="10" RETAIN="24" >
<DISTRIBUTION>SYSGTCPIOQ:TS200</DISTRIBUTION>
</HISTORY>
```

The Linux OS agent collects KLZ_Disk table data every 5 minutes with the data exported every 15 minutes.

```
<HISTORY TABLE="KLZ_Disk" INTERVAL="5" EXPORT="15" />
```

The Windows OS agent collects NT_Logical_Disk table data every thirty seconds and exports to the Warehouse Proxy agent every minute. The agent maintains one hour of short term history data in the local binary file. This history collection is designated for a stream analytic application.

```
<HISTORY Table="NT_Logical_Disk" Interval="30" IntervalUnit="S"
 Retain="1" Export="1" Use="A" />
```

<WAREHOUSE>

Optional. Use the warehouse element to specify the location of the Warehouse Proxy agent to which historical data is exported. The agent does not support multiple <WAREHOUSE> specifications.

LOCATION=

This parameter specifies the location of each Warehouse Proxy agent to which historical data is exported. A primary location and multiple secondary locations can be specified with each separated by a semicolon (;). This location is only used by the agent when it has full autonomy and the KHD_WAREHOUSE_LOCATION parameter is not specified.

Specify a registered listening address of the Warehouse Proxy agent that will transfer historical data from the agent to the Tivoli Data Warehouse. The syntax is *family protocol:network address[port number]*. Review the warehouse proxy agent's RAS1 trace log to determine the registered addresses.

The following RAS1 log excerpt shows the warehouse proxy agent registering a listening address:

```
khdxrpcr.cpp,621,"register_interface") Registering
   "Candle_Warehouse_Proxy": ip.pipe:#9.44.255.253 [63358]
```

Note: In order for historical data to be successfully exported, the monitoring agent must have the same communications protocol enabled that was specified for the warehouse proxy location. See "Monitoring your operating system via a System Monitor Agent" in the *IBM Tivoli Monitoring Installation and Setup Guide* for more information on the KDC_FAMILIES_OVERRIDE parameter.

Examples:

```
<WAREHOUSE LOCATION="ip.pipe:DEPT-XP[63358]" />
<WAREHOUSE LOCATION="ip.pipe:#9.44.255.253[63358]" />
```

Exported enterprise situation XML specification

Use the situation definitions from the *situation_name*.xml files that result from the CLI **tacmd bulkExportSit** and **tacmd viewSit** commands to populate the agent's private situation configuration file.

See *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) for the tacmd syntax and examples.

If you already have enterprise situations for a Tivoli Enterprise Monitoring Agent, you can run the bulk export situation command or the view situation command to get situation definitions for the specified agent in the XML format that is acceptable to the private situation configuration file. Not all exported situations are valid; only those that use the *VALUE or *MISSING formula functions. See "Situation limitations" on page 329 for other restrictions.

Elements

XML tags are case-insensitive. All other parameters are case-sensitive. For example, you can enter <SITNAME>, <SitName>, or <sitname>.

<TABLE>
This is the root element of the exported situation XML file. In the private situation configuration file, the TABLE tagging (and everything between) from the exported situation is processed as a private situation definition.

<ROW>
This is the child element to follow TABLE.

<SITNAME>
Monitoring situation name. The situation name must begin with a letter and can be up to 31 letters and numbers and _ underscores. Within each set of SITNAME begin and end tags is the complete situation definition. Example:

```
<SITNAME>Free_DiskSpace_Low</SITNAME>
```

Be aware that all situations, whether private or enterprise, must have unique names. Otherwise, actions invoked upon one situation are applied to the other situation with the same name.

<PDT>

The situation criteria is specified within the <PDT> predicate element and the `<![CDATA[   ]]>` element. Each expression has three parts, starting with *IF *VALUE or *IF *MISSING, followed by **attribute-table-name.attribute-name**, the logical operator (such as *NE), and the attribute threshold value or, for the MISSING function, a comma-separated list. Exported enterprise situations always begin with *IF and it is acceptable, but not required to include *IF in the formula.

For the attribute, the detailed attribute name is used in the format of attribute-table- name dot attribute-name. The product attribute file defines the agent product attribute tables and associated attributes, for example, **knt.atr** or **kux.atr** files residing in the ATTRIB directory for a distributed agent installation.

The operator defines the logical operation of filter value and data. The supported operators are: *EQ for equal, *NE for not equal, *GE for greater than or equal to, *LE for less than or equal to, *LT for less than, and *GT for greater than. Within the <PDT> element, the command is enclosed in Character Data tags to exclude it from XML parsing. This example shows a formula that triggers an alert when the available disk space is 35% or below:

```
<PDT> <![CDATA[*IF *VALUE NT_Logical_Disk.%_Free *LE 35]]> </PDT>
```

For multiple expressions, use the *AND and *OR connectors. All connectors in the formula must be the same, either all *AND or all *OR. A mix of logical *AND and *OR connectors is not supported. Example:

```
<PDT> <![CDATA[*IF *VALUE NT_Process.%_Processor_Time *GE 65 *AND
*VALUE NT_Process.Priority_Base *NE 0 *AND
*VALUE NT_Process.Process_Name *NE _Total]]> </PDT>
```

Wildcards are not supported in private situations. For example, *VALUE NT_Process.Process_Name *EQ DB2* to find all processes that start with "DB2" is invalid.Exported enterprise situations with scaled attributes are not normalized when running as private situations. You must normalize the values manually. For example, this enterprise situation expression `Avg Disk Queue Length >= 0.004` is for a floating point attribute with a scale of 3. When the situation is exported with the **tacmd viewSit** command, the export monitoring criteria is shown as:

```
<PDT>  <![CDATA[*IF *VALUE NT_Physical_Disk.Avg_Disk_Queue_Length
*GE 0.004]]> < PDT>
```

When this same definition is specified in a private situation, the value comparison value is interpreted as a zero value.

```
<PRIVATECONFIGURATION>
<PRIVATESIT>
<SITUATION>SCALE_TEST</SITUATION>
<CRITERIA><![CDATA[ *IF *VALUE NT_Physical_Disk.Avg_Disk_Queue_Length
*GE 0.004 ]]></CRITERIA>
<INTERVAL>000030</INTERVAL>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

Normalize the value by shifting the decimal point to the right by three places: 0.004 is 4 or a value such as the one shown here.

```
<CRITERIA><![CDATA[ *IF *VALUE NT_Physical_Disk.Avg_Disk_Queue_Length
*GE 4.123 ]]></CRITERIA>
```

<CMD>

Optional. Defines the action command or script to invoke when the situation is true. Enclose the command in the `<![CDATA[   ]]>` section. Example:

```
<CMD><![CDATA[netstat >.\logs\netstat.dat]]></CMD>
```

<AUTOSOPT>

This is required if an action command <CMD> is specified. It defines reflex automation action command execution options, in order XYZ, between begin and end tags. The default is NNN:

◉ **Only take action on first item**

◉ **Don't take action twice in a row (wait until situation goes false then true again)**

◉ **Execute the Action at the Managed System (Agent)**

X=Y Run command for each item.

X=N Run command on first item only.

Y=Y Run command for each sample interval.

Y=N Do not run command twice in a row.

Z=N This is always set to N for private situations, and means to run the command at the agent. If the exported option is set to Y, the setting will be ignored and be treated as N.

<DISTRIBUTION>

Required for products with subnodes (subagents). Specifies a managed system name or multiple managed system names separated by a comma (,). The default is the agent managed system name or all known subagents. Managed system groups are not supported including the predefined managed system groups, which are prefixed with an asterisk (*).

<LSTCCSID>
Optional. Specifies the IBM Code Character Set ID. **en_US** is the only value allowed.

<LSTDATE>
Optional. Situation last updated timestamp. If it is unspecified then the current data time is automatically generated. The format is CYYMMDDHHMMSSmmm (as in 1090715074501000 for July 15, 2009 at 07:45:01) where:

C = Century (1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

<LSTRELEASE>
Optional. Specifies the situation version.

<LSTUSRPRF>
Optional. This is the ID of the user who last updated this situation definition. If it is unspecified then the current logon user ID is used.

<SITINFO>
Optional. Defines the situation qualifiers for EIF events. Within the <SITINFO> element, enclose the situation formula in `<![CDATA[   ]]>` tagging, such as `<![CDATA[ SEV=Critical ]]>`. Alternatively, this defines qualifiers using parameters. Multiple qualifiers are delimited by a semicolon (;).

ATOM= Optional. For multiple-row attribute groups. This is the catalog COLUMN name to use as the display item, which causes an event to be generated for each subset of rows with the same display item value.

COUNT= Optional. This is called "situation persistence" in the Tivoli Enterprise Portal. Specify the number of intervals that the situation must remain true before an event is opened. Valid values are positive numbers greater than 0.

SEV= Optional. The severity to assign to the EIF event: Fatal, Critical, Warning, Minor, Harmless, Informational, or Unknown.

TFWD=[Y|N] Optional. **Y** is the default. If you want to send only SNMP alerts and no EIF events, set this attribute to **N**.

TDST= Optional. Specify one or more EIF receiver destinations to send the event to. You can enter up to five valid destination server IDs, each separated by a comma (,). Valid destinations are defined in the `pc_eventdest.xml` file. If no TDST parameter is specified, the EIF event is sent to all default event destinations defined (destination entries with a default="Y" setting) in the event destination configuration file.

<TEXT>
Situation description. Within the <TEXT> element, enclose the situation formula in `<![CDATA[   ]]>` tagging.

<REEV_TIME>
Specifies the situation sample interval in HHMMSS format. A value of 0 zero indicates a pure-event situation. The default interval is 15 minutes,

001500; the minimum is 30 seconds, 000030; and the maximum is 23 hours, 59 minutes, and 59 seconds, 235959. Example:

```
<REEV_TIME>000500</REEV_TIME>
```

Ignored elements

The following elements in the exported XML specification are not used except where noted:

<FULLNAME> (processed for EIF)
<ADVISE>
<AFFINITIES>
<ALERTLIST>
<AUTOSTART>
<DESTNODE />
<HUB />
<LOCFLAG />
<NOTIFYARGS>
<NOTIFYOPTS>
<OBJECTLOCK>
<PRNAMES>
<QIBSCOPE>
<REEV_DAYS> (over 1 day unsupported)
<REFLEXOK>
<SENDMSGQ>
<SITINFO> (processed for EIF)
<SOURCE>

Exported enterprise situation example

The NT_System_File_Critical situation exported with **tacmd bulkExportSit** or **tacmd viewSit** is saved in the file, NT_System_File_Critical.xml:

```
<TABLE>
 <ROW>
  <SITNAME>NT_System_File_Critical</SITNAME>
  <FULLNAME>
    <![CDATA[ ]]>
  </FULLNAME>
  <ADVISE>
   <![CDATA[ ADVICE("knt:"+$ISITSTSH.SITNAME$);]]>
  </ADVISE>
  <AFFINITIES>%IBM.STATIC021 01000000000</AFFINITIES>
  <ALERTLIST>*NO</ALERTLIST>
  <AUTOSOPT>NNN</AUTOSOPT>
  <AUTOSTART>*YES</AUTOSTART>
  <CMD>
   <![CDATA[ *NONE  ]]>
  </CMD>
  <DESTNODE />
  <HUB />
  <LOCFLAG />
  <LSTCCSID />
  <LSTDATE>0961009010101000</LSTDATE>
  <LSTRELEASE />
  <LSTUSRPRF>IBM</LSTUSRPRF>
  <NOTIFYARGS />
  <NOTIFYOPTS />
  <OBJECTLOCK />
  <PDT>
    <![CDATA[ *IF *VALUE NT_System.File_Data_Operations/Sec *GE 100000 ]]>
  </PDT>
  <PRNAMES />
  <QIBSCOPE>E</QIBSCOPE>
  <REEV_DAYS>0</REEV_DAYS>
  <REEV_TIME>001500</REEV_TIME>
  <REFLEXOK />
  <SENDMSGQ>*NONE</SENDMSGQ>
  <SITINFO>
   <![CDATA[ SEV=Critical ]]>
  </SITINFO>
  <SOURCE />
  <TEXT>
   <![CDATA[ Knt:KNT1359 ]]>
  </TEXT>
  <DISTRIBUTION>*NT_SYSTEM</DISTRIBUTION>
 </ROW>
</TABLE>
```

In the private situation configuration file, a set of <PRIVATESIT> and </PRIVATESIT> tags are created, then the contents of NT_System_File_Critical.xml pasted inside the tags. This is an nt_situations.xml private situation configuration file after the exported NT_System_File_Critical situation definition was added above another private situation definition, Check_Process_CPU_Usage. The redundant elements (see "Ignored elements" earlier) and unused elements (AUTOSOPT and CMD, LSTCCSID, LSTRELEASE, DISTRIBUTION) from the exported situation were removed, although leaving them in the file does no harm because the XML parser ignores them:

```
<PRIVATECONFIGURATION>
<TABLE>
 <ROW>
  <SITNAME>NT_System_File_Critical</SITNAME>
  <LSTDATE>0961009010101000</LSTDATE>
  <LSTUSRPRF>IBM</LSTUSRPRF>
  <PDT>
   <![CDATA[ *IF *VALUE NT_System.File_Data_Operations/Sec *GE 100000 ]]>
  </PDT>
  <REEV_TIME>001500</REEV_TIME>
  <SITINFO>
   <![CDATA[ SEV=Critical ]]>
  </SITINFO>
  <TEXT>
   <![CDATA[ Knt:KNT1359 ]]>
  </TEXT>
  </ROW>
</TABLE>
<PRIVATESIT>
  <SITNAME>Check_Process_CPU_Usage</SITNAME>
  <PDT>
   <![CDATA[ *IF *VALUE NT_Process.%_Processor_Time *GE 65 *AND
   *VALUE NT_Process.Priority_Base *NE 0 *AND
   *VALUE NT_Process.Process_Name *NE _Total]]>
  </PDT>
  <REEV_TIME>000300</REEV_TIME>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

Tip: Each exported situation is given its own XML file. If your private situations will initially result from an export of your enterprise situations, create an XML with PRIVATECONFIGURATION begin and end tags, then paste the TABLE begin and end tags and everything contained in them into the file for each situation that you want to include. For exported situations, the TABLE tags are equivalent to the PRIVATESIT tags.

Private situation examples

Define private situations for monitoring criteria that is pertinent to your local agent environment and not dependent on or relevant to the enterprise environment. These examples can be used as a template for your private situations.

Tip: Sample private situation configuration files are provided on the Tivoli Monitoring Agent installation media in the `PrivateConfigSamples` directory.

Linux OS lz_situations.xml

```
<PRIVATECONFIGURATION>
<!--  Situation Description: Percentage of time the processor is busy
is extremely high -->
<PRIVATESIT>
  <SITUATION>Linux_High_CPU_Overload_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_CPU.Idle_CPU *LT 10  *AND  *VALUE Linux_CPU.CPU_ID
   *EQ Aggregate ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
  </PRIVATESIT>
<!-- Situation Description: Percentage of packet collisions during data
transmission is high -->
<PRIVATESIT>
  <SITUATION>Linux_High_Packet_Collisons_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_Network.Collision_Percent *GT 10 ]]>
  </CRITERIA>
  <INTERVAL>000500</INTERVAL>
</PRIVATESIT>
<!-- Situation Description: Percentage of available i-nodes is low -->
<PRIVATESIT>
  <SITUATION>Linux_Low_Pct_Inodes_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_Disk.Inodes_Used_Percent *GT 80 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!-- Situation Description: Percentage of space available on a filesystem
is low -->
<PRIVATESIT>
  <SITUATION>Linux_Low_Pct_Space_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_Disk.Space_Available_Percent *LT 15 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!-- Situation Description: Tests if the SSH Daemon, sshd, is up running -->
<PRIVATESIT>
  <SITUATION>Linux_Process_Missing_sshd_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *IF *MISSING Linux_Process.Process_Command_Name
   *EQ ("/usr/sbin/sshd") ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!-- Situation Description: Percentage of Processor time used by
a process high -->
<PRIVATESIT>
  <SITUATION>Linux_Process_High_CPU_pr</SITUATION>
  <CRITERIA>
```

```
   <![CDATA[  *VALUE Linux_Process.Busy_CPU_Pct *GT 60 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!-- Situation Description: High number of stopped processes on this system -->
<PRIVATESIT>
  <SITUATION>Linux_Process_Stopped_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_Process.State *NE Running  *AND
   *VALUE Linux_Process.State *NE Sleeping  *AND
   *VALUE Linux_Process.State *NE Disk  *AND
   *VALUE Linux_Process.State *NE Trace ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Percentage of rejected RPC server or
client calls is high -->
<PRIVATESIT>
  <SITUATION>Linux_RPC_Bad_Calls_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_RPC_Statistics.RPC_Client_Calls_Retransmitted *GT 30
   *OR  *VALUE Linux_RPC_Statistics.RPC_Server_Calls_Rejected *GT 30 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: The swap space paging activity on this system
is extremely high  -->
<PRIVATESIT>
  <SITUATION>Linux_System_Thrashing_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE Linux_System_Statistics.Pages_paged_out_per_sec *GT 400
   *OR  *VALUE Linux_System_Statistics.Pages_paged_in_per_sec *GT 400 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

UNIX OS ux_situations.xml

```
<PRIVATECONFIGURATION>
<!--  Situation Description: Reports High CPU processes  -->
<PRIVATESIT>
  <SITUATION>UNIX_CMD_Runaway_Process_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *IF *VALUE Process.CPU_Utilization *GT 95 ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Process CPU utilization is greater than
or equal to 85% -->
<PRIVATESIT>
  <SITUATION>UNIX_CPU_Critical_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *IF *VALUE Process.CPU_Utilization *GE 85 *AND *VALUE
   Process.Command *NE kproc *AND *VALUE Process.Command *NE swapper ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Notes typical I/O bound processor (NFS) -->
<PRIVATESIT>
  <SITUATION>UNIX_HD_Exces_IO_Wait_prv</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE System.Wait_I/O *GT 20  ]]>
  </CRITERIA>
  <INTERVAL>000200</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Tests if the Internet Services Daemon, inetd,
is up running -->
<PRIVATESIT>
  <SITUATION>UNIX_Process_Missing_inetd_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *MISSING Process.Command *EQ  ("/usr/sbin/inetd") ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Checks the System CPU, Idle, I/O Wait,
and Load Averages for the Busy state  -->
<PRIVATESIT>
  <SITUATION>UNIX_System_Busy_Warning_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE System.System_CPU *GT 50 *AND
   *VALUE System.Idle_CPU *GT 0 *AND *VALUE System.Wait_I/O *GT 0 *AND
   *VALUE System.Load_Average_5_Min *GT 1 ]]>
  </CRITERIA>
  <INTERVAL>000200</INTERVAL>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

Windows OS nt_situations.xml

```
<PRIVATECONFIGURATION>
<!--  Situation Description: One of the NT Logs is close to capacity  -->
<PRIVATESIT>
  <SITUATION>NT_Log_Space_Low_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Monitored_Logs_Report.%_Usage *GE 95 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!-- Situation Description: Test if the NT Scheduler process is running -->
<PRIVATESIT>
  <SITUATION>NT_Missing_Scheduler_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *MISSING NT_Process.Process_Name *EQ ("schedule") ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Percent of the Page File in use is too high -->
<PRIVATESIT>
  <SITUATION>NT_Paging_File_Critical_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Paging_File.%_Usage *GE 80 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Percent of the Page File in use is rising  -->
<PRIVATESIT>
  <SITUATION>NT_Paging_File_Warning_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Paging_File.%_Usage *GE 75 *AND
   *VALUE NT_Paging_File.%_Usage *LT 80  ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Percent of the time the disk drive is busy
is too high -->
<PRIVATESIT>
  <SITUATION>NT_Phys_Disk_Busy_Crit_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Physical_Disk.%_Disk_Time *GT 90 *AND
   *VALUE NT_Physical_Disk.Disk_Name *NE _Total ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
```

```
<!--  Situation Description: Percent of the time the disk drive is busy
is rising  -->
<PRIVATESIT>
  <SITUATION>NT_Phys_Disk_Busy_Warn_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Physical_Disk.%_Disk_Time *GT 80 *AND
   *VALUE NT_Physical_Disk.%_Disk_Time *LE 90 *AND
   *VALUE NT_Physical_Disk.Disk_Name *NE _Total  ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Percentage of processor time used is too high  -->
<PRIVATESIT>
  <SITUATION>NT_Proc_CPU_Critical_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Process.%_Processor_Time *GE 65 *AND *VALUE
   NT_Process.Priority_Base *NE 0 *AND *VALUE NT_Process.Process_Name
   *NE _Total  ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Percentage of processor time used is high  -->
<PRIVATESIT>
  <SITUATION>NT_Proc_CPU_Warn_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Process.%_Processor_Time *GE 50 *AND
   *VALUE NT_Process.%_Processor_Time *LT 65 *AND
   *VALUE NT_Process.Priority_Base *NE 0 *AND
   *VALUE NT_Process.Process_Name *NE _Total  ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: A Service Error was reported  -->
<PRIVATESIT>
  <SITUATION>NT_Service_Error_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_Event_Log.Source *EQ "Service Control Manager"
   *AND *VALUE NT_Event_Log.Type *EQ Error  ]]>
  </CRITERIA>
  <INTERVAL>001000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Rate of operations to file system devices
per second is too high  -->
<PRIVATESIT>
  <SITUATION>NT_System_File_Critical_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_System.File_Data_Operations/Sec *GE 100000 ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Rate of operations to file system devices per second
 is rising -->
<PRIVATESIT>
  <SITUATION>NT_System_File_Warn_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE NT_System.File_Data_Operations/Sec *GE 10000 *AND
   *VALUE NT_System.File_Data_Operations/Sec *LT 100000  ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

Tivoli Data Warehouse Summarization and Pruning sy_situations.xml

```
<PRIVATECONFIGURATION>
<!--  Situation Description: No connectivity to Warehouse database  -->
<PRIVATESIT>
  <SITUATION>KSY_DB_Connectivity_Fail_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KSY_CONNECTIVITY.DB_Connectivity *EQ No ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Failures occurred in pruning  -->
<PRIVATESIT>
  <SITUATION>KSY_Pruning_Failures_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KSY_SUMMARIZATION_STATISTICS.Pruning_Failures *GT 0 ]]>
  </CRITERIA>
  <INTERVAL>000000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Failures occurred in summarization  -->
<PRIVATESIT>
  <SITUATION>KSY_Summ_Failures_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KSY_SUMMARIZATION_STATISTICS.Summarization_Failures
   *GT 0  ]]>
  </CRITERIA>
  <INTERVAL>000000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: No connectivity to the
Tivoli Enterprise Portal Server  -->
<PRIVATESIT>
  <SITUATION>KSY_TEPS_Conn_Fail_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KSY_CONNECTIVITY.TEPS_Connectivity *EQ No  ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

Tivoli Data Warehouse warehouse_situations.xml

```
<PRIVATECONFIGURATION>
<!--  Situation Description: No connectivity to warehouse database  -->
<PRIVATESIT>
  <SITUATION>KHD_DB_Connectivity_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KHD_DB_INFO.DB_Connectivity *EQ No ]]>
  </CRITERIA>
  <INTERVAL>001500</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Critical errors during the execution
of the Warehouse Proxy  -->
<PRIVATESIT>
  <SITUATION>KHD_Error_Critical_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KHD_LAST_ERROR_DETAILS.Error_Severity *EQ Critical ]]>
  </CRITERIA>
  <INTERVAL>000000</INTERVAL>
</PRIVATESIT>
<!--  Situation Description: Fatal errors during the execution
of the Warehouse Proxy  -->
<PRIVATESIT>
  <SITUATION>KHD_Error_Fatal_pr</SITUATION>
  <CRITERIA>
   <![CDATA[  *VALUE KHD_LAST_ERROR_DETAILS.Error_Severity *EQ Fatal ]]>
  </CRITERIA>
  <INTERVAL>000000</INTERVAL>
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

Private history

Private history is the collection and short-term storage of data from a local monitoring agent. Define historical collection in a private situation configuration file for an agent, then use the Agent Service Interface to view the short-term history.

Private history is configured in the private situation configuration file

Local historical data collection is defined in the local private situation configuration file for each attribute group that you want to save historical data for. You can define private history with or without private monitoring situations. There can be only one active history data collection per application table (attribute group).

Use the <HISTORY> tag to specify each attribute group that you want to collect historical data for. Optionally, you can use the EXPORT parameter to specify an interval in minutes for exporting data to the Tivoli Data Warehouse. You can also export data for analytical consumption by using the USE parameter.

Use the <WAREHOUSE> tag to specify a Warehouse Proxy agent to which historical data is exported.

See “Private situation XML specification” on page 336.

Agent Operation Log

All XML validation error messages are saved to the Agent Operation Log. The private history is completely separate and independent of historical data collection and the Tivoli Data Warehouse configuration within IBM Tivoli Management Services. Each private short-term history table data resides in its own history binary file.

Short-term history file names

The table name for an attribute group is also the history binary file name prefixed with **PVTHIST_**; one unique history binary file per table. As part of the private history configuration, you can set the RETAIN attribute to manage the history file size. You can configure an alternative private history file location with the CTIRA_HIST_DIR agent configuration parameter.

Short-term history file directory

The agent outputs all private history files to this subdirectory:

Windows <install_dir> \TMAITM6\logs

Linux UNIX <install_dir>/<arch>/<pc>/hist

You can configure an alternative private history file location with the CTIRA_HIST_DIR agent configuration parameter.

Short-term history file maintenance

The short-term history file conversion utilities, such as **krarloff** (KPDXTRA on z/OS), are provided to move data out of the historical files to delimited text files.

z/OS considerations

The Persistent Data Store (PDS) facility of the Tivoli Enterprise Monitoring Server on z/OS provides a mechanism for Tivoli Monitoring applications to access historical data in the same manner as SQL table data. OMEGAMON XE products leverage the PDS to store and retrieve historical data through the PDS component without use of Tivoli Management Services.

The PDS dictionary contains application table definitions.

- Each table is identified by an application name, usually the application product code, table name, and assigned file group.
- Table column definitions follow the table definition and include column name, data type, and data length. Table columns are related to the table using the same identifier.

The following sample PDS dictionary table definition is from the Tivoli OMEGAMON XE for Mainframe Network product KN3 table KN3BPG:

```
CREATE ID=N303 APPL=KN3      TABLE=KN3BPG     GROUP=KN3
ADDCOL ID=N303 COL=TMZDIFF    TYP=INTEGER    LEN=4    REQ
ADDCOL ID=N303 COL=WRITETIME  TYP=CHARACTER  LEN=16   BIT=72   REQ
ADDCOL ID=N303 COL=ORIGINNODE TYP=CHARACTER  LEN=32   REQ
ADDCOL ID=N303 COL=SYSID      TYP=CHARACTER  LEN=4    REQ
ADDCOL ID=N303 COL=TIMESTAMP  TYP=CHARACTER  LEN=16   REQ
ADDCOL ID=N303 COL=CATDESC    TYP=INTEGER    LEN=2    REQ
ADDCOL ID=N303 COL=CATPCT     TYP=INTEGER    LEN=4    REQ
ADDCOL ID=N303 COL=POOLNAME   TYP=CHARACTER  LEN=4    REQ
ADDCOL ID=N303 COL=CATEGORY   TYP=INTEGER    LEN=4    REQ
ADDCOL ID=N303 COL=SAMPLES    TYP=INTEGER    LEN=4    REQ
ADDCOL ID=N303 COL=INTERVAL   TYP=INTEGER    LEN=4    REQ
```

A table belongs to a PDS group and a number of VSAM files are allocated for a PDS file group for storing table data. The PDS OVERRIDE statement can be used to modify the table or group assignment (or both) and properties. The KN3 group specification is illustrated here:

```
OVERRIDE TABLE=KN3BPG APPL=KN3 WRAP=0 GROUP=KN3
GROUP=KN3 FILE=DSN:CCAPI.COMMON.NV.CIDSSP13.RKN3HIS3
GROUP=KN3 FILE=DSN:CCAPI.COMMON.NV.CIDSSP13.RKN3HIS2
GROUP=KN3 FILE=DSN:CCAPI.COMMON.NV.CIDSSP13.RKN3HIS1
```

The PDS stores table data using the application name, table name, WRITETIME, and any indexed columns as the VSAM file key. For Private History, using KN3BPG table – VTAM_Buffer_Usage_By_Category as an example, the following two configuration steps are required:

1. Add application tables that require history data collection as new tables to PDS dictionary, in data set RKANPARU member KN3PDICT:
   a. Make a copy of the KN3BPG table definition.
   b. Change TABLE=KN3BPG to TABLE=ZN3BPG.
   c. Change ID=N303 to a unique ID (for example, N399).
2. Add application tables OVERRIDE statement in data set RKANPARU member KN3PG.
   a. Copy the table KN3BPG OVERRIDE statement, if any.
   b. Change TABLE=KN3BPG to TABLE=ZN3BPG.

After completion of the two configuration steps, Private History can be configured and retrieved by adding the following information to <pc>_situations.xml as shown in this example:

```
<PRIVATECONFIGURATION>
 <HISTORY TABLE="VTAM_Buffer_Usage_By_Category" Interval="15" Retain="24" />
</PRIVATECONFIGURATION>
```

Service Interface Request example:

```
<HISTREAD>
 <SQLTABLE>
  <TABLENAME>KN3BPG</TABLENAME>
  <FILTER><![CDATA[ *VALUE WRITETIME  *GE 1090728020000000 *AND
   *VALUE WRITETIME  *LE 1090728080000000]]&gt;</FILTER>
  <OUTLIMIT>5000</OUTLIMIT>
 </SQLTABLE>
</HISTREAD>
```

Both enterprise- and private history table data are stored and read by the PDS from the same VSAM datasets using the unique key. Alternatively, you can assigned private history to its own PDS file group and allocate separate VSAM dataset for the private history group.

Enterprise situation override XML specification

You can temporarily override the thresholds set for an enterprise situation on-demand or with a schedule. If you define situation overrides in the Tivoli Enterprise Monitoring Agent's thresholds XML specification, you can manage them locally.

Attention: This information does not apply to private situations. For information about private situations see "Private situation XML specification" on page 336.

Any updates made to the local XML thresholds file take effect after the agent is restarted. Situation overrides that go through the Tivoli Enterprise Monitoring Server (they were defined in the Tivoli Enterprise Portal or with the CLI tacmd setOverride) or applied through the Agent Service Interface take effect immediately.

After reading the XML document, the agent synchronizes the defined threshold override specifications against all data collection requests of all defined table definitions. All threshold parameters, calendar, and situation updates and deletion take effect immediately. The agent outputs the complete threshold override specification XML document to the named local threshold file.

Default situation override path and file name

Windows *install_dir*\TMAITM6*pc*_thresholds.xml

Linux UNIX *install_dir*/bin/*pc*_thresholds.xml

z/OS *PC*THRESH in the RKANDATV dataset

IBM i ctira_sit_path/hostname_pc_thresholds.xml

See the "Environment variables" topic in the *IBM Tivoli Monitoring Installation and Setup Guide* for the agent environment variables that enable local situation override operation.

You must create and manually write override definitions in the same file that is created in CENTRAL mode. The names of the columns to be used when specifying overrides is taken from the attributes file (such as C:\ibm\ITM\TMAITM6\ATTRLIB\knt.atr for the Windows OS agent).

Another way to look up the names of the columns to be used when specifying overrides is through ASI. Open ASI > Queries and select the table name. ASI returns complete a table schema including the table display name and display names for all table columns.

Elements

Enclose all values in double quotation marks, for example, "NT_Available_Bytes_Warning").

<OVERRIDE>

Begin <override> and end </override> tags define this as a dynamic threshold configuration document.

ObjName=

Specify the situation override document name.

<CALENDAR>

Optional. Specify the named calendar definition. Alternatively, you can specify a scheduled override in the <threshold> element.

Name=

Specify the symbolic calendar name.

Action=

Optional. Specify the calendar definition disposition. Value Update creates or replaces named calendar. Value Delete removes existing named calendar.

Start= Stop=

Optional. Use these attributes to apply the override starting at the same time and for the same duration. For example, start="08:15" stop="17:30" causes the override to take effect during the hours of 8:15 AM to 5:30 PM; start="21:45"" stop="05:15" causes the override to take effect from 9:45 PM to 5:15 AM on the next day. If calendar= is not defined, the start=, stop=, and cron= values are used.

Cron=

Optional. Specify a time definition in **minute hour day month day-of-week** format, where **minute** is from 0 to 59, **hour** is 0 to 23, **day** is 1 to 31, **month** is 1 to 12, and **day-of-week** is 0 to 6 (Sunday can be either 0 or 7). Separate each field with a space and use any combination of these symbols:

- Use an asterisk (*) to mean all legal values for the field. For example, * in the month field means every month.
- Enter multiple field values separated by a comma (,).
- Use a hyphen (-) to denote a value range.
- Names can also be used for the **month** and **day-of-week** fields. Use the first three letters of the particular day or month.
- Step value, preceded with a slash (/), is the number to skip. For example, */3 in the hour field means every three hours or (0,3,6,9,12,15,18,21). Step value is not valid for the minute field.

The CRON definition must specify a time range (begin time to end time). If calendar= is not defined, the start=, stop=, and cron= values are used.

LastUpdate=

Optional. Last update 16 digits timestamp. The timestamp is ignored if it is earlier than the existing set timestamp. Default: **0 0 0 0 0**.

ObjName=

Optional. Specify the override document name.

<SITUATION>

Define the situation threshold configuration.

Name=
Specify the situation name.

Table=
Optional. Specify the attribute table name if you prefer to use the attribute name for the key or threshold definition instead of table column name. Use either SQL table name or attribute table name.

Action=
Optional. Specify the situation definition disposition. If the specification does not exist, value Update creates situation specification; otherwise, matching overrides modified. Value Delete removes entire situation override specification.

LastUpdate=
Optional. Last update 16 digits timestamp. Ignored if earlier than existing set timestamp.

Calendar=
Optional. Specify a named calendar definition. The calendar applies to all thresholds in this situation.

Priority=
Situation override priority. A lower numerical value denotes a higher priority. Agent replaces lower priority override with higher priority update and rejects update of equal priority. Default: 2147483647

ObjName=
Optional. Specify override document name.

`<KEY> or <TRIGGER>`
Optional. Define a table column containing a data value to uniquely distinguish a data row in a multiple-row sample. Nested <key> definitions imply AND condition; <key> definitions of the same level imply OR condition.

Column=
Column name. For example, *column=USAGE*. If you have subagents that you want to apply the override on, you can specify column ORIGINNODE as the key and the subnode Managed System name as the key value.

Attr=
Attribute name. As an alternative to specifying a column name, you can specify the attribute name. If you use attribute name, then you must specify the table name in the <situation> element or specify the attribute name in **table-name.attribute-name** format, such as *attr=NT_Paging_File.%_usage*.

Value=
Column or attribute filter data value. The attribute value can also specified between begin and end tags without using the Value parameter. However, the parameter style is preferred.

`<THRESHOLD>`
Define the threshold specification.

Column=
Column name. For example, *column=CONATTMP*.

Attr=
Attribute name. As an alternative to specifying a column name, you can specify the attribute name. If you use attribute name, then you must specify the table name in the <situation> element or specify the

attribute name in **table-name.attribute-name** format, such as *attr=HTTP_Service.Connection_Attempts*.

Position=
Optional. Attribute sequence position in the situation logic construct. Starting with value of 1. Value zero (0) implies all attribute occurrence in conjunctive and/or disjunctive situation logic. This parameter is useful in specifying a particular override attribute in logic containing several occurrences of the same attribute. For example A1 > 80% AND A2 < 95%. Default: **0**

Operator=
Optional. Logic operation uniquely qualify defining attribute in situation construct containing multiple occurrences of the same attribute. Operators values are: EQ, NE, GE, LE, GT, LT. In the above example, A1 can also be qualified using Operator=GT.

Value=
Column or attribute threshold value. Attribute value can also specified in between begin and end tags without using Value parameter. However, parameter style is preferred.

Calendar=
Optional. Specify a named calendar definition. The calendar overrides any calendar specified in the <situation> element and any start=, stop=, and cron= attributes.

Start= Stop=
Optional. Use these attributes to apply the override starting at the same time and for the same duration. For example, start=“08:15” stop=“17:30” causes the override to take effect during the hours of 8:15 AM to 5:30 PM; start=“21:45”" stop=“05:15” causes the override to take effect from 9:45 PM to 5:15 AM on the next day. If calendar= is not defined, the start=, stop=, and cron= values are used.

Cron= Optional. Specify a time definition in **minute hour day month day-of-week** format, where **minute** is from 0 to 59, **hour** is 0 to 23, **day** is 1 to 31, **month** is 1 to 12, and **day-of-week** is 0 to 6 (Sunday can be either 0 or 7). Separate each field with a space and use any combination of these symbols:

- Use an asterisk (*) to mean all legal values for the field. For example, * in the month field means every month.
- Enter multiple field values separated by a comma (,).
- Use a hyphen (-) to denote a value range.
- Names can also be used for the **month** and **day-of-week** fields. Use the first three letters of the particular day or month.
- Step value, preceded with a slash (/), is the number to skip. For example, */3 in the hour field means every three hours or (0,3,6,9,12,15,18,21). Step value is not valid for the minute field.

The CRON definition must specify a time range (begin time to end time). If calendar= is not defined, the start=, stop=, and cron= values are used.

<DEFAULT>
Optional. Define one or more default filter thresholds apply to multiple row samples. This is desirable if <key> tags are defined.

Example

```
<overrides>
  <situation name="Check_Event" table="NT_Event_Log">
    <threshold attr="Source"
               value="Symantec Antivirus"
               start="08:00" stop="17:00" />
  </situation>
  <situation name="NT_Available_Bytes_Critical" table="NT_Memory">
    <threshold attr="Available_Bytes"
               value="750000"
               start="08:00" stop="17:30"
               cron=" * * * * 1-5" />
  </situation>
  <situation name="NT_Disk_Space_Low">
   <threshold name="FREEMGBTES"
               value="10"
               cron="31-59 8-20 */2 * *"
    </threshold>
  </situation>
  <situation name="NT_Log_Space_Low">
      <threshold name="USAGE"
                 value="75"
                 start="08:00" stop="18:00"
                 cron="* * * * MON,WED,FRI"
     </threshold>
  </situation>
  <situation name="Message_Queue_Warning" table="Queue_Statistics">
    <KEY column="ORIGINNODE" value="SYSG:NETQ3">
      <threshold attr="Queue_Depth"
                 value="10"
                 cron="0-30 8-17 * 3,6,9,12 *"
      </threshold>
    </KEY>
  </situation>
  <situation name="NT_Process_CPU_Critical" table="NT_Process">
    <KEY attr="Process_Name" value="_Total">
      <threshold attr="%_Processor_Time"
                 value="70"
                 start="06:00" stop="21:30"
                 cron="* * * * 1-5" />
    </KEY>
  </situation>
  <situation name="NT_System_File_Critical" table="NT_System">
      <threshold attr="File_Data_Operations/Sec"
                value="50000"
                cron="* 6-22 * * SAT,SUN"
      </threshold>
  </situation>
  <situation name="DISKFULL">
    <key column="INSTCNAME" value="C:">
      <threshold column="PCFREE">5</threshold>
    </key>
    <key column="INSTCNAME" value="D:">
      <threshold column="PCFREE">10</threshold>
    </key>
    <default>
    <threshold column="PCFREE">0</threshold>
    </default>
  </situation>
  <situation name="Windows_Events">
    <key column="SOURCE" value="MSFTPSVC">
    <key column="EVENTID" value="10">
      <threshold column="SOURCE">MSFTPSVC</threshold>
    </key>
    <key column="EVENTID" value="100">
      <threshold column="SOURCE">MSFTPSVC</threshold>
    </key>
    </key>
    <key column="SOURCE" value="EventLog">
    <key column="EVENTID" value="6005">
      <threshold column="SOURCE">EventLog</threshold>
    </key>
    <key column="EVENTID" value="6009">
      <threshold column="SOURCE">EventLog</threshold>
    </key>
    </key>
    <default>
      <threshold column="SOURCE">NOPASS</threshold>
    </default>
  </situation>
</overrides>
```

SNMP alerts

Tivoli Enterprise Monitoring Agents and Tivoli System Monitor Agents can be configured to send alerts to an SNMP receiver like Netcool/OMNIbus, using the Netcool/OMNIbus SNMP Probe, or Tivoli NetView. Sample OMNIbus rules files are provided to illustrate some key integration ideas.

SNMP alert configuration

Configure a monitoring agent and an SNMP trap configuration file to emit life cycle events or situation events to an SNMP event receiver.

Trap configuration file

A trap configuration file must be present when the agent is started to enable the agent to emit SNMPv1/v2 traps or SNMPv3 informs for configured situations. If a correctly named trapcnfg.xml file is present in the agent's local configuration directory, the agent emits the traps that are defined in the file when it is started. The file is named `pc_trapcnfg.xml`, where *pc* is the 2-character product code of the agent and resides in the *`install_dir`*`/localconfig/`*`pc`* directory. The file must be named `pc_trapcnfg.xml`, where *pc* is the two-character product code, such as ux for the UNIX OS agent.

IBM i The IBM i agent can send SNMPv1/v2 traps, but it cannot send SNMPv3 informs.

z/OS On z/OS, the default name for the file is *PC*TRAPS in the RKANDATV dataset.

Agent parameters

IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG parameter in the agent environment file can be set to specify a different name and path to the trap configuration file. SNMP alerts are emitted only for situations that are configured in the trap configuration XML file for that agent type. You can specify the complete path or the path relative to the local configuration directory.

To specify the complete path, the PDS should be listed at the end (or omitted and allowed to default to RKANDATV).

IRA_EVENT_EXPORT_SNMP_TRAP=N disables agent SNMP alerts even if the `pc_trapcnfg.xml` file is present.

XML specification

The trap configuration file can include these XML elements:

SNMP

TrapDest

TrapAttrGroup

Situation

StatTrap

SNMP is the top-level XML element. TrapDest, TrapAttrGroup, and Situation are elements within the SNMP begin and end tags.

Sample trap configuration file

Review this sample `nt_trapcnfg.xml` for a Windows OS agent to see how a trap configuration file might be composed. It is located in the *`install_dir`*`\localconfig\nt` directory to enable trap emission for the Windows OS agent. The file is configured to send status traps receiver using SNMPv1 trap on host nt2003infra and to send informs for the individually defined situation events to a Netcool/OMNIbus SNMP probe using SNMPv3 running on host 10.21.32.234.

```
<!–C:\IBM\ITM\localconfig\nt\nt_trapcnfg.xml /-->
<SNMP>
   <TrapDest name="AgentStatMon" Address=" nt2003infra " Version="v1"
   Community="{AES256:keyfile:a}P0hUrmUhCgfFwimS+Q6w+w==" Stat="Y" />

   <TrapDest name="Probe1" Version="v3" Address="10.21.32.234"
   SecLevel="authPriv" User="AuthPrivMD5DES" AuthType="MD5"
   AuthPassKey="{AES256:keyfile:a}yifHSbFcTKHBqvORpzxS6A=="
   PrivType="DES" PrivPassKey=
   "{AES256:keyfile:a}1le2SxljJR1MOIi0EDIvig==" Stat="N" />

   <TrapAttrGroup Table="NT_Paging_File" TrapAttrList="Server_Name,
   %_Usage" />

   <Situation name="NT_Log_Space_Low_pr" sev="2" cat="0"
   mode="HY"
   target="Probe1" />
   <Situation name="NT_Missing_Scheduler_pr" sev="5" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Paging_File_Critical_pr" sev="5" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Paging_File_Warning_pr" sev="2" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Phys_Disk_Busy_Critical_pr" sev="5" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Phys_Disk_Busy_Warn_pr" sev="2" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_System_File_Warn_pr" sev="2" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Proc_CPU_Critical_pr" sev="5" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Proc_CPU_Warn_pr" sev="2" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_Service_Error_pr" sev="2" cat="0"
   mode="RC" target="Probe1" />
   <Situation name="NT_System_File_Critical_pr" sev="5" cat="0"
   mode="HY" target="Probe1" />
   <Situation name="NT_System_File_Warn_pr" sev="2" cat="0"
   mode="HY" target="Probe1" />

   <StatTrap name="EE_HEARTBEAT" sev="1" interval="15" cat="3" />
   <StatTrap name="EE_AUTO_ENTER" sev="1" cat="3" />
   <StatTrap name="EE_AUTO_EXIT" sev="1" cat="3" />
   <StatTrap name="EE_AUTO_USE_LIMIT" sev="5" cat="3" />
   <StatTrap name="EE_TEMS_RECONNECT_LIMIT" sev="5" cat="3" />
   <StatTrap name="EE_TEMS_CONNECT" sev="1" cat="4" />
   <StatTrap name="EE_TEMS_DISCONNECT" sev="1" cat="4" />
   <StatTrap name="EE_SIT_STOPPED" sev="1" cat="4" />
</SNMP>
```

Trap configuration XML specification

Use the SNMP, TrapDest, TrapAttrGroup, Situation, and StatTrap elements in SNMP XML files to configure traps for any agent type that you want to specify for the event receiver.

XML tags are case-insensitive. All other parameters are case-sensitive. For example, you can enter ADDRESS, Address, or address.

Note: You can use the `SNMP.xsd` schema file, located in the `samples/PrivateConfigSamples` directory on the Agent DVD image, to validate SNMP trap configuration XML files. However, the schema file requires the XML tag names to be in uppercase.

SNMP element

The SNMP element of the trap configuration XML specification is the top-level XML element. TrapDest, TrapAttrGroup, and Situation are elements within the SNMP begin and end tags.

```
<SNMP>
 <TrapDest name="OMNIbus2" Address="nswin21a" Stat="Y" />
 <situation name="*" target="OMNIbus2" />
</SNMP>
```

TrapDest element

Use TrapDest elements in a trap configuration XML file to define a trap receiver.

The TrapDest element requires the name and address attributes. Default values are used for any other attributes that are not specified.

```
<TrapDest name="LABEL" Address="HOSTNAME"/>
```

Table 33. TrapDest element XML specification

Attribute	Description	Required	Default	SNMPv1/v2 or SNMPv3
Name=	Alphanumeric label that is used to identify the Trap Destination.	Required		
Address=	Trap receiver's TCP/IP address or hostname.	Required		All
IP=	ip protocol: “4” | “6” 4 is IPv4; 6 is IPv6	Optional	"4"	All
Port=	Trap receiver TCP/IP trap listening port.	Optional	"162"	All
BindAddress=	Used to specify which local interface to use for SNMP traffic.The interface specified must match the IP setting.	Required if the host has multiple network interfaces defined. Otherwise the trap send might fail with error number 22.	First available	All

Table 33. TrapDest element XML specification (continued)

Attribute	Description	Required	Default	SNMPv1/v2 or SNMPv3
Version=	Specify SNMP trap version. Valid string values are (case insensitive) : **v1**, **v2**, **v3**	Optional	v1	All
Type=	Trap | Inform Type must match the Version. Version= "v1" | "v2" Type Must be "Trap" Version= "3" Type Must be "Inform"	Optional	Matches version	All
Stat=	Stat is used on a destination to send all status traps to that receiver when Stat="Y". Set Stat to "N" to disable all status alerts for the TrapDest. Also, set it to "N" if you want only a subset of Status alerts to be sent to the TrapDest. Individual Status Alerts can be sent to specific TrapDest using the StatTrap element.	Optional	"Y"	All
Community=	Specify trap community name string. Should be encrypted using itmpwdsnmp, but clear-text is also allowed. (1-63 characters)	Optional	public	v1 and v2
SecModel=	Specify the security model. Only USM supported.	Optional	USM	v3
SecLevel=	Specify the Authentication and Privacy levels. The levels supported are: **noAuthNoPriv** – no authentication and no privacy **authNoPriv** – authentication no privacy **authPriv** – authentication and privacy (not supported on z/OS monitoring agents)	Required for v3.		v3

Table 33. TrapDest element XML specification (continued)

Attribute	Description	Required	Default	SNMPv1/v2 or SNMPv3
User=	Specify the account name	Required for v3		v3
AuthType=	Specify the authentication protocol. The protocols supported are: MD5 and SHA	Required for v3 SecLevel= authNoPriv or authPriv		v3
AuthPassKey=	Specify the authentication password Should be encrypted using itmpwdsnmp, but clear-text is also allowed. (1-63 characters)	Required for v3 SecLevel= authNoPriv or authPriv		v3
PrivType=	Specify the privacy protocol. The protocol supported are: DES	Required for v3 SecLevel= authPriv		v3
PrivPassKey=	Specify the privacy password. Should be encrypted using itmpwdsnmp, but clear-text is also allowed. (1-63 characters)	Required for v3 SecLevel= authPriv		v3
Timeout=	Specify the timeout (in seconds, integer) for the acknowledgement of SNMPv3 message (minimum 1)	Optional	2	v3
Retries=	Specify the number of retransmissions when a timeout occurs (min 0, max 5)	Optional	3	v3

TrapAttrGroup element

Use the TrapAttrGroup element in a trapcnfg.xml file to specify which attributes from an attribute group to include in situation event traps.

In this syntax example, situations written for the Windows OS Paging File attribute group will send an SNMP trap with the server name, usage percentage and the usage peak values to the event receiver.

```
<TrapAttrGroup Table="NT_Paging_File" TrapAttrList="Server_Name,
%_Usage,%_Usage_Peak" />
```

This element can be used to decrease the amount of attribute data sent in each trap request, reduce the possibility of trap fragmentation, and reduce the received data to include only what is relevant.

The TrapAttrGroup element sets the default attributes that will be sent for all situation that run against the Table. Individual situations can override the TrapAttrGroup settings by specifying a TrapAttrList attribute in the situation element.

If a TrapAttrGroup element is not defined for an attribute table, all attributes in the situation's data row are added to the sitAttributeList varbind of the traps sent for situations based on this attribute table. Attributes used in the situation predicate are added first and remaining attributes are added until the PDU maximum length of 1500 bytes is reached.

Table 34. TrapAttrGroup element XML specification

Attribute	Description
Table=	The name of the attribute table. For manually creating this file, you can look in the agent's attribute file, k*pc*.atr to identify the table names, where *pc* is the two-character product code.
TrapAttrList=	A comma delimited list of attributes to be included in the **sitAttributeList varbind** of the traps sent for situations based on this attribute table.

Situation element

Use situation elements in a trap configuration XML file to define the trap sent for the situation.

```
<situation name="Situation_ID" target="TrapDest_Name" />
```

The Situation element requires the name and target attributes. Default values are used for any other attributes that are not specified. The * asterisk wildcard can be specified for the situation name or target or both:

- Specifying the wildcard for situation name represents all situations. For example the following line sends traps for all defined true situations to the defined TrapDest named trapProbe1:

  ```
  <situation name="*" target="trapProbe1" />
  ```

  Hysteresis mode behavior cannot be specified if a * wildcard is used for situation name.
- Specifying the wildcard for the target parameter enables sending the situation specified in the situation name field to all defined targets:

  ```
  <situation name="NT_Disk_Low" target="*" />
  ```
- Specifying the wildcard for both situation name and target enables the sending of all traps to all defined trap receivers.
- Named situations have precedence over wildcard definitions. If a situation definition includes a wildcard and another situation definition names a situation or the target, the first occurrence of the named situation definition is honored. Example:

  ```
  <TrapDest name="MyReceiver" Address="UAHOST1" Version="v1" />
  <TrapDest name="OMNIbus1" Address="OMNIbus1" Version="v2"
   Community="{AES256:keyfile:a}P0hUrmUhCgfFwimS+Q6w+w==" />
  <TrapDest name="OMNIbus2" Version="v3" Address="9.42.10.164"
   SecLevel="authPriv" User="SnmpUser" AuthType="SHA"
   AuthPassKey="{AES256:keyfile:a}vgpNvf5Vx3XbPj1sKRRvYg==" PrivType="DES"
   PrivPassKey="{AES256:keyfile:a}OK5YOWvRIkPOw9k4JRy9ag==" />
  <situation name="*" target="OMNIbus2" />
  <situation name="My_Missing_Process" target="MyReceiver" />
  ```

```
<situation name="NT_AA_Missing_Test" target="OMNIbus1" />
<situation name="NT_AA_Missing_Test" target="OMNIbus2" />
<situation name="NT_ABC_Missing_Test" target="*" />
```

The **My_Missing_Process** situation sends a trap to **MyReceiver** instead of **OMNIbus2**. And **NT_ABC_Missing_Test** is sent to **MyReceiver**, **OMNIbus1**, and **OMNIbus2** instead of solely to **OMNIbus2** because the situation is defined explicitly rather than using the wildcard.

If a situation is defined more than once, the first occurrence of a situation definition has precedence. Looking again at the example, **NT_AA_Missing_Test** is sent to **OMNIbus1** and not **OMNIbus2** because the first occurrence of the definition for the same situation specifies **OMNIbus1**.

Table 35. Situation element XML specification

Attribute	Description	Required	Default
Name=	This is the ID or short name of the situation.	Required	
Target=	Specify a previously defined TrapDest. “*” implies send trap to all defined destinations.	Required	
Sev=	Specify trap severity. The standard trap severities are: 0 – Cleared 1 – Indeterminate 2 – Warning 3 – Minor 4 – Major 5 – Critical	Optional	2
Cat=	Specify trap category. The standard trap categories are: 0 - Threshold 1 – Network Topology 2 – Error 3 – Status 4 – Node Configuration 5 – Application Alert 6 – All Category 7 – Log Only 8 – Map 9 – Ignore	Optional	0

Table 35. Situation element XML specification (continued)

Attribute	Description	Required	Default
Mode=	Used to specify behavior for SNMP trap emission on sampled situations. The standard modes are: RC – Rising Continuous, whereby traps are sent on each true evaluation of a situation. (Pure events are always RC.) No specific clearing trap will be sent. HY – Hysteresis, whereby a trap is sent the first time the sampled situation evaluates as true. A clearing trap will be sent once the sampled value no longer meets the criteria of the situation. Hysteresis mode requires the situation be named; not specified with a * wildcard.	Optional	RC
Pred=	The situation predicate (formula) is sent in the trap's autoSit-Predicates varbind. The Pred attribute allows you to omit the situation predicate by setting Pred="N". This can be useful if you do not care to receive the predicate or if a complex predicate is taking up too much of the trap PDU, and you want more room to send situation attributes in the sitAttributeList varbind.	Optional	Y
Table=	Table name of the attribute group. Used with the TrapAttrlist to identify a subset of attributes used to construct the **`sitAttributeList varbind`**.	Required only if a **`TrapAttrList`** is used.	
TrapAttrList=	A comma delimited list of attributes to be included in the sitAttributeList varbind of the traps sent for situation. Values specified here will override any TrapAttrList values specified in a TrapAttrGroup element for the table that the situation is running against.	Optional	

Note: Situations for multiple-row attribute groups that include a display item are limited to sending one trap for the first row that evaluates to true, but not for any subsequent rows.

StatTrap

Use the StatTrap element in an SNMP trap configuration file to modify the default configuration of the predefined agent life cycle status traps.

In this syntax example, the predefined trap for EE_HEARTBEAT was modified to specify severity 1 (Indeterminate) for the event, a 30-minute sampling interval, and trap category 3 (Status).

```
<StatTrap name="EE_HEARTBEAT" sev="1" interval="30" cat="3" />
```

There are eight predefined agent life cycle traps and their default values are given in this table. By default, these traps are sent to all TrapDest trap destinations where the Stat attribute is "Y". If the Stat attribute is omitted from a TrapDest element the default value is "Y".

Table 36. Agent life cycle status traps

Attribute	**Description**	**Severity**	**Category**
EE_HEARTBEAT	A heartbeat indicates that the agent is running and events emitted can reach the trap destination. This is the only status trap with a set interval: 15 minutes.	1 – Indeterminate	3 – Status
EE_AUTO_ENTER	The agent has entered autonomous mode.	1 – Indeterminate	3 – Status
EE_AUTO_EXIT	The agent has exited autonomous mode.	1 – Indeterminate	3 – Status
EE_AUTO_USE_LIMIT	The agent has reached the storage limit specified by the IRA_AUTONOMOUS_LIMIT environment variable. Additional events generated while the agent is disconnected from the monitoring server may not be uploaded on reconnect.	1 – Indeterminate	3 – Status
EE_TEMS_RECONNECT_LIMIT	Agent has reached the retry limit specified by the CTIRA_MAX_RECONNECT_TRIES environment variable. The agent will no longer attempt to connect to a monitoring server and will shutdown. In IBM Tivoli Monitoring 6.2.2 or later, the default value of CTIRA_MAX_RECONNECT_TRIES has been changed to 0, so the agent will never shutdown.	1 – Indeterminate	3 – Status
EE_TEMS_CONNECT	The agent has successfully connected to the monitoring server.	1 – Indeterminate	4 - Node Configuration
EE_TEMS_DISCONNECT	The agent has lost connection with the monitoring server.	1 – Indeterminate	4 - Node Configuration
EE_SIT_STOPPED	The situation has stopped	1 – Indeterminate	4 - Node Configuration

Use the StatTrap element to configure agent life cycle traps.

Table 37. StatTrap element XML specification

Status trap	Description	Required	Default
Name=	This trap name must be the name of a predefined Life-Cycle status trap. EE_HEARTBEAT EE_AUTO_ENTER EE_AUTO_EXIT EE_AUTO_USE_LIMIT EE_TEMS_RECONNECT_LIMIT EE_TEMS_CONNECT EE_TEMS_DISCONNECT EE_SIT_STOPPED	Optional	
Target=	Specify a previously defined TrapDest. An asterisk (*) implies send trap to all defined destinations. If no Target is defined, all TrapDest with Stat=”Y” will receive the status trap.	Required	
Sev=	Specify trap severity. The standard trap severities are: 0 – Cleared 1 – Indeterminate 2 – Warning 3 – Minor 4 – Major 5 – Critical	Optional	Varies
Cat=	Specify trap category. The standard trap categories are: 0 - Threshold 1 - Network Topology 2 – Error 3 – Status 4 - Node Configuration 5 - Application Alert 6 - All Category 7 - Log Only 8 – Map 9 - Ignore	Optional	Varies
Interval=	Interval specifies in minutes how often the EE_HEARTBEAT status trap is emitted. Interval is ignored for the other status traps because they are pure events.	Optional	15 for EE_HEARTBEAT 0 for all others

SNMP PassKey encryption: itmpwdsnmp

Use the **itmpwdsnmp** CLI command to interactively encrypt a password or add it to the SNMP trap configuration XML file to encrypt all SNMP passwords.

The **itmpwdsnmp** uses GSKIT to either interactively encrypt a string or to encrypt all SNMP password strings in a trap configuration xml file.

itmpwdsnmp [[-b |-n]your_agent_trapcnfg.xml][-?]

where:
no arguments specifies interactive mode
-b specifies to create a backup file. There is no prompting to delete the backup file.
-n specifies that no backup file is to be created.

your_agent_trapcnfg.xml is a trap configuration xml file that contains plaintext SNMP password strings.
-? displays usage

If a **-b** or **-n** backup option is not specified when encrypting a Trap Configuration xml file, you are prompted to delete the backup. The backup of the original input Trap Configuration xml file is created in the same directory as the original with a date and timestamp appended to the original file name.

Windows `install_dir\TMAITM6\itmpwdsnmp.bat`
Linux UNIX `install_dir/bin/itmpwdsnmp`

CLI examples

This command will interactively encrypt a string:

```
itmpwdsnmp

Enter string to be encrypted:
********
Confirm string:
********
{AES256:keyfile:a}GbHOlF7KPYZS80Rripx4QQ==
```

Then copy the encrypted string into the trap configuration file.

This command encrypts all SNMP password strings in the trap configuration file and then removes the backup of the original file:

```
itmpwdsnmp -n nt_trapcnfg.xml
```

Program Summary

Community strings encrypted 1

AuthPassKey strings encrypted 2

EncryptPassKey strings encrypted 1

MIB for SNMP alerts and agent emits

Tivoli monitoring agents emit three types of SNMP messages: **agentStatusEvent** to convey agent operational status, **agentSitSampledEvent** for situations that sample at intervals and become true, and **agentSitPureEvent** for situations that receive unsolicited notifications.

They are defined in the canbase.mib and cansyssg.mib files that are available on the IBM Tivoli Monitoring IBM Tivoli Monitoring Agents installation media.

agentStatusEvent
The agentStatusEvent is a monitoring agent operational status information trap generated by the Tivoli Autonomous Agent SNMP Event Exporter to inform and notify about a specific agent operational event.

agentSitSampledEvent
A sampled situation event was detected. This trap was generated by the Tivoli Autonomous Agent SNMP Event Exporter in response to a situation threshold being exceeded at the time of the data sampling.

agentSitPureEvent
A pure situation event was detected. This trap was generated by the Tivoli Autonomous Agent SNMP Event Exporter in response to a situation threshold being exceeded. The variables in a pure event trap are identical to those for a sampled event trap except there is no agentSit-SampleInterval because pure events are not sampled; rather the arrival of

unsolicited data from the monitored attribute group causes the situation to become true. A situation created with an attribute group for a system log, for example, opens a pure event when a log entry arrives.

OMNIbus configuration for SNMP

You must configure your IBM Tivoli Netcool/OMNIbus environment for it to receive the SNMP alerts of situation events from Tivoli Enterprise Monitoring Agents and Tivoli System Monitor Agents. The Tivoli Monitoring Agent DVD installation media has the Management Information Base (mib) and sample rules files that you add to the probe configuration.

Configuring OMNIbus to receive SNMP alerts

Configure the SNMP Probe to accept the SNMP traps and informs of situation events from Tivoli monitoring agents.

Before you begin

Have the IBM Tivoli Monitoring V6.2.2 or later Agents DVD available. Verify that Tivoli Netcool/OMNIbus V7.x is installed and that the SNMP Probe is installed.

Do not configure an enterprise situation for emitting SNMP alerts to the SNMP Probe if the hub monitoring server is also configured to forward events for the same situation to the Netcool/OMNIbus Probe for Tivoli EIF because OMNIbus deduplication will not detect that they are the same event.

About this task

Complete these steps to prepare your OMNIbus environment to receive SNMP alerts for situation events from Tivoli monitoring agents.

Procedure

1. Copy the Tivoli Monitoring rules file and lookup file.
   a. Locate the `mibs/sample_rules/omnibus` directory on the Tivoli Monitoring V6.2.2 or later Agents installation media.
   b. Copy these management information base files to `$OMNIHOME/probes/`*`arch`*`/` on the computer where the SNMP Probe is installed:
   `ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules`
   `ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.lookup`
2. Reference the files in the rules that the SNMP Probe is using.
   a. Open the default rules file in a text editor. The default rules file is `$OMNIHOME/probes/`*`arch`*`/mttrapd.rules` unless specified otherwise in the mttrapd properties file (Step 3).
   b. Add the lookup table reference as the first definition:

   ```
   include "<path_to_lookup_file>/
   ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.lookup"
   ```

   Table definitions must appear at the start of a rules file, before any processing statements. If you are adding this statement to mttrapd.rules, position it after the comments at the head of the file and before the first processing statement. The fully qualified filename must be enclosed in double quotes. Environment variables like %OMNIHOME% or $OMNIHOME can be used. The Linux and UNIX filename convention, with the / forward slash to delimit the path, is also used by Windows.
   c. Add the rules reference in the order in which it should be processed.

```
include "<path_to_rules_file>/
ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules"
```

This statement should be added in the rules file in the location where it should be processed. For example, if adding the include to the default mttrapd.rules file, you would want the default rules to first "Check if an SNMPv2 trap and convert to SNMPv1 style tokens". The next block of code in the default mttrapd.rules handles Generic traps. The include statement for the ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules should go after this, possibly as the last line of mttrap.rules. You will best know where to include the rules if you are familiar with the SNMP Probe and your event space.

3. Review and edit the SNMP Probe properties file:
   a. Open `$OMNIHOME/probes/`*`arch`*`/mttrapd.props` in a text editor.
   b. Set the `Protocol` property to `"UDP"` or `"ALL"`. Tivoli Monitoring SNMP alerts are sent using UDP.
   c. Set the `RulesFile` property if the default rules file for the probe is not `mttrapd.rules`.
   d. Set the `MIBDirs` property to the path where the mib files will reside.
4. Make the Tivoli Monitoring mib files available to the SNMP Probe:
   a. Locate the `mibs` directory on the Tivoli Monitoring installation media.
   b. Copy `canbase.mib` and `cansyssg.mib`, to the mib location specified in mttrapd.props by the MIBDirs property.
   c. The `canbase.mib` and `cansyssg.mib` include some common SNMP mibs. These mibs must also be available to the SNMP probe:
      RFC1155-SMI
      RFC1213-MIB
      SNMPv2-TC
      RFC-1212
      RFC-1215
      If these mibs are not already present in the location specified in `mttrapd.props` by the `MIBDirs` property, they are publicly available and can be downloaded from the Internet.
5. If you are integrating Tivoli Monitoring, Tivoli Business Service Manager, and Netcool/OMNIbus, the Netcool/OMNIbus SNMP Probe rules should also include an additional rules file, `tbsm_snmp_event.rules`, that sets the OMNIbus **BSM_Identity** attribute. The `mibs/sample_rules/omnibus/tbsm` directory on the Tivoli Monitoring Agent installation media (V6.2.2 and higher) contains the `tbsm_snmp_event.rules` file and the readme file that describes how to use it with the SNMP Probe and how to use the `itm_tbsm_update.sql` file to add the **BSM_Identity** attribute to the Netcool/OMNIbus database schema.

Results

You should now have these files installed on the probe system:

`ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules`

`ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.lookup`

`can*.mib` files that are provided on the Tivoli Monitoring installation media

What to do next

To activate the new rules and begin receiving alerts from Tivoli Monitoring agents, recycle the SNMP Probe.

Sample OMNIbus rules for SNMP alerts

The IBM Tivoli Monitoring V6.2.2 or later Agents installation media has a sample rules files that you add to the Netcool/OMNIbus SNMP Probe configuration.

Tivoli Monitoring SNMP trap mib

The `ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules` file contains a sample mapping of the IBM Tivoli Monitoring SNMP trap variables to the Default alerts.status fields in OMNIbus.

The `ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.lookup` file contains these tables:

`SituationCategory` maps the Tivoli Monitoring **`$autoSit-Category`** to OMNIbus **`@AlertGroup`**.

`SituationSeverity` maps the Tivoli Monitoring **`$autoSit-Severity`** to OMNIbus **`@Type`**: 1 - Problem; 2 - Resolution; and 13 - Information. It also changes the severity of an autoSit-Severity=0 clearing trap to 1 so that the OMNIbus generic_clear automation will correlate events.

`SituationSource` enumerates the **`$agentSit-Source`** that identifies whether the situation was an enterprise situation defined at the Tivoli Enterprise Monitoring Server or a private situation defined in the Private Situation Configuration file located in the agent installation directory, `<`*`tema_install_dir`*`>/localconfig/k`*`pc`*. This table is also use to determine event Class.

Notes on creating the @Identifier & @AlertKey

The `ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules` use the Tivoli Netcool/OMNIbus Deduplication Automation and Generic Clear Automation. These automations rely on several alert fields, including the Identifier and the AlertKey fields, each of which can be up to 255 characters. The Netcool/OMNIbus rules file standard for setting the Identifier alert field for an SNMP alert is:

```
 @Identifier = @Node + " " + @AlertKey + " " + @AlertGroup + " " + @Type + " "
 + @Agent + " " + @Manager + " " + $specific-trap
```

Because the AlertKey is included in the information that is used to construct the Identifier, you might encounter truncation problems with 255-character AlertKeys used to create your Identifier.

As implemented in the ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules:

```
@Identifier = @Node + " " + @AlertKey + " " + $autoSit-Category + " " + @Type  +
" " + @Agent + " " + @Manager + " " + $specific-trap
```

`$autoSit-Category` is an enumeration of the `@AlertGroup` (24 bytes) and is substituted for `@AlertGroup` to save 23 bytes in the final Identifier. These are the maximum field lengths of the components used to construct the Identifier:

Field	Size
@Node Max length	32
$autoSit-Category fixed length	1
@Type Max length	2
@Agent Max length	31
@Manager fixed length	13
$specific-trap fixed length	2
6 space delimiters	6

Field	Size
Total	87

This leaves 168 characters for the @AlertKey (255-87=168). If @AlertKey is defined as $agentSit-Name + " (" + $sitDisplayItem + ")", then `$sitDisplayItem` must be less than 133 characters (168-35=133).

Field	Size
agentSit-Name	32
space delimiter	1
parentheses	2
Total	35

A best practice is to limit `$sitDisplayItem` to 128 characters to maintain consistency with the IBM Tivoli Monitoring EIF probe rules. The sample rules enforce this limit using

```
$sitDisplayItem=substr($sitDisplayItem, 1, 128)
```

Situations written for attribute groups (such as Event Log) that generate pure events can be deduplicated by using the `$agentSit-Name`, but many might require additional information to uniquely identify the event. Use the `$sitDisplayItem` attribute to construct this additional data. The AlertKey will then be

```
$agentSit-Name + " (" + $sitDisplayItem + ")"
```

Use case statements based on the `$agentSit-Table` field to identify all events based on a specific table.

Use case statements based on the `$agentSit-Name` if individual situations need unique `$sitDisplayItems`.

The **extract** command can be used to extract the value of any of the name value pairs from the `$sitAttributeList` using regex pattern matching. An example is provided in the Sample rules for `agentSitPureEvent` traps based on the NTEVTLOG `$agentSit-Table`.

```
$sitDisplayItem=extract($sitAttributeList,"Description=.(.+).;.*?")
```

This command extracts the value of the Description key and removes the quotes.

Compatibility notes

@ExtendedAttr

OMNIbus V7.2 and greater defines the `@ExtendedAttr` column in the ObjectServer. The **nvp** functions are provided to allow manipulation of name-value pairs in the `@ExtendedAttr` alert field. The sitAttributeList varbind is formatted to allow direct mapping into the `@ExtendedAttr`, but this function is commented out to allow the rules to parse when the MTTRAPD probe connects to an OMNIbus ObjectServer V7.0 or V7.1. Uncomment the two lines in the ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules file that set `@ExtendedAttr` if you are forwarding events to OMNIbus V7.2 or greater.

```
#           @ExtendedAttr = $sitAttributeList
```

@Class

The @Class alert field is used to associate Tivoli Netcool/OMNIbus Tools with Events displayed in the Tivoli Netcool/OMNIbus EventList.

For Tivoli Netcool/OMNIbus 7.2x and below, see the Netcool/OMNIbus documentation for more information on creating and editing classes. By default, these class values are not defined in your ObjectServer.

Setting @Class to a value that is *not* defined in the OMNIbus ObjectServer causes no problems, but if you prefer to not set the @Class, uncomment this line in the ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules file to clear the @Class field before the event is forwarded to OMNIbus.# `@Class = ""`

Enabling OMNIbus heartbeat automation

For Tivoli Enterprise Monitoring Agents that send situation events as SNMP alerts or EIF events to Netcool/OMNIbus, you can enable OMNIbus automation to have an event sent when the EE_HEARTBEAT is overdue. EE_HEARTBEAT life cycle status events are sent to the receiver at regular intervals to confirm that the monitoring agent is running and alerts can reach their destination.

About this task

HEARTBEAT events from SNMP and EIF are displayed on the OMNIbus event console as they arrive and the count is incremented as new events arrive from the agent.

The `itm_heartbeat.sql` file contains sample automations for processing the Autonomous Agent Heartbeats for both EIF and SNMP. Run the SQL file to enable the automation.

Procedure

1. Copy `itm_heartbeat.sql` from the Tivoli Monitoring Agent DVD `mibs/sample_rules/omnibus` directory.
2. Place the copy in the Netcool/OMNIbus installation path and run the following command:
   - Windows Where "*user*" is a valid user name, "*password*" is the corresponding password, and "*server*" is the ObjectServer name

     ```
     %NCHOME%\bin\redist\isql.exe -U "user" -P "password" -S "server"
     < itm_heartbeat.sql
     ```
   - Linux UNIX Where "*servername*" is the ObjectServer name, "*username*" is a valid user name, and "*psswrd01*" is the corresponding password

     ```
     $NCHOME/omnibus/bin/nco_sql -server servername -user username
      -password psswrd01 < itm_heartbeat.sql
     ```

Results

After the OMNIbus automation is installed, the automation registers the heartbeats from managed systems as they arrive. Individual heartbeats are no longer displayed and counted in the event console but, if an expected heartbeat is overdue, the automation raises a "Heartbeat missing" alert with:

```
Summary = 'Heartbeat Missed for:' + heartbeat_missed.Node +
' last received at ' + to_char(heartbeat_missed.LastOccurrence)
```

What to do next

The default interval for sending the EE_HEARTBEAT status 15 minutes. You can adjust the value by modifying the interval attribute for the heartbeat status event in the trapcnfg.xml file for SNMP alerts and in the eifdest.xml file for EIF events configuration file.

Especially with SNMP, one missed heartbeat does not necessarily indicate a problem, thus the default is to raise an alert after the heartbeat is overdue: (2 x heartbeat interval) + 2 minutes. You can edit this in the itm_heartbeat.sql with the entry,

```
-- 2 heartbeats plus 2 minutes grace before agent missed
    set time_of_expiry = (new.ExpireTime * 2 * 60 + 120) + getdate();
```

For example, add two more minutes and the setting looks like this:

```
-- 2 heartbeats plus 4 minutes grace before agent missed
    set time_of_expiry = (new.ExpireTime * 2 * 60 + 240) + getdate();
```

EIF events

Send private situation events directly from a Tivoli Monitoring Agent to an EIF receiver without going through the Tivoli Enterprise Monitoring Server.

EIF event configuration

Configure a monitoring agent and a local EIF event configuration XML file to emit life cycle events or private situation events, or both, to one or more EIF receivers such as the IBM Tivoli Enterprise Console event server or Netcool/OMNIbus Probe for Tivoli EIF.

Restriction: EIF is not supported on iSeries® to send events directly from the iSeries agents.

Usage note: In code examples below use the following substitutions:

- `<install_dir>` is the directory where IBM Tivoli Monitoring is installed
- `<pc>` is the agent 2 letter product code in lowercase
- `<PC>` is the agent 2 letter product code in uppercase

EIF event configuration

The default setting of the IRA_EVENT_EXPORT_EIF environment variable is Y, which causes the EIF emitter to start during agent startup. The following files must be present and configured on the system where the monitoring agent is installed before EIF event forwarding can take place:

- A private situation configuration file to define the situations that generate events when the comparison criteria evaluate to true. The private situations are started and stopped as part of the agent startup and shutdown procedure.
- An event destination configuration file to define the event listeners for receiving the EIF emitted events.

Additionally, you can have an event mapping file to control the event data being sent to the EIF receiver.

Agent parameters

IRA_EVENT_EXPORT_EIF=Y parameter in the agent environment file is set to enable the EIF event export facility. Change the value to **N** to disable the facility.

IRA_EIF_DEST_CONFIG=<filename> parameter in the agent environment file is set to specify the location of the EIF destination configuration XML file. The default is `<install_dir>/localconfig/<pc>/<pc>_eventdest.xml`.

IRA_LOCALCONFIG_DIR parameter in the agent environment file can be set to specify a different directory path for the EIF destination or optional event mapping file, or both. The default is `<install_dir>/localconfig/<pc>`.

IRA_EIF_MSG_LOCALE=en_US parameter in the agent environment files is set to American English by default. For agents that support globalized message text for the message slot in the generated event using a predefined mapping file and language resource bundles, the default language locale can be specified.

Agent EIF event destination configuration XML specification

The EIF event destination XML file is used to specify the event destination server or servers and their configurations. The root element is <EventDest> and contains <Destination> and <Server> elements, as well as the optional <StatEvent> element to configure the EIF heartbeat interval: how often the agent sends a heartbeat event to the EIF receiver.

The event destination configuration file resides in the following default location:

Windows `<install_dir>\localconfig\<pc>\<pc>_eventdest.xml`

Linux UNIX `<install_dir>/localconfig/<pc>/<pc>_eventdest.xml`

z/OS `RKANDATV`, with member name `<PC>EVDST`

Agent EIF event mapping configuration XML specification

The optional EIF event mapping file configuration XML file can be used to customize the EIF events that are generated. If no event mapping file is provided, events are formatted by generic mapping. Event mapping files can be product provided or user defined. User defined event maps, if any, have precedence over product provided maps. The name and location of the predefined event mapping file, if any:

Windows `<install_dir>\TMAITM6\EIFLIB\k<pc>.map` for 32-bit agents; `<install_dir>\TMAITM6_x64\EIFLIB\k<pc>.map` for 64-bit agents.

Linux UNIX `<install_dir>/<platform>/<pc>/tables/EIFLIB/k<pc>.map`

z/OS `RKANDATV`, with member name `K<PC>MAP`.

If you create a user defined event mapping file, store it in the following location:

Windows `<install_dir>\localconfig\<pc>\<pc>_eventmap.map`

Linux UNIX `<install_dir>/localconfig/<pc>/<pc>_eventmap.map`

z/OS `RKANDATV`, with file name <PC>EVMAP

Sample event map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<itmEventMapping:agent>
   <event_mapping>
      <situation name="Flipper*" mapAllAttributes="Y">
         <class name="ITM_ABC"/>
            <slot slotName="msg">
               <literalString value="The time now is
```

```
                $Local_Time.Timestamp.TIMESTAMP$ on $hostname$"/>
            </slot>
        </situation>
    </event_mapping>
</itmEventMapping:agent>
```

Heartbeat event of agent's online status sent to the EIF event receiver

The EventDest configuration XML file has an optional element for specifying a heartbeat interval. After each interval, the agent status is tested and the result then sent as online-offline status to the EIF receiver (or receivers) specified in the EventDest file.

Tivoli Enterprise Monitoring Agents installed before Version 6.2.2 Fix Pack 1

When a Tivoli Monitoring Version 6.2.2 Fix Pack 1 or later OS agent of has been installed, any Tivoli Enterprise Monitoring Agents installed on that computer are eligible to use the autonomous EIF event forwarding feature, even if they were installed prior to Version 6.2.2 Fix Pack 1. Be aware, however, that monitoring agents that were installed before Version 6.2.2 Fix Pack 1 need some files that were not included in the agent install bundle but that are part of the application support provided for a Tivoli Enterprise Monitoring Server installation: the baroc file, optional event mapping file, and resource bundle files. For these earlier version agents to use the EIF facility to forward events, the following steps must be taken:

1. Install the earlier version agent in a monitoring server environment to access the baroc and optional event mapping files. The agent predefined baroc and optional event mapping files can be found in the <install_dir>/CMS/TECLIB or <install_dir>/CNPS/teclib directory.
2. Copy the provided k<pc>.map event mapping file, if any, to the EIFLIB directory of the agent installation.
3. If the Tivoli Enterprise Console event server is used as an event receiver, copy the baroc file for each agent to the system where the event server is installed. Compile and load the baroc on the event server.

Agents running within the hub Tivoli Enterprise Monitoring Server Address Space on z/OS

On z/OS systems it is possible to configure agents to run within the same address space of the hub monitoring server. Because the EIF event forwarder function (OTEA) can also be enabled at the hub monitoring server, some precautions must be taken to avoid the cross interference between the Event Forwarder at the hub monitoring server and the EIF event export directly from the monitoring agent. One area that can have potential overlap is a custom event mapping file. Currently on the hub monitoring server, users can code their own event mapping (in addition to those that can be created using the Tivoli Enterprise Portal Situation editor and stored in a monitoring server table). The name of these map files must be in the form, **Q<xx>MAP** (where <xx> is any two alphanumeric characters) and reside in the RKANDATV dataset. In order to support user defined event mapping files for autonomous agents, which also reside in the RKANDATV data set, a different naming convention must be adopted. For user defined event mapping files for autonomous agents, the file naming convention is **<pc>EVMAP** (where <pc> is the 2 letter agent product code).

EIF event mapping XML specification

The EIF event map is an XML file that specifies how the events for one or more private situations are to be translated. Create a custom event mapping file to modify the data being sent to the EIF receiver.

Event mapping file format

```
<itmEventMapping:agent>
  <id>xx</id>
  <version>n.n</version>
  <event_mapping>
    <situation>
      <slot>
        <mappedAttribute/>
          or
        <mappedAttributeEnum/>
          or
        <literalString/>
      </slot>
      :    one or more slot tags
    </situation>

or

    <attributeTable>
      <slot>
        <mappedAttribute/>
          or
        <mappedAttributeEnum/>
          or
        <literalString/>
      </slot>
      :    one or more slot tags
    </attributeTable>
  </event_mapping>
</itmEventMapping:agent>
```

Elements

XML tags are case-insensitive. All other parameters are case-sensitive.

Note: You can use the `EIF.xsd` schema file, located in the `samples/PrivateConfigSamples` directory on the Agent DVD image, to validate EIF event mapping XML files. However, the schema file requires the XML tag names to be in uppercase.

<itmEventMapping:agent>
: itmEventMapping:agent is the root element identifying this as an event mapping definition for the monitoring agent.

<id> Syntax:

`<id>`*pc*`</id>`

ID is the two character product code, such as “UX” for the UNIX OS agent. For user defined event maps, it is recommended to use “99” as the ID.

<version>
Syntax:

`<version>`*nnnn*`</version>`

Optional. Use this element to specify the version of the event mapping file.

<valueList>

Syntax:

```
<valueList name="valueListName">
```

Optional. Use the valueList element to define a value list of one or more value items where *valueListName* is the name of the list.

<valueItem>

Syntax:

```
<valueItem name="item_value">
```

This element is required when a valueList is being defined. ValueItem specifies a valid item value for the named valueList.

<event_mapping>

Syntax:

```
<event_mapping>
```

The event_mapping element encloses a group of mapping entries.

<situation>

Syntax:

```
<situation name="situation_name" [mapAllAttributes="Y"]
```

The situation element specifies a DM mapping entry whose key is situation_name. The situation_name string can contain wildcard characters (* asterisk and ? question mark) except for in the first character position.

mapAllAttributes="Y" instructs the EIF event forwarder to construct the EIF event slots like the generic mapping except the slots explicitly specified via the <slot> tag within this mapping entry. This attribute is useful for cases where only a few of the slots in the event must be customized (such as the msg slot). This alleviates the need to explicitly specify every slot to be included in the EIF event.

<attributeTable>

Syntax:

```
<attributeTable name="attribute_table_name"
 [truncated="Y"] [freeSpace="nnnn"]
```

truncated="Y" causes "ITM_Agent: Private Situation: Truncated" to be assigned to the "source" slot of the EIF event instead of "ITM Agent: Private Situation". This is an indicator that not all the attributes in the event data can fit in the EIF event due to size limitations, as defined by the event mapping generator.

freeSpace="*nnnn*" is a value determined by the event map generator as the maximum size available in the EIF event buffer after all the slots defined in this event map are built. Its value is used by the EIF event emitter to determine how much raw event data to include in the situation_eventdata slot.

<class>

Syntax:

```
<class name="eif_class_name" [valueList="valueList_name"]
 [defaultClass="default_eif_class_name"]>
```

name= specifies the EIF class name to be used for the generated EIF event. The eif_class_name string can contain a substitution variable (attribute name) to generate EIF class names that are dynamic depending on the value of the named attribute during run time. See "Dynamic EIF classname" on page 385.

valueList= specifies a `valueList` to be searched for dynamic EIF class name generation.

defaultClass= specifies the default EIF class name to be used for the EIF event if the `eif_class_name` string contains a substitution variable but the value of the attribute has no match in the specified `valueList`.

<slot> Syntax:

```
<slot name="slot_name">
```

Optional. Define a slot in the EIF event. The name of the slot is the *`slot_name`*.

<mappedAttribute>
Syntax:

```
<mappedAttribute name="attribute_name" [multiplier="nnn"]>
```

Optional. Specify the value source for the slot being defined. This is the value of the attribute with the name `attribute_name` in the event data, if available. Otherwise, a null value is used. If the `multiplier=` attribute is specified and the value of the attribute is numeric, the value assigned for the slot is the attribute value multiplied by the number specified.

<mappedAttributeEnum>
Syntax:

```
<mappedAttributeEnum name="attribute_name">
```

Optional. MappedAttributeEnum is similar to the mappedAttribute tag except that if the attribute is defined as an enumeration in the attribute file, the enumerated display text is used as the slot value instead of the raw attribute value. If no enumerated display text is defined that matches the attribute value, the raw attribute value is used.

<literalString>
Syntax:

```
<literalString value="text">
```

Optional. Use the text as the value for the slot being defined. When defining a "msg" slot, you can specify variable substitution within the text (described next).

Custom msg slot

If the value of the msg slot is defined as a literal string (<literalString> element), it can include substitution variables. Substitution variables are designated by the `$variable$` syntax. When formatting the msg slot, the EIF event forwarder replaces the `$variable$` symbol with its replacement value.

Valid variables:

$AttrGroup.Attribute$
Attribute substitution requires a fully qualified name (i.e. both attribute group and attribute name separated by a period). The variable token will be replaced by the value of the named attribute in the event data. If the named attribute cannot be found in the event data, a null string will be used.

$AttrGroup.Attribute.TIMESTAMP$
This is the same as the $AttrGroup.Attribute$ syntax but with a **.TIMESTAMP** suffix qualifier. This is an indication to the EIF event forwarder that the attribute value is a time stamp (defined as a timestamp type in the attribute file) and should be formatted as a displayable timestamp format: MM/DD/YYYY HH:MM:SS. If the attribute value is not a valid timestamp, the raw attribute value will be used.

$slotname$
Event slot substitution replaces the variable token with the value of the named slot after event mapping has been performed.

The following is an example taken from a predefined event mapping file where the msg slot is customized for DM parity.

```
<slot slotName="msg">
  <literalString value="Distributed Monitoring $sub_source$/$monitor$
  on host $hostname$ $NT_LogicalDisk.Timestamp$"/>
</slot>
```

If the value for the `sub_source` and `monitor` slots have values “tmpdisk” and “Disk Read Bytes/sec”, the msg slot text is similar to this example:

```
Distributed Monitoring tmpdisk/Disk Read Bytes/sec on host
 elaix04 08/14/2009 10:23:11
```

Dynamic EIF classname

The EIF event class name is defined by the **name=** attribute of the **<class>** element. The EIF class name string can contain a substitution variable for dynamic generation of the EIF class name. The substitution variable can appear anywhere within the EIF class name string, except at the beginning. The substitution variable has a syntax of `$attributeGroup.attribute$`. At run time, the EIF event forwarder searches the designated `valueList`, if one exists, for the value of the attribute specified in the substitution variable. If the attribute value is found in the `valueList`, the variable (and delimiting $ dollar sign) is replaced by the attribute value (after being normalized) in the EIF class name string. If no match is found in the `valueList` or the designated `valueList` is not defined, the EIF class name defined in the **defaultClass=** attribute is used as the EIF class name for the event. If no **defaultClass=** is specified, the variable in the EIF class name is replaced with a null string.

If the variable references a numeric attribute, no scaling or precision operation is performed. The string representation for the numeric field in the situation event record will be used without any adjustment. If the variable references an enumerated attribute, any text representation of the enumeration, is used as the value for the variable.

When the situation is not true (status is not “Y”), the situation status record does not contain any event attribute data. Consequently, there is no way to determine the value of any substitution variablea in the class name. The EIF event forwarder uses the **defaultClass=** attribute if one is specified. Otherwise, it uses the EIF event class of the EIF event last sent for the same situation name.

This is the relevant part of a sample event mapping definition uses to map a situation event “Test_Syslog” to a set of EIF events based on the value of the “Message_Number” attribute.

```
<situation name="Test_Syslog">
  <class name="SAP_Syslog_$R/3_System_Log.Message_Number$"
  valueList="SyslogIDList" defaultClass="SAP_Syslog_Default" />
  :
  :
</situation>
```

This example has a “SyslogIDList” value list with valueItems AB0, AB1, A08, BV7, EAS, and R45 and a “Test_Syslog” situation that monitors for message IDs AB0, AB1, AB2, BV7, and BV8, The “Test_Syslog” situation evaluates to true for each of these message ids. The generated EIF events are of the following classes:

1. AB0: SAP_Syslog_AB0 x
2. AB1: SAP_Syslog_AB1
3. AB2: SAP_Syslog_Default
4. BV7: SAP_Syslog_BV7
5. BV8: SAP_Syslog_Default

Normalizing the attribute value
A variable within the EIF event class name can reference any valid attribute in the event whose value might contain characters that are not valid for use in an EIF event class name. Before performing variable substitution in the event class name, the EIF event forwarder replaces any UTF-8 multi-byte characters and invalid characters with a single _ underscore. For example the white space characters, < > ( ) & /, are replaced by the _ underscore character.

Example

```
<itmEventMapping:agent
  xmlns:itmEventMapping="http://www.ibm.com/tivoli/itm/agentEventMapping"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/tivoli/itm/agentEventMapping
  agentEventMap.xsd">

  <id>NT</id>
  <version>6.2.0</version>
  <event_mapping>
    <situation name="NT_LDDBPS*">
      <class name="w2k_LogDskDskBytesPerSec"/>
      <slot slotName="source">
        <literalString value="SENTRY"/>
      </slot>
      <slot slotName="probe">
        <literalString value="DskBytesPerSec"/>
      </slot>
      <slot slotName="probe_arg">
        <mappedAttribute name="NT_Logical_Disk.Disk_Name"/>
      </slot>
      <slot slotName="collection">
        <literalString value="w2k_LogicalDisk"/>
      </slot>
      <slot slotName="monitor">
        <literalString value="Disk Bytes/sec"/>
      </slot>
      <slot slotName="units">
        <literalString value="(per second)"/>
      </slot>
      <slot slotName="value">
        <mappedAttribute name="NT_Logical_Disk.Disk_Bytes/Sec"/>
      </slot>
      <slot slotName="effective_value">
        <mappedAttribute name="NT_Logical_Disk.Disk_Bytes/Sec"/>
      </slot>
      <slot slotName="msg">
        <literalString value="Distributed Monitoring $sub_source$/Disk
        Bytes/sec on host $hostname$ $NT_Logical_Disk.Timestamp.TIMESTAMP$"/>
      </slot>
    </situation>
  </event_mapping>
</itmEventMapping:agent>
```

Sample EIF files are provided on the Tivoli Monitoring Agent installation media in the `PrivateConfigSamples/EIF` directory.

EIF event destination configuration XML specification

Use the EventDest, Server, and Destination elements in the EIF destination XML file to configure the destination servers for the EIF events sent by the monitoring agent.

Elements

The elements and their attributes are not case-sensitive. For example, you can enter EVENTDEST=, EventDest=, or eventdest=.

<EventDest>

EVENTDEST is the root element identifying this as an event destinations definition for the monitoring agent.

<Destination>

Start of an event destination definition. Specify the destination index. Optional attributes enable you to specify the destination type, a default server, the maximum cache file size, and the option to clear the cache file on restart.

id= Required. Destination index, from 0 to 999. Default: **0**

type= Optional. Destination type: T=IBM Tivoli Enterprise Console; M=Netcool/OMNIbus. The maximum event size generated will be 4K and 32K for type=T and type=M, respectively. Default: **T**

default= Optional. The server entered here is designated as the default destination. Default: **N**

clear_cache= Optional. Use this attribute to specify whether the existing EIF cache file should be cleared when the destination is instantiated. clear_cache="Y" will cause the EIF event cache file to be cleared. On z/OS systems, the EIF event cache is always cleared because z/OS EIF supports only an in core event cache. Default: **Y**

max_cache_size= Optional. Specifies the maximum event cache physical file size in kilobytes. Default: **4096**

stat= Optional. Specify whether the destination shall receive life cycle events. Default: **N**

master_reset= Optional. Specify whether a master reset event shall be sent during agent startup. Default: **N**

<Server>

Defines the event servers for the destination: one primary and up to seven secondary servers. Specify each event server hostname or IP address and the port. The first <server> definition is the primary listener. Any additional <server> definitions are backup servers.

location= Specifies the hostname or IP address of the event listener.

port= Optional. Specifies the listening port of the IBM Tivoli Enterprise Console event server or Netcool/OMNIbus Probe for Tivoli EIF. The default port number for the IBM Tivoli Enterprise Console event server is **5529**. However, a setting of **port=0** can be used for a IBM Tivoli Enterprise Console on Linux or operating systems such as UNIX and indicates that the event listener is using portmapper. The default port number for the EIF Probe is **9998**. If no value is specified, the port number is defaulted to **0** in the server location definition.

SSL= Optional. Specify whether events will be sent encrypted over a Secure Socket Layer (SSL) connection or events will be sent unencrypted over a non-SSL connection. SSL connections are only supported for Netcool/OMNIbus event destinations (type="M"). If enabled for a IBM Tivoli Enterprise Console event destination (type="T"), a default non-SSL connection is used. Specify Y to enable, and N to disable. Default: **N**.

<StatEvent>

Optional. Use the StatEvent element to send the online or offline status of the agent to the event server. By default, heartbeat monitoring is disabled.

name= Specifies the name of the heartbeat event.

interval= Optional. The interval, expressed in minutes, at which the heartbeat event is sent. A zero interval means no heartbeats are to be sent. Default: **15**.

Examples: Both stanzas name the heartbeat event, EE_HEARTBEAT. The first stanza specifies a 5-minute interval and the second stanza disables the heartbeat event.

```
<StatEvent name="EE_HEARTBEAT" interval="5"/>
```

```
<StatEvent name="EE_HEARTBEAT" interval="0"/>
```

The destination server receives an EIF event with a class name of "ITM_Heartbeat" containing a slot called "interval" whose value is the heartbeat interval. SNMP events received contain an attribute "AlertGroup" whose value is "ITM_Heartbeat" and an attribute "HeartbeatInterval" whose value is the heartbeat interval. You can customize the provided heartbeat rules or write your own to handle the heartbeat events.

Note: Multiple EE_HEARTBEAT events are not supported. If multiple destinations are configured to receive EE_HEARTBEAT, the same EE_HEARTBEAT is sent to each destination.

Example

The following example is an event destination configuration file containing one event destination:

```
<EventDest>
 <Destination  id="0" type="M" master_reset="Y" stat="Y" default="Y">
  <Server location="server.ibm.com" port="9998" />
 </Destination>
 <StatEvent name="EE_HEARTBEAT" interval="5" />
</EventDest>
```

The following example is an event destination configuration file containing two event destinations:

```
<EventDest>
 <Destination id="0" type="M" default="Y" master_reset="Y" stat="Y">
  <Server location="omniserver.ibm.com" port="9998" />
 </Destination>
 <Destination id="1" type="T" default="Y" master_reset="Y" stat="N">
```

```
  <Server location="tecserver.ibm.com" port="5529" />
 </Destination>
 <StatEvent name="EE_HEARTBEAT" />
</EventDest>
```

Here, the second destination will not receive life cycle events because stat parameter is set to "N".

The following example is an event destination configuration file containing two event destinations defined to use SSL connections:

```
<EventDest>
    <Destination id="0" type="M" default="Y" master_reset="Y" stat="Y" >
        <Server location="server1.ibm.com" port="9998" SSL="Y" />
    </Destination>
    <Destination id="1" type="M" default="Y" master_reset="Y" stat="Y" >
        <Server location="server2.ibm.com" port="9998" SSL="Y" />
        <Server location="server3.ibm.com" port="9998" SSL="N" />
    </Destination>
</EventDest>
```

Here, the second destination illustrates that the same SSL value is not required when multiple server locations are defined where a SSL connection is used for the primary server and a non-SSL connection is used for the secondary server.

Tip: Sample EIF files are provided on the Tivoli Monitoring Agent installation media in the `PrivateConfigSamples/EIF` directory.

Common slots for EIF emitted events

Review the descriptions of the set of common slots to understand the private situation event information at the EIF receiver.

All emitted EIF events will have a common set of slots in addition to the slots from the event attribute data. All attributes, except hidden attributes, that are defined in the attribute table used by the event are included in the emitted event (subject to the total event and slot size limitation). The set of common slots are explained in the following table.

Table 38. Set of common slots for emitted EIF events.

Slots	Values and meaning
adapter_host	Base EVENT class attribute. Same as hostname (see below). This is application-specific data related to the event, if any.
appl_label	Use to indicate the source of the event is from a private situation or agent online status. The value has the following syntax: **`source : sit_type : event_type`** where **`source`** is always "A" for agent **`sit_type`** is "P" for private situation or "E" for enterprise situation **`event_type`** is "S" for situation events or "L" for life cycle status events For example, A : P **Note:** For enterprise situation events, the appl_label value is not set. Thus, there is no appl_label="A:E:S".

Table 38. Set of common slots for emitted EIF events. (continued)

Slots	Values and meaning
cms_hostname	Not used or null for agent emitted event. **Note:** Because the Tivoli Enterprise Monitoring Server is not used for EIF emitted events, the IBM Tivoli Enterprise Console event server logs no error message in the event synchronization synch_trace.log file after a private situation event has been closed.
cms_port	Not used or null for agent emitted event.
fqhostname	Base EVENT class attribute that contains the fully qualified hostname, if available.
hostname	Base EVENT class attribute that contains the TCP/IP hostname of the managed system where the event originates, if available.
integration_type	Indicator to help performance. • `N` for a new event, the first time the event is raised • `U` for update event, subsequent event status changes
master_reset_flag	Master reset indicator set for master reset events. Value is NULL for all other events: • `R` for agent restart • Otherwise, NULL
msg	Base EVENT class attribute that contains the situation name and formula, without the use of customization.
origin	Base EVENT class attribute contained in the TCP/IP address, if available, of the managed system where the event originates. The address is in dotted-decimal format.
severity	Base EVENT class attribute that contains the resolved severity.
situation_displayitem	Display item of associated situation, if available.
situation_eventdata	Raw situation event data starting from the second event data row, if any. Event data attributes are in key-value pair format. The event data can be truncated because of the total event size and slot size limit, which is 2 KB.
situation_group	One or more situation group names (up to 5) that the situation is a member of.
situation_fullname	Display name of situation if one was defined for the private situation.
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.
situation_status	Current status of the situation event: "Y" situation is true "N" situation is false "P" situation stopped
situation_time	Timestamp of the situation event.
situation_type	Situation event type **S** for sampled event; **P** for pure event.
situation_thrunode	Managed system name of the agent.
source	Base EVENT class attribute that contains **ITM Agent: Private Situation** or **ITM Agent: Private Situation:Truncated**

Table 38. Set of common slots for emitted EIF events. (continued)

Slots	Values and meaning
sub_origin	Base EVENT class attribute that contains the value of display item, if any.
sub_source	Base EVENT class attribute that contains the origin managed system name for the associated situation.

EIF life cycle event

In addition to emitting situation start or stop and status events, the Event Integration Facility event emitter generates additional life cycle events that are not private situation related.

Life cycle events are emitted when the agent or situation changes state, as shown in the *EIF life cycle events table*. The heartbeat event is a life cycle event that needs no state change to be emitted: it is sent at regular intervals to confirm that the agent is running.

Table 39. EIF life cycle events.

Event	Meaning
EE_AUTO_ENTER	The situation has entered autonomous mode operation.
EE_AUTO_EXIT	The situation has exited autonomous mode operation.
EE_CONFIG_UPDATE	Generated when the configuration file is pulled from a server.
EE_HEARTBEAT	Agent heartbeat.
EE_TEMS_CONNECT	The agent is connected to the monitoring server.
EE_TEMS_DISCONNECT	The agent is disconnected from the monitoring server.
EE_TEMS_RECONNECT_LIMIT	Agent reconnect to the monitoring server limit has been exceeded.
EE_SIT_STOPPED	The situation is stopped. This is optional. **Note:** The situation_status slot for EIF events sends “P” automatically for a stopped situation.

All life cycle EIF events are ITM_StatEvent, which is a derived class of Event, with the following slot values:

Table 40. EIF life cycle event ITM_StatEvent class slot values.

Slot	Value
source	“ITM Agent: Status Event”
appl_label	“A:E:L” for stopped enterprise situation; “A:P:L” for all others.
hostname	Hostname or IP address of agent machine
fqhostname	Fully qualified hostname if available
origin	The IP address of the agent computer
situation_name	Life cycle status value, such as EE_AUTO_ENTER. If the life cycle event is a EE_SIT_STOP, the situation_displayitem slot contains the situation name being stopped.
situation_time	Datetime the life cycle event occurred
date	Date of life cycle event

Table 40. EIF life cycle event ITM_StatEvent class slot values. (continued)

Slot	Value
severity	“HARMLESS”
msg	Message describing the life cycle event

EIF heartbeat event

The Event Integration Facility event destination XML file can include a StatEvent element to send the monitoring agent's online or offline status to the event server. You can customize the provided heartbeat rules or write your own to handle the heartbeat events.

The destination server receives an EIF event with a class name of “ITM_Heartbeat” containing a slot called “interval” whose value is the heartbeat interval. SNMP events received contain an attribute “AlertGroup” whose value is “ITM_Heartbeat” and an attribute “HeartbeatInterval” whose value is the heartbeat interval. The situation_eventdata slot is also set to the heartbeat interval.

The IBM Tivoli Enterprise Console **ITM_Heartbeat** class is available for customizing the heartbeat rule. The class is in the `om_tec.baroc` file that gets installed with the event synchronization on the IBM Tivoli Enterprise Console event server. It is on the IBM Tivoli Monitoring Tools DVD. Status events are kept separate from situation events so that rules can be written to apply only to the specific class or type.

Example: The ITM_Heartbeat EIF event has a 1-minute interval (`interval='1';` and `situation_eventdata='1';`) and is characterized as a Heartbeat Event:

```
ITM_Heartbeat;
 interval='1';
 source='ITM Agent: Heartbeat Event';
 sub_source='EE_HEARTBEAT';
 situation_name='**';
 situation_origin='SuperServer:TEST';
 situation_time='09/30/2009 09:03:24.000';
 situation_eventdata='1';
 appl_label='A:P:L';
 hostname='SuperServer.raleigh.ibm.com';
 fqhostname='SuperServer.raleigh.ibm.com';
 origin='9.25.111.201';
 severity='HARMLESS';
 date='09/30/2009';
 msg='Heartbeat Message';END
```

Master reset event

A master reset event can be configured to be sent when the monitoring agent is recycled. Upon receiving the master reset event, the included Netcool/OMNIbus or IBM Tivoli Enterprise Console rule closes all the opened events from this particular agent and its subnodes.

Table 41. Master reset event content

Slot	Value
source	“ITM Agent: Private Situation”
appl_label	“A:P:S”
master_reset_flag	“R”
hostname	Hostname or IP address of agent machine

Table 41. Master reset event content (continued)

Slot	Value
fqhostname	Fully qualified hostname if available
origin	ip address of agent machine
situation_name	"**"
situation_origin	Manage system name of agent
situation_time	Datetime the life cycle event occurred
date	Date of event
situation_status	"N"
severity	"MINOR"
msg	Message describing that agent has been restarted. er_reset_flag

Sending private situation events by using TLS/SSL communication

You can now send your private situation events to a Netcool/OMNIbus EIF receiver probe using TLS/SSL communication. The destination Netcool/OMNIbus Probe for Tivoli EIF must be at version 12.0 or later.

Complete the following steps to send private situation events by using TLS/SSL communication:

1. For any monitoring agent, define one or more private situations in the agent's situation XML file. See "Private situations" on page 333.
2. Define one or more Netcool/OMNIbus event destinations (type="M") in the monitoring agent's event destination XML file. Specify SSL="Y" for the associated <Server> element. See "EIF event destination configuration XML specification" on page 387.

   For more information about configuring TLS/SSL in Netcool/OMNIbus, see "Configuring an EIF receiver application for SSL" in the *IBM Tivoli Netcool/OMNIbus Event Integration Facility Reference*.
3. Edit the monitoring agent's environment file, where *pc* is the two-character product code:

   Windows *install_dir*\TMAITM6\k*pc*cma.ini.

   Linux UNIX *install_dir*/config/*pc*.ini. For system monitoring agents, the configuration file is *pc*.environment.

   z/OS member name K*PC*ENV in &hilev.&rte.RKANPARU.

   Set the following environment variables in the monitoring agent's environment file:

   Note: The indicated environment variable settings apply to every secure connection that the agent establishes with all target destinations (such as the monitoring server and the Warehouse Proxy agent) and not just the TLS/SSL connection established with the destination Netcool/OMNIbus EIF probe.

   - IRA_EVENT_EXPORT_EIF=Y (Default)
   - KDEBE_FIPS_MODE_ENABLED=Y or N (Default)

Specify the comparable value as defined for *channel_name*SSLFIPSMode=ON|OFF in the EIF probe's configuration file. For example, if *channel_name*SSLFIPSMode=ON, then set KDEBE_FIPS_MODE_ENABLED=Y.

- ITM_AUTHENTICATE_SERVER_CERTIFICATE=Y or N (Default)

  Specify the comparable value as defined for *channel_name*SSLRequireClientAuthentication=ON|OFF in the EIF probe's configuration file. For example, if *channel_name*SSLRequireClientAuthentication=ON, then set ITM_AUTHENTICATE_SERVER_CERTIFICATE=Y.

  Enabling server certificate authentication ensures that the EIF probe is a trusted entity because it is required to present a CA-signed digital certificate.

  Note: Enabling server certificate authentication for the monitoring agent means that any secure connection initiated by the agent requires that all target destinations (such as the monitoring server and the Warehouse Proxy agent) to present a valid CA-signed digital certificate in order for the connection to be established.

4. IBM Tivoli Monitoring Version 6.3 and higher support TLSv1.1 and TLSv1.2 in addition to SSL V3 and TLSv1.0. The default ciphers supported for each protocol are:

   TLSv1.0 and TLSv1.1 (in preference order):

   - TLS_RSA_WITH_AES_128_CBC_SHA
   - TLS_RSA_WITH_AES_256_CBC_SHA
   - SSL_RSA_WITH_3DES_EDE_CBC_SHA

   TLSv1.2 (in preference order):

   - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
   - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
   - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
   - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
   - TLS_RSA_WITH_AES_128_GCM_SHA256
   - TLS_RSA_WITH_AES_256_GCM_SHA384
   - TLS_RSA_WITH_AES_128_CBC_SHA256
   - TLS_RSA_WITH_AES_256_CBC_SHA256
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
   - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
   - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

   To override the TLSv1.0, 1.1, or 1.2 cipher specs, use the correlating environment variable, GSK_TLSV10_CIPHER_SPECS, GSK_TLSV11_CIPHER_SPECS, or GSK_TLSV12_CIPHER_SPECS, to provide a list of comma separated ciphers. This override action is different than overriding the SSL V3 cipher list. Refer to the relevant TLS RFCs for the cipher spec name. In the following example the ciphers are separated by commas with no other spaces appearing on one line:

   ```
   GSK_TLSV11_CIPHER_SPECS=TLS_RSA_WITH_DES_CBC_SHA,TLS_RSA
   _WITH_3DES_EDE_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_
   RSA_WITH_AES_256_CBC_SHA
   ```

   Ensure that at least one of these ciphers has been specified in the *channel_name*SSLcipherList parameter of the EIF probe's configuration file. If this parameter value does not match any of the default TLS and SSL ciphers, specify a cipher override by using the KDEBE_V3_CIPHER_SPECS environment variable defined in the agent's environment file.

By default, the EIF probe's configuration file specifies SSL_RSA_WITH_3DES_EDE_CBC_SHA, which matches one of the monitoring agent ciphers and therefore you typically do not need to customize your agent's cipher list. However, if the EIF probe's *channel_name*SSLCipherList parameter does not match any of the monitoring agent's ciphers, then you must use KDEBE_V3_CIPHER_SPECS to specify the same cipher so that the TLS/SSL exchange can complete. The format of the environment variable is as follows:

KDEBE_V3_CIPHER_SPECS=*nn*

where *nn* is the cipher's short name.

The following table lists the cipher's short name and corresponding long name that would be defined for the *channel_name*SSLCipherList parameter.

Short name	Long name
01	SSL_RSA_WITH_NULL_MD5
02	SSL_RSA_WITH_NULL_SHA
03	SSL_RSA_EXPORT_WITH_RC4_40_MD5
04	SSL_RSA_WITH_RC4_128_MD5
05	SSL_RSA_WITH_RC4_128_SHA
06	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
09	SSL_RSA_WITH_DES_CBC_SHA
0A	SSL_RSA_WITH_3DES_EDE_CBC_SHA
2F	TLS_RSA_WITH_AES_128_CBC_SHA
35	TLS_RSA_WITH_AES_256_CBC_SHA

For example, if *channel_name*SSLCipherList=SSL_RSA_WITH_DES_CBC_SHA is defined in the EIF probe's configuration file, set KDEBE_V3_CIPHER_SPECS=09 in the agent's environment file.

Note: The KDEBE_V3_CIPHER_SPECS variable is ignored when KDEBE_FIPS_MODE_ENABLED=Y is defined. As a result, the default TLS and SSL ciphers are used.

5. Recycle the monitoring agent to process the changes to the agent's environment file, the private situation XML file, and the event destination XML file.

Certificate management

If the Netcool/Omnibus EIF probe uses a CA-signed digital certificate and *channel_name*SSLRequireClientAuthentication=YES is specified in the probe's configuration file, you must ensure that the monitoring agent's key database has imported a corresponding CA-signed digital certificate.

Configuring a monitoring agent's key database requires using a certificate management tool, which can be run in either GUI or CLI mode. Both modes of operation require a Java Runtime Environment available on the local system where the management tool is invoked. Typical environments require a minimum of IBM JRE V6. You also must ensure that the JAVA_HOME environment variable points to your IBM Java location. See "Setting the JRE for GSKit and starting Key Manager" on page 242.

IBM Tivoli Monitoring and Netcool/OMNIbus rely on GSKit for their SSL implementations. IBM Tivoli Monitoring V6.3 or later installs GSKit V8, which provides the GUI utility in the `gsk8ikm` binary and the CLI utility in the `<gskittoolcmd>` binary. Netcool/OMNIbus is based on GSKit V8, which runs only in CLI mode; for GUI mode, you must use the iKeyman utility, which is included in IBM JRE V6 or later.

IBM Tivoli Monitoring requires a CMS-type key database, whereas Netcool/OMNIbus requires a Java Key Store (JKS) database. The `keyfile.kdb` CMS key database file is installed in the *install_dir*`\keyfiles` directory. However, you cannot use this database in its current form if you require a CA-signed digital certificate when sending events over an SSL connection to the Netcool/OMNIbus EIF probe.

Complete agent certificate management tasks by using the iKeyman utility. Instructions in the example show how to do the following tasks:

- Create a new CMS key database, which can be located either in *install_dir*`\keyfiles` or in some other directory
- Import the CA-signed digital certificate
- Use the newly created database in place of the product-provided one

Example

The example monitoring agent runs on a Windows system and includes a key database file called `omnieif.kdb` with a password of `ITMPWD`, a previously configured Netcool/OMNIbus keystore file called `omni.jks` with a password of `EIFPWD`, and a certificate label named `eifca`. A copy of the `omni.jks` file is locally available in the *install_dir*`\keyfiles` directory.

GSKit keystroke configuration (GUI mode): To invoke the GSKit GUI tool on a Windows system, complete the following steps:

1. Run the *install_dir*`\GSK8\bin\gsk8ikm.exe` command file. The IBM Key Management GUI is displayed. If an error occurs, verify that a JRE is installed and that JAVA_HOME is set correctly.
2. In the menu bar, click **Key Database File** > **New**. Enter the following information and click **OK**:

   Key database type: `CMS`

   File Name: `omnieif.kdb`

   Location: *install_dir*`\keyfiles\`
3. Set the keystore password and click **OK**:

   Password: `ITMPWD`

   Confirm Password: `ITMPWD`

   ☑ Expiration time: `366 Days`

   ☑ Stash the password to a file
4. Ensure that **Personal Certificates** is displayed in the **Key database content** menu. **Import** and then click **OK**:

   Key file type: `JKS`

   File Name: `omni.jks`

   Location: *OMNIbus_keystroke_dir*`\`
5. Enter the password to open the source key database: `EIFPWD`. Click **OK**.

6. Select keys from the key list of the source key database. Select the label `eifca`. Click **OK**.
7. When prompted with, `Would you like to change any of these labels before completing the import process?` click **OK** without changing any labels.
8. Exit the IBM Key Management window.
9. Edit the monitoring agent's environment file and set the following values:

   KDEBE_KEYRING_FILE=*install_dir*\keyfiles\omnieif.kdb

   KDEBE_KEYRING_STASH=*install_dir*\keyfiles\omnieif.sth

   KDEBE_KEY_LABEL=eifca
10. Restart the agent and the new CMS key database is used.

GSKit keystore configuration (CLI mode): If an "IBM Key Management" GUI utility is not available, you can use the GSKit's CLI tool on Windows to perform the certificate import function. By using the same values chosen in the GUI example, the commands are as follows:

1. From the command line, `cd` to the *install_dir*`\keyfiles` directory and create the database file:

   ```
   install_dir\GSK8\bin\gsk8cmd.exe -keydb -create -db omnieif.kdb -pw
   ITMPWD -type CMS -stash -expire 366
   ```
2. Run the following command to import the Netcool/OMNIbus certificate:

   ```
   install_dir\GSK8\bin\gsk8cmd.exe -cert -import -file
   OMNIbus_keystore_dir\omni.jks –pw EIFPWD -label eifca -type JKS -target
   omnieif.kdb -target_pw ITMPWD
   ```

   As in the GUI example, you must update these values in the agent's environment file.

   KDEBE_KEYRING_FILE=*install_dir*\keyfiles\omnieif.kdb

   KDEBE_KEYRING_STASH=*install_dir*\keyfiles\omnieif.sth

   KDEBE_KEY_LABEL=eifca

Agent Service Interface

Use the agent service interface for retrieving information from an installed agent, whether it is a Tivoli Enterprise Monitoring Agent or Tivoli System Monitor Agent. After logging into the local operating system, you can get reports of agent information, private situations, private history, queries, and attributes, and such requests as configuration load list commands.

The Agent Service Interface is accessed through the IBM Tivoli Monitoring Service Index utility. The interface operates as an Internet server, accepting and validating requests, dispatching requests to the agent for processing, and gathering and formatting reply data using the HTTP or HTTPS application protocol over TCP/IP.

z/OS IBM i The Agent Service Interface is not available for installation on IBM i and z/OS operating systems. You can, however, use the ITMSUPER Tools that are included in the IBM Support Assistant (ISA), a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. To install the ISA software, go to IBM Support Assistant (http://www-01.ibm.com/software/support/isa). As part of the tool set, **itmsa.htm** provides direct access to agent reports on IBM i, z/OS, Windows, Linux, and UNIX platforms; and **itmsuper.htm** provides Web Services tools that are accessible through the hub monitoring server.

Starting the Agent Service Interface

Start the Agent Service Interface from your browser to get a menu of choices for reporting agent information, getting situation status, displaying short-term history, and for making service requests in XML.

Before you begin

You must have an administrator user ID for the operating system where the monitoring agent is installed to access the Agent Service Interface and its functions.

About this task

Follow these steps to start the IBM Tivoli Monitoring Service Index utility and then log onto the Agent Service Interface for the agent that you want to get information about.

Procedure

1. Start the IBM Tivoli Monitoring Service Index by entering `http://`*`<host name>`*`:1920` or `https//`*`<host name>:`*`3661`, where host name is the fully qualified name or IP address of the computer where the agent is installed. A list of the started services is displayed.
2. Click the **IBM Tivoli <*PC*> Agent Service Interface** (where *pc* is the two-character component code) link for the application to work with.
3. As prompted, enter the administrator-level user name and password for the operating system.

Results

After you have been authenticated, the Agent Service Interface Navigator page is displayed with links to **Agent Information**, **Situations**, **History**, **Queries**, and **Service Interface Request**. The Navigator page is the `navigator.htm` file that is installed at this location by default:

Windows *install_dir*\localconfig\html

Linux UNIX *install_dir*/localconfig/HTML

Monitoring agents that use subnodes, such as Agentless Monitors, VMware VI Agent, and subnodes created with Agent Builder, have some reporting limitations in the Agent Service Interface:

- Queries link is unavailable
- Situations listing shows all the situations for an agent, including the subnodes; filtering by subnode is not available
- Private history shows all the collected historical data for the selected attribute on the agent, including any subnodes; filtering by subnode is not available.

Access Authorization Group Profile

The Access Authorization Group Profile (AAGP) contains the access authorization group definitions and user ID assignments that are established by the security administrator.

The security administrator can define any access authorization group name with the exception of the **Restricted** group, which is mandatory. Each access authorization group has at least one agent component category, such as Service Interface (SIAPI element) and services published by that agent component. Each agent component calls upon the AAGP facility to get the user ID, component category, and the requested service name for access authorization. After authorization, the agent component executes the requested service; otherwise the agent returns a status of unauthorized.

The AAGP is not an authentication service and it assumes that the user ID provided has been authenticated. The same assumption is made for the Service Interface because all users must first sign onto the system with a valid ID and password. However, the agent can perform work on behalf of other agents or the Tivoli Enterprise Monitoring Server, such as automation actions, and the user ID on hand could be unknown to the local system. In such a scenario, the agent considers that the virtual user is a trusted Tivoli Monitoring member and therefore authentic and calls upon AAGP for authorization. Alternatively, the AAGP can be enhanced to leverage a centralized authentication and authorization service where such facility becomes available.

Access Authorization Group types

The following default AAGP groups are predefined and they are automatically loaded upon agent startup.

Restricted group
The default group. The Service Interface category in this group consists of services that provide system information, operation configuration, workload monitoring, and historical data reporting capabilities. All users are in this mandatory group, including those that are not specifically defined.

Operation group
This group includes **Restricted** group category services and the Service Interface services that provide operation control, configuration management, and application customized access capabilities.

Administrative group
This group has access to all Service Interface capabilities, with the addition of File Object and dynamically updating the AAGP.

Table 42. Access Authorization Group permissions for Service Interface commands

Service Interface API	Restricted	Operation	Administrative
AgentInfo	x	x	x
AttrList	x	x	x
ReadAttr	x	x	x
ListSubnode	x	x	x
TableSit	x	x	x
SitStat	x	x	x
SitSummary	x	x	x
HistRead	x	x	x
Report	x	x	x
PvtControl		x	x
CnfgCommand		x	x
ConfigurationArtifact		x	x
PrivateConfiguration		x	x

Table 42. Access Authorization Group permissions for Service Interface commands (continued)

Service Interface API	Restricted	Operation	Administrative
Overrides		x	x
AAGP			x
ListAAGP		x	x
FileObj			x

The **Restricted** group definition is required. If it is not included in the AAGP, the agent default specification shown in Table 42 on page 399 is in effect.

Specifying the keyword *NONE for a component category prohibits all non-explicit users from accessing that component service. For example, `<SIAPI>*NONE</SIAPI>` specified in **Restricted** group disallows general access to the Agent Service Interface.

FileObj allows you to *push* or *pull* files on an agent using an HTTP request. For Centralized Configuration, FileObj is the API that is used to allow a monitoring agent to act as a central configuration server. The Agent Service Interface is available in the basic services of the monitoring agent and can be used to serve files or you can send HTTP requests to any agent to push or pull files. The AAGP function provides additional security. By default, only **root** on Linux or UNIX and **Administrator** on Windows are members of the AD group that has permission to use the FileObj API. See the example in “Monitoring agent as the central configuration server” on page 427.

If the AAGP contains no `<AAGROUP>` specification, the agent default specification shown in Table 42 on page 399 is in effect. Valid groups are RE, OP, and AD. There is no need to define R (restricted) group users because all users are automatically assigned to the **Restricted** group unless otherwise defined by the AAGP.

Access Authorization Group Profile XML specification

The security administrator defines the agent User Group Authorization Profile in simple XML specification format:

`<AAGP>`

This element identifies the XML file as an agent Access Authorization Group Profile document. All AAGP specifications must be enclosed by begin <AAGP> and end </AAGP> root-level element tags. The contents of the AAGP file are merged with the existing AAGP being used by the agent and you can add users to the default Access Authorization Groups. If you prefer to completely replace the existing AAGP, use the REFRESH attribute with the AAGP element.

`REFRESH="Y"`

Deletes the current active AAGP and replaces it with the AAGP definition from this AAGP specification.

`LOCAL="LOCK | UNLOCK"`

Optional, with no default value. LOCK and UNLOCK are only accepted when an AAGP update originated from the ASI.

LOCAL="LOCK" locks the local AAGP configuration, which cannot be updated by either the ASI or by a Centralized Configuration Facility AAGP file download.

LOCAL="UNLOCK" unlocks the local AAGP configuration and AAGP can now be updated by ASI or a Centralized Configuration Facility file download. UNLOCK is valid only when LOCK is in effect, otherwise it is ignored. In other words, <AAGP LOCAL="UNLOCK"> must be preceded by a prior <AAGP LOCAL="LOCK"> operation.

<AAGP LOCAL="LOCK"></AAGP> and <AAGP LOCAL="UNLOCK"></AAGP> can be independent stand-alone AAGP ASI transactions.

<AAGROUP>
Defines an Access Authorization Group. Begin <AAGROUP> and end </AAGroup> element tags enclose a set of group definitions.

<GROUPNAME>
Defines the Access Authorization Group name. Specify the name between begin <GROUPNAME> and end </GROUPNAME> element tags. The group name can be up to 32 characters and the first two characters must be unique among all user group names.

<INCLUDE>
Optional. Specifies the AAGROUP definitions to be included in this AAGROUP. Enclose the AAGROUP name within begin <INCLUDE> and end </INCLUDE> tags.

<SIAPI>
Specifies the agent Service Interface API name and is not case-sensitive. Only the component category is defined at this time. Enclose the name within begin <SIAPI> and end </SIAPI> tags.

<other>
The <other> element is not available in the current release; it is reserved for future use. It specifies the other agent component services to be managed.

<AAUSER>
Defines an authorized user ID and its associated Access Authorization Group by name. Enclose each user definition within begin <AAUSER> and end </AAUSER> tags.

<ID>
Specifies an authorized user sign-on ID and is not case-sensitive. Enclose the user ID within begin <ID> and end </ID> tags.

<ASSIGN>
Specifies the Access Authorization Group assignment and is not case-sensitive. Valid AAGP types are RE (**Restricted**), OP (**Operation**), and AD (**Administrative**). You can enter the full group name or the first character. Enclose the AAGP type within begin <ASSIGN> and end </ASSIGN> tags.

Example

```
<AAGP>
   <AAGROUP>
   <GROUPNAME>Restricted</GROUPNAME>
   <SIAPI>AgentInfo</SIAPI>
   <SIAPI>AttrList</SIAPI>
   <SIAPI>ReadAttr</SIAPI>
   <SIAPI>ListSubnode</SIAPI>
   <SIAPI>TableSit</SIAPI>
     <SIAPI>ListTable</SIAPI>
   <SIAPI>SitStats</SIAPI>
   <SIAPI>SitSummary</SIAPI>
   <SIAPI>HistRead</SIAPI>
   <SIAPI>Report</SIAPI>
```

```
 <REFLEXAUTO>ExecAction</REFLEXAUTO>
 </AAGROUP>
 <AAGROUP>
 <GROUPNAME>Operation</GROUPNAME>
 <INCLUDE>Restricted</INCLUDE>
 <SIAPI>PvtControl</SIAPI>
 <SIAPI>CnfgControl</SIAPI>
 <SIAPI>CnfgCommand</SIAPI>
 <SIAPI>ConfigurationArtifact</SIAPI>
 <SIAPI>PrivateConfiguration</SIAPI>
 <SIAPI>Overrides</SIAPI>
   <SIAPI>XMSClientSpec</SIAPI>
 <SIAPI>ListAAGP</SIAPI>
   <CLI>ExecCommand</CLI>
 <TAKEACTION>ExecAction</TAKEACTION>
</AAGROUP>
<AAGROUP>
 <GROUPNAME>Administrative</GROUPNAME>
 <INCLUDE>Operation</INCLUDE>
 <SIAPI>FileObj</SIAPI>
 <SIAPI>AAGP</SIAPI>
</AAGROUP>
<AAUSER>
 <ID>default</ID>
 <ASSIGN>OP</ASSIGN>
</AAUSER>
</AAGP>
```

Access Authorization Group methodology

All valid system users are automatically authorized for **Restricted** group access. Authorized, **Administrative** group, and other groups users are defined by the enterprise security administrator through the AAGP. The following procedure illustrates AAGP methodology.

1. The enterprise security administrator creates a customized AAGP and stores it at a secure central configuration server. The predefined authorization group content can be customized and additional custom authorization groups added. For example, `<AUTOCMD>KILL</AUTOCMD>` could be included in the **Operation** group.
2. The monitoring agent starts and activates the default AAGP. An administrative ID is defined as a member of the **Administrative** group by default: *Administrator* on Windows; *root* on Linux or UNIX.
3. The monitoring agent leverages Centralized Configuration and retrieves its own customized AAGP from a central configuration server. The agent always chooses the HTTPS protocol for this operation. If there is no AAGP included in agent's configuration load list or if the AAGP cannot be downloaded from the central configuration server, the agent operates in this mode until the next restart.
4. Agent components check the AAGP for authorization, which provides the user ID, component category, and service name. The AAGP grants or denies access based on the access authorization group and user ID assignment.
5. The monitoring agent checks for AAGP updates periodically as specified in the configuration load list or when the Service Interface configuration command is issued.
6. The monitoring agent does not save or store the User Authorization Profile locally.

Local AAGP Persistent Configuration

An administrator might need to make AAGP customization changes to meet ad-hoc challenges. For example, adding temporary contractor user IDs or rearranging category services per local environment access restrictions. The steps below describe the procedure for implementing local AAGP configuration changes that persist across agent restarts:

1. Logon to the local system through the Agent Service Interface using an administrator user ID.
2. Use the <ListAAGP> transaction to retrieve the agent's current AAGP specification.
3. Edit the ListAAGP output for your new requirements. For example, adding another authorized user ID.
4. Submit the updated AAGP definitions to the agent through the Agent Service Interface.
5. The agent processes the input AAGP definitions, which then become the current active AAGP definition in effect. The agent also outputs a copy of the AAGP configuration XML to a local file, either `$ITM_HOME$/localconfig/pc/pc_aagpcnfg.txt` on distributed systems, or `KpcAAGPX.UKANDATV` in encrypted format on z/OS systems.
6. At the next agent startup, the agent searches for the local persistent AAGP configuration file, decrypts and reads the file, and then processes all of its AAGP XML statements. These actions restore the previously customized AAGP definitions. If the agent cannot find the local persistent AAGP configuration file, the default AAGP definitions take effect.
7. While the locally customized AAGP configuration is in effect, the agent suspends all AAGP updates from the Centralized Configuration Facility, ensuring that the local AAGP customization values remain unchanged. The administrator must submit an <AAGP Resume=”Y”> request through the Agent Service Interface to unlock the local AAGP persistent configuration. This allows the Centralized Configuration Facility to resume its support of AAGP update operations.

Local AAGP Authorization Control

If the administrator wants to enforce strict authorization controls at an agent endpoint and not allow any automation requests from executing, unless the user ID is defined to the local system, then you can use the following procedure to enable local AAGP authorization control:

1. Logon to local system through the Agent Service Interface using an administrator user ID.
2. Use the <ListAAGP> transaction to retrieve the agent's current AAGP specification.
3. Edit the ListAAGP output and delete the default user definition.

   ```
   <AAUSER>
      <ID>default</ID>
      <ASSIGN>OP</ASSIGN>
   </AAUSER>
   ```

4. The default user definition instructs AAGP to create an internal secret user ID that is granted the authority to run automation. Without the default user ID definition, AAGP extracts the user ID from each command request sent to the agent. If the extracted user ID is undefined to AAGP, the command request is

rejected with an authorization error. This capability gives the local administrator full control over which automation is allowed to run at an agent endpoint.

5. Submit the updated AAGP definitions to the agent through the Agent Service Interface.

Agent Service Interface - Agent Information

Select **Agent Information** from the Agent Service Interface menu to retrieve a report of pertinent data about the agent, including the environment file settings.

HOSTNAME

This is the fully qualified name of the computer, such as **myitm.raleigh.ibm.com**.

NODENAME

This is the name of the managed system, such as **Primary:MYITM:NT**.

SUBSYSID

If the agent has subnodes (subagents), this is the name. Otherwise, the subsystem ID is **Primary**.

NODEINFO

This is the type of system and operating platform, such as **Win2003~5.2-SP2**.

PRODUCT

This is the two-character product code of the agent, such as **NT**.

VERSION

This is the installed version of the agent, such as **06.22.00**.

LEVEL A=00:WINNT C=06.22.00.00:WINNT G=06.22.00.00:WINNT

PATCHLEVEL A=00:WINNT;C=06.22.00.00:WINNT;G=06.22.00.00:WINNT;

AFFINITY

This is value that identifies the affinity of the agent to the Tivoli Management Services components. For example, **%IBM.STATIC021 000000000A00000u0a4**.

BOOTTIME

This is the day of the week, the calendar date and time when the agent completed startup, such as **Wed Jul 29 15:15:33 2009**.

ENVFILE

This is a list of the current parameter settings in the agent environment file. If you need to change any of the values, you can open the environment file through Manage Tivoli Monitoring Services or in a text editor on distributed systems.

Here is an example of the Windows OS environment file as it is displayed in Agent Information report:

```
* CANDLE_HOME=d:\IBM\ITM
* KBB_RAS1=ERROR
* KBB_VARPREFIX=%
* KBB_VARPREFIX=$
* KBB_RAS1_LOG=d:\IBM\ITM\tmaitm6\logs\$(computername)_nt_kntcma_$
(sysutcstart)-.log INVENTORY=d:\IBM\ITM\tmaitm6\logs\$(computername)
_nt_kntcma.inv COUNT=03 LIMIT=5 PRESERVE=1 MAXFILES=9
* TIMEOUT=600
* ITMDEPLOY_AGENTDEPOT=d:\IBM\ITM\tmaitm6\agentdepot
* ICCRTE_DIR=d:\IBM\ITM\GSK8
* CSV1_PATH=d:\IBM\ITM\GSK8\lib
* CSV2_PATH=d:\IBM\ITM\GSK8\bin * KBB_VARPREFIX=$
* PATH!=$(CSV1_PATH);$(CSV2_PATH);$(PATH)
* KEYFILE_DIR=d:\IBM\ITM\keyfiles
* KDEBE_KEYRING_FILE=d:\IBM\ITM\keyfiles\keyfile.kdb
* KDEBE_KEYRING_STASH=d:\IBM\ITM\keyfiles\keyfile.sth
* KDEBE_KEY_LABEL=IBM_Tivoli_Monitoring_Certificate
* KBB_IGNOREHOSTENVIRONMENT=Y
* JAVA_HOME=d:\IBM\ITM\java\java70\jre
* KBB_IGNOREHOSTENVIRONMENT=N
* PATH=d:\IBM\ITM\GSK8\LIB;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\
System32\Wbem;D:\IBM\SQLLIB\BIN;D:\IBM\SQLLIB\FUNCTION;D:\IBM\SQLLIB\
SAMPLES\REPL;d:\IBM\ITM\bin;d:\IBM\ITM\bin\dll;d:\IBM\ITM\InstallITM;
d:\IBM\ITM\TMAITM6;d:\IBM\ITM\InstallITM
```

Agent Service Interface - Situations

Select the **Situations** option of the Agent Service Interface to see the status and statistics of each situation, including private situations and situations distributed to any subnodes, for the monitoring agent.

The Situations report gives some vital statistics about each situation on the agent. The setting of the agent environment variable IRA_EVENT_EXPORT_SIT_STATS determines the level of detail given.

Situation name
Above each situation summary page is the name of the situation. If this is a private situation, the name will be appended with **_pr**.

TYPE
Sampled or Pure. A situation is sampled if it samples data at regular intervals. Pure events are unsolicited notifications. The Windows Event Log and Windows File Change attribute are examples of attribute groups that report pure events.

INTERVAL
The interval between data samples, in seconds. If situations for this attribute group trigger pure events, there is no sampling interval and the value shows as 0.

ROWSIZE
This is the row size.

FIRSTSTARTTIME
This is the day of the week, calendar day, and time when the situation is initially started after the agent starts.

LASTSTARTTIME
This is the day of the week, calendar day, and time when the situation was most recently started.

LASTSTOPTIME
This is the day of the week, calendar day, and time when the situation was most recently stopped.

FIRSTEVENTTIME
This is the day of the week, calendar day, and time of the first occurrence that the situation became true and opened an event since the situation was started.

LASTTRUETIME
This is the day of the week, calendar day, and time when the situation most recently became true and opened an event.

LASTFALSETIME
This is the day of the week, calendar day, and time when the situation state evaluated to false after an earlier sampling evaluated to true.

TIMESRECYCLED
This is the number of times the situation was stopped and started since the agent has been online.

TIMESAUTONOMOUS
This is the number times since startup that the situation entered autonomous state because the enterprise monitoring agent was disconnected from its monitoring server, followed by the DAY statistics:

DAY

DATE that the most recent statistical data was collected. If this is an enterprise situation, this is since the agent was most recently connected.

TRUESAMPLES is the number of times the situation evaluated to true while the agent was disconnected from the monitoring server.

FALSESAMPLES is the number of times the situation evaluated to false after a prior true while the agent was disconnected from the monitoring server.

TRUERATIO is the percentage of the number of times the situation evaluates to true compared with the false state.

FALSERATIO is the percentage of the number of times the situation evaluates to false compared with the true state.

HOURROWS is the number of rows of data that have been reported.

HOURTRUE is the number of hours that the situation remained true while the agent was disconnected from the monitoring server.

HOURFALSE is the number of hours that the situation remained false while the agent was disconnected from the monitoring server.

All situations are shown for an agent, including the subnodes. In this sample TestLab agent with subnodes called ComputerA and ComputerB, ten situations would be listed:

TestLab

SubNodeA (4 unique situations, plus 2 situations that are also on SubNodeB)

SubNodeB (4 unique situations, plus 2 situations that are also on SubNodeA)

Agent Service Interface - History

Select **History** in the Agent Service Interface to display the private history data samples that have been saved for the selected attribute group table.

You can filter the report to show only the attributes that you are interested in by clearing the check box next to any unwanted attributes. Select a start date and time and an end date and time, then click **Report**. The report is displayed in a table below the attributes, showing historical data samples for the attribute group, one column per attribute and one row per sampling, for the time period specified, up to 5000 rows. If you do not see the rows you are interested in within the 5000 limit, you can generate another report after narrowing the time range.

All collected historical data is shown for the agent, including any subnodes.

Agent Service Interface - Queries

Select the **Queries** option of the Agent Service Interface to query the k*pc*.atr file for the selected attribute group, shown by table name. One report is a list of the attributes, their column name and display name, and characteristics. The other report shows the current sampled values of the attributes.

Select a table name from the ▼ list to see the component attributes and a report of the sampled data.

Table name
This is the table name for the attribute group taken from the <*install_dir*>/TMAITM6/ATTRLIB/k*pc*.atr file (where *pc* is the two-character product code).

Name
This is the column name for the attribute. It is not used in private situations or private history, but is what you would see if you were to look up the stored data in the Tivoli Data Warehouse.

Display
This is the detailed name of the attribute, formatted as *Attribute_Group_Name.Attribute_Name*, and is what you enter in the private situation and private history definitions. For example, KHD_CONFIG.Connection_Pool_Size or NT_Registry.Server_Name.

Type
Displayed in this column is a number that represents the type of attribute this is, such as 4 to denote an integer-enumerated attribute. The type is not used directly in a private situation or private history definition, but informs you of the required format for the attribute value.

Length
This is the number of bytes or maximum number of bytes possible for the attribute value. For TIMESTAMP attributes, 16 indicates the following format: CYYMMDDHHMMSSmmm, such as 1090819160501000 for the 21st century on August 19, 2009 at 4:05:01 PM

Minimum
The lowest possible value for the attribute is displayed here. If the field is empty, there is no minimum value for the attribute.

Maximum
The highest possible value for the attribute is shown in this column. If the field is empty, there is no maximum value for the attribute.

ENUMS
This is the enumeration and what it represents for an attribute. Some enumerated attributes have multiple enumerations. When composing a private situation for an enumerated attribute, use the actual value in the formula and not the display value (what you would see in the Tivoli Enterprise Portal).

These two reports show the results of a query to the Windows IP Address attribute group, table name NTIPADDR. The first report is a listing of the attributes in the table as they appear in the k*pc*.atr file. When creating private situations or private history definitions, you must use the name shown in the **Display** column.

Table 43. Agent Service Interface - Queries sample attribute listing

Name	Display	Type	Length	Minimum	Maximum	ENUMS
ORIGINNODE	NT_IP_Address.System_Name	2	64			

Table 43. Agent Service Interface - Queries sample attribute listing (continued)

Name	Display	Type	Length	Minimum	Maximum	ENUMS
TIMESTAMP	NT_IP_Address.Timestamp	2	16			
INTFNAME	NT_IP_Address.Network_ Interface_Name	3				Not Available for Windows 2000
IPADDRESS	NT_IP_Address.IP_Address	2	50			
DNSNAME	NT_IP_Address.DNS_Name	10	388			No DNS Entry
IPVERSION	NT_IP_Address.IP_Version	4		-2147483648	2147483647	4 IPv4 6 IPv6 10 IPv4_IPv6
MACADDRESS	NT_IP_Address.MAC_Address	2	28			

The second report is displayed with the current sampled values of the attribute group.

Table 44. Agent Service Interface - Queries sample report

ORIGINNODE	TIMESTAMP	INTFNAME	IPADDRESS	DNSNAME	IPVERSION	MACADDRESS
Primary:East:NT	1090819142128111	11a_b_g Wireless LAN Mini PCI Adapter	9.52.100.111	East.ibm.com	4	00054e48f5bd
Primary:East:NT	1090819142128111	MS TCP Loopback interface	127.0.0.1	NO_DNS_ ENTRY	4	000d608b2938

Agent Service Interface - Service Interface Request

Select **Service Interface Request** in the Agent Service Interface to enter commands in XML format for information about the agent, such as attribute group definitions.

Agent Service Interface request - Agent information

This is a request of agent identification information. The data retrieved is in three sections: agent ID, which includes computer hostname, managed system name, subnode list, and operating system information; product ID, which includes product name, version, maintenance and patch level data, product affinity and features; and environment ID, which includes the current environment variable settings.

Request input

Table 45. Agent Service Interface <AGENTINFO> request.

Tag	Description
<AGENTINFO>	Enter begin and end AGENTINFO tags to make an agent property request..

Sample request:

```
<AGENTINFO>
</AGENTINFO>
```

Report output

Table 46. Agent Service Interface <AGENTINFO> request output.

Output tag	Description
<HOSTNAME>	Agent host name
<NODENAME>	Agent Managed System name
<SUBSYSID>	Agent Subsystem ID
<NODEINFO>	Agent system OS information
<PRODUCT>	ITM product name
<VERSION>	Agent version
<LEVEL>	Agent installation and maintenance level
<PATCHLEVEL>	Agent maintenance patch level
<AFFINITY>	Agent affinity in effect
<BOOTTIME>	Agent boot time
<ENVFILE>	Agent configuration file enclosed by CDATA[] control data tags

Sample output: The agent returns property data

```
<AGENTINFO>
  <HOSTNAME>dyang7</HOSTNAME>
  <NODENAME>Primary:DYANG7:NT</NODENAME>
  <SUBSYSID>Primary</SUBSYSID>
  <NODEINFO>WinXP~5.1-SP2</NODEINFO>
  <PRODUCT>NT</PRODUCT>
  <VERSION>06.22.00</VERSION>
 <LEVEL>A=00:WINNT C=06.21.00.00:WINNT G=06.21.00.00:WINNT</LEVEL>
 <PATCHLEVEL>A=00:WINNT;C=06.21.00.00:WINNT;G=06.21.00.00:WINNT;
 </PATCHLEVEL>
 <AFFINITY>000800000000000000000000000000004000000u0a4</AFFINITY>
  <BOOTTIME>Mon Mar 02 22:48:27 2009</BOOTTIME>
  <ENVFILE>
   <![CDATA[
   CANDLE_HOME=C:\IBM\ITM
   KBB_RAS1=ERROR
   KBB_VARPREFIX=%
   TIMEOUT=600
   ITMDEPLOY_AGENTDEPOT=C:\IBM\ITM\tmaitm6\agentdepot
   IRA_AUTONOMOUS_MODE=Y
   CTIRA_HEARTBEAT=1440
   CTIRA_RECONNECT_WAIT=60
   IRA_DUMP_DATA=Y
   IRA_DEBUG_TRANSCON=N
   IRA_DEBUG_EVENTEXPORT=N
   IRA_DEBUG_AUTONOMOUS=Y
   IRA_DEBUG_SERVICEAPI=Y
   IRA_DEBUG_PRIVATE_SITUATION=Y
   IRA_EVENT_EXPORT_LISTSTAT_INTERVAL=300
   IRA_EVENT_EXPORT_SNMP_TRAP=Y
   ICCRTE_DIR=C:\IBM\ITM\GSK8
   CSV1_PATH=C:\IBM\ITM\GSK8\lib
   PATH!=$(CSV1_PATH);$(PATH)
   KEYFILE_DIR=C:\IBM\ITM\keyfiles
   KDEBE_KEYRING_FILE=C:\IBM\ITM\keyfiles\keyfile.kdb
   KDEBE_KEYRING_STASH=C:\IBM\ITM\keyfiles\keyfile.sth
   KDEBE_KEY_LABEL=IBM_Tivoli_Monitoring_Certificate
   JAVA_HOME=C:\Program Files\IBM\Java70\jre
   PATH=C:\IBM\ITM\GSK8\LIB;\;C:\WINDOWS\system32;C:\WINDOWS;
   C:\WINDOWS\System32\Wbem;c:\perl\bin;C:\Infoprint;
    C:\IBM\ITM\InstallITM ]]>
  </ENVFILE>
</AGENTINFO>
```

Agent Service Interface request - Agent subnode list

Use the <LISTSUBNODE> request in the Service Interface Request to get a listing of all known subnodes on this computer.

Request input

Table 47. Agent Service Interface <LISTSUBNODE> request.

Tag	Description
<LISTSUBNODE>	Enter begin and end LISTSUBNODE tags to make a subnode list request.

Sample request:

```
<LISTSUBNODE>
</LISTSUBNODE>
```

Report output

Table 48. Agent Service Interface <LISTSUBNODE> request output.

Output tag	Description
<SUBNODELIST>	Subnode list.
<NODECOUNT>	Subnode count.
<NAME>	Subnode name.

Sample output: The agent returns a listing of all known subnodes of the agent

```
<SUBNODELIST>
  <NODECOUNT>3</NODECOUNT>
  <NAME>dyang7ASFSdp:UAGENT00</NAME>
  <NAME>dyang7:TS100</NAME>
  <NAME>dyang7:TS200</NAME>
  </SUBNODELIST>
```

Agent Service Interface request - Attribute files list

Use <ATTRLIST> in a Service Interface Request to get a listing of all known attribute files (.atr) that are available on this computer.

Request input

Table 49. Agent Service Interface <ATTRLIST> request.

Tag	Description
<ATTRLIST>	Enter begin and end ATTRLIST tags to make an attribute file list request.

Sample request:

```
<ATTRLIST>
</ATTRLIST>
```

Report output

Table 50. Agent Service Interface <ATTRLIST> request output.

Output tag	Description
<LISTATTRFILE>	List of available attribute file names.
<ATTRCOUNT>	The total number of attribute files in the list.
<NAME>	Name fo the attribute file.

Sample output: The agent returns a listing of all known attribute files that are available on the computer

```
<LISTATTRFILE>
  <ATTRCOUNT>16</ATTRCOUNT>
  <NAME>DM3ATR00</NAME>
 <NAME>TS1ATR00</NAME>
 <NAME>TS2ATR00</NAME>
 <NAME>UAGATR00</NAME>
 <NAME>kdy.atr</NAME>
 <NAME>khd.atr</NAME>
 <NAME>kib.atr</NAME>
 <NAME>knt.atr</NAME>
 <NAME>kr2.atr</NAME>
 <NAME>kr3.atr</NAME>
 <NAME>kr4.atr</NAME>
 <NAME>kr5.atr</NAME>
 <NAME>kr6.atr</NAME>
 <NAME>ksh.atr</NAME>
 <NAME>ksy.atr</NAME>
 <NAME>kum.atr</NAME>
</LISTATTRFILE>
```

Agent Service Interface request - Attribute file contents

Use <READATTR> in a Service Interface Request to get a listing of the contents of the specified attribute file (.atr) on this computer.

Request input

Table 51. Agent Service Interface <READATTR> request.

Tag	Description
<READATTR>	Enter begin and end READATTR tags to make an attribute file request.
<ATTRFILE>	Attribute file name.

Example of a request for the Linux OS Agent `klz.atr` attribute file:

```
<READATTR>
  <ATTRFILE>klz.atr</ATTRFILE>
</READATTR>
```

Report output

Table 52. Agent Service Interface <READATTR> request output.

Output tag	Description
<ATTRFILE>	Attribute file name.
<ATTRDATA>	Attribute file records.

This example shows an excerpt from the `klz.atr` attribute file contents:

```
<ATTROUTPUT><ATTRFILE>klz.atr</ATTRFILE>
<ATTRDATA>
<![CDATA[// 1130415173632000 KLZ/6.2.0 = KLZ/01
//   DO NOT EDIT THIS GENERATED FILE
//
entr AFF
afid %IBM.STATIC134
asym LINUX_SYSTEM
atxt "Linux OS"
acod KLZ
//
entr ATTR
name KLZ_User_Login.System_Name
affi "%IBM.STATIC134          0000000008000000000"
appl KLZ
aplv 01
stmp 1130415173632000
cvrm 6.2.0
lvrm 0.0.0
tabl KLZLOGIN
mult 1
samp 3
doop Y
colu ORIGINNODE
type 2
slng 64
msid KLZ####
opgr 2048
atid 00030001
//
entr ATTR
name KLZ_User_Login.Timestamp
unit CYYMMDDHHMMSSmmm
affi "%IBM.STATIC134          0000000008000000000"
colu TIMESTAMP
type 2
slng 16
msid KLZ####
opgr 2048
atid 00030002
//
entr ATTR
name KLZ_User_Login.User_Name
affi "%IBM.STATIC134          0000000008000000000"
colu USRNAME
pkey 1
atom Y
type 10
slng 96
msid KLZ####
opgr 2048
atid 00030003
]]></ATTRDATA></ATTROUTPUT>
```

Agent Service Interface request - Attribute group report

Use <REPORT> in a Service Interface Request to get a report of the attribute group specified in the TABLENAME attribute, such as UNIXOS or NTPROCESS.

Request input

Table 53. Agent Service Interface <REPORT> request

Tag	Description
<REPORT>	Enter begin and end REPORT tags to retrieve application table data for the table specified.

Table 53. Agent Service Interface <REPORT> request (continued)

Tag	Description
<SQLTABLE>	The SQLTABLE begin and end tags enclose the TABLENAME tagging pair to identify the SQL table definition set.
<TABLENAME>	The TABLENAME begin and end tags enclose the table name to report. This is the name as it appears bracketed by begin and end tags. If you are not sure what the spelling is of the table, you can find it in the **tabl** field of the agent .atr file, located in the <*install_dir*>/TMAITM6/ATTRLIB directory.
<OUTPUT>	Optional. Use OUTPUT begin and end tags and their subordinate tags to filter and refine the report. **<COLUMN>** Define selected column name bracketed by begin and end tags. **<FILTER>** Define output data rows filter criteria with begin and end tags. The filter follows the same syntax as the private situation <CRITERIA> element. See "Private situation XML specification" on page 336.

Sample request 1: Report all attributes in the UNIX OS table

```
<REPORT>
 <SQLTABLE>
 <TABLENAME>UNIXOS</TABLENAME>
 </SQLTABLE>
</REPORT>
```

Sample request 2: Summary report of the Windows Process attribute group with a filter and columns specified

The request is for the values in the *_Total* row.

```
<REPORT>
 <SQLTABLE>
 <TABLENAME>NTPROCESS</TABLENAME>
   <OUTPUT>
    <COLUMN>ORIGINNODE</COLUMN>
    <COLUMN>TIMESTAMP</COLUMN>
    <COLUMN>INSTCNAME</COLUMN>
    <COLUMN>IDPROCESS</COLUMN>
    <COLUMN>PCTPRCSTME</COLUMN>
    <COLUMN>THREADCNT</COLUMN>
    <COLUMN>WRKINGSET</COLUMN>
   </OUTPUT>
 <FILTER>
 <![CDATA[ *VALUE INSTCNAME  *EQ _Total]]>
 </FILTER>
 </SQLTABLE>
</REPORT>
```

Report output

Table 54. Agent Service Interface <REPORT> request output.

Output tag	Description
<REPORTDATA>	Identify output report data set.
<ROWCOUNT>	Output table row count.
<ROW>	Identify an output row data.
<NAME>	Define output column name enclosed by begin and end tags.
<DATA>	Specify output column data value enclosed by begin and end tags.

Numeric output

The report does not format numeric values; they remain unformatted.

For example, if you were to get a report containing an attribute with a scale factor of 2, a value of **7** for that attribute would show in a table view in the Tivoli Enterprise Portal as **0.07**. You can look up the scale factor, shown as `scal` in the attribute definition, in the attribute file:
Windows `<`*`install_dir`*`>\TMAITM6\ATTRLIB\k`*`pc`*`.atr`
Linux UNIX `<`*`install_dir`*`>/`*`platform`*`/<`*`pc`*`>/tables/ATTRLIB/k`*`pc`*`.atr`, where *`platform`* is the operating system and *`pc`* is the product code.

Enumerated values are also unformatted, so values shown in the report as **1** and **2**, for example, would show their text equivalent (such as **Started** and **Stopped**) in the portal client. Enumerated attributes are defined in the `k`*`pc`*`.atr` attribute file: `vale` for the display value; `vali` for the unformatted value.

Sample output 1: UNIX OS output from a simple <REPORT> request

```
<REPORTDATA><SQLTABLE><TABLENAME>UNIXOS</TABLENAME>
 <ROWCOUNT>1</ROWCOUNT><ROW><COLUMN><NAME>ORIGINNODE</NAME>
 <DATA><![CDATA[fvaix26:KUX]]></DATA></COLUMN><COLUMN>
 <NAME>SAMPLENO</NAME><DATA>0</DATA></COLUMN><COLUMN>
 <NAME>ROWNO</NAME><DATA>0</DATA></COLUMN><COLUMN><NAME>TIMESTAMP</NAME>
 <DATA><![CDATA[1090629105627000]]></DATA></COLUMN><COLUMN>
 <NAME>SYSTEMTYPE</NAME><DATA><![CDATA[AIX]]></DATA> </COLUMN><COLUMN>
 <NAME>SYSTEMVERS</NAME><DATA><![CDATA[5.3]]> </DATA></COLUMN><COLUMN>
 <NAME>TOTREALMEM</NAME><DATA>3915776</DATA> </COLUMN><COLUMN>
 <NAME>TOTVIRTMEM</NAME><DATA>8634368</DATA> </COLUMN><COLUMN>
 <NAME>SYSUPTIME</NAME><DATA>6633819</DATA> </COLUMN><COLUMN>
 <NAME>NOUSRSESS</NAME><DATA>1</DATA> </COLUMN><COLUMN>
 <NAME>NOSYSPROCS</NAME><DATA>112</DATA> </COLUMN><COLUMN>
 <NAME>NETADDR</NAME>
  <DATA><![CDATA[9.42.11.174]]> </DATA></COLUMN><COLUMN>
 <NAME>UNIXUSRCPU</NAME><DATA>1</DATA> </COLUMN><COLUMN>
 <NAME>UNIXSYSCPU</NAME><DATA>1</DATA> </COLUMN><COLUMN>
 <NAME>UNIXIDLCPU</NAME><DATA>98</DATA> </COLUMN><COLUMN>
 <NAME>UNIXWAITIO</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>VMINRUNQ</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>VMINPGWAIT</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>VMPGFAULTS</NAME><DATA>1538</DATA> </COLUMN><COLUMN>
 <NAME>VMPGRCLM</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>VMPGIN</NAME><DATA>2</DATA></COLUMN> <COLUMN>
 <NAME>VMPGOUT</NAME><DATA>1</DATA></COLUMN> <COLUMN>
 <NAME>VMPGSIN</NAME><DATA>1</DATA></COLUMN> <COLUMN>
 <NAME>VMPGSOUT</NAME><DATA>0</DATA></COLUMN> <COLUMN>
 <NAME>VMFREEMEM</NAME><DATA>7614492</DATA></COLUMN> <COLUMN>
 <NAME>VMFREESWAP</NAME><DATA>1019876</DATA> </COLUMN><COLUMN>
 <NAME>PSWITCH</NAME><DATA>5357</DATA> </COLUMN><COLUMN>
 <NAME>SYSCALL</NAME><DATA>42598</DATA> </COLUMN><COLUMN>
 <NAME>SYSFORK</NAME><DATA>337</DATA> </COLUMN><COLUMN>
 <NAME>SYSEXEC</NAME><DATA>274</DATA> </COLUMN><COLUMN>
 <NAME>BREAD</NAME><DATA>0</DATA></COLUMN> <COLUMN>
 <NAME>BWRITE</NAME><DATA>0</DATA></COLUMN><COLUMN>
 <NAME>LREAD</NAME><DATA>0</DATA></COLUMN><COLUMN>
 <NAME>LWRITE</NAME> <DATA>0</DATA></COLUMN><COLUMN>
 <NAME>PHREAD</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>PHWRITE</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>RCVINT</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>XMTINT</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>MDMINT</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>NETCONNECT</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>NETSOCKET</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>NETLOAD1</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>NETLOAD2</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>NETLOAD3</NAME><DATA>2</DATA> </COLUMN><COLUMN>
 <NAME>MEMFREE</NAME><DATA>108812</DATA> </COLUMN><COLUMN>
 <NAME>MEMUSED</NAME><DATA>3806964</DATA> </COLUMN><COLUMN>
 <NAME>VMSCAN</NAME><DATA>0</DATA></COLUMN> <COLUMN>
 <NAME>VMUSEDPRC</NAME><DATA>119</DATA></COLUMN> <COLUMN>
 <NAME>VMFREEPRC</NAME><DATA>881</DATA></COLUMN> <COLUMN>
 <NAME>CPUBUSY</NAME><DATA>2</DATA></COLUMN> <COLUMN>
 <NAME>SYSREAD</NAME><DATA>5694</DATA></COLUMN> <COLUMN>
 <NAME>SYSWRITE</NAME><DATA>749</DATA></COLUMN> <COLUMN>
 <NAME>NSYSTHRD</NAME><DATA>-1</DATA></COLUMN> <COLUMN>
 <NAME>PRUNABLE</NAME><DATA>112</DATA></COLUMN> <COLUMN>
 <NAME>PRUNNING</NAME><DATA>-1</DATA></COLUMN> <COLUMN>
 <NAME>PSLEEPING</NAME><DATA>0</DATA></COLUMN> <COLUMN>
 <NAME>PIDLE</NAME><DATA>0</DATA></COLUMN><COLUMN>
 <NAME>PZOMBIE</NAME><DATA>0</DATA></COLUMN><COLUMN>
 <NAME>PSTOPPED</NAME><DATA>0</DATA></COLUMN><COLUMN>
 <NAME>THRDRUNQ</NAME><DATA>-1</DATA></COLUMN><COLUMN>
 <NAME>THRDWAIT</NAME><DATA>-1</DATA></COLUMN><COLUMN>
 <NAME>BOOTTIME</NAME>
  <DATA><![CDATA[1090413161248000]]> </DATA></COLUMN><COLUMN>
 <NAME>PENDIOWT</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>STARTIO</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>DEVINT</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>UPTIME</NAME>
  <DATA><![CDATA[076d18:43:39]]> </DATA></COLUMN><COLUMN>
 <NAME>ZATTRIB</NAME><DATA><![CDATA[ ]]> </DATA></COLUMN><COLUMN>
 <NAME>ZVALUE</NAME><DATA><![CDATA[ ]]> </DATA></COLUMN><COLUMN>
 <NAME>SWAPFREE</NAME><DATA>7436</DATA> </COLUMN><COLUMN>
 <NAME>PGINRATE</NAME><DATA>9</DATA> </COLUMN><COLUMN>
 <NAME>PGOUTRATE</NAME><DATA>6</DATA> </COLUMN><COLUMN>
 <NAME>PGSCANRATE</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGINS1</NAME><DATA>1</DATA> </COLUMN><COLUMN>
 <NAME>AVPGINS5</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGINS15</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGINS60</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGOUT1</NAME><DATA>3</DATA> </COLUMN><COLUMN>
 <NAME>AVPGOUT5</NAME><DATA>3</DATA> </COLUMN><COLUMN>
 <NAME>AVPGOUT15</NAME><DATA>3</DATA> </COLUMN><COLUMN>
 <NAME>AVPGOUT60</NAME><DATA>3</DATA> </COLUMN><COLUMN>
 <NAME>AVPGSCAN1</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGSCAN5</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGSCAN15</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPGSCAN60</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>AVPRRUNQ60</NAME><DATA>0</DATA> </COLUMN><COLUMN>
 <NAME>NETADDR6</NAME>
  <DATA><![CDATA[No DNS Entry]]> </DATA></COLUMN><COLUMN>
 <NAME>ZID</NAME><DATA>-1</DATA> </COLUMN><COLUMN>
 <NAME>ZONE</NAME><DATA><![CDATA[-1]]> </DATA></COLUMN>
</ROW></SQLTABLE>
</REPORTDATA>
```

Sample output 2: Report with a filter and columns specified

This is the output from the sample request of Windows Process attributes in the *_Total* row.

```
<REPORTDATA>
 <STATUS>0</STATUS>
 <SQLTABLE>
  <TABLENAME>NTPROCESS</TABLENAME>
  <ROWCOUNT>1</ROWCOUNT>
  <ROW>
   <COLUMN>
    <NAME>ORIGINNODE</NAME>
    <DATA>Primary:DYANG7:NT</DATA>
   </COLUMN>
   <COLUMN>
    <NAME>TIMESTAMP</NAME>
    <DATA>1090303122813634</DATA>
```

```
      </COLUMN>
      <COLUMN>
       <NAME>INSTCNAME</NAME>
       <DATA>_Total</DATA>
      </COLUMN>
      <COLUMN>
       <NAME>PCTPRCSTME</NAME>
       <DATA>99</DATA>
      </COLUMN>
      <COLUMN>
       <NAME>IDPROCESS</NAME>
       <DATA>0</DATA>
      </COLUMN>
      <COLUMN>
       <NAME>THREADCNT</NAME>
       <DATA>1057</DATA>
      </COLUMN>
      <COLUMN>
       <NAME>WRKINGSET</NAME>
       <DATA>1088495616</DATA>
      </COLUMN>
     </ROW>
    </SQLTABLE>
   </REPORTDATA>
```

Agent Service Interface request - Agent table and situation list

Use <TABLESIT> in a Service Interface Request to get a report of the attribute group specified in <TABLENAME> attribute and the situations that are running for the group.

Request input

Table 55. Agent Service Interface <TABLESIT> request

Tag	Description
<TABLESIT>	Enter begin and end TABLESIT tags to retrieve the agent table and situation list.
<SQLTABLE>	The SQLTABLE begin and end tags enclose the TABLENAME tagging pair to identify the SQL table definition set.
<TABLENAME>	The TABLENAME begin and end tags enclose the table name to report. This is the name as it appears bracketed by begin and end tags. If you are not sure what the spelling is of the table, you can see it in the Agent Service Interface Queries report or find it in the **tabl** field of the agent .atr file, located in the <*install_dir*>/TMAITM6/ATTRLIB directory. A value of ***ALL** implies all known agent tables.

Sample request 1: Active Windows OS situations for Process and Logical Disk

```
<TABLESIT>
 <SQLTABLE>
 <TABLENAME>NTPROCESS</TABLENAME>
 </SQLTABLE>
 <SQLTABLE>
 <TABLENAME>WTLOGCLDSK</TABLENAME>
 </SQLTABLE>
</TABLESIT>
```

Report output

Table 56. Agent Service Interface <TABLESIT> request output.

Output tag	Description
<SITUATION>	Defines the output situation set.
<NAME>	Specifies the situation name.
<TYPE>	E – Enterprise situation; P – Private situation
<STATUS>	Returns the status code bracketed by begin and end tags

Sample output: The Windows OS agent returns all running Process and Logical Disk situations

```
<TABLESIT>
 <SQLTABLE>
 <TABLENAME>NTPROCESS</TABLENAME>
  <SITUATION>
   <NAME>Check_Process_CPU_Usage</NAME>
  </SITUATION>
  <TYPE>P</TYPE>
  <SITUATION>
   <NAME>Check_Process_CPU_Usage</NAME>
 </SITUATION>
  <TYPE>P</TYPE>
  <SITUATION>
   <NAME>Is_KFC_Running</NAME>
 </SITUATION>
 <TYPE>E</TYPE>
 </SQLTABLE> <SQLTABLE>
  <TABLENAME>WTLOGCLDSK</TABLENAME>
  <SITUATION>
  <NAME>Check_DiskSpace_Low</NAME>
 </SITUATION>
 <TYPE>P</TYPE>
 </SQLTABLE>
</TABLESIT>
```

Agent Service Interface request - Private situation control

Create a service interface <PVTCONTROL> request to start, stop, or recycle a private situation on the monitoring agent.

Private situations start running automatically when the monitoring agent they are written for, whether a Tivoli Enterprise Monitoring Agent or a Tivoli System Monitor Agent, is started. The PVTCONTROL command enables you to start, stop, or recycle the specified situation without having to stop and restart the agent.

Request input

Table 57. Agent Service Interface <PVTCONTROL> request.

Tag	Description
<PVTCONTROL>	Specify private situation control request.
<PVTCOMMAND>	Specify private situation command.
<PVTSITNAME>	Specify private situation name.
<PVTACTION>	START – Start a known situation request. STOP – Stop an active situation. RECYCLE – Stop and restart an active situation.

Sample request 1: Recycle private situation Check_DiskSpace_Low

```
<PVTCONTROL>
 <PVTCOMMAND>
  <PVTSITNAME>Check_DiskSpace_Low</PVTSITNAME>
  <PVTACTION>RECYCLE</PVTACTION>
 </PVTCOMMAND>
</PVTCONTROL>
```

Report output

Table 58. Agent Service Interface <PVTCONTROL> request output.

Output tag	Description
<STATUS>	Return status code bracketed by begin and end tag.

Sample output 1: Recycle private situation Check_DiskSpace_Low returns the command status

```
<PVTCONTROL>
 <STATUS>0 Request completed successfully</STATUS>
</PVTCONTROL>
```

Agent Service Interface request - Situation summary

Use the situation summary command to request a listing of the private situations that are running on the monitoring agent.

Request input

Table 59. Agent Service Interface <SITSUMMARY> request.

Tag	Description
<SITSUMMARY>	Situation summary report.

```
<SITSUMMARY>
</SITSUMMARY>
```

The output from the request looks like the private situation configuration files shown in the “Private situation examples” on page 351.

Report output

Table 60. Agent Service Interface <SITSUMMARY> request output.

Output tag	Description
<NUMBSIT>	The number of private situations running on the monitoring agent.
<SITUATION>	Define situation selection properties.
<NAME>	Specifies the situation name.
<SOURCE>	Indicates the source of the event. The value “E” means enterprise situation. The value “P” means private situation.
<TYPE>	Situation type – Sample or Pure-Event.
<INTERVAL>	Situation sample interval; or 0 – Pure-Event.
<ROWSIZE>	Sample data row size.
<FIRSTSTARTTIME>	The situation's initial start time.
<LASTSTARTTIME>	The situation's most recent start time.
<LAST STOPTIME>	The situation's last stop time.

Table 60. Agent Service Interface <SITSUMMARY> request output. (continued)

Output tag	Description
<FIRSTEVENTTIME>	The time that the situation first opened an event.
<LASTTRUETIME>	The last time the situation evaluated to **True**.
<LASTFALSETIME>	The last time the situation evaluated to **False**.
<TIMESRECYCLED>	Number of times the situation restarted.
<TIMESAUTONOMOUS>	Number of times the situation entered autonomous mode.
<PREDICATE>	The situation formula.

Sample output: The agent returns private situation data

```
<SITSUMMARY>
<NUMBSIT>12</NUMBSIT>
<SITUATION>
<NAME>NT_AMS_Alert_Critical</NAME>
<SOURCE>E</SOURCE><TYPE>Event</TYPE><INTERVAL>0</INTERVAL>
<ROWSIZE>484</ROWSIZE>
<FIRSTSTARTTIME>Sun Aug 18 21:34:41 2013</FIRSTSTARTTIME>
<LASTSTARTTIME>Sun Aug 18 21:34:41 2013</LASTSTARTTIME>
<LASTSTOPTIME>NA</LASTSTOPTIME><FIRSTEVENTTIME>NA</FIRSTEVENTTIME>
<LASTTRUETIME>NA</LASTTRUETIME><LASTFALSETIME>NA</LASTFALSETIME>
<TIMESRECYCLED>0</TIMESRECYCLED><TIMESAUTONOMOUS>1</TIMESAUTONOMOUS>
<PREDICATE><![CDATA[Alert_Text = 5  OR  Alert_Text = 8  OR
Alert_Text = 7  OR  Alert_Text = 6  OR  Alert_Text = 9  OR
Alert_Text = 11 ]]>
</PREDICATE>
</SITUATION>
<SITUATION>
<NAME>NT_Invalid_Logon_Attempt</NAME>
<SOURCE>E</SOURCE><TYPE>Event</TYPE><INTERVAL>0</INTERVAL>
<ROWSIZE>3132</ROWSIZE>
<FIRSTSTARTTIME>Sun Aug 18 21:34:41 2013</FIRSTSTARTTIME>
<LASTSTARTTIME>Sun Aug 18 21:34:41 2013</LASTSTARTTIME>
<LASTSTOPTIME>NA</LASTSTOPTIME><FIRSTEVENTTIME>NA</FIRSTEVENTTIME>
<LASTTRUETIME>NA</LASTTRUETIME>
<LASTFALSETIME>Sun Aug 18 21:37:05 2013</LASTFALSETIME>
<TIMESRECYCLED>0</TIMESRECYCLED><TIMESAUTONOMOUS>1</TIMESAUTONOMOUS>
<PREDICATE><![CDATA[Event_ID = 529  OR  Event_ID = 4625 ]]>
</PREDICATE>
</SITUATION>
<SITUATION><NAME>NT_Log_Space_Low</NAME>
<SOURCE>E</SOURCE><TYPE>Sampled</TYPE><INTERVAL>900</INTERVAL>
<ROWSIZE>1256</ROWSIZE>
<FIRSTSTARTTIME>Sun Aug 18 21:34:41 2013</FIRSTSTARTTIME>
<LASTSTARTTIME>Sun Aug 18 21:34:41 2013</LASTSTARTTIME>
<LASTSTOPTIME>NA</LASTSTOPTIME><FIRSTEVENTTIME>NA</FIRSTEVENTTIME>
<LASTTRUETIME>Sun Aug 18 21:34:49 2013</LASTTRUETIME>
<LASTFALSETIME>NA</LASTFALSETIME><TIMESRECYCLED>0</TIMESRECYCLED>
<TIMESAUTONOMOUS>1</TIMESAUTONOMOUS>
<PREDICATE><![CDATA[%_Usage >= 95 ]]>
</PREDICATE>
</SITUATION>
...
...
</SITSUMMARY>
```

Agent Service Interface request - Agent monitoring statistics

Use the agent monitoring statistics command to request information about the monitoring agent activity.

Request input

Table 61. Agent Service Interface <AGENTSTAT> request.

Tag	Description
<AGENTSTAT>	Specify Agent statistic request.
<SITUATION>	Define situation selection properties.
<NAME>	Specify the situation name or *ALL for all known situations. Default: ***ALL**
<DAYS>	Optional. Specify the period to display, such as 1 for today's data. Up to 7 days history data can be retrieved.
<DETAILS>	Optional. Yes – output hourly detail data No- output state information only Default: **No**

Sample request 1: Retrieve today's situation state statistics

```
<AGENTSTAT>
 <SITUATION>
  <NAME>*ALL</NAME>
 </SITUATION>
</AGENTSTAT>
```

Sample request 2: Retrieve today's NT_Service_Error situation statistics

```
<AGENTSTAT>
 <SITUATION>
  <NAME>NT_Service_Error</NAME>
  <DAYS>1</DAYS>
  <DETAILS>Y</DETAILS>
</SITUATION>
</AGENTSTAT>
```

Report output

Table 62. Agent Service Interface <AGENTSTAT> request output.

Output tag	Description
<TYPE>	Situation type – Sample or Pure-Event.
<INTERVAL>	Situation sample interval; or 0 – Pure-Event.
<ROWSIZE>	Sample data row size.
<FIRSTSTARTTIME>	The situation's initial start time.
<LASTSTARTTIME>	The situation's most recent start time.
<LAST STOPTIME>	The situation's last stop time.
<FIRSTEVENTTIME>	The time that the situation first opened an event.
<LASTTRUETIME>	The last time the situation evaluated to **True**.
<LASTFALSETIME>	The last time the situation evaluated to **False**.
<TIMESRECYCLED>	Number of times the situation restarted.
<TIMESAUTONOMOUS>	Number of times the situation entered autonomous mode.
<DAY>	Begin daily metrics.
<DATE>	Date description.
<TRUESAMPLES>	True sample row count.
<FALSESAMPLES>	False sample row count.
<TRUERATIO>	Percent of true samples.

Table 62. Agent Service Interface <AGENTSTAT> request output. (continued)

Output tag	**Description**
<FALSERATIO>	Percent of false samples.
<HOURROWS>	Hourly sample row count.
<HOURTRUE>	Hourly true sample row count.
<HOURFALSE>	Hourly false sample row count.
<STATUS>	Return status code bracketed by begin and end tag.

Sample output from sample request 2: The agent returns today's NT_Service_Error situation statistics

```
<SITSTATS>
 <SITUATION>
  <NAME>NT_Service_Error</NAME>
  <TYPE>Event</TYPE>
  <INTERVAL>0</INTERVAL>
  <ROWSIZE>3124</ROWSIZE>
  <FIRSTSTARTTIME>Thu Mar 12 23:09:36 2009</FIRSTSTARTTIME>
  <LASTSTARTTIME>NA</LASTSTARTTIME>
  <LASTSTOPTIME>NA</LASTSTOPTIME>
  <FIRSTEVENTTIME>NA</FIRSTEVENTTIME>
  <LASTTRUETIME>NA</LASTTRUETIME>
  <LASTFALSETIME>Fri Mar 13 22:53:31 2009</LASTFALSETIME>
  <TIMESRECYCLED>0</TIMESRECYCLED>
  <TIMESAUTONOMOUS>0</TIMESAUTONOMOUS>
 <DAY>
   <DATE>Fri Mar 13 00:00:00 2009</DATE>
    <TRUESAMPLES>0</TRUESAMPLES>
   <FALSESAMPLES>80</FALSESAMPLES>
   <TRUERATIO>0.00%</TRUERATIO>
   <FALSERATIO>100.00%</FALSERATIO>
    <HOURROWS>0 0 0 0 0 0 0 12 2 2 4 6 0 4 4 6 0 2 5 4 0 15 14 0
     </HOURROWS>
   <HOURTRUE>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     </HOURTRUE>
   <HOURFALSE>0 0 0 0 0 0 0 12 2 2 4 6 0 4 4 6 0 2 5 4 0 15 14 0
   </HOURFALSE>
 </DAY>
 </SITUATION>
</SITSTATS>
```

Agent Service Interface request - History report

Create a service interface <HISTREAD> request to start, stop, or recycle a private situation on the monitoring agent.

Historical data from Tivoli Enterprise Monitoring Agents is displayed in the Tivoli Enterprise Portal when you select a time span for a table view or other query-based view. Outside the portal, you can see historical data from an enterprise monitoring agent or private history data from an enterprise monitoring agent or system monitor agent by getting a History report from the Agent Service Interface or by creating a HISTREAD service interface request.

Request input

Table 63. Agent Service Interface <HISTREAD> request.

Tag	**Description**
<HISTREAD>	Retrieve history table data
<SQLTABLE>	Identify SQL table definition set.

Table 63. Agent Service Interface <HISTREAD> request. (continued)

Tag	Description
<TABLENAME>	Defines table name bracketed by begin and end tags. Maximum table name consists of 10 characters.
<PVTHIST>	Optional. Specify reading private history. No direct agent to read enterprise short term history
<OUTPUT>	Optional. Define output table column selection.
<COLUMN>	Optional. Define selected column name bracketed by begin and end tags.
<FILTER>	Optional. Define output data rows filter criteria bracketed by begin and end tags. See Private Situation <CRITERIA> specification for details. Use from and to WRITETIME column to specify history data read range. Use ORIGINNODE column to select specific MSN history data
<OUTLIMIT>	Optional. Define the output record limit bracketed by begin and end OUTLIMIT tags to safeguard against to much output volume.

Sample request 1: Get Windows OS agent Process history data using filter and column selections

```
<HISTREAD>
 <SQLTABLE>
 <TABLENAME>NTPROCESS</TABLENAME>
 <OUTPUT>
  <COLUMN>ORIGINNODE</COLUMN>
  <COLUMN>TIMESTAMP</COLUMN>
  <COLUMN>INSTCNAME</COLUMN>
  <COLUMN>IDPROCESS</COLUMN>
  <COLUMN>PCTPRCSTME</COLUMN>
  <COLUMN>THREADCNT</COLUMN>
  <COLUMN>WRKINGSET</COLUMN>
 </OUTPUT>
 <FILTER>
 <![CDATA[ *VALUE ORIGINNODE *EQ Primary:DYANG3:NT  *AND
   *VALUE WRITETIME  *GE 1090408224500000  *AND
   *VALUE WRITETIME  *LE 1090408234500000]]>
 </FILTER>
 <OUTLIMIT>5000</OUTLIMIT>
 </SQLTABLE>
</HISTREAD>
```

Report output

Table 64. Agent Service Interface <HISTREAD> request output.

Output tag	Description
<HISTREADDATA>	Identify output report data set.
<STATUS>	Return status code bracketed by begin and end tag
<ROWCOUNT>	Output table row count
<ROW>	Identify an output row data
<NAME>	Define output column name enclosed by begin and end tags.
<DATA>	Specify output column data value enclosed by begin and end tags.

Sample output 1: The Windows OS agent returns Process history data for the seven specified attributes on April 8 from 10:45 PM to 11:45 PM

```
<HISTREADDATA>
 <SQLTABLE>
  <TABLENAME>NTPROCESS</TABLENAME>
  <ROWCOUNT>212</ROWCOUNT>
  <ROW>
   <COLUMN>
  <NAME>ORIGINNODE</NAME>
    <DATA><![CDATA[Primary:DYANG3:NT]]></DATA>
   </COLUMN>
   <COLUMN>
  <NAME>TIMESTAMP</NAME>
    <DATA><![CDATA[1090408224551430]]></DATA>
   </COLUMN>
   <COLUMN>
   <NAME>INSTCNAME</NAME>
    <DATA>![CDATA[Idle]]> </DATA>
   </COLUMN>
   <COLUMN>
    <NAME>IDPROCESS</NAME>
   <DATA>0</DATA>
   </COLUMN>
   <COLUMN>
   <NAME>PCTPRCSTME</NAME>
   <DATA>74</DATA>
   </COLUMN>
   <COLUMN>
   <NAME>THREADCNT</NAME>
   <DATA>1</DATA>
   </COLUMN>
   <COLUMN>
   <NAME>WRKINGSET</NAME>
   <DATA>16384</DATA>
   </COLUMN>
  </ROW>
  ...
  ...
 </SQLTABLE>
</HISTREADDATA>
```

Agent Service Interface request - Configuration control

The Service Interface Request can be used to process configuration load list requests.

Authorization

The Service Interface Request recognizes the complete configuration load list XML syntax. The requests that are allowed depend on group permissions:

- If your user ID is a member of the **Operation** group in the Access Authorization Group Profile (AAGP), you can use the <CNFGCOMMAND> element to refresh files using an existing configuration load list, and can issue <CNFGACTION> Reboot, Reload, and Download requests.
- If your user ID is a member of the **Administrative** group in the AAGP, you can submit any valid configuration load list request using the syntax in the Configuration load list XML specification.

Elements

The elements and their attributes are case-insensitive. For example, you can enter <CNFGCOMMAND>, <CnfgCommand>, or <cnfgcommand>.

<CNFGCOMMAND>
Specify configuration command request. The following example of a configuration control request reloads the contents of the current configuration load list:

```
<CNFGCOMMAND>
  <CNFGACTION>RELOAD</CNFGACTION>
</CNFGCOMMAND>
```

<CNFGACTION>
Specify the configuration command action:

Reboot
This attribute downloads the configuration load list.

Reload
Reload is used to perform an immediate resend of the current agent load list, thereby downloading all specified agent artifacts if they have been updated. See the example under <CnfgCommand>.

Download
Download is used to specify the file to send. See the examples under <CnfgFile> and <CnfgDisp>.

<CNFGFILE>
Optional. Specific last two segments of file name when <CNFGACTION> is Download.

File name must exist in the load list. Download file alert.txt

```
<CNFGCOMMAND>
  <CNFGACTION>DOWNLOAD</CNFGACTION>
  <CNFGFILE>alert.txt</CNFGFILE>
</CNFGCOMMAND>
```

<CNFGDISP>
Optional. Specific well-known configuration file disposition as identification of file name when <CNFGACTION> is Download. The file definition must exist in load list. The following example request downloads the private situation configuration XML file:

```
<CNFGCOMMAND>
  <CNFGACTION>DOWNLOAD</CNFGACTION>
  <CNFGDISP>PVTSIT</CNFGDISP>
</CNFGCOMMAND>
```

<STATUS>
Use this element to return the status code within the beginning <status> and ending </status> tags. The following example causes the agent to return the command status:

```
<CNFGCOMMAND>
  <STATUS>600 - Configuration control command completed successfully
  </STATUS>
</CNFGCOMMAND>
```

Centralized Configuration

Use the Centralized Configuration feature to maintain monitoring agent configuration files in a central location that agents can collect from.

Centralized Configuration overview

Centralized Configuration provides the ability to update local configuration files on many monitoring agents without connection to a Tivoli Enterprise Monitoring Server.

These are some of the benefits of Centralized Configuration:

- Ensures consistent agent installations
- Reduces installation support and configuration complexity
- Improves agent deployment efficiency
- Enhances Tivoli Monitoring scalability
- Leverages users' web management skills

Centralized Configuration is a Tivoli monitoring agent or (preferably) web server acting as a repository of agent configuration files that are pulled by monitoring agents on the same or different computers using their local *configuration load list*. The repository can contain such files as the configuration XML for SNMP alerts and EIF events, automation scripts, and any other pertinent agent operational files. The configuration load list specifies the central server location and the configuration files to get.

The *central configuration server* can be a Tivoli Monitoring agent at version 6.2.2 Fix Pack 2 or later, or it can be any web server, such as WebSphere, IBM HTTP, Microsoft IIS, or Apache. For security reasons, IBM recommends you choose a web server.

You can have multiple central configuration servers and logically arrange them as a hierarchy of central configuration servers.

After Centralized Configuration has been initiated by the agent, the default behavior is to pull any file updates from the designated central configuration server every hour. You can also get updates on-demand by entering the load list as an Agent Service Interface request.

Agents dynamically activate newly downloaded well-known configuration files, such as private situations and configuration load lists, without agent restart. Other configuration changes that require the agent to restart to enable the new changes to take effect can include an agent restart specification so that these configuration updates take effect immediately without intervention.

The basic tasks for implementing Centralized Configuration for existing agents are:

1. Decide on a strategy to organize and distribute configuration files from a central repository and create the configuration files.
2. Configure the central configuration server.
3. Enable the agent to collect the initial configuration load list.
4. Update configuration files as needed on the central configuration server.

Centralized Configuration design

The Centralized Configuration structure that you define depends on the size and organization of your monitored enterprise, the types of monitoring agents you want to maintain, what kinds of updates, and how often.

Configuration load list

Any newly installed monitoring agent or an existing agent can participate in Centralized Configuration by using a configuration load list. The configuration load list is an XML configuration file that is unique to the running agent. It tells the agent how to connect to one or more central configuration servers and what files to download from those servers.

Administrators maintain and update the configuration file in this centralized repository. New agents collect initial configuration files from this location and periodically or on-demand contact the server to collect any updates or changes.

Central configuration server

The configuration load list contains one or more ConfigServer elements that tell the agent how to connect to a central configuration server. Central configuration servers contain a repository of files that are served to the monitoring agents using HTTP.

The server authenticates requests, examines the configuration article last update time stamp (set to GMT), and, if the client copy of the requested file is older than the server copy, the server returns the configuration article contents to the agent. Otherwise, it returns HTTP status 304 - Object Not Modified. The server returns other HTTP status if an error is encountered during article processing.

When deploying a central configuration server, consider these factors:

User access

Administrators require access to maintain the configuration load list that are to be distributed. The central configuration clients need access to the central configuration server to collect the updates. If you already have a web server and are familiar with access and maintaining files there, you can use any available web server to act as a central configuration server. This includes the web server for the Agent Service Interface.

Directory structure

Identify a directory structure on the server and client that enable you organize your files, to minimize any duplication on the server, and to reduce the number of files that must be maintained.

Keyword substitution

Using keyword substitution in the configuration load list can simplify the organization of configuration files.

For example, if all of the Linux OS agents run one set of private situations with events defined and the Windows OS agents run another set, the @PRODUCT@ keyword can be used to direct the agents to the correct directory and file on the central configuration server. Place the files in the *install_dir*/localconfig directory or as specified by the IRA_SERVICE_INTERFACE_CONFIG_HOME environment variable. For example,

📂 common
    cnfglist.xml

📂 lz
    lz_situations.xml
    lz_trapcnfg.xml

📂 nt
    nt_situations.xml
    nt_trapcnfg.xml

The cnfglist.xml file can use the @PRODUCT@ keyword to direct the agents to the correct files. Example:

```
<ConfigurationArtifact>
  <ConfigServer
   Name="CENTRAL-CONFIG-SERVER"
   URL="http://icvr5a05.tivlab.raleigh.ibm.com/"
   User="itmuser"
   Password="{AES256:keyfile:a}vHBiEqmmvylNPs90Dw1AhQ==" />
  <ConfigFile
   Server="CENTRAL-CONFIG-SERVER"
   Name="cnfglist.xml"
   Path="common"
   Disp="CNFGLIST"
   Activate="YES" />
  <ConfigFile
   Server="CENTRAL-CONFIG-SERVER"
   Name="@PRODUCT@_situations.xml"
   Path="@PRODUCT@"
   Disp="PVTSIT"
   Activate="YES" />
  <ConfigFile
   Server="CENTRAL-CONFIG-SERVER"
   Name="@PRODUCT@_trapcnfg.xml"
   Path="@PRODUCT@"
   Disp="TRAPCNFG"
   Activate="RESTART" />
</ConfigurationArtifact>
```

Other keywords can be used to create the granularity required for each agent to get the correct files.

Web server as the central configuration server

Tivoli monitoring agents can use existing web servers to collect configuration files. You can use any web server that the agents can access with HTTP or HTTPS as a central configuration server.

To use a web server, create a user ID that has permission to access the files on the central configuration server and reference those credentials in the configuration load list. The URL specification is slightly different than when the central configuration server is a monitoring agent, but that is the only difference in the configuration load list. The ConfigServer element looks like this:

```
<ConfigServer
   Name="CENTRAL-CONFIG-SERVER"
   URL="http://webserver.domain.com/"
   User="itmuser"
   Password="{AES256:keyfile:a}vHBiEqmmvylNPs90Dw1AhQ==" />
```

Monitoring agent as the central configuration server

All monitoring agents (enterprise or system monitor) that run on Windows, Linux and UNIX platforms contain an HTTP-based Service Interface that can be used as a central configuration server. Monitoring agents on z/OS and i5 do not provide an HTTP Service interface, so cannot be used as central configuration servers.

The advantage of using a monitoring agent as the central configuration server rather than a web server, is that you do not need to maintain a web server and you can use several agents to form a set of cascading central configuration server to provide some workload balancing.

User access to the central configuration server

Access to an agent's Service Interface is controlled using its Access Authorization Group Profile (AAGP). The ability to request or place files on an agent requires more security rights than viewing metrics or historical data. By default, any valid ID on the host computer where the agent is running can access the Agent Service Interface. Those users can view metrics, situations, and historical data collected by the agent. However, the default behavior gives only the **Administrative** ID on the host the permission to access central configuration files through the agent's Service Interface.

The following user IDs are default members of the **Administrative** group on the platform where the agent is running:

Windows Administrator

Linux UNIX root

z/OS ITMUSER

IBM i QSECOFR

You can create configuration load lists and use the **Administrative** credentials to connect to the monitoring agent that is acting as the central configuration server. The ID does not have to exist on the client agents for them to connect to the central configuration server.

On Windows systems, the agent automatically discovers all user accounts belonging to the AAGP Administrators group; the agent adds those accounts to the AAGP Administrative group at startup. On UNIX and Linux systems, the agent automatically discovers all user IDs having root (superuser) authority; the agent adds them to AAGP Administrative group at startup. (Userid QSECOFR on IBM i and ITMUSER on z/OS remain unchanged at this time.) As results:

- Customers do not need to create special user account to use agent services.
- Administrator/root ID exposure is avoided.
- Any authorized account on a system automatically gains access to privileged agent services such as file access, AAGP customization, and so on.

Although user IDs can be stored in encrypted format, in most cases you want to define a user ID on the system that will be hosting the central configuration server and add that user to the agent's AAGP.

The following example gives the steps for creating a user on the Linux OS agent and granting administrative access to the central configuration server there:

1. Create the user on the system: The user named “itmuser” is created on the Linux operating system and the Linux OS agent (lz) is the central configuration server.
2. Create an AAGP.xml file in the *install_dir*/`localconfig` directory that adds the new ID to the **Administrative** group:

   ```
   <AAGP>
     <AAUSER>
       <ID>itmuser</ID>
       <ASSIGN>AD</ASSIGN>
     </AAUSER>
   </AAGP>
   ```

   The default local configuration directory can be changed with the IRA_SERVICE_INTERFACE_CONFIG_HOME environment variable.

   Tip: Consider having each agent collect an AAGP file so that you can contact the agent directly through its Agent Service Interface to perform configuration actions with a non-root ID. By connecting directly to an agent's Service Interface with AD permission, you can provide credentials to connect to a new central configuration server, put or get files, or force immediate refreshes of configuration files.

3. On the monitoring agent that functions as the central configuration server, edit the configuration load list to add a DISP="AAGP" ConfigFile entry to load the AAGP XML file.

   This load list must use credentials (Linux UNIX root; Windows Administrator) to connect to itself to add the new user ID to the AAGP. The edited configuration load list looks like this:

   ```
   <ConfigurationArtifact>
      <ConfigServer
       Name="CENTRAL-CONFIG-SERVER"
       URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
       User="root"
       Password="{AES256:keyfile:a}vHBiEqmmvylNPs90Dw1AhQ==" />
      <ConfigFile
       Server="CENTRAL-CONFIG-SERVER"
       Name="AAGP.xml"
       Path="/"
       Disp="AAGP" />
   </ConfigurationArtifact>
   ```

4. Save the load list in *install_dir*/localconfig/lz/lz_cnfglist.xml. This directory is the default location on Linux systems and can be changed with the IRA_SERVICE_INTERFACE_CONFIG_LOADLIST environment variable.
5. Start the Linux OS agent.

All of the central configuration clients that connect to this central configuration server can now use the credentials for “itmuser” rather than “root”.

```
  <ConfigServer
    Name="CENTRAL-CONFIG-SERVER"
    URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
    User="itmuser"
    Password="{AES256:keyfile:a}vHBiEqmmvylNPs90Dw1AhQ==" />
```

Cascading central configuration servers

Other monitoring agents can collect an Access Authorization Group Profile (AAGP) update from a central configuration server and then be used to distribute files to other agents.

Cascading servers must be configured so that they distribute configuration load lists that specify their Agent Service Interface for the ConfigServer element's URL.

Cascading servers use the DISP=CUSTOM ConfigFile elements to download the content that they distribute to other agents.

Agent service interface

The Service Interface for an agent can be used to enter and process configuration load list requests on demand. See “Agent Service Interface request - Configuration control” on page 423.

Configuration load list XML specification

Use the XML syntax from the configuration load list XML specification to create a load list for Centralized Configuration.

Default configuration load list path and file name

The following configuration load list file names are the defaults, where *pc* is the two-character product code:

Windows *install_dir*\localconfig\pc\pc_cnfglist.xml

Linux UNIX *install_dir*/localconfig/*pc*/*pc*_cnfglist.xml

z/OS IRA_SERVICE_INTERFACE_CONFIG_LOADLIST= *PC*CFGLST.RKANDATV

IBM i /QIBM/UserData/IBM/ITM/localconfig/a4/a4_cnfglist.xml

Use the IRA_SERVICE_INTERFACE_CONFIG_LOADLIST agent environment variable to change the default path and file name. See "Environment variables for Centralized Configuration" on page 439.

Elements

XML element tags and their attributes are **not case-sensitive**, but values are **case-sensitive**. For example, you can enter <CONFIGSERVER>, <ConfigServer>, or <configserver>.

<ConfigurationArtifact>

ConfigurationArtifact is the root element that identifies this XML file as a load list configuration document. Enter <ConfigurationArtifact> at the beginning of the file and </ConfigurationArtifact> at the end.

Example of a load list file:

```
<ConfigurationArtifact >
  <ConfigServer
   Name="AGOURALAB"
   URL="http://9.55.100.99/"
   User="smith"
   Password="{AES256:keyfile:a}hjZM0YaLzpd5JQC5DeboJg==" />
  <ConfigFile
   Server="AGOURALAB"
   Name="Private_Situations.xml"
   Path="ITM/Config/@HOSTNAME@"
   Disp="PVTSIT"
   Activate="YES" />
  <ConfigFile
   Server="AGOURALAB"
   Name="TRAPCNFG.xml"
   Path="ITM/Config/common"
   Disp="trapcnfg"
   Activate="RESTART" />
  <ConfigFile
   Server="AGOURALAB"
   Name="@PRODUCT@_eventdest.xml"
   Path="DataStore/@PRODUCT@"
   Disp="EIFCNFG"
   Activate="YES" />
  <ConfigFile
   Server="AGOURALAB"
   Name="@PRODUCT@_eventmap.xml"
   Path="DataStore/@PRODUCT@"
   Disp="EIFMAP"
   Activate="YES" />
</ConfigurationArtifact>
```

<ConfigServer>

Define a central configuration server with the following attributes:

Name=
Defines a symbolic name of a SERVER statement. The name can be up to 32 characters and must be unique in the load list. Duplicate names are not permitted.

URL= Defines the URL used to connect to the central configuration server, which can be either of the following types:

- An Agent Service Interface acting as a central configuration server and specified as:

```
HTTP-method://Hostname:port
///agent-ServicePoint/agent-ServicePoint
```

- A web server acting as a central configuration server and specified as:

```
HTTP-method://Hostname:[port]
/[path]
```

where:

HTTP-method
is http or https

Hostname
is the central configuration server host name. Use the IP address if the Hostname is not guaranteed to be resolved by local DNS.

port is the target central configuration server listening port of the Tivoli Monitoring Service Index (KDH component code) or of the web server if the port is different than the default 80. The Tivoli Monitoring Service Index default port is 1920 for HTTP or 3661 for HTTPS, either of which can be customized with the KDC_FAMILIES environment variable in the target agent.

agent-ServicePoint
is the TMS/Engine-registered Agent Service Interface name, such as `system.myhost_nt` for the Windows OS agent and `myhost_lz` for the Linux OS agent. You can customize the service name of the target agent to a more functionally recognized name using the IRA_SERVICE_INTERFACE_NAME agent environment variable. For example, `https://9.48.123.13:3661///Paris-CSF-A/Paris -CSF-A`. Omit the agent-instance-name definition if you are using a generic web server configuration repository.

Path is an additional target central configuration server path definition. For example, `http://9.48.132.40:80/ITM/config`.

User= Optional. Specifies the target central configuration server server host system account user ID.

Password=
Optional. Specify user password in plain text or use the itmpwdsnmp utility program to encrypt user password and specify the output AES data string here. See “SNMP PassKey encryption: itmpwdsnmp” on page 372.

AltServer=

Optional. Specifies an alternate server name. The agent constructs the file request URL using the alternate server definition when it cannot contact or log on to this server. The alternate server definition cannot include an additional alternate server specification.

The following HTTP status codes cause the agent to retry the request using the **AltServer** specification:

401	Unauthorized
403	Forbidden
404	Object not found
500	Internal server error
503	Service unavailable

If you are using the AltServer attribute to specify an alternate central configuration server, be sure to define the alternate <ConfigServer> before the <ConfigServer> definition that references it. Example:

```
<ConfigServer
 Name="CENTRAL-CONFIG-ALTERNATE"
 URL="http://lnxhostB:1920///lnxhostB_lz/lnxhostB_lz"
 User="itmconfig"
 Password="{AES256:keyfile:a}vHBiEqmmvylNPs90DhQ==" />
<ConfigServer
 Name="CENTRAL-CONFIG-REPOSITORY"
 URL="http://lnxhostA:1920///lnxhostA_lz/lnxhostA_lz"
 User="itmconfig"
 Password="{AES256:keyfile:a}hjZM0YaLzpd5JQC5DJg=="
 AltServer="CENTRAL-CONFIG-ALTERNATE" />
```

See also the example in ““Environment variables in the configuration load list” on page 437.”

<ConfigFile>

Identifies a previously defined <ConfigServer> by name. The agent sends the request to this **Server** for downloading this particular file.

You can reference environment variables when defining <ConfigFile> element attributes. Where the variables are resolved depends the type of central configuration server:

- When connection is to a monitoring agent acting as the central configuration server, environment variables used to define ConfigFile, Name, and Path attributes are resolved at the server.
- When connection is to a web server acting as the central configuration server, *all* environment variables are resolved at the client.

Server=

Identifies a previously defined <ConfigServer> by name. The agent sends the request to this **Server** for downloading this particular file.

Name=

Specifies the file name at the server location. The name can include environment variables if they can be recognized and resolved by the agent.

Path= Specifies the file location path on the target <ConfigServer>. The path can include environment variables that are resolved at the server when connecting to a central configuration server. The variables are resolved at the central configuration client when the connection is to a web server.

Path is relative from the HTTP root of the configuration server. When using a monitoring agent as the configuration server, the *install_dir*/localconfig directory is the HTTP root. The configuration file anchor can be overridden by the IRA_SERVICE_INTERFACE_CONFIG_HOME environment variable.

Disp= Optional. Specifies the local disposition of known agent configuration files. When the DISP attribute is specified, the agent places the downloaded ConfigFile (or ConfigFiles) in the correct location on the central configuration client system based on the environment settings of the client agent. The DISP attribute also enables activation options that are appropriate for each ConfigFile. The agent knows the local location and name of these agent configuration files and, upon downloading them, saves them according to the agent specification.

If the Disp attribute is omitted, CUSTOM is used by default. The placement of a file when using Disp=CUSTOM is restricted to locations within the Tivoli Monitoring *install_dir*. This restriction is in place to enhance security. Other Disp file types are placed wherever the monitoring agent expects the file to be, even if the specified location is outside the *install_dir* directory structure. If you specify Disp=CUSTOM, it is helpful to use the @ITMHOME@ keyword to specify the LocalPath.

The following disposition values are currently defined:

CNFGLIST
Configuration load list file.

PVTSIT
Private Situation configuration XML file.

TRAPCNFG
Agent SNMP trap configuration XML file.

EIFCNFG
EIF configuration file.

EIFMAP
EIF event mapping file.

UAMETA
Universal Agent application Meta files.

PASCAP
Proxy Agent Service (Agent Management Services) common agent package (CAP) file. The PASCAP Disp can be used only on a Tivoli Monitoring OS Agent that supports Agent Management Services. It downloads and activates the CAP file if the product is installed. Options=NOPRODUCTCHECK can be used to force the placement of the CAP file even if the product that the CAP file manages is not installed. (See Tivoli Agent Management Services installation and configuration.)

`AAGP`
Access Authorization Group Profile, which contains access authorization group definitions and user ID access authorization group assignments that were defined by the security administrator. (See "Access Authorization Group Profile" on page 398.)

`CUSTOM`
CUSTOM is the default value used if DISP is omitted. The **LocalName** and **LocalPath** attributes should be specified for DISP=CUSTOM. CUSTOM files can be downloaded only to a location within the agent's installation directory structure.

Options=
Specifies the configuration file handling option:

`NOPRODUCTCHECK`
Valid only with DISP=PASCAP. Bypass product installation requirement before the CAP file download operation. Currently this is the only option value defined.

LocalName=
Specifies the local system file name. LocalName is used only if DISP is omitted or is set to CUSTOM. If LocalName is omitted when Disp=CUSTOM, the NAME attribute is used. LocalName is ignored for all other Disp values because the agent identifies the location of the file using default values or override parameters.

LocalPath=
Specifies the local system file path. LocalPath is used only if DISP is omitted or is set to CUSTOM. If LocalPATH is omitted when Disp=CUSTOM, the PATH attribute is used. LocalPath is ignored for all other Disp values because the agent identifies the location of the file using default values or override parameters.

Activate=

`NO` Replace current file with downloaded copy, but do not activate. The downloaded file must be newer than the existing configuration file. This is the default value.

`RESTART`
Restart the agent after a successful file download.

The Disp=TRAPCNFG type requires the agent to restart in order to read the updated configuration files. RESTART is also available for Disp=CUSTOM.

RESTART can be used with PVTSIT to force a replacement of the private situation definition rather than merging the definition with currently active situations.

RESTART is not supported on z/OS or i5. The agent process must be restarted in another manner to activate the new configuration.

Agent Management Services
The RESTART code uses Agent Management Services to recycle the agent:

- The agent must be under the management of the Agent Management Services. This is accomplished by specifying <managerType>ProxyAgentServices</managerType> in the agent's common agent package (CAP) file or by using the AMS Start Management take action command

- The OS agent must be running on the system so that the agent watchdog is available to recycle the agent.

Expect the following results if the agent is not under the management of the Agent Management Services or the OS agent is not running when the ConfigFile is retrieved from the central configuration server:

- The agent writes a message to the log file that the restart is bypassed.
- If SNMP or EIF events at the monitoring agent are enabled, an autonomous lifecycle status event is generated.
- The file is activated the next time the agent is restarted.

By default, Agent Management Services is enabled for all OS agents except the zLinux OS agent, which is disabled. The Agent Management Services watchdog for the zLinux OS agent must be enabled manually to take advantage of RESTART capability. Otherwise, the zLinux OS agent must be restarted in another manner to activate the new configuration.

See Chapter 14, "Agent Management Services," on page 317 for details on using Agent Management Services to monitor the availability of agents.

YES

Specifying "YES" instructs the agent to download and replace the existing configuration file. New changes are implemented dynamically without agent restart. This option is valid only for the following DISP values: CNFGLIST, PVTSIT, EIFCNFG and EIFMAP.

Specifying YES does not automatically cause the EIF emitter to start or recycle. The EIF configuration file must be physically downloaded from the Centralized Configuration server to the agent in order to trigger file activation.

If you are using private situations and use Activate="YES", then specifying REFRESH="N" in your private situation configuration file instructs the agent to merge the downloaded file into the current file and then activate the file. If you specify REFRESH="Y" in your private situation configuration file, the agent replaces the local file with the downloaded file and then activate the file.

Table 65. Configuration load list <ConfigFile> element and the Activate options available for the Disp type.

Disp=	Activate="Yes"	Activate="Restart"	Activate="NO"
CNFGLIST	Default	N/A	N/A
PVTSIT	Available	Available	Default
TRAPCNFG	N/A	Available	Default
EIFCNFG	Available	Available	Default
EIFMAP	Available	Available	Default
PASCAP	CAP files are activated on download, therefore the Activate attribute does not apply.		

Table 65. Configuration load list <ConfigFile> element and the Activate options available for the Disp type. (continued)

Disp=	Activate="Yes"	Activate="Restart"	Activate="NO"
AAGP	Default	N/A	N/A
CUSTOM	N/A	Available	Default

<ConfigParm>

Optional. Specifies an agent environment variable override value that updates the environment settings immediately and takes effect at the next operation interval or instance.

Interval=

Override or set IRA_SERVICE_INTERFACE_CONFIG_INTERVAL.

Backup=

Override or set IRA_SERVICE_INTERFACE_CONFIG_BACKUP.

NumbTasks=

Override or set IRA_SERVICE_INTERFACE_CONFIG_POOL_SIZE.

MaxWait=

Override or set IRA_SERVICE_INTERFACE_CONFIG_MAX_WAIT.

Configuration load list keyword substitution

Use keyword substitution to create a configuration load list that can be consistently applied to different agents and locations.

Tivoli monitoring agents recognize certain keywords in the configuration load list attributes and substitute them using run time values from the central configuration client.

With the exception of @ITMHOME@, any characters that are not alphanumeric are changed to hyphens (-) in the output. The keywords for the configuration load list are:

@PRODUCT@ This is the monitoring agent's lowercase, two-character product code. Example: On a Windows OS agent, @PRODUCT@_trapcnfg.xml resolves to nt_trapcnfg.xml

@ITMHOME@ This is the 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@ This is the Managed System Name (not the subnode name). Examples: If the agent's Managed System Name is primary:icvw3d62:nt, @MSN@ resolves to primary-icvw3d62-nt.

@TASKNAME@ This is the monitoring agent's process name. Examples: klzagent; kntcma.

@VERSION@ This is the monitoring agent's product version. Example: If the agent's version is 6.2.2 Fix Pack 2, @VERSION@ resolves to 06-22-02.

@HOSTNAME@ This is the computer host name. Example: myhost.

@IPADDRESS@ This is the 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@ This is the operating system type. Valid values include: z/OS, Tandem, AS/400®, Win98, Win95, WinNT, Win2K, WinXP, Windows, Win2003, WinVista, Windows7, and Win2008. The OSTYPE for all other platforms is obtained using the `uname` command. You can also find the OSTYPE for a given computer in the RAS1 log listed under System Type or by bringing up the Service Console and looking in the upper right-hand corner of the display.

@OSVERSION@ This is the operating system version. Examples: Red Hat Enterprise Linux Version 5 (64bit) resolves to 2-6-18-128-el5; Windows 2003 (32bit) with ServicePack 2 resolves to 5-2-sp2.

@SYSTEMID@ This is the computer system identifier. Example: System ID icvr4a04.mylab.mycity.ibm.com is output as icvr4a04-mylab-mycity-ibm-com.

See the keyword organization and syntax examples under "Central configuration server" on page 426.

Environment variables in the configuration load list

You can reference an environment variable in the configuration load list instead of entering a fixed value for an attribute. Variable substitution enables you to apply the same load list definition in different environments.

Enclose environment variables in percent signs (%) when you reference them in a configuration load list. You might expect this requirement is for Windows only, because environment variables are delimited with percent signs on that operating system, but the percent sign delimiters are used to identify environment variables within the configuration load list on all platforms.

Environment variables in the configuration load list are resolved at the central configuration server or at the central configuration client, depending on whether a monitoring agent or a web server is acting as the central configuration server:

- When a monitoring agent is acting as the central configuration server, environment variables in the configuration load list are resolved using the environment at the central configuration server.
- When a web server is acting as the central configuration server, environment variables in the configuration load list are resolved using the environment at the central configuration client (the monitoring agent making the request).

To illustrate how variable substitution might be used in the configuration load list, consider an environment with two monitoring agents acting as central configuration servers. The environment variable SALES_CNFG_FILES is used to identify the directory that contains configuration files for use by systems that belong to the Sales Department of our company. The Primary central configuration server is a windows OS agent on a server called **winhost**

```
IRA_SERVICE_INTERFACE_CONFIG_HOME=C:\IBM\ITM\ConfigFiles
```

and the configuration for the sales department are in a directory called `C:\IBM\ITM\ConfigFiles\Sales`, so we set

```
SALES_CNFG_FILES=Sales
```

The Development Department also has a central configuration server configured on a Linux OS agent on their server **linuxhost**. They keep a current copy of the configuration files for the sales systems too; their agent is used as an alternate central configuration server if any new agents are deployed while **winhost** is unavailable. Their Linux OS agent uses the default value for IRA_SERVICE_INTERFACE_CONFIG_HOME, which is `/opt/IBM/ITM/localconfig`.

They locate the configuration files for the sales department in `/opt/IBM/ITM/localconfig/eastcoast/sales`, so they set:

```
SALES_CNFG_FILES=eastcoast/sales
```

This is the bootstrap configuration load list for the Sales Department:

```
<ConfigurationArtifact>
 <ConfigServer Name="BACKUP-CONFIG-SERVER"
  URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
  User="itmuser"
  Password="{AES256:keyfile:a}8wNnAEj6uLMTTOeaC+2rfQ==" />
 <ConfigServer Name="CENTRAL-CONFIG-SERVER"
  URL="http://winhost:1920///primary.winhost_nt/primary.winhost_nt/"
  User="itmuser"
  Password="{AES256:keyfile:a}8wNnAEj6uLMTTOeaC+2rfQ=="
  AltServer="BACKUP-CONFIG-SERVER" />
  <ConfigFile Server="CENTRAL-CONFIG-SERVER"
   Name="cnfglist.xml"
   Path="%SALES_CNFG_FILES%"
   Disp="CNFGLIST"
   Activate="YES" />
</ConfigurationArtifact>
```

When agents connect to **winhost**, they collect `C:\ibm\ITM\ConfigFiles\Sales\cnfglist.xml`. If **winhost** is unavailable, agents connect to **linuxhost** and collect `/opt/IBM/ITM/localconfig/eastcoast/sales/cnfglist.xml`.

By using the environment variable in the specification, additional customization is possible on the individual central configuration servers.

Bootstrap configuration load list

When placing the initial configuration load list, you can place a file that immediately identifies all the components that the monitoring agent needs to collect from the central configuration server. However, this means that you must place a unique configuration load list file on every agent. One of the items that a central configuration client should always collect from the configuration server is the load list. This allows the load list to be modified from the configuration server.

Because you will be identifying the complete configuration load list anyway, to initialize Centralized Configuration operations, you only need to tell the agent where to connect to get the first configuration load list. A simple configuration load list that contains one ConfigServer element and one ConfigFile element to define the DISP=CNFGLIST file can be used to bootstrap Centralized Configuration operations.

In this example, the ConfigServer URL identifies the central configuration server location and the user name and password to gain access to that server. The ConfigFile element points to the server named in the ConfigServer element and identifies the configuration load list file as cnfglist.xml, which is in the path for the product.

```
<ConfigurationArtifact>
  <ConfigServer
   Name="CENTRAL-CONFIG-SERVER"
   URL="http://winhost:1920///primary.winhost_nt/primary.winhost_nt/"
   User="Administrator"
   Password="{AES256:keyfile:a}vHBiEqmmvylNPs90Dw1AhQ==" />
  <ConfigFile
   Server="CENTRAL-CONFIG-SERVER"
   Name="cnfglist.xml"
   Path="@PRODUCT@"
   Disp="CNFGLIST"
   Activate="YES" />
</ConfigurationArtifact>
```

After creating a bootstrap configuration load list that identifies the correct load list for the agent, place the file and restart the agent.

Environment variables for Centralized Configuration

Environment variables can be used for customizing the agent environment for Centralized Configuration: to bootstrap the central configuration server; to control central configuration client operations; and, when a monitoring agent acts as a central configuration server, to control operations.

For enterprise monitoring agents, the environment variables are set in the agent's environment file; for system monitor agents, the environment variables are set in the *pc*_silent_install.txt response file. For more information about installing the system monitor agent, see "Monitoring your operating system via a System Monitor Agent" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Bootstrap central configuration server

Upon startup, a monitoring agent first looks for its configuration load list XML file. If the configuration load list does not exist, the agent reviews its environment file for the following variables. The agent constructs the initial, or *bootstrap*, load list from these environment values to connect to a central configuration server and download a configuration load list.

IRA_CONFIG_SERVER_URL
Specifies the server URL. For example, http://9.52.111.99.

IRA_CONFIG_SERVER_USERID
Specifies the server user ID. Default: itmuser.

IRA_CONFIG_SERVER_PASSWORD
Specifies the user password either in plain text or AES encrypted password string.

IRA_CONFIG_SERVER_FILE_PATH
Specifies the path to the configuration load list on the central configuration server. The default is loadlist/@PRODUCT@. See "Configuration load list keyword substitution" on page 436 for a list of keywords.

IRA_CONFIG_SERVER_FILE_NAME
Specifies the name of the configuration load list file on the central configuration server. Default: **cnfglist.xml**.

Central configuration client operations

The following agent environment variables affect how the agent operates as a client for Centralized Configuration. Use them to specify a different configuration load list file from the default, how often to connect to the central configuration server to check for updates, and whether to download only the configuration files that have changed since the last time you downloaded or all the files.

IRA_SERVICE_INTERFACE_CONFIG_LOADLIST
Use this variable to override the default configuration load list. Specify the full path and file name of the configuration load list. The following configuration load list file names are the defaults, where *pc* is the two-character product code:

Windows *install_dir*\localconfig\pc\pc_cnfglist.xml

Linux UNIX *install_dir*/localconfig/*pc*/*pc*_cnfglist.xml z/OS IRA_SERVICE_INTERFACE_CONFIG_LOADLIST=*PC*CFGLST.RKANDATV

IBM i /QIBM/UserData/IBM/ITM/localconfig/a4/a4_cnfglist.xml

IRA_SERVICE_INTERFACE_CONFIG_INTERVAL

Specifies how often the agent attempts to check with the central configuration server for updates. Specify the interval in minutes. One day is 1440 minutes; one week, which is also the maximum, is 10080 minutes. Default: **60** minutes.

IRA_SERVICE_INTERFACE_CONFIG_BYPASS_TIMESTAMP

When set to N, the agent downloads and replaces only the configuration files that have a newer UTC time stamp than that of the local version. Default: **N**. Setting this parameter to **Y** instructs the agent to bypass the timestamp and always download the file after every interval.

As a best practice, synchronize system times across the network to ensure that any monitoring agents running with their time ahead of the central configuration server do not miss an update if the file is changed within the time difference after download.

IRA_SERVICE_INTERFACE_CONFIG_BACKUP

When a new configuration file is downloaded, the agent renames the existing local file to a backup copy by appending suffix 1 through 5 to the file name and moving the file to a backup directory. This variable specifies the number of backup versions to keep. The minimum is 0 for no backup; the maximum is 5 backups. Default: **2** backups.

IRA_SERVICE_INTERFACE_CONFIG_BACKUP_DIR

Use this environment variable to establish a different backup directory from the default. These are the default backup directories:

Windows *install_dir*\localconfig\pc\backup

Linux UNIX *install_dir*/localconfig/*pc*/backup

z/OS RKANDATV DD dataset

IBM i /QIBM/UserData/IBM/ITM/localconfig/a4/backup

IRA_SERVICE_INTERFACE_CONFIG_MAX_WAIT

Defines the maximum wait time in seconds for downloading all configuration files specified in the load list from the central configuration repository. The valid time range is between 15 – 300 seconds. Default: **60** seconds.

IRA_SERVICE_INTERFACE_CONFIG_PASCAP_FACTOR

Linux UNIX central configuration client only. The time multiplication factor that the monitoring agent uses to calculate the Agent Management Services common agent package (CAP) file output delay interval. The delay interval is derived from the KCA_DISCOVERY_INTERVAL environment variable multiplied by this factor. The agent enforces a minimum factor value of 1. Default: **1.5**.

IRA_SERVICE_INTERFACE_CONFIG_POOL_SIZE

The agent creates a task thread pool to handle multiple load list articles concurrently. The agent puts requests on a FIFO queue served by pool tasks. Default: **10** tasks.

Central configuration server operations

The following agent configuration parameters affect how the agent operates as a server for Centralized Configuration:

IRA_SERVICE_INTERFACE_CONFIG_HOME

If an agent is being used as a configuration server, this setting can be used to override the default central configuration repository location. The default location used to place files to be served by the central configuration server is: `*install_dir*/localconfig`.

The web server HTTP file root is defined by the web server administrator and cannot be altered. Relative path specifications such as ../../ to reference artifacts outside of defined repository location cannot be used. In addition, the new repository location cannot be the root directory.

IRA_SERVICE_INTERFACE_NAME

Specify the preferred agent service interface name to define a more functionally recognized name to replace the agent generated default name in the format of `k`*`pc`*`agent`, where *pc* is the two-character product code, such as `kntagent` or `kmqagent`; or *`pc`*`agent`, such as `kuxagent02` to identify a second installed UNIX OS Agent instance on a system.

Default:

Windows system.*hostname_pc*
Linux UNIX *hostname_pc*

Example: The default agent-ServicePoint for a Windows OS Agent is: ***hostname*_nt**. The URL to connect to a central configuration server on a Windows OS agent running on host system winhost1 is: https://winhost1:3661///winhost1_nt/winhost1_nt. If IRA_SERVICE_INTERFACE_NAME= **ConfigServer-A**, the URL is `https://winhost1:3661///ConfigServer-A/ConfigServer-A`

KDE_TRANSPORT

Use the KDE_TRANSPORT environment variable to specify a different port from the default 1920 for HTTP or 3661 for HTTPS.

Do not change the default port settings, especially on multifunction UNIX and Linux systems, because 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.

Note: The KDE_TRANSPORT variable supersedes and overrides the KDC_FAMILIES variable. If a KDC_FAMILIES variable exists in the file, copy the KDC_FAMILIES settings that you want to keep to the new KDE_TRANSPORT variable. For more information, see the topic on "Controlling port number assignments" on the portal server in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Problem determination

These diagnostic options can be set for the agent component. The output is the ras1 log file.

IRA_DEBUG_SERVICEAPI=Y

Agent component name: Service Interface

IRA_DEBUG_PRIVATE_SITUATION=Y

Agent component name: private situation

IRA_DEBUG_TRANSCON=Y

Agent component name: Transport Conduit

KDH_DEBUG=D

Agent component name: Tivoli Monitoring Service Index HTTP Service.

Enable password encryption in configuration files on z/OS

Agent autonomy configuration XML files include user credentials with passwords that can be entered in plain text. Securing access to these configuration files is usually adequate to secure the credentials. You can also add a layer of security by storing passwords in encrypted format within the configuration file.

Before you begin

If you are enabling SNMP alerts from the agent, SNMP v1 & v2c Community Strings and SNMP v3 Authentication and Privacy Passwords can be stored in encrypted format in the *PC*TRAPS.RKANDATV trap configuration file.

If you are enabling Centralized Configuration, the ConfigServer password attributes can be encrypted when they are stored in a *PC*CFGLST.RKANDATV Configuration Load List file or using the IRA_CONFIG_SERVER_PASSWORD parameter in the K*PC*ENV environment file.

On Windows, Linux, and UNIX systems, password and community strings are encrypted and decrypted using the GSKIT encryption utilities provided by the Tivoli Management Services infrastructure. On z/OS, GSKit is known as the Integrated Cryptographic Service Facility, or ICSF. If these strings are stored in encrypted format on z/OS, the ICSF subsystem must be available on the z/OS system and the ICSF modules must be added to the z/OS monitoring agent startup PROC so that the strings can be decrypted for use by the agent.

Procedure

1. Verify that you have at least one IBM cryptographic coprocessor installed and that the ICSF is installed.
2. Create a KAES256 member in the RKANPARU data set in the z/OS agent runtime environment. Be sure to use the same encryption key that is used throughout your environment. If the z/OS Configuration Tool has already created a KAES256 member with the same encryption key for a Tivoli Enterprise Monitoring Server on z/OS and the z/OS agent is configured in the same runtime environment as the monitoring server, you can skip this step.
   - Copy the KAES256 member from the monitoring server's RKANPARU data set to the z/OS agent's RKANPARU data set.
   - Alternatively, you can copy the KAES256.ser file from the `keyfiles` directory of the distributed system where you will execute the itmpwdsnmp tool to encrypt password and community strings. Upload the KAES256.ser file to the KAES256 member of the z/OS agent's RKANPARU data set in binary mode. KAES256.ser is 48 bytes on distributed systems and is padded with blanks in the KAES256 member of the RKANPARU data set.
   - For instructions on using the z/OS Configuration Tool to create the KAES256 member, see the “Configuring hub and remote monitoring servers on z/OS” topic in *Configuring the Tivoli Enterprise Monitoring Server on z/OS*.
3. Concatenate ICSF modules to the existing startup PROC RKANMODL DDNAME of the z/OS agent. Edit the z/OS agent startup PROC and add ICSF support to the RKANMODL DDNAME. The following example illustrates RKANMODL where CSF.SCSFMOD0 is the data set that contains ICSF decryption modules:

```
//RKANMODL DD DISP=SHR,DSN=my_load_modules
// DD DISP=SHR,DSN=TDOMPT.&LVMLVL..MODL
// DD DISP=SHR,DSN=TDOMPT.&CMSLVL..MODL
// DD DISP=SHR,DSN=CSF.SCSFMOD0
```

4. Restart the monitoring server or the z/OS monitoring agent or both.

What to do next

Use the `itmpwdsnmp` utility to create the encrypted password and community strings. The utility is available only in the Tivoli Enterprise Monitoring Agent framework on distributed platforms. The agent framework can be installed from the Tivoli Monitoring Base DVD or Tivoli Monitoring Agent DVD. Run the `itmpwdsnmp` tool in interactive mode on the distributed system to encrypt the passwords that will be placed in the configuration files. For instructions, see "SNMP PassKey encryption: itmpwdsnmp" on page 372.

Centralized Configuration sample setup

This sample setup illustrates planning considerations and the steps required to prepare your Tivoli Monitoring environment and files for Centralized Configuration.

Create a directory structure

Decide what kind of files you want to serve and create a directory structure on the computer that allows the keywords that can be used in the configuration load list to enable each client agent to collect the correct files.

The default home directory where files are served from on the agent used as a central configuration server is *`install_dir`*`/localconfig` on all platforms. You can relocate the directory using the IRA_SERVICE_INTERFACE_CONFIG_HOME environment variable in the agent's environment file.

For our sample setup, we relocate the central configuration server home directory to

`install_dir``/configserver`

and create these subdirectories:

`install_dir``/configserver/common` contains files that are the same on all agents

`install_dir``/configserver/nt` contains files used by the Windows OS agents, which are located using the @PRODUCT@ keyword

`install_dir``/configserver/lz` contains files used by the Linux OS agents, which are located using the @PRODUCT@ keyword

`install_dir``/configserver/ux` contains files used by the UNIX OS agents, which are located using the @PRODUCT@ keyword

`install_dir``/configserver/myfiles` contains other files that you might want to distribute

Keywords @OSTYPE@ and @OSVERSION@ are useful to serve different files to different groups of systems. For example, on UNIX systems use @OSTYPE@ to separate AIX situations from Solaris situations. See "Configuration load list keyword substitution" on page 436.

Obtain the root password for the central configuration server

For our sample setup, store the password in the configuration load list that the central configuration server uses to load its AAGP file every time the agent starts.

Encrypt the password using the `itmpwdsnmp` utility that is available on any agent at V6.2.2 or later.

Windows `C:\ibm\ITM\TMAITM6\itmpwdsnmp.bat`

Linux UNIX `/opt/IBM/ITM/bin/itmpwdsnmp`

Here is an example of the display at the Linux command line:

```
itmpwdsnmp

Enter the password to be encrypted:
Confirm string:
{AES256:keyfile:a}qNf3u5TzYsiNXRacS4/sXQ==
```

Select the ID that agents use to access the central configuration server

For our sample setup, it is **itmuser**. The password is stored in the configuration load list that is used by agents to connect to the central configuration server.

Encrypt the password for the itmuser ID using the `itmpwdsnmp` utility. (Define this ID on each agent. The ID is added to the agent's AAGP.)

Windows `C:\ibm\ITM\TMAITM6\itmpwdsnmp.bat`

Linux UNIX `/opt/IBM/ITM/bin/itmpwdsnmp`

Create the AAGP.xml file that adds the Administrative ID to the AD group

The **Administrative** ID used to access the central configuration server is added to the predefined AD authorization group.

For our sample setup, we save the `AAGP.xml` file saved on the central configuration server in the *`install_dir`*`/configserver/common` directory.

```
<AAGP>
 <AAUSER>
  <ID>itmuser</ID>
  <ASSIGN>AD</ASSIGN>
 </AAUSER>
</AAGP>
```

This AAGP.xml file sets the AAGP on the central configuration server. For our sample setup, the central configuration server will serve that same file to agents that connect to it. This simplifies our sample setup. Nonetheless, you can have different sets of agents collect unique AAGP files with different sets of IDs and groups so that a different set of permissions is in place for working with the Agent Service Interface on those agents. The AAGP they download is used when connecting *to* their Agent Service Interface. The agents use the IDs defined in the AAGP that the agent collected to connect to the central configuration server.

Create the configuration load list for the central configuration server

For our sample setup, we use a Linux OS agent as the central configuration server, so we create *`install_dir`*`/localconfig/lz_cnfglist.xml`, which is the default location for the agent's configuration load list. (Having the load list file in the localconfig directory is one reasons we moved the default central configuration repository location with IRA_SERVICE_INTERFACE_CONFIG_HOME. The agent could use the same localconfig files that it serves to other agents, but it might be more convenient to keep the files separate that the central configuration server distributes.) The `cnfglist.xml` allows the central configuration server to load the AAGP to itself:

```
<ConfigurationArtifact>
 <ConfigServer Name="CENTRAL-CONFIG-SERVER"
  URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
  User="root"
  Password="{AES256:keyfile:a}qNf3u5TzYsiNXRacS4/sXQ==" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
```

```
  Name="AAGP.xml"
  Path="common"
  Disp="AAGP" />
</ConfigurationArtifact>
```

Create a generic bootstrap configuration load list

Create a generic bootstrap configuration load list that agents will use to find the specific load list that provides the complete list of files they should collect. This step is not required, but it lets you change how you organize files on the central configuration server. There are many ways to do this.

For our sample setup, we create *install_dir*/configserver/common/bootstrap_cnfglist.xml with the following settings:

```
<ConfigurationArtifact>
 <ConfigServer Name="CENTRAL-CONFIG-SERVER"
  URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
  User="itmuser"
  Password="{AES256:keyfile:a}2fhfCvELPHKm94/OkbOjyw==" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="cnfglist.xml"
  Path="@PRODUCT@"
  Disp="CNFGLIST"
  Activate="YES" />
</ConfigurationArtifact>
```

There is no need to include the AAGP for the agent in the bootstrap CNFGLIST. The agent uses this file to locate the unique configlist that the agent should use. This CNFGLIST is only in effect for a few seconds. In our sample, the agent looks in the directory identified by the @PRODUCT@ keyword for a file called cnfglist.xml. It is best practice to look for the bootstrap CNFGLIST in the config server to allow agents to identify their CNFGLIST by changing this file directly on the config server (rather than changing the mechanism on each agent for beginning central config operations).

Create a CNFGLIST that all windows OS agents will use

For our sample, create the configuration load list in *install_dir*\configserver\nt\confglist.xml:

```
<ConfigurationArtifact>
 <ConfigServer Name="CENTRAL-CONFIG-SERVER"
  URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
  User="root"
  Password="{AES256:keyfile:a}qNf3u5TzYsiNXRacS4/sXQ==" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="AAGP.xml"
  Path="common"
  Disp="AAGP" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="cnfglist.xml"
  Path="@PRODUCT@"
  Disp="CNFGLIST"
  Activate="YES" />
<ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="@PRODUCT@_situations.xml"
  Path="@PRODUCT@"
  Disp="PVTSIT"
  Activate="YES" />
<ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="@PRODUCT@_trapcnfg.xml"
  Path="@PRODUCT@"
  Disp="TRAPCNFG"
  Activate="RESTART" />
</ConfigurationArtifact>
```

Create a CNFGLIST that all Linux OS agents will use

For our sample, create the configuration load list in *install_dir*/configserver/lz/cnfglist.xml. These are the agents that we want to collect all of the files in *install_dir*/configserver/myfiles.

```
<ConfigurationArtifact>
 <ConfigServer Name="CENTRAL-CONFIG-SERVER"
  URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
  User="itmuser"
  Password="{AES256:keyfile:a}2fhfCvELPHKm94/OkbOjyw==" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="AAGP.xml"
  Path="common"
  Disp="AAGP" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="cnfglist.xml"
  Path="@PRODUCT@"
  Disp="CNFGLIST"
  Activate="YES" />
<ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="myfile1.sh"
  Path="myfiles"
  LocalPath="@ITMHOME@/tmp" />
<ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="myfile2.sh"
  Path="myfiles"
  LocalPath="@ITMHOME@/tmp" />
</ConfigurationArtifact>
```

Create the other files that you want to distribute

Create configuration load lists for the other agent product codes and other files that you want to deploy and place them on the central configuration server.

Enable the monitoring agents to start using Centralized Configuration

“Centralized Configuration startup” on page 447 describes several ways to start using Centralized Configuration.

For our sample setup, you can set these environment variables in the client agent's environment file:

```
IRA_CONFIG_SERVER_URL=http://linuxhost:1920///linuxhost_lz/linuxhost_lz
IRA_CONFIG_SERVER_USERID=itmuser
IRA_CONFIG_SERVER_PASSWORD={AES256:keyfile:a}2fhfCvELPHKm94/OkbOjyw==
IRA_CONFIG_SERVER_FILE_PATH=common
IRA_CONFIG_SERVER_FILE_NAME=bootstrap_cnfglist.xml
```

Or you can create this *install_dir*/localconfig/*pc*/*pc*_confglist.xml file:

```
<ConfigurationArtifact>
 <ConfigServer Name="CENTRAL-CONFIG-SERVER"
  URL="http://linuxhost:1920///linuxhost_lz/linuxhost_lz/"
  User="itmuser"
  Password="{AES256:keyfile:a}2fhfCvELPHKm94/OkbOjyw==" />
 <ConfigFile Server="CENTRAL-CONFIG-SERVER"
  Name="bootstrap_cnfglist.xml"
  Path="common"
  Disp="CNFGLIST"
  Activate="YES" />
</ConfigurationArtifact>
```

Centralized Configuration startup

After you have designed the configuration load list for each agent and created the central configuration server, you are ready for the agents to connect and start using Centralized Configuration.

You can enable an agent to use Centralized Configuration by editing the agent environment variables to specify a default central configuration server, by placing the configuration load list file on the computer and restarting the agent, or by submitting a service interface request.

Initiating Centralized Configuration with agent environment variables

You can use the agent environment variables for Centralized Configuration to identify the location of the central configuration server and download the initial (bootstrap) configuration load list. These environment variables are used only if no local configuration load list exists.

After the initial configuration load list file is established, the monitoring agent uses that file and no longer attempts to use the environment variables. Deleting the local configuration load list (on the system monitor agent you also must run a silent installation) causes the agent to again use the environment variables to download the bootstrap configuration load list. (See “Bootstrap configuration load list” on page 438.)

Initiating with enterprise monitoring agent environment variables:

Use the Tivoli Enterprise Monitoring Agent's Centralized Configuration environment variables to point to the central configuration server and download the initial configuration load list.

Procedure

1. On the computer where the enterprise monitoring agent is installed, open the agent environment file from Manage Tivoli Enterprise Monitoring Services or from the command line:
   - Start Manage Tivoli Enterprise Monitoring Services:
     Windows Click **Start → Programs →IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services**
     Linux UNIX Where *ITM_dir* is the IBM Tivoli Monitoring installation directory, change to the `*ITM_dir*/bin` directory and run `./itmcmd manage [-h *ITM_dir*]`. Right-click the monitoring agent and click **Advanced → Edit ENV File**.
   - At the command line, change to the agent configuration directory and open the environment file in a text editor. Where *pc* is the two-character product code:
     Windows `*install_dir*\TMAITM6[_x64]\k*pc*env`
     Linux UNIX `*install_dir*/config/*pc*.ini`
2. Specify how to connect to a central configuration server and download the initial (bootstrap) configuration load list:

   IRA_CONFIG_SERVER_URL
   Specifies the server URL. For example, http://9.52.111.99.

   IRA_CONFIG_SERVER_USERID
   Specifies the server user ID. Default: itmuser.

IRA_CONFIG_SERVER_PASSWORD
Specifies the user password either in plain text or AES encrypted password string.

IRA_CONFIG_SERVER_FILE_PATH
Specifies the path to the configuration load list on the central configuration server. The default is loadlist/@PRODUCT@. See "Configuration load list keyword substitution" on page 436 for a list of keywords.

IRA_CONFIG_SERVER_FILE_NAME
Specifies the name of the configuration load list file on the central configuration server. Default: **cnfglist.xml**.

3. Save the environment file and recycle the agent to apply the changes.

Setting Centralized Configuration environment variables during system monitor agent installation:

You can use a system monitor agent's Centralized Configuration environment variables in the silent response file to point to the central configuration server and download the initial (bootstrap) configuration load list.

About this task

System monitor agents are started at the end of their silent installation. Without local configuration files, the agents run but do not run private situations or send SNMP alerts or EIF events. Using Centralized Configuration, the agent can retrieve these files and begin using them immediately. The system monitoring agent installation uses entries in the silent response file to create entries in the agent's environment file.

For more information about the silent response file, how to configure it, and how to invoke it, see "Monitoring your operating system via a System Monitor Agent" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Procedure

1. Locate the *pc*_silent_install.txt response file (such as ux_silent_install.txt) on the Tivoli Monitoring Agent installation media and make a copy of it.
2. Open the silent response file in a text editor.
3. Specify how to connect to the central configuration server and download the initial configuration load list.

SETENV_ IRA_CONFIG_SERVER_URL
Specifies the server URL. For example, http://9.52.111.99.

SETENV_ IRA_CONFIG_SERVER_USERID
Specifies the server user ID. Default: itmuser.

SETENCR_IRA_CONFIG_SERVER_PASSWORD
Specifies the user password as an AES encrypted password string. If you want to enter it in plain text, prefix the environment variable with SETENV_ instead of SETENCR_.

SETENV_ IRA_CONFIG_SERVER_FILE_PATH
Specifies the path to the configuration load list on the central configuration server. The default is loadlist/@PRODUCT@. See "Configuration load list keyword substitution" on page 436 for a list of keywords.

SETENV_ IRA_CONFIG_SERVER_FILE_NAME
Specifies the name of the configuration load list file on the central configuration server. Default: **cnfglist.xml**.

The *SETENV_parameter=value* statements create *parameter=value* statements in the agent environment file; *SETENCR_parameter=value* statements create *parameter={AES256:keyfile:a}encryptedvalue* statements.

4. Invoke the silent installation procedure as described in the Example of a silent installation file named nt_silent_installcc.txt that is invoked on a Windows system:

```
silentInstall.cmd -p nt_silent_installcc.txt
```

Example of a silent installation file named ux_silent_installcc.txt in the /opt/IBM/sma path on a UNIX system:

```
silentInstall.sh -h /opt/IBM/sma/ -p ux_silent_installcc.txt
```

Example

This example shows how a copy of nt_silent_install.txt from the installation media might be edited to install a system monitor agent on the local computer and configure it for Centralized Configuration:

Before

```
;License Agreement=I agree to use the software only in accordance with
the installed license.
;SETENV_IRA_CONFIG_SERVER_URL=http://configserver.domain.com:1920
;SETENV_IRA_CONFIG_SERVER_USERID=itmuser
;SETENCR_IRA_CONFIG_SERVER_PASSWORD=plaintext_or_encrypted_using_itmpwdsnmp
;SETENV_IRA_CONFIG_SERVER_FILE_PATH=initloadlist/@PRODUCT@
;SETENV_IRA_CONFIG_SERVER_FILE_NAME=cnfglist.xml
```

After

```
License Agreement=I agree to use the software only in accordance with
the installed license.
SETENV_IRA_CONFIG_SERVER_URL=http://mysystem.mydomain.ibm.com:1920
SETENV_IRA_CONFIG_SERVER_USERID=itmuser
SETENCR_IRA_CONFIG_SERVER_PASSWORD={AES256:keyfile:a}encryptedpassword
SETENV_IRA_CONFIG_SERVER_FILE_PATH=bootstraploadlist
SETENV_IRA_CONFIG_SERVER_FILE_NAME=cnfglist.xml
```

Initiating Centralized Configuration with a load list file

If you have created a configuration load list file, initiate Centralized Configuration by placing it in the proper location and starting the agent. You can do this manually, using a non-agent deploy bundle, or through the command line interface using the **tacmd putfile**.

Initiating by manually placing the load list file:

You can initiate Centralized Configuration by placing the configuration load list file in the agent configuration directory and recycling the agent.

Before you begin

Create the configuration load list file using the XML tagging described in “Configuration load list XML specification” on page 429.

Procedure

1. Access the system locally and place the configuration load list *pc*_cnfglist.xml in the agent's *install_dir*/localconfig/*pc* directory, where *pc* is the two-character product code.

   Windows *pc*_cnfglist.xml

   Linux UNIX *pc*_cnfglist.xml

   z/OS *PC*CFGLST.RKANDATV

   IBM i a4_cnfglist.xml
2. Recycle the agent.

Results

During startup, the configuration load list file is read for the central configuration server connection URL and the files to download from there.

Initiating with remote deployment of non-agent bundles:

If your environment contains a large number of existing agents that are connected to a Tivoli Enterprise Monitoring Server, you might prefer to use remote deployment to distribute the configuration load list to the agents. You can use the Agent Builderto build non-agent deployment bundles.

Procedure

1. Use Agent Builder to create a non-agent deploy bundle. See the *IBM Tivoli Monitoring Agent Builder User's Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3/builder/agentbuilder_user.htm) for details.
   a. Add the common bootstrap configuration load list confglist.xml file to the bundle.
   b. Create your own copy command that copies the file to the correct location for the agent that you plan to deploy to. Here is an example of an installation command for the Linux OS agent:

      ```
      cp |DEPLOYDIR|/cnfglist.xml |CANDLEHOME|/localconfig/lz/lz_cnfglist.xml
      ```
   c. Recycle the agent to start using the new configuration load list after it is deployed.
      - Optionally, create a post-installation command that uses the Agent Management Services watchdog to recycle the agent. Create a command for each platform that you plan to support because you must specify the fully qualified path to the pasctrl utility that is located in the agent's binary architecture directory.
      - Here are some post-installation commands that could be used:

        Windows

        ```
        install_dir\TMAITM6[_x64]\kcapasctrl.exe recycle nt
        ```

        Linux

        ```
        install_dir/lx8266/lz/bin/pasctrl.sh recycle lz
        ```

        UNIX

        ```
        install_dir/aix526/ux/bin/pasctrl.sh recycle ux
        ```
      - Non-OS agents still use the watchdog from the OS agent. A multi-instance DB2 agent on Windows, for example, that is managed by Agent Management Services requires that you specify the instance name

in the post-installation command. Therefore, including the restart in the deploy bundle might not be the best method but can be done if standard instance names are used.

```
install_dir\tmaitm6\kcapasctrl.exe recycle –o db2inst1 ud
```

- You could also include a script in the deploy bundle that contains more advanced logic to recycle agent instances and call that script in a post-installation command.

```
Deploy_dir/afterscript.sh
```

d. Generate the remote deployment bundle.

e. If the agent was not restarted using a post-installation command in the deploy bundle, recycle the agent to activate the configuration load list.
   - See “Starting, stopping, and recycling an agent through the Tivoli Enterprise Portal” on page 291.
   - OS agents can be recycled in the portal client using the AMS Recycle Agent Instance TakeAction
   - The AMS Recycle Agent Instance Take Action command can also be run using the CLI tacmd executeAction. Here are some examples for the OS agents:

     Windows

     ```
     tacmd executeaction -n "AMS Recycle Agent Instance" -t nt –m
     Primary:winhost:NT -c value="Monitoring Agent for Windows OS,
     kntcma.exe,,"
     ```

     Linux

     ```
     tacmd executeaction -n "AMS Recycle Agent Instance" -t lz –m
     linuxhost:LZ -c value="Monitoring Agent for Linux OS,klzagent,,"
     ```

     UNIX

     ```
     tacmd executeaction -n "AMS Recycle Agent Instance" -t ux –m
     unixhost:KUX -c value="Monitoring Agent for Unix OS,kuxagent,,"
     ```

2. Add the bundle to your depot using the CLI **tacmd addbundles**.
3. Deploy the bundle to the agents using Add Managed System in the portal client (see “Adding an agent through the Tivoli Enterprise Portal” on page 289) or through the CLI using **tacmd addSystem** from the monitoring server (see *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).
4. If the agent was not restarted using a post-installation command in the deploy bundle, recycle the agent to activate the configuration load list as described in step 1.e.

Initiating with tacmd putfile:

You can initiate Centralized Configuration using the CLI **tacmd putfile** to transfer the configuration load list to the monitoring agent.

About this task

Take these steps to push the configuration load list to the monitoring agent where you want to initiate Centralized Configuration. The *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) describes the tacmds and their syntax and includes examples.

Procedure

1. Log onto the to the hub monitoring server:

   ```
   tacmd login –s myhubserver –u myusername –p mypassword –t 1440
   ```

   where *myhubserver* is the fully qualified host name of the hub monitoring server, and *myusername* and *mypassword* is a valid user ID for logging onto the monitoring server operating system.
2. Push the file: tacmd putfile –m Primary:winhost:NT -s C:\config\cnfglist.xml -d C:\IBM\ITM\localconfig\nt\nt_cnfglist.xml -t text
3. Recycle the agent to activate the configuration load list. See Starting, stopping, and recycling an agent through the Tivoli Enterprise Portal
   - OS agents can also be recycled in the portal client using the AMS Recycle Agent Instance Take Instance: Open the Agent Management Services workspace of the OS Agent; right-click the OS agent in the “Agent Runtime Status” table view and click **Take Action - Select**; select **AMS Recycle Agent Instance**.
   - The AMS Recycle Agent Instance Take Action command can also be run using the CLI **tacmd executeAction**. Here are some samples for the OS agents:

     Windows

     ```
     tacmd executeaction -n "AMS Recycle Agent Instance" -t nt -m
     Primary:winhost:NT -c value="Monitoring Agent for Windows OS,
     kntcma.exe,,"
     ```

     Linux

     ```
     tacmd executeaction -n "AMS Recycle Agent Instance" -t lz -m
     linuxhost:LZ -c value="Monitoring Agent for Linux OS,klzagent,,"
     ```

     UNIX

     ```
     tacmd executeaction -n "AMS Recycle Agent Instance" -t ux -m
     unixhost:KUX -c value="Monitoring Agent for Unix OS,kuxagent,,"
     ```

Initiating Centralized Configuration with a service interface request

You can submit service interface requests interactively in a browser or use kshsoap to submit service interface requests from a script. Use the service interface to initiate Centralized Configuration by submitting the configuration load list as a request.

Initiating in the Agent Service Interface:

Use the Agent Service Interface to initiate Centralized Configuration by submitting the configuration load list as a request.

About this task

Complete these steps to enter the configuration load list as a request in the Agent Service Interface.

Procedure

1. Open a browser and access the agent's Service Index with the URL, `http://`*`hostname`*`:1920` or `https://`*`hostname`*`:3661`, where `hostname` is the fully-qualified name or IP address of the computer where the monitoring agent is installed.
2. Click the **IBM Tivoli *pc* Agent Service Interface** link for the agent, where *pc* is the two-character product code.

3. As prompted, enter the user name and password. The ID must be a member of the Access Authorization Group Profile's **Administrative** group on the central configuration client agent.
4. Select the link to the Service Interface Request.
5. Paste the contents of the configuration load list XML file into the Agent Service Interface Request text box, then submit the request.

What to do next

You can use the Agent Service Interface to submit the configuration load list as a service interface request whenever you want to refresh the local configuration on-demand.

Initiating using the Service Interface API (kshsoap):

The Service Interface is an API that allows the creation of a custom interface. Sample HTML files are provided with the agents to demonstrate the function of the Service Interface. The API can also be accessed programmatically using Java, Visual Basic, Perl, HTML and other languages. The Tivoli Enterprise Monitoring Server includes a command line utility called *kshsoap* that can be used within a script to submit these service interface requests. You can use kshsoap to initiate Centralized Configuration.

Procedure

1. Create a file called **request.xml** with <UUSER> and <UPASS> elements to specify the credentials that kshsoap requires to connect to the Agent Service Interface. This ID must be defined on the target systems that you plan to submit the request to. As well, the ID must be a member of the **Administrative** group in the target agent's Access Authorization Group Profile. Example:

   ```
   <ConfigurationArtifact>
     <UUSER>root</UUSER>
     <UPASS>{AES256:keyfile:a}ENRUCXLW40LpR0RtGSF97w==</UPASS>
       <ConfigServer
           Name="CENTRAL-CONFIG-SERVER"
           URL="http://winhost:1920///system.winhost_nt/system.winhost_nt/"
           User="Administrator"
           Password="{AES256:keyfile:a}vHBiEqmmvylNPs90Dw1AhQ==" />
       <ConfigFile
           Server="CENTRAL-CONFIG-SERVER"
           Name="cnfglist.xml"
           Path="@PRODUCT@"
           Disp="CNFGLIST"
           Activate="YES" />
   </ConfigurationArtifact>
   ```

   The <UUSER> and <UPASS> elements can be replaced by <UNAME> and <UWORD> if you do not want your password encrypted and prefer to enter it in plain text.
2. Create another text file named **urls.txt** containing the URLs of the Agent Service Interface Request. Example:

   ```
   http://linuxhost:1920///linuxhost_lz/linuxhost_lz
   http://unixhost:1920///unixhost_ux/unixhost_ux
   ```
3. Use kshsoap to send **request.xml** to the Service Interfaces listed in **urls.txt**. On a Windows-based monitoring server, kshsoap.exe is in the `install_dir\CMS` directory; on a Linux or UNIX-based monitoring server, kshsoap.exe is located in the `install_dir/interp/ms/bin` directory.
   - Windows

```
install_dir\CMS\kshsoap path_to_file\request.xml path_to_file\urls.txt
```

- Linux UNIX

  ```
  install_dir/interp/ms/bin/kshsoap
  path_to_file/request.xml path_to_file/urls.txt
  ```

Agent autonomy on z/OS

Throughout the agent autonomy topics are references to files and exceptions on z/OS-based monitoring agents. This topic consolidates that information.

Central configuration server

The central configuration server must be on a distributed system; the z/OS system is not supported.

Configuration load list

The monitoring agent downloads all items in the configuration load list at agent startup. The agent uses the initial file download timestamp as reference and begins keeping track of the configuration file last modified time. A change to the server copy of the file occurring after this timestamp is downloaded and the timestamp for when the file was last modified is updated.

Default names for z/OS monitoring agent local configuration members in the RKANDATV data set

Where *PC* is the two-character product code:

*PC***CFGLST**

Local Configuration Load data set member name

*PC***THRES**

Local threshold override file name.

*PC***TRAPS**

Local SNMP trap configuration file name

*PC***SICNFG**

Local agent private situation configuration file name

*PC***EIF** Local Agent EIF eventmap configuration file name

*PC***EVMAP**

Local Agent EIF destination configuration file name

Activate="RESTART" is not supported on z/OS

The RESTART option is used to restart the agent after a successful file download. It is not supported on the z/OS or i5 operating systems. The agent process must be restarted in another manner to activate the new configuration.

Multiple agents running in the same address space

Any override parameters defined in the `KDSENV` member of the *`&hilev.&rte.`*`RKANPARU` data set are used for all agents running within the address space. This works well for `IRA_EIF_DEST_CONFIG`, because all agents will likely share the same EIF event destination. The other override parameters can also be used, but the data set members identified might need to combine definitions for multiple agents, which is not recommended. The best practice is to use the default naming convention for local configuration data set members when running multiple agents in the same address space.

Password encryption

Local configuration XML files include user credentials with passwords that can be entered in plain text. Securing access to these configuration files is usually adequate to secure the credentials. You can also add a layer of security by storing passwords in encrypted format within the configuration file.

If you are enabling SNMP alerts from the agent, SNMP v1 & v2c Community Strings and SNMP v3 Authentication and Privacy Passwords can be stored in encrypted format in the *PC*TRAPS.RKANDATV trap configuration file.

If you are enabling Centralized Configuration, the ConfigServer password attributes can be encrypted when they are stored in a xxCFGLST.RKANDATV Configuration Load List file or using the IRA_CONFIG_SERVER_PASSWORD environment variable.

See the topic on “Password encryption in configuration files on z/OS” in *IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/zcommonconfig/zcommonconfig.htm) for instructions.

Chapter 16. Managing historical data

Tivoli Management Services provides tools for collecting and saving data samples, displaying historical reports, uploading data to a relational database for long-term storage or converting short-term history to delimited flat files, and data aggregation and pruning. These topics describe the components used for historical data collection, how they store data, and best practices for configuring data collection and maintaining the database.

The “Tivoli Data Warehouse solutions” topics in the *IBM Tivoli Monitoring Installation and Setup Guide* describe how to install and configure a Tivoli Data Warehouse and its requisite agents, the warehouse proxy and the summarization and pruning agent.

The “Historical collection configuration” topics in the *Tivoli Enterprise Portal User's Guide* describe how to configure historical data collections for attribute groups, how to get a report of historical data for a specified time range, how to apply historical baselines to a chart for trend analysis, and how to model situation thresholds using historical data.

The *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) describes the **tacmd** history commands that can be entered at the command line for configuring historical data collections, viewing their definitions, and exporting and importing the historical data collection definitions.

About historical data collection

To make historical data available for reporting and analysis, you must set up historical data collections. One or more of these collections are configured for each attribute group that you want to collect historical data for, then distributed to the managed systems that you specify.

Historical data collection

Configuration programs allow you to specify the collection of historical data. The historical data is stored in short-term history files either at the Tivoli Enterprise Monitoring Server or at the monitoring agent. You can choose to specify that historical data to be sent to the Tivoli Data Warehouse database for long-term storage. The data model is the same across the long-term and short-term historical data.

You can create another copy of a collection definition for an attribute group, then configure the copy for different values for any of these criteria: collection interval, warehouse interval, managed system distribution, or attribute filtering. What remains the same for every historical collection that is defined for an attribute group, however, is the Collection Location (TEMA or TEMS) and the settings for Summarization and Pruning.

Distribute to Managed System (Agent) or Managing System (TEMS)

Each historical data collection has a method of distribution:

Managed System (Agent) is the default method and requires that any managed system in the distribution connect to a Tivoli Enterprise Monitoring Server Version 6.2.2 or later. The distribution goes to a subset of managed systems: the managed systems of that agent type that are assigned individually or as part of a managed system group. Alternatively, you can choose to assign managed systems for distribution to a historical configuration group that the collection belongs to.

Managing System (TEMS) is the method that was used for distribution in IBM Tivoli Monitoring Version 6.2.1 and earlier; it is the required method for distribution if the managed system connects to a V6.2.1 or earlier monitoring server. The distribution is to managed systems of that agent type that are connected to the Tivoli Enterprise Monitoring Server. If the Managing System (TEMS) method has been chosen for a collection definition, that collection becomes ineligible for membership in a historical configuration group.

If you have upgraded to Tivoli Management Services Version 6.2.2 or later from a release prior to Version 6.2.2, you get a historical collection definition for each attribute group that was configured and the distribution method is unchanged: **Managing System (TEMS)**. If you would like to use the distribution techniques that are available when distribution is by managed system, change the distribution method for each collection definition to **Managed System (Agent)**.

Historical configuration object groups

Part of a historical collection definition is the distribution list, where the managed systems are specified to save historical data samples for. You can add the distribution directly to the historical collection, indirectly through a historical configuration object group, or a combination of the two.

- **Direct distribution** involves assigning individual managed systems or managed system groups or both to the historical collection. The advantage of this method is that the distribution applies only to this collection and you can easily add and remove managed systems as needed.
- **Indirect distribution** involves assigning managed systems or managed system groups or both to the historical configuration group that the historical collection is a member of. The advantage of this method is that you can establish one distribution list and apply it to multiple historical collections simply by adding those collections to the historical group membership.

Use historical configuration groups as a way to assign the same distribution list to multiple historical collection definitions. You can then control collection for the group rather than having to select historical collection definitions individually. This feature is available when the Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server are at Version 6.2.2 or later and the distribution method for the collection is set to **Managed System (Agent)**.

Warehouse schema

The data warehouse has one or more tables for each product, with column names that relate to the data contents. This platform follows a simple data model that is based on the concept of attributes. An attribute is a characteristic of a managed object (node). For example, Disk Name is an attribute for a disk, which is a managed object.

Attributes can be single-row or multiple-row. Single-row attributes gather only one set of data, such as the local time attributes because there is only one set of values for local time at any one time. Multiple-row attributes can gather multiple sets of data, such as the Avg_Queue attribute that returns one set of data for each queue that exists on the system. Each attribute belongs to an attribute group, and each attribute item stores data for a particular property of the attribute group.

A table is generated for each attribute group and the table names are used for collection of historical data. The individual monitoring agent user guides contain complete descriptions of the attribute groups specific to that agent.

Warehouse proxy

Managed systems to which data collection configurations have been distributed send data to the Tivoli Data Warehouse through the warehouse proxy agent, a multi-threaded server process that can handle concurrent requests from multiple monitoring agents. If the warehouse proxy is not reachable, the agent tries the transmission at the next warehouse interval (next hour or next day, depending on the setting). If, at the next interval, the warehouse proxy does not send back its status during transmission, the transaction is restarted. Then the data is resent to the warehouse proxy after 15 minutes. If the warehouse proxy sends back a status indicating a failure, the transaction will be restarted at the next warehouse interval.

You can have multiple warehouse proxy agents in a monitored environment. Install multiple warehouse proxy agents in a large environment to spread the work of receiving historical data from the monitoring agents and inserting it into the warehouse database.

If you do not intend to save historical data to a data warehouse, you do not need to install and configure the warehouse proxy and the summarization and pruning agent. If the data warehouse is not used, then it is necessary to use additional programs to trim short-term history files.

Warehouse summarization and pruning

The warehouse summarization and pruning agent provides the ability to customize the length of time for which to save data (pruning) and how often to aggregate data (summarization) in the data warehouse. With summarized data, the performance of queries can be improved dramatically. And with data summarization and data pruning working together, the amount of disk space used can be better managed.

Warehouse summarization is controlled on a per-table (attribute group) basis. How the rows in each table are summarized is determined by a set of attributes in each table that are designated as *primary keys*. There is always one primary key, the ORIGINNODE (often called Server Name or System Name), which means that data is summarized by the managed resource. One or more additional primary keys are provided to further refine the level of summarization for that table. For example, in an OS agent disk table, a primary key might be the *logical disk name*, which allows historical information be reported for each logical disk in the computer.

There can be only one summarization and pruning agent in the managed environment; it connects directly to the Tivoli Data Warehouse.

Historical data collection configuration

After your configured historical data collections begin saving data samples, make provisions to manage it. Without additional action, the history data files can grow unchecked, using up valuable disk space.

Defining historical data collection

The Historical Collection Configuration window is available through the Tivoli Enterprise Portal. You can specify the collection and storage of historical data at either the Tivoli Enterprise Monitoring Server or at the remote system where the monitoring agent is installed. For flexibility in using historical data collection, you can:

- Configure multiple collections for the same attribute group. Each collection has a different distribution, can have a different collection interval, and can have a different warehouse interval.
- Reduce the amount of data collected to only what passes a filter that you create. For example, collect only the data samples with processor busy time greater than 80%.
- Configure a historical collection for all managed systems on a specific monitoring server when distribution is set to "Managing System (TEMS)", and for any managed system or set of managed systems when distribution is set to "Managed System (Agent)".
- Set the **Collection Location** to save short-term history at the monitoring server or at the monitoring agent for each historical collection.
- Set the **Collection Interval**, how often to send data samples to the short-term history file, from once a minute to once a day for each historical collection.
- Set the **Warehouse Interval**, how often to save data into the Tivoli Data Warehouse, from every 15 minutes to once a day for each historical collection.
- Determine how and when to summarize and prune the data that is stored in the data warehouse. Summarization and pruning is configured for each attribute group that has one or more historical collections defined.
- Start collection on a managed system by adding it (or a managed system group it belongs to) to the distribution list of a historical collection or to a historical configuration group that the collection is a member of.
- Stop collection on a managed system by removing it (or a managed system group it belongs to) from the distribution list of a historical collection or from a historical configuration group that the collection is a member of.
- Create historical configuration groups with a distribution list and assign collections to the group that you want to use the distribution.

Defining historical data collections from the command line

The historical data collection can also be configured using the command-line interface `tacmd hist` commands:

histconfiguregroups

histcreatecollection

histdeletecollection

histeditcollection

histlistattributegroups

histlistcollections

histlistproduct

histstartcollection

histstopcollection

histunconfiguregroups

histviewattributegroup

histviewcollection

bulkExportSit (to export historical data collections) bulkImportSit (to import historical data collections)

If you have a test environment, you can write scripts that use tacmds for configuring historical data collections and run the script on other test computers or on the production system so that you do not need to repeat the same configuration for each system. For more information about these commands, see *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm).

Avoid redundant data collection

It is possible to collect data twice or more for the same attribute group on a managed system. This happens if you have configured multiple historical collections for the same attribute group and distribute them to the same managed system. Not only does this create more data traffic and use storage space unnecessarily, your summarization values are skewed. The skewing happens because the additional values sent by multiple collections for the same attribute group are factored into the summarization calculation.

For a given historical collection, you need not be concerned about inadvertently assigning the same managed system to the historical collection distribution multiple times. The historical function is aware of the managed systems the collection is distributed to and collects a data sample only once for each managed system. A managed system is included in the distribution of a historical data collection when it is:

- directly referenced in the collection definition
- in a managed system group that is referenced in the collection definition
- in the distribution of a historical configuration group that the historical collection is a member of
- in a managed system group that is in the distribution of a historical configuration group that the historical collection is a member of

Create filter formulas for granular data collection

The History Collection Configuration editor has a **Filter** tab with a formula editor for writing filter criteria that specifies the data to collect. Historical collection of a data sample only occurs if the values in the data row meet the filter criteria. For example, if the attribute value for % Disk Write Time is greater than 50%, the data sample is saved to short-term history; otherwise the sample is not saved.

The filter criteria is configurable for each collection definition. Applying filters to historical data collection can help reduce network traffic and wasted disk space and improve summarization and pruning performance.

Be aware that filtered data collection can affect the results of trending calculations that are performed with chart baselining and situation modeling and of query-based views that include historical data. For example, a filter to collect only rows where the process uses 80% or more CPU means that a calculation of average values will be only of values 80% and higher, rather than of all values.

Trimming short-term history files

If you have chosen to upload data through the warehouse proxy to the Tivoli Data Warehouse, then the short-term history files on the monitoring server or monitoring agent are automatically trimmed after upload.

Managing System (TEMS) collection type

Historical data collection can be specified for individual monitoring servers, products, and attribute groups. However, all agents of the same type that report directly to the same monitoring server must have the same history collection options. Also, for a given attribute group, the same history collection options are applied to all monitoring servers for which that collection is currently enabled.

Collection location

The **Collection Location** is where the short-term historical data file resides: at the TEMA (Tivoli Enterprise Monitoring Agent) or the TEMS (Tivoli Enterprise Monitoring Server). The default location is TEMA, which minimizes the performance impact on the monitoring server from historical data management. However, there are some product and attribute group combinations that are only collected at a specific place, either the monitoring server or the monitoring agent.

z/OS On OMEGAMON XE products, the persistent data store is used to store short-term history, so it must be configured at the collection location. For any given agent, do not vary the collection location: collect all historical data for the product either at the monitoring agent or monitoring server. For agents that are configured in the same address space as the monitoring servers (required for OMEGAMON XE for z/OS and OMEGAMON XE for Storage on z/OS), configure the persistent data store in the same address space, and specify TEMS as the collection location.

Aggregate and prune warehouse data

The Summarization and Pruning Agent is a mechanism for managing data in the Tivoli Data Warehouse. The data in the warehouse is a historical record of activity and conditions in your enterprise. Summarization of the data is the process of aggregating your historical data into time-based categories, for example, hourly, daily, weekly, and so on. Summarizing data allows you to perform historical analysis of the data over time. Pruning of the data keeps the database to manageable size and thus improves performance. Pruning of the database should be performed at regular intervals.

Important: You can run only one summarization and pruning agent even if you have multiple monitoring servers that are sharing a single Tivoli Data Warehouse database. Running multiple summarization and pruning agents causes database deadlocks and conflicts because the multiple instances might attempt to summarize or prune the data in the tables simultaneously.

Converting short-term history files to delimited flat files

If you choose not to use the Tivoli Data Warehouse, then you must institute roll-off jobs to regularly convert and empty out the history data files. Roll-off programs are provided. In addition to trimming the history data files, these scripts produce flat files which can be used with third-party tools to produce trend analysis reports and graphics. There is also an environment variable for setting the maximum size of history files.

See “Limiting the growth of short-term history files” on page 496.

Some attribute groups are ineligible for historical data collection

Some agents do not enable collection of history data for all of their attribute groups. This is because the product development team for that agent has determined that collecting history data for certain attribute groups is not appropriate or might have a detrimental effect on performance. This might be because of the vast amount of data that is generated. Therefore, for each product, only attribute groups that are available for history collection are shown in the History Collection Configuration window when you click a Monitored Application.

Changing the directory for short-term history files

When historical data has been configured to be collected at the agent (TEMA; not TEMS), use the agent environment variable CTIRA_HIST_DIR to change the directory where historical data is collected. You might, for example, want to store the history files on a disk that provides more storage capacity than the default history data file location provides.

Before you begin

The directory must be an existing directory, you must specify the full path, and your operating system user ID must have write permission for this directory. If the directory does not exist, the agent will not collect historical data.

About this task

Take these steps to edit the CTIRA_HIST_DIR agent environment variable to establish a different directory to store the short-term history files.

Procedure

- Windows
  1. In the Manage Tivoli Enterprise Monitoring Services window, right-click the monitored application and click **Advanced** → **Edit Variables**.
  2. In the Override Local Variable Settings window, click **Add**.
  3. Scroll through the variable ▾ list and select **CTIRA_HIST_DIR**
  4. In the Value field, replace `@LogPath@` with the full path to the directory where you want the short-term history to be saved.
  5. Click **OK** to see CTIRA_HIST_DIR in the **Variable** column and the new path in the **Value** column; and click **OK** again to close the window. The value is recorded in the K*pc*ENV file, such as KNTENV.
  6. Recycle the agent to have your changes take effect.
- Linux UNIX
  1. Change to the `<`*`itm_install_dir`*`>/config` directory and open *`pc`*`.ini` in a text editor, where *pc* is the two-character product code. For example, `/opt/IBM/ITM/config/ux.ini` for the UNIX OS agent. For a list of product codes, see “IBM Tivoli product, platform, and component codes” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
  2. On a new line, add this environment variable followed by the full path to the location where you want the short-term history to be saved:

     ```
     CTIRA_HIST_DIR=
     ```
  3. Save and close the file.
  4. Recycle the agent to have your changes take effect.

Performance impact of historical data requests

The impact of historical data collection and warehousing on Tivoli Management Services components is dependent on multiple factors, including collection interval, frequency of roll-off to the data warehouse, number and size of historical tables collected, system size, and more.

Impact of large amounts of historical data on the monitoring server or agent

The default location for storing short-term historical data is at the monitoring agent, although in certain configurations the monitoring server might be preferable.

This topic presents factors to consider when determining
- The attribute groups to collect historical data on
- Where to save the short-term data files
- How frequently to send historical data samples to the short-term collection location
- Whether to warehouse data from the attribute group and, if so, how frequently to send the data from short-term history files to the data warehouse

The collection location can be negatively impacted when large amounts of data are processed. This occurs because the warehousing process on the monitoring server or the monitoring agent must read the large row set from the short-term history files. The data must then be transmitted by the warehouse proxy to the data warehouse. For large datasets, this impacts memory, CPU resources, and, especially when collection is at the monitoring server, disk space.

Because of its ability to handle numerous requests simultaneously, the impact on the monitoring server might not be as great as the impact on the monitoring agent. Nonetheless, when historical collection is at the monitoring server, the history data file for one attribute group can contain data for many agents (all the agents storing their data at the monitoring server) thus making a larger dataset. As well, requests against a large dataset also impact memory and resources at the Tivoli Enterprise Portal Server.

When historical data is stored at the agent, the history file for one attribute group contains data only for that agent and is much smaller than the one stored at the monitoring server. The most recent 24 hours worth of data comes from short-term history files. Beyond 24 hours, the data is retrieved from the Tivoli Data Warehouse. (You can change the break point with the KFW_REPORT_TERM_BREAK_POINT portal server environment variable.) This action is transparent to the user; however, requests returning a large a amount of data can negatively impact the performance of monitoring servers, monitoring agents, and your network.

If a query goes to the short-term history file and retrieves a large amount of data, this retrieval can consume a large amount of CPU and memory and users can experience low system performance while the data is being retrieved. When processing a large data request, the agent might be prevented from processing other requests until this one has completed. This is important with many monitoring agents because the agent can typically process only one view query or situation at a time.

A best practice that can be applied to the historical collection, to the view query, or both is to use filters to limit the data *before* it gets collected or reported. For historical collections, pre-filtering is done in the **Filter** tab of the Historical Collection Configuration editor or the filter option of the CLI **tacmd histcreatecollection** command, as described in the *IBM Tivoli Monitoring Command Reference* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_cmdref.htm) and Creating a historical collection in the *Tivoli Enterprise Portal User's Guide*. For workspace views, pre-filtering is done in the Query editor by creating another query from a predefined query and adding a filter to the specification, as described in Creating another query to the monitoring server in the *Tivoli Enterprise Portal User's Guide*.

Requests for historical data from large tables

Requests for historical data from tables that collect a large amount of data have a negative impact on the performance of the Tivoli Management Services components involved. To reduce the performance impact on your system, set a longer collection interval or create a filter (or both) for tables that collect a large amount of data.

For optimal performance, best practice is to ensure the short term history file does not exceed 1 GB.

You specify the collection interval and filter criteria in the History Collection Configuration window. To find out the disk space requirements for tables in your IBM Tivoli Monitoring product, see the specific agent's documentation.

While displaying a query-based view, you can set the Time Span interval to obtain data from previous samplings. Selecting a long time span interval for the report time span adds to the amount of data being processed, and might have a negative impact on performance. The program must dedicate more memory and CPU cycles to process a large volume of report data. In this instance, specify a shorter time span setting, especially for tables that collect a large amount of data.

If a report rowset is too large, the report request can drop the task and return to the Tivoli Enterprise Portal with no rows because the agent took too long to process the request. However, the agent continues to process the report data to completion, and remains blocked, even though the report data is not viewable.

There can also be cases where the historical report data from the z/OS Persistent Data Store might not be available. This can occur because the Persistent Data Store might be not be available while its maintenance job is running.

Scheduling the warehousing of historical data

The same issues with requesting large rowsets for historical reports apply to scheduling the warehousing of historical data only once a day. The more data being collected and stored, the more resources required to read data into memory and to transmit to the data warehouse. If possible, make the warehousing rowset smaller by spreading the warehousing load over each hour, that is, by setting the warehousing interval to one per hour, rather than one day.

Using a data mart to improve long or complex queries

This section describes the how a data mart can be used to increase the performance of your primary datastore.

Within the Tivoli Management Services infrastructure, the warehouse proxy regularly inserts new data from the short-term history files into the data warehouse tables. This detailed data is derived by queries from historical views to report this information and can be derived by queries from an external reporting tool. Any active datastore needs to balance read and write activity to maximize performance of the datastore. The data warehouse has periodic write activity balanced with frequent read activity for formatting and creating reports. Under some circumstances (especially formatting reports over long durations or executing complex queries), the database read and write activity can become unbalanced and result in abnormal wait times. Under these circumstances, you can significantly improve performance by adding a secondary datastore, commonly called a *data mart*, for reports from causing long or complex data queries.

Depending upon the reporting requirements, there are two mechanisms that can be used, exploiting the open interfaces that are included with the warehouse:

1. If the complete database is required, use the Database Replication Facilities of the Tivoli Data Warehouse RDBMS.
2. Write and schedule SQL extract scripts, similar to ETL Scripts in Tivoli Data Warehouse V1.x, to extract desired data elements at a scheduled interval from the Tivoli Data Warehouse and populate a reporting database. This reporting database can be optimized for use by an external reporting tool, just like data marts were used in Tivoli Data Warehouse V1.x. These scripts can be SQL Scripts, shell scripts, or PERL scripts.

Sample data mart SQL script for IBM Tivoli Monitoring

The following SQL script is an sample script of how you can create and populate a data mart. Your actual script needs to be revised to reflect your environment.

```
--------------------------------------------------------
-- Example data mart SQL Script for TDW 2.1
--------------------------------------------------------
-- This scripts demonstrates the creation and population
-- of a data mart (similar to the data marts in TDW 1.x)
-- starting from the "flat" tables in TDW 2.1.
-- This script can be run using the DB2 UDB CLP:
-- db2 -tvf myscript
--------------------------------------------------------
-- 1. Create hourly "flat" table from TDW 2.1 (simulated)
-- One row per hour per Windows system
drop table itmuser."Win_System_H";
create table itmuser."Win_System_H" (
 WRITETIME                      CHAR( 16 ),
 "Server_Name"                  CHAR( 64 ),
 "Operating_System_Type"        CHAR( 16 ),
 "Network_Address"              CHAR( 16 ),
 "MIN_%_Total_Privileged_Time"  INTEGER,
 "MAX_%_Total_Privileged_Time"  INTEGER,
 "AVG_%_Total_Privileged_Time"  INTEGER,
 "MIN_%_Total_Processor_Time"   INTEGER,
 "MAX_%_Total_User_Time"        INTEGER,
 "AVG_%_Total_User_Time"        INTEGER );

-- 2. Insert example data

insert into itmuser."Win_System_H" values (
'1050917030000000', 'Primary:WinServ1:NT', 'Windows_2000', '8.53.24.170',
20, 40, 30, 10, 30, 20 );

insert into itmuser."Win_System_H" values (
'1050917040000000','Primary:WinServ1:NT','Windows_2000','8.53.24.170',
20, 40, 30, 10, 30, 20 );

insert into itmuser."Win_System_H" values (
'1050917030000000','Primary:WinServ2:NT','Windows_2000','8.53.24.171',
20, 40, 30, 10, 30, 20 );

insert into itmuser."Win_System_H" values (
'1050917040000000','Primary:WinServ2:NT','Windows_2000','8.53.24.171',
  20, 40, 30, 10, 30, 20 );

-- 3. Create a dimension table for the hosts
-- primary key is Server_ID, a generated value
-- alternate key is Server_Name, Network_Address
```

```
drop table itmuser."D_Win_System";
create table itmuser."D_Win_System" (
 "Server_ID" INTEGER GENERATED ALWAYS AS IDENTITY
    PRIMARY KEY NOT NULL,
 "Server_Name"                    CHAR( 64 ),
 "Operating_System_Type"          CHAR( 16 ),
 "Network_Address"                CHAR( 16 ) );

-- 4. Create an hourly fact table for the System facts
-- Server_ID is a foreign key to D_Win_System

drop table itmuser."F_Win_System_H";
create table itmuser."F_Win_System_H" (
 WRITETIME                      CHAR( 16 ) NOT NULL,
 "Server_ID"                    INTEGER NOT NULL,
 "MIN_%_Total_Privileged_Time"  INTEGER,
 "MAX_%_Total_Privileged_Time"  INTEGER,
 "AVG_%_Total_Privileged_Time"  INTEGER,
 "MIN_%_Total_Processor_Time"   INTEGER,
 "MAX_%_Total_User_Time"        INTEGER,
 "AVG_%_Total_User_Time"        INTEGER,
 constraint SERVID foreign key ("Server_ID")
  references itmuser."D_Win_System" ("Server_ID")
);

-- 5. Insert into the dimension table
-- only insert rows that do not already exist

insert into itmuser."D_Win_System" (
 "Server_Name",
 "Operating_System_Type",
 "Network_Address" )
select
 "Server_Name",
 min("Operating_System_Type") as "Operating_System_Type",
 "Network_Address"
from
  itmuser."Win_System_H" h
where
 not exists ( select 1 from
 itmuser."D_Win_System" d
 where d."Server_Name" = h."Server_Name"
 and d."Network_Address" = h."Network_Address"
 )
group by
 "Server_Name",
 "Network_Address"
;

-- 6. Check values in dimension table
select * from itmuser."D_Win_System"
;

-- 7. Insert into the fact table
-- only insert rows that do not already exist
insert into itmuser."F_Win_System_H"
select
 h.WRITETIME   ,
 d."Server_ID" ,
 h."MIN_%_Total_Privileged_Time" ,
 h."MAX_%_Total_Privileged_Time" ,
 h."AVG_%_Total_Privileged_Time" ,
 h."MIN_%_Total_Processor_Time"  ,
 h."MAX_%_Total_User_Time"       ,
 h."AVG_%_Total_User_Time"
from
```

```
 itmuser."Win_System_H" h,
 itmuser."D_Win_System" d
where d."Server_Name" = h."Server_Name"
 and d."Network_Address" = h."Network_Address"
 and not exists ( select 1 from
 itmuser."F_Win_System_H" f
  where f.WRITETIME = h.WRITETIME
  and f."Server_ID" = d."Server_ID"
  )
;

-- 8. Check values in fact table
select * from itmuser."F_Win_System_H"
;

-- 9. Repeat"5. Insert into the dimension table"
-- and "7. Insert into the fact table" on a daily basis
```

See the IBM Redbooks® publication, *Introduction to Tivoli Enterprise Data Warehouse* at http://www.redbooks.ibm.com/ for references and additional sample SQL extract scripts.

Tivoli Data Warehouse and short-term history configuration

This section addresses some of the short-term history configurations in relation to the Tivoli Data Warehouse database.

Naming of the Tivoli Data Warehouse history tables and columns

The history tables in the Tivoli Data Warehouse database have the same names as the group names of history tables and columns. For example, Windows NT history for group name NT_System is collected in a short-term file having the name WTSYSTEM. Historical data in this file, WTSYSTEM, is stored to the database in a table named NT_System.

The warehouse proxy uses the complete product attribute name to create DBMS table and column identifiers. This includes any special characters found in an attribute name. When the length of an attribute name exceeds the maximum table or column name length supported by a DBMS product, the warehouse proxy uses the internal table and column names as defined in the product attribute file.

The WAREHOUSEID table is located in the Tivoli Data Warehouse database. It contains records that describe any attribute or table names that exceed the DBMS maximum name length and that have been converted to internal table or column names. You can query this table to find out the correct name for a table or a column that has been internally converted. Each attribute group name in this table has a RECTYPE value of "TAB". Only the TABLENAME and OBJECTNAME values are filled in. Each attribute column name has a RECTYPE value of "COL". All other column values in WAREHOUSEID are filled in. The WAREHOUSE ID table has these definitions:

RECTYPE CHAR(3)

Indicates the type of record: "TAB" for table; "COL" for column.

TABLENAME CHAR(20)

Indicates an internal table name.

OBJECTNAME CHAR(140)

Indicates an attribute group name.

COLUMNNAME CHAR(20)
Indicates an internal column name.

ATTRNAME CHAR(140)
Indicates an attribute name.

The warehouse proxy automatically creates an associated index for each data table in the warehouse database. The index is based on WRITETIME and ORIGINNODE (whose display name can be "Server_Name," "System_Name," and so on, depending on the table) and the TMZDIFF (time zone difference) columns. The index name is the short name of the table, with an "_IDX" suffix.

Use of double quotes to ensure correct access to all data

All data warehouse table or column names for all major DBMS products are created by surrounding them with the DBMS-supported quoted identifier characters. When referencing historical data in the warehouse database, you must use the double-quote character to ensure correct access to that data. Some database products, such as Microsoft SQL Server, do not require the use of double quotes.

If you created SQL queries or stored procedures prior to IBM Tivoli Monitoring V6.2.1 for use with the previous version of the historical data collection program, these now might need to be modified. The SQL needs to take into account the fact that some relational database products (such as Oracle) require all table and column names to be surrounded by double quotation marks to access IBM history data, some agents might have changed their data characterizations or added new columns.

Warehouse proxy ATTRLIB directory

The ATTRLIB directory in the warehouse proxy is automatically created for you at product installation time. On a Windows system, this directory is located in `*ITM_dir*\tmaitm6\attrlib`. On an operating system such as UNIX, this directory is located in *ITM_dir*`/hd/tables`.

During installation, if the warehouse proxy is installed on the same computer where other agents are installed, the agent product attribute files that are accessible to the installation program are added to the ATTRLIB directory. The warehouse proxy uses the attribute file in only one specific condition: when the monitoring agent version is earlier than version 6.1.0.

The attribute file allows determination of the table or column internal name when the length of an attribute name exceeds the maximum table or column name length that a warehouse DBMS product supports. In that condition only, the attribute file must be in the ATTRLIB directory. If the warehouse proxy is installed on a separate computer and you have a monitoring agent that is not at the latest level, you must copy the attribute file of that agent to the ATTRLIB directory where the warehouse proxy is installed.

If you see an error message stating that an export failed because a particular product attribute file was missing from this directory, locate the missing product attribute file and copy it into the ATTRLIB directory.

Changes in the set of collected attributes

When changes are detected in the set of collected attributes, such as when a new version of an agent with added attributes is deployed, the historical program performs these functions:

- If warehousing is specified in the current historical data collection request, all collected historical data for the table is exported to the data warehouse. If the warehousing operation is successful, all short-term history data and meta files are deleted.

  If the operation fails (for example, if the warehouse proxy is not available), the short-term historical data and meta files are renamed. On the z/OS operating system environment, if a generic table is used to store the data, the short-term historical data for a table are deleted regardless of whether the warehousing operation is successful or not.
  - Windows and UNIX operating system environments

    On these operating system environments, the history data and meta files are renamed with the **.prv** and **.prvhdr** suffixes respectively.
  - IBM i operating system environment

    On this operating system environment, the history data and meta files are renamed with the **P** and **Q** suffixes respectively.

    If the renamed files already exist, they are deleted prior to the renaming operation (that is, only one generation of changed short-term history files is kept).
- If warehousing is NOT specified in the current historical data collection request, the history data and meta file are renamed as described above. On z/OS, if a generic table is used to store the data, all short-term history data for a table together with its meta record are deleted.

Tivoli Data Warehouse range partition migrations

Range partitioning is a database data organization feature that can significantly improve pruning and query performance in large Tivoli Data Warehouse databases. You can migrate your existing tables to partitioned tables to take advantage of the performance improvements provided with partitioned tables.

To use partitioned tables, the Summarization and Pruning agent and Warehouse Proxy agent must both be configured with partitioning enabled and the Tivoli Data Warehouse must allow partitioning.

The migration and required cleanup is handled using scripts generated by the schema publication tool in migrate mode. The scripts provide the following functions:

tdw_migrate_setup.sql
This script creates a stored procedure to redefine the source table to a new partitioned table and creates the control tables required for migration, such as the WAREHOUSE_MIGRATION_STATUS table.

tdw_migrate_step1.sql
This script invokes the stored procedure created in the setup script. The stored procedure renames the source table to MIGRATING_<*short table name*>, creates the new partitioned table, loads the data from the source table to the new table, and then renames the source table to DONE_<*short table name*>.

tdw_migrate_step2.sql

This script recreates the indices on the new partitioned tables, deletes the source tables, and grants SELECT to PUBLIC on the tables.

You can also migrate tables partitioned using a partitioning scheme different than the Tivoli Data Warehouse partitioning scheme. Only a table partitioned with the Tivoli Data Warehouse scheme can be managed by the Summarization and Pruning agent. If you want to continue to use your user-defined partitioning scheme, use the `KSY_TABLE_FILTER` variable to list only the tables you want migrated.

A migrated table's partitions are defined based on the table's retention period and the forward partitions parameter. The *forward partitions parameter* is a configuration parameter defined in the Summarization and Pruning configuration file using the variable KSY_PARTITIONS_UPWARD. The *retention period* is the pruning parameter defined on the attribute group you select through the History Configuration dialog in the Tivoli Enterprise Portal or through the command line.

For more information about range partitioning, see “Tivoli Data Warehouse range partitioning” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

The status tables WAREHOUSELOG and WAREHOUSEAGGREGLOG, can also be migrated. These tables can be filtered by specifying them in the `KSY_TABLE_FILTER` variable or by product code `KHD` for WAREHOUSELOG and `KSY` for the WAREHOUSEAGGREGLOG in the `KSY_PRODUCT_FILTER` variable. These tables are treated as detailed tables if the `KSY_SUMMARIZATION_FILTER` is used.

Prerequisites and best practices

Before you begin, ensure the following criteria is met:

- Ensure you have sufficient disk space, since during the migration two copies of the same data are present until the source table is deleted. You need at least twice the disk space of the tables being migrated.
- Define pruning for all the tables being migrated *before* the migration, so that data can be properly pruned after the tables are partitioned.
- Stop the Summarization and Pruning agent and Warehouse Proxy agent during the migration period. Also stop any reports or external users of the tables being migrated.
- Ensure that your Tivoli Enterprise Portal Server application support is updated. The portal server and agent application support must match.

Best practice is to migrate the tables in batches. This practice reduces the amount of disk space required for migration and the amount of time the Summarization and Pruning agent and Warehouse Proxy agent need to be offline. A batch of tables can be migrated within a maintenance window.

Migrating non-partitioned tables to partitioned tables for DB2 on Linux, UNIX, and Windows

Use the following steps to migrate non-partitioned tables if you are using DB2 on Linux, UNIX, or Windows.

Before you begin

Review the “Prerequisites and best practices.” You must ensure you have enough disk space.

The Tivoli Data Warehouse user must have the following privileges:

- CREATE TABLE
- LOAD
  The DB2 migration uses the load utility to copy data. To grant load authority, login to DB2 as a user with SYSADM or DBA authority and issue the `db2 grant load on database to user <`*`Tivoli Data Warehouse user ID`*`>` SQL command.
- Execution privileges for the ADMIN_CMD procedure
  To grant the authority, login to DB2 as a user with SYSADM or DBA authority and issue the `db2 grant execute on procedure sysproc.admin_cmd to user <Tivoli Data Warehouse user ID>` SQL command.

The privileges required for migration can be revoked after all the desired migrations are complete.

About this task

The migration of non-partitioned tables to partitioned tables in a Tivoli Data Warehouse DB2 on Linux, UNIX, and Windows database, uses a stored procedure generated by the schema publication tool. The stored procedure itself uses the DB2 LOAD utility.

Procedure

1. Configure the Summarization and Pruning agent to partition tables. For detailed steps, see "Specifying range partitioned tables for the Summarization and Pruning Agent" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. Stop all Warehouse Proxy agent and Summarization and Pruning agent instances.
3. Backup the Tivoli Data Warehouse database.
4. Edit the schema publication tool response file.
   a. Open the response file:

      Windows

      `install_dir``\TMAITM6\tdwschema.rsp` for 32-bit Windows systems

      `install_dir``\TMAITM6_x64\tdwschema.rsp` for 64-bit Windows systems

      Linux UNIX

      `install_dir``/`*`arch`*`/bin/tdwschema.rsp`

   b. Configure the following variables:

      `KSY_PRODUCT_SELECT = migrate`

      `KSY_PRODUCT_FILTER =` *`list of products to migrate`*
      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, see the *IBM Tivoli Monitoring Command Reference*).

      `KSY_TABLE_FILTER =` *`list of tables to migrate`*
      An optional filter to indicate only specific tables. This filter can be used in addition to the `KSY_PRODUCT_FILTER` variable. Use the following command to get the list of tables that are available for a given product. Replace each space in the attribute group name with an underscore character. For a list of table names, use the following command:
      `tacmd histListAttributeGroups -t <productcode>`

`KSY_SUMMARIZATION_SELECTION =` *`list of aggregation periods to migrate`*
An optional filter that can be used in addition to the `KSY_PRODUCT_FILTER` and `KSY_TABLE_FILTER` variables. This variable has an additional option when the migrate mode is used. The **R** option allows you to migrate the detailed tables. Other options are as follows:

R: Migrate detailed tables only

H: Hourly

D: Daily

W: Weekly

M: Monthly

Q: Quarterly

Y: Yearly

Filters can be combined. For example, to migrate the detail, hourly, and daily tables for the Windows OS agent:

```
KSY_PRODUCT=KNT
KSY_SUMMARIZATION_SELECTION=R,H,D
```

`KSY_SQL_OUTPUT_FILE_PATH =` *`optional file path for SQL output`*
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 variable, refer to the comments in the response file.

5. Ensure the Tivoli Enterprise Portal Server is started.
6. If you have not already, export the CANDLEHOME variable. Execute the following commands:

   Windows

   ```
   set CANDLE_HOME=install_dir
   ```

   Linux UNIX

   ```
   CANDLEHOME=/install_dir
   export CANDLEHOME
   ```
7. Run the schema publication tool script to generate the scripts required for migration.

   Windows

   `install_dir``\TMAITM6\tdwschema.bat` for 32-bit Windows systems

   `install_dir``\TMAITM6_x64\tdwschema.bat` for 64-bit Windows systems

   Linux UNIX

   `install_dir``/`*`arch`*`/bin/tdwschema.sh`

   The following scripts are generated for migration: tdw_migrate_setup.sql, tdw_migrate_step1.sql, and tdw_migrate_step2.sql.
8. Execute the tdw_migrate_setup.sql script and view the results to ensure it executed successfully.

   ```
   db2 -td# -f tdw_migrate_setup.sql
   ```

   Use the tdw_migrate_setup.sql script only once, even when migrating in batches. Executing this script more than once breaks the ability of the migration process to restart if an error occurs. This script contains drop statements that might fail if the objects do not already exist. Do not consider these failures as errors, they can be ignored. The expected failures might return the following messages: DB21034E and SQL0204N.

9. Execute the tdw_migrate_step1.sql script and view the results to ensure it executed successfully.

   ```
   db2 -tf tdw_migrate_step1.sql
   ```

   If any errors occur after this script is executed, the errors must be resolved before running the tdw_migrate_step2.sql script. Continue to re-execute this script until all errors are resolved.

   The following return codes apply:

   - `-2`: Table already partitioned
   - `-1`: Invalid parameter passed
   - `0`: No errors occurred
   - `1`: Renaming of the non partitioned table to MIGRATING_* failed
   - `2`: Creation of the partitioned table failed
   - `3`: Load data in the partitioned table failed
   - `4`: Renaming of the source table to DONE_* failed
10. Execute the tdw_migrate_step2.sql script and view the results to ensure it executed successfully.

    ```
    db2 -tf tdw_migrate_step2.sql
    ```

    You can execute this script multiple times to resolve any errors. This script has no effect on tables that did not migrate successfully.

    Note: If you are migrating in batches, the tdw_migrate_step1.sql and tdw_migrate_step2.sql scripts are executed for each batch.
11. Backup the database. You must complete this step since the load utility was used in a non-recoverable mode to improve migration performance. Migrated tables cannot be restored from a backup until a new backup is made.

Results

The tables you specified are now partitioned, and the source tables have been deleted.

What to do next

If any errors occurred during the migration, review the WAREHOUSE_MIGRATION_STATUS table. For detailed information, see the *IBM Tivoli Monitoring Troubleshooting Guide*.

Migrating non-partitioned tables to partitioned tables for DB2 on z/OS

Use the following steps to migrate non-partitioned tables if you are using DB2 on z/OS.

Before you begin

Review the “Prerequisites and best practices” on page 471. You must ensure you have enough disk space.

You must have DB2 on z/OS V9 or later to use this procedure.

The schema publication tool can only be executed on a distributed platform, such as UNIX or Windows. The scripts generated must be executed from a distributed platform with a DB2 client connected to the remote DB2 z/OS database.

The generated migration scripts must be executed from a DB2 client.

The Tivoli Data Warehouse user must have one or more of the following privileges in order to execute the tdw_migrate_setup.sql script:

- Ownership and EXECUTE privilege for the packages, DSNADMJS and DSNADMJF, used in the setup script
- SYSADM authority
- PACKADM authority for the package collection
- Daemon authority
  If the BPX.DAEMON is active, the stored procedures loaded into an address space must be defined to the RACF program control. Otherwise, the following error is returned: EDC5139I Operation not permitted. For detailed information about this issue, see APAR II13698 in the IBM Support Portal.

The privileges required for migration can be revoked after all the desired migrations are complete.

You must define the stored procedures to DB2 using the DSNTIJSG sample installation job, then ensure that all stored procedures are defined to RACF program control. Additionally, you must define the necessary application environment in WLM to run these stored procedures, and also specify a WLMENV value. For more information on defining the stored procedures to DB2, see the *DB2 for z/OS Installation and Migration Guide* and *DB2 for z/OS Administration Guide* for DB2 9 or later, in the DB2 for z/OS Information Center.

About this task

The migration of non-partitioned tables to partitioned tables in a Tivoli Data Warehouse DB2 on z/OS database uses a stored procedure generated by the schema publication tool. The stored procedure itself uses a JCL job which uses the DB2 LOAD utility. A JCL job is created and submitted for each table that is being migrated.

Procedure

1. Configure the Summarization and Pruning agent to partition tables. For detailed steps, see "Specifying range partitioned tables for the Summarization and Pruning Agent" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. Stop all Warehouse Proxy agent and Summarization and Pruning agent instances.
3. Backup the Tivoli Data Warehouse database.
4. Catalog the z/OS database using the following commands:
```
db2 catalog tcpip node <node_Name> remote <DB_server_hostname>
  server <port_number> ostype OS390
db2 catalog dcs database <db_name> as <db_name>
db2 catalog db <database_name_on_server> as <alias_on_client_database_name>
  at node <node_Name> authentication dcs
```
5. Edit the schema publication tool response file.
   a. Open the response file:

      Windows

      install_dir\TMAITM6\tdwschema.rsp for 32-bit Windows systems

      install_dir\TMAITM6_x64\tdwschema.rsp for 64-bit Windows systems

      Linux UNIX

      install_dir/*arch*/bin/tdwschema.rsp

b. Configure the following variables:

   `KSY_PRODUCT_SELECT = migrate`

   `KSY_PRODUCT_FILTER =` *`list of products to migrate`*
   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, see the *IBM Tivoli Monitoring Command Reference*).

   `KSY_TABLE_FILTER =` *`list of tables to migrate`*
   An optional filter to indicate only specific tables. This filter can be used in addition to the `KSY_PRODUCT_FILTER` variable. Use the following command to get the list of tables that are available for a given product. Replace each space in the attribute group name with an underscore character. For a list of table names, use the following command:
   `tacmd histListAttributeGroups -t <productcode>`

   `KSY_SUMMARIZATION_SELECTION =` *`list of aggregation periods to migrate`*
   An optional filter that can be used in addition to the `KSY_PRODUCT_FILTER` and `KSY_TABLE_FILTER` variables. This variable has an additional option when the migrate mode is used. The **R** option allows you to migrate the detailed tables. Other options are as follows:

   R: Migrate detailed tables only

   H: Hourly

   D: Daily

   W: Weekly

   M: Monthly

   Q: Quarterly

   Y: Yearly

   Filters can be combined. For example, to migrate the detail, hourly, and daily tables for the Windows OS agent:

   ```
   KSY_PRODUCT=KNT
   KSY_SUMMARIZATION_SELECTION=R,H,D
   ```

   `KSY_SQL_OUTPUT_FILE_PATH =` *`optional file path for SQL output`*
   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 variable, refer to the comments in the response file.

6. If you have not already, export the CANDLEHOME variable. Execute the following commands:

   Windows

   ```
   set CANDLE_HOME=install_dir
   ```

   Linux UNIX

   ```
   CANDLEHOME=/install_dir
   export CANDLEHOME
   ```

7. Ensure the Tivoli Enterprise Portal Server is started.

8. Run the schema publication tool script to generate the scripts required for migration.

   Windows

   `install_dir``\TMAITM6\tdwschema.bat` for 32-bit Windows systems

install_dir\TMAITM6_x64\tdwschema.bat for 64-bit Windows systems

Linux UNIX

install_dir/*arch*/bin/tdwschema.sh

The following scripts are generated for migration: tdw_migrate_setup.sql, tdw_migrate_step1.sql, and tdw_migrate_step2.sql.

9. Execute the tdw_migrate_setup.sql script and view the results to ensure it executed successfully.

   ```
   db2 -td# -f tdw_migrate_setup.sql
   ```

   Use the tdw_migrate_setup.sql script only once, even when migrating in batches. Executing this script more than once breaks the ability of the migration process to restart if an error occurs. This script contains drop statements that might fail if the objects do not already exist. Do not consider these failures as errors, they can be ignored. The expected failures might return the following messages: DB21034E and SQL0204N.

10. In the tdw_migrate_step1.sql script update the INSERT INTO WAREHOUSE_MIGRATION_CONFIG statement with the following information:
    - Specify the user ID and password required to execute the stored migration procedure. You can specify NULL for the user ID and password in the following circumstances:
      - The operating system is z/OS Version 1 Release 13 or later, and the authorization ID that is associated with the stored procedure address space has daemon authority.
      - The operating system is z/OS Version 1 Release 13 or later, and the authorization ID that is associated with the stored procedure address space does not have daemon authority but is authorized to the BPX.SRV.*userid* SURROGAT class profile, where *userid*is the authorization ID of the stored procedure. In this case, you must install APAR OA36062. For more information see the *DB2 for z/OS Administration Guide*.
    - Specify the JCL prefix library where the system LOAD and UNLOAD utilities are located.

11. Execute the tdw_migrate_step1.sql script and view the results to ensure it executed successfully.

    ```
    db2 -tf tdw_migrate_step1.sql
    ```

    If any errors occur after this script is executed, the errors must be resolved before running the tdw_migrate_step2.sql script. Continue to re-execute this script until all errors are resolved.

    The following return codes apply:
    - -5: Invalid system name specified
    - -4: Invalid job class specified
    - -3: Prefix library is null
    - -2: Table already partitioned
    - -1: Invalid parameter passed
    - 0: No errors occurred
    - 1: Rename source table failed
    - 2: Create partitioned table failed
    - 3: Create or submit migrate JCL job failed
    - 4: Query migrate JCL job status failed
    - 5: Fetch migrate JCL job output failed

- 7: Load failed
- 8: Rename source table failed

Return code 6 is intentionally left blank.

12. Execute the tdw_migrate_step2.sql script and view the results to ensure it executed successfully.

    ```
    db2 -tf tdw_migrate_step2.sql
    ```

    You can execute this script multiple times to resolve any errors. This script has no effect on tables that did not migrate successfully.

    Note: If you are migrating in batches, the tdw_migrate_step1.sql and tdw_migrate_step2.sql scripts are executed for each batch.

    When the tdw_migrate_step2.sql script is executed, the rows from the WAREHOUSE_MIGRATION_CONFIG, WAREHOUSE_JCLJOB_MIGRATION_STATUS, and WAREHOUSE_JCLJOB_OUTPUT table are deleted.
13. Backup the database. You must complete this step since the load utility was used in a non-recoverable mode to improve migration performance. Migrated tables cannot be restored from a backup until a new backup is made.

Results

The tables you specified are now partitioned, and the source tables have been deleted.

What to do next

If any errors occurred during the migration, review the WAREHOUSE_MIGRATION_STATUS, WAREHOUSE_JCLJOB_MIGRATION_STATUS, and WAREHOUSE_JCLJOB_OUTPUT tables. For detailed information, see the *IBM Tivoli Monitoring Troubleshooting Guide*.

Migrating non-partitioned tables to partitioned tables for Oracle

Use the following steps to migrate non-partitioned tables if you are using Oracle.

Before you begin

Review the “Prerequisites and best practices” on page 471. You must ensure you have enough disk space.

The Tivoli Data Warehouse user must be directly granted the following system privileges, the privileges cannot be granted via a role:

- ALTER ANY TABLE
- CREATE ANY TABLE
- DROP ANY TABLE
- LOCK ANY TABLE
- SELECT ANY TABLE
- Execution privileges for the DBMS_REDEFINITION package
  To grant authority use the `grant execute on DBMS_REDEFINITION TO <`*`Tivoli Data Warehouse user ID`*`>` command.

The privileges required for migration can be revoked after all the desired migrations are complete.

About this task

The migration of non-partitioned tables to partitioned tables in a Tivoli Data Warehouse Oracle database uses a stored procedure generated by the schema publication tool. The stored procedure itself uses the DBMS_REDEFINITION package to load data from the non-partitioned table to the partitioned tables.

Procedure

1. Configure the Summarization and Pruning agent to partition tables. For detailed steps, see "Specifying range partitioned tables for the Summarization and Pruning Agent" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. Stop all Warehouse Proxy agent and Summarization and Pruning agent instances.
3. Backup the Tivoli Data Warehouse database.
4. Edit the schema publication tool response file.
   a. Open the response file:

      Windows

      `install_dir\TMAITM6\tdwschema.rsp` for 32-bit Windows systems

      `install_dir\TMAITM6_x64\tdwschema.rsp` for 64-bit Windows systems

      Linux UNIX

      `install_dir/arch/bin/tdwschema.rsp`

   b. Configure the following variables:

      `KSY_PRODUCT_SELECT = migrate`

      `KSY_PRODUCT_FILTER = list of products to migrate`
      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, see the *IBM Tivoli Monitoring Command Reference*).

      `KSY_TABLE_FILTER = list of tables to migrate`
      An optional filter to indicate only specific tables. This filter can be used in addition to the KSY_PRODUCT_FILTER variable. Use the following command to get the list of tables that are available for a given product. Replace each space in the attribute group name with an underscore character. For a list of table names, use the following command:
      tacmd histListAttributeGroups -t <productcode>

      `KSY_SUMMARIZATION_SELECTION = list of aggregation periods to migrate`
      An optional filter that can be used in addition to the KSY_PRODUCT_FILTER and KSY_TABLE_FILTER variables. This variable has an additional option when the migrate mode is used. The **R** option allows you to migrate the detailed tables. Other options are as follows:

      R: Migrate detailed tables only

      H: Hourly

      D: Daily

      W: Weekly

      M: Monthly

      Q: Quarterly

Y: Yearly

Filters can be combined. For example, to migrate the detail, hourly, and daily tables for the Windows OS agent:

```
KSY_PRODUCT=KNT
KSY_SUMMARIZATION_SELECTION=R,H,D
```

`KSY_SQL_OUTPUT_FILE_PATH = optional file path for SQL output`
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 variable, refer to the comments in the response file.

5. Ensure the Tivoli Enterprise Portal Server is started.
6. If you have not already, export the CANDLEHOME variable. Execute the following commands:

   Windows

   ```
   set CANDLE_HOME=install_dir
   ```

   Linux UNIX

   ```
   CANDLEHOME=/install_dir
   export CANDLEHOME
   ```

7. Run the schema publication tool script to generate the scripts required for migration.

   Windows

   install_dir`\TMAITM6\tdwschema.bat` for 32-bit Windows systems

   install_dir`\TMAITM6_x64\tdwschema.bat` for 64-bit Windows systems

   Linux UNIX

   install_dir/*arch*`/bin/tdwschema.sh`

   The following scripts are generated for migration: tdw_migrate_setup.sql, tdw_migrate_step1.sql, and tdw_migrate_step2.sql.

8. Execute the tdw_migrate_setup.sql script and view the results to ensure it executed successfully.

   ```
   sqlplus <TDW userid>/<password>@<Oracle SID> @./tdw_migrate_setup.sql
   ```

   Use the tdw_migrate_setup.sql script only once, even when migrating in batches. Executing this script more than once breaks the ability of the migration process to restart if an error occurs. This script contains drop statements that might fail if the objects do not already exist. Do not consider these failures as errors, they can be ignored. The expected failures might return the following messages: DB21034E and SQL0204N.

9. Execute the tdw_migrate_step1.sql script and view the results to ensure it executed successfully.

   ```
   sqlplus <TDW userid>/<password>@<Oracle SID> @./tdw_migrate_step1.sql
   ```

   If any errors occur after this script is executed, the errors must be resolved before running the tdw_migrate_step2.sql script. Continue to re-execute this script until all errors are resolved.

   If the tdw_migrate_step1.sql script succeeds, a message is provided. For example:

   ```
   Partitioning table "AIXTST"."KSY_TABLE_STATISTICS"

   PL/SQL procedure successfully completed.
   ```

```
Table AIXTST.KSY_TABLE_STATISTICS successfully migrated.

PL/SQL procedure successfully completed.
```

If this script encounters an error, a message is provided on the standard output. For example:

```
Code: -20002 Message: ORA-20002:
Table "ITMUSER630"."K4X_USGS_STREAM_FLOW" is already partitioned.
```

The following error messages apply:

- `20000`: Invalid parameter passed
- `20001`: Source table does not exist
- `20002`: Table already partitioned
- `20003`: Error when determining if source table can be partitioned
- `20004`: Creation of target partitioned table failed
- `20005`: Unable to drop target table when redefinition was aborted or finishing
- `20006`: Migration aborted
- `20007`: Error while finishing the table redefinition
- `20008`: Error during final table rename

10. Execute the tdw_migrate_step2.sql script and view the results to ensure it executed successfully.

    ```
    sqlplus <TDW userid>/<password>@<Oracle SID> @./tdw_migrate_step2.sql
    ```

    You can execute this script multiple times to resolve any errors. This script has no effect on tables that did not migrate successfully.

    Note: If you are migrating in batches, the tdw_migrate_step1.sql and tdw_migrate_step2.sql scripts are executed for each batch.
11. Optionally, you can backup the database.

Results

The tables you specified are now partitioned, and the source tables have been deleted.

What to do next

If any errors occurred during the migration, review the error messages from the execution of the scripts.

Summarization and pruning configuration

After installation of Tivoli Management Services is complete, one of the initial setup tasks is to configure the summarization and pruning agent for general behavior, such as the summarization and pruning schedule and frequency. As well, you must specify summarization and pruning for the attribute groups that historical data is being collected for in your monitored application.

About the Summarization and Pruning agent

This topic gives you some background information to help in planning and configuring the Summarization and Pruning agent.

The Tivoli Enterprise Portal enables you to set up summarization and pruning for selected attribute groups in the **History Collection Configuration** window or from the command line using `tacmd histconfiguregroups` (see *IBM Tivoli Monitoring Command Reference*). For information about setting up data connections for the warehouse proxy and the Summarization and Pruning agent, see *IBM Tivoli Monitoring Installation and Setup Guide*.

Planning to summarize and prune your collected data

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.

History Collection Configuration window

The **History Collection Configuration** window in the Tivoli Enterprise Portal has options for specifying the time period to be aggregated and the same or different time period to be pruned: Yearly, Quarterly, Monthly, Weekly, Daily, and Hourly.

Configure Summarization and Pruning Agent window

You configure the Summarization and Pruning agent itself through the Manage Tivoli Enterprise Monitoring Services window. To see the utilization of resources or to determine the hours of peak loads, you can define a set of hours as *shifts*, for example 9 AM to 5 PM. To specify whether a particular day is a normal work day or a vacation day, you can classify the days that are not normal work days as *vacation days*. Be aware that defined shifts and vacation days increase the amount of data in the Tivoli Data Warehouse.

If the Tivoli Data Warehouse and all the agents that are collecting data are not in the same time zone, the *Timezone Indicator* identifies the time zone to use. If you chose to use the Tivoli Data Warehouse time zone, all data is changed to reflect the time zone of the Tivoli Data Warehouse. If you choose the agent's time zone, the data stays unchanged, with the original time zone for that agent.

Summarization tables in the Tivoli Data Warehouse

The following are names of the summarization tables. The *x* represents the original table name of the detailed data. The summarization interval that is chosen for the particular attribute group is appended to the original table name. Names can be different between the detailed data table and summarized table name due to database name length restrictions.

Yearly	x_Y
Quarterly	x_Q
Monthly	x_M
Weekly	x_W
Daily	x_D
Hourly	x_H

The table shows the names of the summarization columns in the detailed tables and what they mean. The *x* represents the original column name. The formula values are set by the agents and can be different for different attribute groups . Attribute names can be different between the detailed data table and summarized table due to database name length restrictions.

Table 66. Summarization functions

Name	Formula
Average	AVG_x
Delta high	HI_x
Delta low	LOW_x
Delta total	TOT_x
Latest (based on the time that the historical data was collected at the monitoring agent)	LAT_x
Maximum	MAX_x
Minimum	MIN_x
Sum	SUM_x

Names can be different between the detailed data table and summarized table name due to database name length restrictions.

Summarization and pruning metrics

The following example describes how the Summarization and Pruning agent calculates metrics that accumulate over time. You can use the results to manage your resources. In this example, the metric represents cache hits since last restart of server.

The total number of cache hits in the last hour is given by the **Total** value. The **Low** value represents the lowest number of cache hits in the hour based on all the detailed data values for the hour. The **High** value represents the highest number of cache hits in the hour based on all the detailed data values for the hour.

With these detailed data values in one hour: 9, 15, 12, 20, 22, delta-based processing has the following rules:

- If the current value is greater than or equal to the previous value, the output equals the previous value minus the current value.
- If the current value is less than the previous value, the output equals the current value.
- Because 15 is greater than 9, the output equals 6.
- Because 12 is less than 15, the output equals 12.
- Because 20 is greater than 12, the output equals 8.
- Because 22 is greater than 20, the output equals 2.
- The TOT_ value is 28, which is the total of outputs.
- The LOW_ value is 2, which is the lowest of outputs.
- The HI_ value is 12, which is the highest of outputs.

Null values in tables and charts of summarized and pruned data

If you see `null` as the value of a table cell or chart point, it means that no value was stored in the Tivoli Data Warehouse. This happens when values that were identified as invalid are reported from a monitoring agent for a given summarization period. The agent support files might have been upgraded or some data cannot be computed on the summarized tables (for instance, counter and delta-based values cannot be calculated if only one value is present).

For example, assume that an invalid value for a particular attribute is -1. If the agent reports -1 for all the collection intervals (1, 5, 15, or 30 minutes; 1 hour; 1 day) up to the point when the summarization and pruning computation is done for a given summarization period (hourly, daily, weekly, monthly, quarterly, or yearly), then there is no data to perform calculations on and a `null` is written for the given summarization.

Capacity planning suggestions for historical data collection on your Tivoli Data Warehouse

Disk capacity planning is a prediction of the amount of disk space to be consumed for each attribute group whose historical data is being collected. Required disk storage is an important factor to consider when you are defining data collection rules and your strategy for historical data collection.

For more information about performance tuning for your DB2 database, go to the IBM Integrated Service Management Library and search for part or all of this phrase: Relational database design and performance tuning for DB2 database servers. For more detailed information on capacity planning and scaling of the Tivoli Data Warehouse, see *IBM Tivoli Monitoring Installation and Setup Guide*.

Summarization after upgrading agent support

After support for an updated product has been applied to the portal sever, it is possible to get a request error message about a missing or unknown column name in the view's status bar after you set a time span with ☑ **Use summarized data** selected.

The resolution is to wait to view the summarized data until after the next scheduled summarization and pruning procedure has taken place. If need be, the summarization and pruning can be rescheduled to run sooner. More information is provided in *IBM Tivoli Monitoring Installation and Setup Guide* and in the Tivoli Enterprise Monitoring Agent User's Guide for your product.

Best practices for summarization and pruning

Use a best practices approach in determining how to summarize and prune the data samples stored in the Tivoli Data Warehouse.

Before enabling historical collection think about your business requirement for the data. There are four common use cases for the historical data. Your needs will vary for each attribute group, so consider the use cases when configuring historical collection: problem determination and debugging; reporting; capacity planning and predictive alerting; and adaptive monitoring.

For performance tuning best practices for the Summarization and Pruning agent, as well as the other monitoring components, see “Performance tuning” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Each of these use cases has different historical requirements. The following sections describe each of these use cases and the types of historical collection that will be desirable.

Problem determination and debugging

These types of metrics are used for problem determination and debugging, which tends to be relatively short term in nature. Occasionally there is a need to compare performance from a long time ago, but most of the time Subject Matter Experts (SMEs) want to go back a few days and evaluate the performance of a server or application and compare it to the current performance. In this case, there is no need for summarization of the data.

Reporting

When configuring historical collection, you need to consider the purpose of your reports. Some reports are used for long term trend analysis, some reports are used to show that SLAs are being met, and some reports are relatively short term to show the health of a server. The primary driver of the historical collection is the duration of the reports. For short term reports, you can use **Detailed** data. For short to medium term reports, use **Hourly** summarized data. For medium to long reports use **Daily** or **Weekly** summarization.

Keep in mind when configuring summarization, that you do not need to configure all intervals. For example, if you want **Weekly** summarization, you do not need to also configure both **Daily** and **Hourly**. Each summarization interval can be configured independently.

For information on using the Summarization and Pruning agent to create, populate, and maintain the dimension tables required for Tivoli Common Reporting, see "Creating and maintaining the dimension tables" on page 523.

Capacity planning and predictive alerting

For capacity planning and predictive analytics, you typically perform long term trend analysis. The Performance Analyzer, for example, uses **Daily** summarization data for the predefined analytic functions. So, in most cases, configure daily summarization. You can define your own analytic functions and use **Hourly** or **Weekly** summarization data.

For the analytic functions to perform well, ensure that you have an appropriate number of data points in the summarized table. If there are too few, the statistical analysis will not be very accurate. You will probably want at least 25 to 50 data points. To achieve 50 data points using **Daily** summarization, you must keep the data for 50 days before pruning. More data points will make the statistical predictions more accurate, but will affect the performance of your reporting and statistical analysis. Consider having no more than a few hundred data points per resource being evaluated. If you use **Hourly** summarization, you get 336 data points every 2 weeks.

Adaptive monitoring (dynamic thresholding)

The situation override capability enables you to analyze historical data to define a threshold that is based on the past performance characteristics. You can define time of day and shifts to analyze the historical data and make recommendations on thresholds.

As an example, evaluate the Prime Shift data for 2 weeks and set the threshold at 1 standard deviation about "normal". Adaptive monitoring uses **Detailed** data to evaluate and make recommendations on thresholds. Therefore, you need to keep a reasonable duration of **Detailed** data in order to perform the evaluation. The duration depends on how the shifts are defined. If you define shifts that include "day of week", then you need to keep the data longer to get an effective analysis of the data. If you are looking only at "Prime Shift" for all weekdays, then you do not need to keep the data as long.

Keep 7 to 30 days of detailed data when comparing all work days. If you compare Monday to Monday, then you need to keep the Detailed data much longer to be able to establish a trend. When comparing a specific day of the week, you will probably need to have at least 60 days of data. Before configuring Adaptive Monitoring, you need to consider the use of the data. There is no value in performing Adaptive Monitoring on certain types of data, such as disk space. You must want to set a static threshold on either the % free space or the amount of disk space available. But CPU monitoring is an excellent candidate for Adaptive Monitoring because it can be very beneficial to learn whether a server is behaving abnormally.

Agent and Attribute Group Considerations

Each Agent and each Attribute Group must be considered separately when defining Historical Collection. Many Tivoli Monitoring products have defined a set of best practice historical collections. They do not include the summarization and pruning intervals, but are a good place to start when setting up historical collection.

When looking at these recommendations, consider whether you plan to use adaptive monitoring, short term problem determination, long term reporting, or capacity planning and predictive analysis. This must be taken into account when configuring the summarization and pruning Intervals.

Summarized and pruned data availability

The first time the summarization and pruning tool is run, you might not get the results you expect. Review the installation and configuration tasks that must take place before you can expect to the data from the Tivoli Data Warehouse summarized and pruned.

The summarization and pruning procedure is dependent on having enough data in the data warehouse to work with, how the data collection and warehousing intervals are set, and whether the summarization and pruning specifications were set in the History Collection Configuration window. These installation and configuration tasks must be completed before summarized and pruned data is available from the warehouse:

1. Install the monitoring agent, then add application support for it on the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server.
2. Configure historical data collection for one or more attribute groups for the agent.
3. Distribute the historical collection to managed systems to start collecting data.
4. For each attribute group that has historical data collection taking place, configure the summarization and pruning intervals.
5. Wait for at least one warehouse interval. Check to make sure data is in the warehouse in the detailed tables. It is not sufficient to query historical data from the Tivoli Enterprise Portal because the first 24 hours comes from the short-term history files and not the data warehouse.
6. Configure the summarization and pruning agent, making sure that the test connection to the database works and that you schedule when the agent should perform work. You can configure the agent earlier, but wait for the scheduled run to complete before expecting the warehoused data to be summarized and pruned.

After the scheduled run time, you should have summary data in the warehouse.

Configuring summarization and pruning for attribute groups

Configure summarization and pruning for the Tivoli Data Warehouse to aggregate data and keep the database size at a manageable level.

Before you begin

The summarization and pruning agent must be installed, configured, and started as described in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Your user ID must have Configure History permission to open the History Collection Configuration window. If you do not have this permission, you will not see the menu item or tool for historical configuration.

About this task

Although summarization and pruning is not mandatory for warehoused data, it keeps the database from growing to an unwieldy size and minimizes the amount of data that gets retrieved to the Tivoli Enterprise Portal. Even if data collection for an attribute group has not been configured, you can set up summarization and pruning. If no collections have been created and distributed for an attribute group, no data goes to the warehouse, and summarization and pruning does not take place.

Procedure

1. If the History Collection Configuration window is not open, click **History Configuration**.
2. Select a **Monitored Application** from the tree.
3. Review the attribute groups in the table. If summarization and pruning has already been configured for an attribute group, the values will be shown in the summarization and pruning cells. Collapse the tree, drag the borders, or scroll the table right to see all the cells.
4. Select one or more attribute groups to configure. You can select multiple groups with Ctrl+click, or Shift+click to select all groups from the first one selected to this point. The settings of the first group selected continue to display, regardless of the settings in any of the other selected groups. This enables you to adjust the configuration controls once and apply the same settings to all selected attribute groups. Use the **Clear all** button if you want to clear all the fields and start over.
5. In the **Summarization** area, select the check box for every time period to be aggregated: Yearly, Quarterly, Monthly, Weekly, Daily, and Hourly.
6. In the **Pruning** area, select the check box for every time period to be pruned: Yearly, Quarterly, Monthly, Weekly, Daily, and Hourly. If you also want to keep the original data samples, select the **Detailed data** check box. In the corresponding fields, specify the number of days, months, or years to keep the data.
7. Click **Apply** to save the configuration for the attribute groups that were selected. The summarization and pruning cells for the attribute group are updated to reflect the new settings.

What to do next

The next time summarization and pruning takes place, the summarization and pruning agent applies the configuration to the long-term data stored in the data warehouse. Wait for the next scheduled time period to elapse before expecting to see any summarized data.

To configure granular summarization and pruning for managed system groups, you must use the tacmd Command Line Interface (CLI). For detailed instructions, see the *IBM Tivoli Monitoring Administrator's Guide*.

Configuring summarization and pruning for managed system groups

Configuring summarization and pruning for managed system groups provides granular control over how the Tivoli Data Warehouse aggregates and prunes data.

Granular summarization and pruning settings are configured only through the command-line interface using the **`tacmd histconfiguregroups`** command specifying the **`-g`** and **`-m`** parameters. Unlike granular summarization and pruning settings, global summarization and pruning settings can be configured through the command-line interface or in the Tivoli Enterprise Portal. If your Summarization and Pruning agent is running in autonomous mode, you cannot configure summarization and pruning for managed system groups.

Before you begin

The summarization and pruning agent must be installed, configured, and started as described in the *IBM Tivoli Monitoring Installation and Setup Guide*.

The managed system groups must be defined with their correlating systems. You can create managed system groups in the Tivoli Enterprise Portal using the Object group editor or through the command-line interface. For detailed syntax of the **`tacmd createsystemlist`** command, see the *IBM Tivoli Monitoring Command Reference*.

About this task

You can assign different summarization and pruning policies to different managed system groups. For example, one group might retain data for 180 days and the other group might retain data for 30 days. Another example is, one group summarizing hourly and the other group summarizing hourly and daily.

A managed system can be part of multiple managed system groups which might have different summarization and pruning settings. In this case, the multiple settings are merged, resulting in a union of all summarizations and retention using the largest found interval. For example:

> Group A has hourly and daily summarizations with 10 days and 30 days retention.
>
> Group B has hourly, daily, and weekly summarizations with 30 days, 60 days, and 6 months retention.

In this example, for a system that is a member of both groups, the union of the summarization settings is hourly, daily, and weekly. The retention settings for the system are 30 days for hourly, 60 days for daily, and 6 months for weekly since those are the largest setting for each summarization.

If a system does not belong to any managed system groups that have summarization and pruning settings defined, the global summarization settings are used. Global summarization is configured through the Tivoli Enterprise Portal or the command-line interface using the **`tacmd histconfiguregroups`**, not specifying the **`-g`** parameter. For more information on configuring global summarization settings, see "Configuring summarization and pruning for attribute groups" on page 486 or the *IBM Tivoli Monitoring Command Reference*.

When you use Tivoli Data Warehouse range partitioning, partitions are only dropped or rotated based on the largest retention interval that is found for the attribute group. After partitions are dropped or rotated, rows are removed by using SQL `DELETE` statements according to the settings for each managed system group or global summarization settings. When you use SQL `DELETE` statements, performance is slower than compared to dropping or rotating a partition. This decrease in performance is because SQL `DELETE` statements require transaction logs and updates to indexes, while dropping partitions does not.

When you use Tivoli Data Warehouse range partitioning, the following scenarios apply:

- If there are only global summarization and pruning settings for a table, the partitions are dropped using the defined retention values.
- If there are any managed system group summarization and pruning settings for a table, both of the following conditions must be true:
  - There are global pruning settings that are defined for the attribute group's detail or summary table.
  - There are pruning settings that are defined for the attribute group's detail or summary table for *all* configured managed system groups.

  When both conditions are not true, the summarization and pruning agent outputs a warning message in its log and self monitoring workspace.

When partitions cannot be dropped:

- The performance of the pruning operation is degraded since rows of data must be deleted by using SQL `DELETE` statements rather than dropping partitions from a table.
- Tables might eventually have partitions that contain no rows of data.

Best practice is to define the retention period globally and for each configured managed system group for the detail data and for all summary tables that might contain data.

Procedure

- Use the **`tacmd histconfiguregroups`** command with the **`-g`** and **`-m`** parameters to configure summarization and pruning for managed system groups.

  For example, the command `tacmd histconfiguregroups -t knt -o NT_Memory_64 -d HD -p H=30d,D=60d -m -g NT_Dev` creates a configuration setting for the agents that belong to the NT_Dev managed system group, with hourly and daily summarization, and retention of 30 days for the hourly table and 60 days for the daily table.
- Use the **`tacmd histviewattributegroup`** command to view the settings for a given attribute group.

  For example, the command `tacmd histviewattributegroup -t knt -o NT_Memory_64` lists the global settings in effect for that attribute group and all the managed system groups that have granular settings on that attribute group.

  To view only granular settings for a given managed system group, use the **`-g`** parameter.

  For example, the command `tacmd histviewattributegroup -t knt -o NT_Memory_64 -g NT_Dev`, lists the settings that are specific to the NT_Dev managed system group.
- Use the **`tacmd histunconfiguregroups`** command with the **`-g`** and **`-m`** parameters to remove the settings for a specific managed system group.

For example, the command `tacmd histunconfiguregroups -t knt -o NT_Memory_64 -m -g NT_Dev` removes the summarization and pruning settings specified for the NT_Dev managed system group.

What to do next

The next time summarization and pruning takes place, the summarization and pruning agent applies the configuration to the long-term data stored in the data warehouse. Wait for the next scheduled time period to elapse before expecting to see any summarized data.

Changing global configuration settings

Use the Configure Summarization and Pruning Agent window to change system-wide configuration settings for data summarization, pruning, or collection.

About this task

Complete these steps to edit the summarization and pruning agent configuration:

Procedure

1. In Manage Tivoli Enterprise Monitoring Services, right-click Summarization and Pruning agent.
2. Click on **Reconfigure**.
3. Click **OK** in the Warehouse Summarization and Pruning Agent: Advanced Configuration window.
4. Click **OK** in the next window.
5. Click **Yes** in the Warehouse Summarization and Pruning Agent window to configure the Summarization and Pruning Agent.
6. Enter the Tivoli Data Warehouse database and Tivoli Enterprise Portal server information in the **Sources** tab:
    a. In the **JDBC drivers** field, click **Add** to invoke the file browser window to select your JDBC driver. Click **OK** to close the browser and add the JDBC drivers to the list You can also highlight an entry in the JDBC drivers list and click **Delete** to remove a driver. This gives you the ability to collect JDBC drivers to communicate with your Tivoli Data Warehouse database. JDBC drivers are installed separately and each database provides a set of these JDBC drivers.

       Note:

       - If your Tivoli Data Warehouse database is on an operating system such as UNIX, find the directory where DB2 is installed and, in the jdbc drivers directory, select the db2jcc.jar and db2jcc_license_cu.jar files. For example, `<*db2_installdir*>/java/db2jcc.jar` and `<*db2_installdir*>/java/db2jcc_license_cu.jar`.
       - If your Tivoli Data Warehouse database is on MS SQL Server 2000 or 2005, install the MS SQL Server 2005 JDBC driver from the Microsoft SQL Server website. You will need the sqljbc.jar file; see the installation instructions for your operating systems from Microsoft to locate the file.
       - If your Tivoli Data Warehouse database uses Oracle, use the ojdbc14.jar file. The location on Windows is `*%ORACLE_HOME%*\jdbc\lib`; the location on operating systems such as UNIX is `*$ORACLE_HOME*/jdbc/lib`.

    b. In the drop down list, select the type of database for your Tivoli Data Warehouse.

   c. If not correct, enter the Tivoli Data Warehouse URL, Driver, Schema, User ID and password.

      Important: During the configuration of the warehouse proxy, a database user (called ITMUser by default) is created. The User ID that you enter here must match that database user.

   d. Click **Test database connection** to ensure you can communicate with your Tivoli Data Warehouse database.

   e. Enter the Tivoli Enterprise Portal Server Host and Port, if you do not want to use the defaults. The **TEP Server Port** field is numeric only.

7. Select the scheduling information in the **Scheduling** tab:
   - Fixed – Schedule the agent to run every *x* days and at what time (at least 5 minutes from now if you want it to run right away). The default is to run every day at 2:00 AM.
   - Flexible – Schedule the agent to run every *x* minutes. In the text box above the **Add** button, you can specify the times when the agent should not run, using the format HH:MM-HH:MM (24-hour clock, such as 12:00-20:00 to not run between 12:00 PM and 8:00 PM), and click **Add** to add the time range to the **Except** box.

   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 what is set on the Scheduling tab. If you select **Flexible**, the Summarization and Pruning agent runs once immediately after it is started and then at the **Run every** interval except during any blackout times.

8. Specify Shift Information and Vacation Settings in the **Work Days** tab:

   a. Select Sunday or Monday as the day the **Week starts on**.

   b. If you want shifts, select **Specify shifts**. The default settings for this field are listed in the **Peak Shift Hours** box on the right side of the window. Change these settings by selecting the hours you want in the **Off Peak Shift Hours** box and clicking the right arrow button to add them to the **Peak Shift Hours** box.

      Important: Specifying shifts is not recommended because it increases the amount of disk space needed on the data warehouse and the amount of processing time needed for summarization and pruning.

      Restriction: Changing the shift information after data has been summarized can create an inconsistency in the data. Previous data collected and summarized cannot be recalculated with the new shift values.

   c. If you want to change your vacation settings, select **Specify vacation days**. Click **Yes** or **No** to specify weekends as vacation days. Select **Add** to open a calendar, then select the vacation days to add. The days selected display in the box below the **Select vacation days** field. If you want to delete any days you have previously chosen, select them and click **Delete**.

      Linux UNIX Right-click to select the month and year.

9. Select the options you want in the **Log Parameters** tab. This tab defines the parameters for pruning the log tables populated by the warehouse proxy and the summarization and pruning agent.

   a. Select □ **Keep WAREHOUSEAGGREGLOG data for** to prune the WAREHOUSEAGGREGLOG table, which is populated by the summarization and pruning agent. After enabling this option, specify the number of days, months, or years to keep data in the table. Data older than the specified time interval will be deleted by the summarization and warehouse pruning agent.

   b. Select □ **Keep WAREHOUSELOG data for** to prune the WAREHOUSELOG table, which is populated by the warehouse proxy. After enabling this option, specify the number of days, months, or years, to keep the data in the table. Data older than the specified time interval will be deleted by the summarization and pruning agent.

10. In the **Additional Parameters** tab select these options:
    a. Specify the maximum rows that can be deleted in a single database transaction. The values are 1 through n. The default is `1000`.
    b. Specify the age of the data that you want summarized in the **Summarize hourly data older than** and **Summarize daily data older than** fields. Values are `0` through n. The default is `1` for hourly data and `0` for daily data.
    c. Choose the time zone you want to use from the **Use timezone offset from** drop down list. If the Tivoli Data Warehouse and agents that are collecting data are all not in the same time zone, and all the data is stored in the same database, use this option to identify the time zone you want to use.
    d. Specify the number of concurrent execution threads that will be used when the summarization and pruning agent is processing data in the **Number of Worker Threads**. The recommended value is twice the number of CPUs. More threads might allow the summarization and pruning agent to finish faster, but will use more resources on the machine that is running the summarization and pruning agent and will use more database resources such as connections and transaction log space.
    e. The summarization and pruning caches the most recent errors that have occurred in memory. This information is provided in an attribute group and can be viewed in workspaces that are provided with the summarization and pruning agent. The **Maximum number of node errors to display** setting specifies the maximum number of errors to store in memory. Only the most recent errors are kept. Once the limit is reached, the oldest errors are dropped.
    f. The summarization and pruning caches information about the most recent runs that were performed. This information is provided in an attribute group and can be viewed in workspaces that are provided with the summarization and pruning agent. The **Maximum number of summarization and pruning runs to display** setting specifies the maximum number of runs to store in memory. Only the most recent runs are kept. Once the limit is reached, the oldest runs are dropped.
    g. The summarization and pruning agent periodically checks that it can communicate with the data warehouse database. The **database connectivity cache time** setting determines how often to perform this check.
    h. To improve performance, the summarization and pruning agent batches updates to the data warehouse database. The **Batch mode** parameter specifies how the batching will be performed. The two options are **single managed system** and **multiple managed systems**.

11. Linux UNIX Click any of these buttons: **Save** after you have all your settings correct; **Reload** to reload the original values; or **Cancel**, at any time, to cancel out of the Configure Summarization and Pruning Agent window.

How to disable the Summarization and Pruning agent

You can disable the Summarization and Pruning agent for your entire enterprise or for particular products or sets of attribute groups.

About this task

If you want to disable summarization and pruning for your entire enterprise:

1. In Manage Tivoli Enterprise Monitoring Services, right-click the Summarization and Pruning agent in the Service/Application column.
2. Select **Stop**.

If you want to turn off summarization and pruning for a particular product or set of attribute groups in the **Historical Collection Configuration** window:

Procedure

1. In the Tivoli Enterprise Portal, click the **Historical Collection Configuration** button that is located on the toolbar.
2. Select the **Product**.
3. Select one or more **Attribute Groups**.
4. Click the **Unconfigure Groups** button.

Error logging for stored data

If the warehouse proxy agent encounters errors during the roll-off of data to the Tivoli Data Warehouse, these errors are recorded in an event log. You can set a trace option to capture additional error messages, and then view the log to help in detecting problems.

Procedure

- Open the event log where the warehouse proxy errors are listed:
  - Windows Start the Event Viewer by clicking **Start** → **Programs** → **Administrative Tools** → **Event Viewer**, then select **Application** from the Log menu. If an error occurs during data roll-off, entries are inserted into the Windows Application Event Log.
  - Linux Open the `ITM_dir/logs/*hd*.log` file.
  - On either platform, errors can also be seen in the WAREHOUSELOG table in the warehouse database.
- Activate the trace option:
  1. In Manage Tivoli Monitoring Services, right-click **Warehouse Proxy** and select **Advanced Edit Trace Parms**.
  2. Select the RAS1 filters. The default setting is **ERROR**.
  3. Accept the defaults for the rest of the fields.
  4. Click **Yes** to recycle the service.
- View the trace log containing the error messages:
  1. In Manage Tivoli Monitoring Services, right-click **Warehouse Proxy** and select **Advanced** → **View Trace Log**. The Log Viewer window displays a list of log files for the warehouse proxy.
  2. Select the appropriate log file in Select Log File. All logs are listed in this window, ordered by most recent file.
  3. Click **OK**.

Collecting Agent Operations Log history

The Agent Operations Log collects the messages occurring on the distributed agents in your enterprise. This log is part of the Tivoli Management Services agent framework. On Windows, if your historical data collection configuration includes the Agent Operations Log attribute group (OPLOG table), you must create directories for the historical data and edit each agent configuration file.

Before you begin

You must manually create history data directories for all agents that are collecting historical data on the same computer and then edit each agent configuration file on the same computer to specify the new path for short-term data collection. This is required on Windows because all agent logs by default are stored in the same *install_dir*\tmaitm6\logs\ directory and each agent creates an agent operations log file named OPLOG to store short term history data. Thus, the same OPLOG history file is being shared by all the agents; if more than one agent process attempts to warehouse history data from the same short term history binary file, the same data could get transferred to the Tivoli Data Warehouse more than once.

For example, the Windows OS and Active Directory monitoring agents are installed. Each process will create and store its operations log history data in a file named C:\IBM\ITM\TMAITM6\logs\OPLOG Now there are at least two processes attempting to share the same history data file. The data from multiple agents can be written to the same file, but the warehouse upload process will encounter problems with this setup. One agent process is not aware that, at any given time, another agent process might be performing the same warehouse data upload from the same short-term history file. This can lead to duplicate history data transferred to the warehouse database.

About this task

For each agent that collects historical data on Windows, complete these steps:

Procedure

1. Create a history child directory of *install_dir*\tmaitm6\logs\.
2. Create a k?? child directory of *install_dir*\tmaitm6\logs\history where ?? is the two-character product code. For example, c:\ibm\itm\tmaitm6\logs\history\k3z would be the path to *IBM Tivoli Monitoring Agent for Active Directory* short-term history files. The system user ID for this agent must have read and write permission for this directory.
3. Open the *install_dir*\tmaitm6\k??cma.ini agent configuration file (where ?? is the two-character product code) in a text editor. See your monitoring product user's guide for the name of the file used for agent configuration.
4. Locate the CTIRA_HIST_DIR=@LogPath@ parameter and append with \history\k?? (where ?? is the two-character product code). For example, CTIRA_HIST_DIR=@LogPath@\history\knt specifies c:\ibm\itm\tmaitm6\logs\history\knt for Windows OS agent historical data collection on this computer.
5. Save the k??cma.ini configuration file.
6. Copy the *install_dir*\tmaitm6\logs\khdexp.cfg warehouse upload status file to the \history\k?? directory. If this file is not copied to the new agent history directory, your existing history data might be warehoused more than once. It is possible that this file does not exist if the history warehousing option has never been enabled.

7. Copy any `.hdr` files and their base name counterparts (no file extension) for the agent to the new location. For example, the `c:\ibm\itm\tmaitm6\logs\history\knt` directory might look like this:

   ```
   khdexp.cfg
   netwrkin
   netwrkin.hdr
   ntprocssr
   ntprocssr.hdr
   wtlogcldsk
   wtlogcldsk.hdr
   wtmemory
   wtmemory.hdr
   wtphysdsk
   wtphysdsk.hdr
   wtserver
   wtserver.hdr
   wtsystem
   wtsystem.hdr
   ```

   Please note that you might be copying history data files from the `tmaitm6\logs` directory that are not managed by the target agent. For example, the directory might contain Oracle database history data, but you are copying the files to the new Windows OS agent history directory. The copied files that are not used by the Windows OS agent will not be needed and can safely be deleted.
8. In Manage Tivoli Enterprise Monitoring Services, right-click the monitoring agent service and click **Reconfigure**, click **OK** twice to accept the settings in the configuration windows, then **Start** the agent.

Conversion process for using delimited flat files

If you chose not to warehouse your data, you must convert your collected data to delimited flat files. Data can be scheduled for conversion either manually or automatically. If you choose to continue to convert data to delimited flat files, schedule data conversion to be automatic. Perform data conversion on a regular basis even if you are collecting historical data only to support short-term history displayed in product reports.

If the KHD_TOTAL_HIST_MAXSIZE environment variable is used, the agent can no longer write any historical data to the short-term history files once the limit is reached. This variable is a limit for the agents.

Data conversion programs

The conversion of short-term history files to delimited flat files is done by running a data rolloff program:

Linux UNIX Windows **krarloff**

z/OS **KPDXTRA**

Columns added to history data files and to meta description files

Four columns are automatically added to the history data files and to the meta description files:

- **TMZDIFF**. The time zone difference from Universal Time (GMT). This value is shown in seconds.
- **WRITETIME.** The CT time stamp when the record was written. This is a 16-character value in the format, where `c` is the century; `yymmdd` is the year, month, and day; and `hhmmssttt` is hours, minutes, seconds, and milliseconds: `cyymmddhhmmssttt`

- **SAMPLES.** The SAMPLES column increments for every value collected during the same sample and then reset to its starting value again. Rows collected on the same sample have different SAMPLES column values.
- **INTERVAL.** The time between samples, shown in milliseconds.

Note: The data warehousing process adds only two columns, TMZDIFF and WRITETIME, to the Tivoli Data Warehouse database.

Meta description files

A meta description file describes the format of the data in the source files. Meta description files are generated at the start of the historical data collection process.

The various operating system environments use different file naming conventions. Here are the rules for some operating system environments:

- IBM i and HP NonStop Kernel: Description files use the name of the data file as the base. The last character of the name is 'M'. For example, for table QMLHB, the history data file name is QMLHB and the description file name is QMLHBM.
- z/OS: Description records are stored in the PDS facility, along with the data.
- UNIX and Linux: Uses the *.hdr file naming convention.
- Windows: Uses the *.hdr file naming convention.

Sample *.hdr meta description file

```
TMZDIFF(int,0,4) WRITETIME(char,4,16)
QM_APAL.ORIGINNODE(char,20,128) QM_APAL.QMNAME(char,148,48)
QM_APAL.APPLID(char,196,12) QM_APAL.APPLTYPE(int,208,4)
QM_APAL.SDATE_TIME(char,212,16)
QM_APAL.HOST_NAME(char,228,48)
QM_APAL.CNTTRANPGM(int,276,4) QM_APAL.MSGSPUT(int,280,4)
QM_APAL.MSGSREAD(int,284,4) QM_APAL.MSGSBROWSD(int,288,4)
QM_APAL.INSIZEAVG(int,292,4) QM_APAL.OUTSIZEAVG(int,296,4)
QM_APAL.AVGMQTIME(int,300,4) QM_APAL.AVGAPPTIME(int,304,4)
QM_APAL.COUNTOFQS(int,308,4) QM_APAL.AVGMQGTIME(int,312,4)
QM_APAL.AVGMQPTIME(int,316,4) QM_APAL.DEFSTATE(int,320,4)
QM_APAL.INT_TIME(int,324,4) QM_APAL.INT_TIMEC(char,328,8)
QM_APAL.CNTTASKID(int,336,4) SAMPLES(int,340,4)
INTERVAL(int,344,4)
```

For example, an entry can have the form, where *int* identifies the data as an integer, *75* is the byte offset in the data file, and *20* is the length of the field for this attribute in the file:

```
attribute_name(int,75,20)
```

Estimating space required to hold historical data tables

The historical data tables for a product are defined in the product's documentation. Refer to the appropriate agent guide for assistance in determining the names of the tables where historical data is stored, their size, and the which are the default tables.

Limiting the growth of short-term history files

Whether your environment includes a data warehouse or is set up for conversion of short-term history to delimited flat files, it is a good idea to set a maximum size for the history files.

Before you begin

Your operating system user ID must have write permission for this directory.

These agent environment variables are not available on z/OS.

About this task

When your configuration includes data roll-off to the Tivoli Data Warehouse, the size of the short-term history files is controlled by the amount of data being collected, the frequency of collection, and the frequency of roll-off to the data warehouse. Yet, it is possible for the warehouse proxy agent or data warehouse to become unavailable, which means the short-term history files can grow unchecked.

Set the KHD_TOTAL_HIST_MAXSIZE and KHD_HISTSIZE_EVAL_INTERVAL environment variables at every Tivoli Enterprise Monitoring Agent where historical data is collected or at the Tivoli Enterprise Monitoring Server if data collection occurs there.

Complete these steps to specify a size limit for the directory where short-term history files are saved and how often that this check should take place:

Procedure

1. Open the environment file for the agent:
   - Windows In the Manage Tivoli Monitoring Services window, right-click the component and click **Advanced → Edit ENV File**. (These are the *install_dir*\TMAITM6\K<pc>ENV files where <pc> is the two-character product code, such as C:\IBM\ITM\TMAITM6\KNTENV.)
   - Linux UNIX Change to the *install_dir*/config directory and open <pc>.ini in a text editor, where <pc> is the two-character product code. For example, /opt/IBM/ITM/config/ux.ini for the UNIX OS agent.

   For a list of product codes see “IBM Tivoli product, platform, and component codes” in the *IBM Tivoli Monitoring Installation and Setup Guide*.
2. Add two new lines to the file, where 5 is the maximum number of megabytes that the directory where the short-term history file is located can grow to; and where 900 (15 minutes) is the number of seconds between evaluation of the directory size:
   ```
   KHD_TOTAL_HIST_MAXSIZE =5
   KHD_HISTSIZE_EVAL_INTERVAL=900
   ```
3. Save and close the file.
4. Recycle the component.

Results

After you set a maximum and the directory limit is reached, no new records are written to the short-term history files, which causes gaps to occur in the data collected. However, if the data is warehoused, the warehouse proxy will trim the short term history files to contain only the last 24 hours of data. This can allow the agent to write historical data again; thus, the limit can be reached again and the process repeats. This process can also cause gaps to appear in the data.

What to do when the short-term history file directory size reaches its limit

When the KHD_TOTAL_HIST_MAXSIZE and KHD_HISTSIZE_EVAL_INTERVAL environment variables have been set for the Tivoli Enterprise Monitoring Agent (or at the Tivoli Enterprise Monitoring Server if data collection occurs there), no more historical data samples are added to the short-term history files if that maximum directory size has been reached.

You must resolve the cause of the unchecked short-term history file growth before the saving of data samples to the history files can resume. When data is collected at the agent you can create a custom SQL query or a situation or both that reports when this condition occurs.

Here is an example of a custom SQL query that you can run:

```
SELECT ORIGINNODE, CATEGORY, SEVERITY, TABLE, TIMESTAMP, MESSAGE
FROM O4SRV.KRAMESG WHERE ORIGINNODE = $NODE$
```

Converting short-term history files to delimited flat files

If you selected the option to store data to a data base, that option is mutually exclusive with running the file conversion programs described in this section. To use these conversion procedures, you must have specified **Off** for the Warehouse option in the History Collection Configuration window of the Tivoli Enterprise Portal.

The conversion procedure empties the history accumulation files and must be performed periodically so that the history files do not take up needless amounts of disk space.

Note: For every ATTRIBUTEGROUPNAME that you plan to use with the krarloff program or **`itmcmd history`**, you must first stop historical collection for that attribute by either using the **`tacmd histstopcollection`** or the History Collection Configuration window in the Tivoli Enterprise Portal.

Corruption of the data can occur if both programs, the historical collection and the krarloff program, try to access the data at the same time. Stopping the historical collection ensures that the saving and roll off of the data cannot be done simultaneously.

Converting files using the krarloff program

The krarloff rolloff program can be run either at the Tivoli Enterprise Monitoring Server or in the directory where the monitoring agent is running, from the directory in which the history files are stored.

Attributes formatting

Some attributes need to be formatted for display purposes. For example, floating point numbers that specify a certain number of precision digits to be printed to the left of a decimal point. These display formatting considerations are specified in product attribute files.

When you use the krarloff rolloff program to roll off historical data into a text file, any attributes that require format specifiers as indicated in the attribute file are ignored. Only the raw number is seen in the rolled off history text file. Thus, instead of displaying 45.99% or 45.99, the number 4599 displays.

The warehouse proxy inserts data according to the type, length, and data precision specified in the attribute files. However, the Tivoli Data Warehouse database displays the correct attribute formatting *only* for those attributes that use integers with floating point number formats.

You can use the krarloff rolloff program to covert the history file of the Tivoli System Monitor Agent into a text file on Linux and UNIX. The krarloff rolloff program is used because the Tivoli System Monitor Agent does not provide the command **itmcmd history** on UNIX and Linux.

Procedure

Windows

Run the krarloff rolloff program from the directory in which the monitoring server or the monitoring agent is run by entering the following at the command prompt:

```
krarloff [-h] [-g] [-x] [-d delimiter] [-m metafile] [-r rename-to-file]
[-o output-file] {-s source | source-file name}
```

Where the [ ] square brackets denote the optional parameters and the { } curly braces denote a required parameter listed below.

Linux

1. Set the environment variables:

   ```
   export PATH=$PATH: $CANDLEHOME/tmaitm6/<interp>/bin/
   export ATTRLIB=$CANDLEHOME/<interp>/lz/tables/ATTRLIB
   export LD_LIBRARY_PATH=$CANDLEHOME/<interp>/gs/lib
   ```

2. Copy the header and data file to another directory:

   ```
   mkdir $CANDLEHOME/<interp>/lz/hist/tmp
   cp PVTHIST_LNXCPU* $CANDLEHOME/<interp>/lz/hist/tmp
   cd tmp
   ```

3. Execute the krarloff program to convert the history file into a text file. For example:

   ```
   krarloff -h -d ";" -m PVTHIST_LNXCPU.hdr -o PVTHIST_LNXCPU.out
   -s PVTHIST_LNXCPU
   ```

UNIX

1. Set the environment variables:

   ```
   export PATH=$PATH: $CANDLEHOME/tmaitm6/<interp>/bin/
   export ATTRLIB=$CANDLEHOME/<interp>/ux/tables/ATTRLIB
   ```

2. Copy the header and data file to another directory:

   ```
   mkdir $CANDLEHOME/<interp>/ux/hist/tmp
   cp PVTHIST_UNIXDISK* $CANDLEHOME/<interp>/ux/hist/tmp
   cd tmp
   ```

3. Execute the krarloff program to convert the history file into a text file. For example:

   ```
   krarloff -h -d ";" -m PVTHIST_UNIXDISK.hdr -o PVTHIST_UNIXDISK.out
   -s PVTHIST_UNIXDISK
   ```

After the conversion is finished, the history file is renamed to `*.old`. For example; `PVTHIST_LNXCPU` becomes `PVTHIST_LNXCPU.old`.

The agent outputs all private history files to this subdirectory:

Windows *install_dir*\TMAITM6\logs

Linux UNIX *install_dir*/*<arch>*/*<pc>*/hist

Restriction: The krarloff rolloff program is not supported for the IBM i agent on AS400 systems.

Krarloff rolloff program parameters

The following table lists the krarloff rolloff program parameters, their purpose, and default values.

Table 67. Parameters for the krarloff rolloff program

Parameter	Default Value	Description
-h	off	Controls the presence or absence of the header in the output file. If present, the header is printed as the first line. The header identifies the attribute column name.
-g	off	Controls the presence or absence of the product group_name in the header of the output file. Add the -g to the invocation line for the krarloff rolloff program to include a group_name.attribute_name in the header.
-x	off	Excludes the SAMPLES and INTERVAL attributes in the output file.
-d	tab	Delimiter used to separate fields in the output text file. Valid values are any single character (for example, a comma).
-m	source-file.hdr	metafile that describes the format of the data in the source file. If no metafile is specified on the command-line, the default file name is used.
-r	source-file.old	Rename-to-filename parameter used to rename the source file. If the renaming operation fails, the script waits two seconds and retries the operation.
-o	source-file.*nnn* where *nnn* is Julian day	Output file name. The name of the file containing the output text file.
-s	none	**Required parameter**. Source short-term history file that contains the data that needs to be read. Within the curly brace, the vertical bar (|) denotes that you can either use an -s source option, or if a name with no option is specified, it is considered a source file name. No defaults are assumed for the source file.

Converting history files to delimited flat files on Windows systems

The history files collected using the rules established in the historical data collection configuration program can be converted to delimited flat files for use in a variety of popular applications to easily manipulate the data and create reports and graphs. Use the Windows **AT** command to schedule file conversion automatically. Use the krarloff rolloff program to manually invoke file conversion. For best results, schedule conversion to run every day.

Conversion process using the AT command

When setting up the process that converts the history files you have collected to delimited flat files, schedule the process automatically the Windows **AT** command, or manually by running the krarloff rolloff program. History file conversion can occur whether or not the Tivoli Enterprise Monitoring Server or the agent is running.

Note: Run history file conversion every 24 hours.

Archiving procedure using the Windows AT command:

To archive historical data files on Tivoli Enterprise Monitoring Servers and on remote managed systems using the **AT** command, use the procedure that follows. To find out the format of the command, enter **AT /?** at the MS/DOS command prompt.

1. For the AT command to function, you must start the Task Scheduler service. To start the Task Scheduler service, select **Settings >Control Panel > Administrative Tools > Services**.

   Result: The Services window displays.
2. At the Services window, select **Task Scheduler**. Change the service Start Type to Automatic. Click **Start**.

   Result: The Task Scheduler service is started.

   An example of using the AT command to archive the history files is as follows:

   ```
   AT 23:30 /every:M,T,W,Th,F,S,Su c:\sentinel\cms\archive.bat
   ```

   In this example, Windows runs the archive.bat file located in c:\sentinel\cms everyday at 11:30 pm. An example of the contents of archive.bat is:

   ```
    krarloff -o memory.txt wtmemory
    krarloff -o physdsk.txt wtphysdsk
    krarloff -o process.txt wtprocess
    krarloff -o system.txt wtsystem
   ```

Location of the Windows executable files and historical data collection table files:

This section discusses the location of Windows programs needed for converting historical data.

The programs are in these locations:
- *install_dir*\cms directory on the Tivoli Enterprise Monitoring Server.
- *install_dir*\tmaitm6 directory on the remote managed systems where the agents were installed.

If your agent history data has been configured to be stored at the agent computer and you want to store your history files on a disk that provides more storage capacity than the default history data file location provides, this location can be overridden using the existing environment variable *CTIRA_HIST_DIR* for your agent. This can not be done when history data is stored at the Tivoli Enterprise Monitoring Server.

If you have multiple instances of the same agent running on the same Windows system, the installer creates a separate directory for the process history files stored at the agent. The default location for agents running on the Windows operating system is C:\IBM\ITM\TMAITM6\LOGS. New directories are created under the TMAITM6\LOGS directory: History*<3 character component code>(KUM, KUD, and so on)**<specified multi-process instance name>*.

For example, if you configure a second instance of the DB2 Monitoring agent called *UDBINST1* on the same Windows system, a directory called C:\IBM\ITM\TMAITM6\LOGS\History\KUD\UDBINST1 is created to store the history data. This instance of the DB2 agent environment variable CTIRA_HIST_DIR is set to this value.

Location of Windows historical data table files:

The following section describes the location of Windows historical data table files.

The krarloff rolloff program needs to know the location of these files.

If you run the monitoring server and agents as processes or as services, the historical data table files are located in the:

- *install_dir*\cms directory on the monitoring server
- *install_dir*\tmaitm6\logs directory on the managed systems

Converting history files to delimited flat files on an IBM i system

The history files collected using the rules established in the historical data collection configuration program can be converted to delimited flat files for use in a variety of popular applications to easily manipulate the data and create reports and graphs. Use the krarloff rolloff program to manually invoke file conversion.

Note: Run history file conversion every 24 hours.

Storing the historical data stored on an IBM i system

User data is stored in the IFS directory set for the configuration variable *CTIRA_HIST_DIR*. The default value for this variable is*/qibm/userdata/ibm/itm/hist*. For each table, there are two files stored on the IBM i system that are associated with historical data collection.

For example, if you are collecting data for the system status attributes, these two files are:

- KA4SYSTS: This is the short-term data that is displayed as output by the IBM i agent.
- KA4SYSTS.hdr: This is the metafile. The metafile contains a single row of column names.

The contents of both files can be displayed using WRKLNK */qibm/userdata/ibm/itm.hist* command.

Conversion process on an IBM i system

The krarloff rolloff program can be run either at the Tivoli Enterprise Monitoring Server or in the directory in which the monitoring agent is running from the directory in which the history files are stored.

Run the krarloff rolloff program by entering the following at the command prompt:

```
call qautomon/krarloff parm (['-h'] ['-g'] ['-x'] ['-d' 'delimiter']
['-m' metafile] ['-r' rename-source-file-to] ['-o' output-file]
{'-s' source-file | source-file )}
```

Where the [ ] square brackets denote the optional parameters and the { } curly braces denote a required parameter.

If you run the krarloff rolloff program from an IBM i system in the directory in which the agent is running, replace `qautomon`with the name of the executable for your agent. For example, the MQ agent uses `kmqlib` in the command string.

Note: Enter the command on a single line.

After running the krarloff rolloff program

In using the system status example above, after running the krarloff rolloff program, file KA4SYSTS becomes KA4SYSTSO. A new KA4SYSTS file is generated when another row of data is available.

KA4SYSTSM remains untouched.

KA4SYSTSH is the file that is displayed as output by the krarloff rolloff program and that contains the data in delimited flat file format. This file can be transferred from the IBM i to the workstation by means of a file transfer program (FTP).

Converting history files to delimited flat files on UNIX Systems

This topic explains how the UNIX **itmcmd history** script is used to convert the saved historical data contained in the history data files to delimited flat files. You can use the delimited flat files in a variety of popular applications to easily manipulate the data to create reports and graphs.

History data conversion overview

The following section describes the procedure of converting historical data tables to other file types for the purpose of being used by other software products.

In the UNIX environment, you use the **itmcmd history** script to activate and customize the conversion procedure used to turn selected Tivoli Monitoring short-term historical data tables into a form usable by other software products. The historical data that is collected is in a binary format and must be converted to ASCII to be used by third party products. Each short-term file is converted independently. The historical data collected by the Tivoli Enterprise Monitoring Server can be at the host location of the Tivoli Enterprise Monitoring Server or at the location of the reporting agent. Conversion can be run at any time, whether or not the Tivoli Enterprise Monitoring Server or agents are active.

Conversion applies to all history data collected under the current *install_dir* associated with a single Tivoli Enterprise Monitoring Server, whether the data was written by the Tivoli Enterprise Monitoring Server or by a monitoring agent.

When you enter:

```
itmcmd history -h
```

at the command-line, this output displays:

```
itmcmd history [ -h install_dir ] -C [ -L nnn[Kb|Mb] ] [ -t masks*,etc ]
   [ -D delim ] [ -H|+H ] [ -N n ] [ -p cms_name ]
   prod_code itmcmd history -A?itmcmd history [ -h install_dir ]
   -A perday|0 [ -W days ] [ -L nnn[Kb|Mb] ] [ -t masks*,etc ]
   [ -D delim ] [ -H|+H ] [ -N n ]
   [ -i instance|-p cms_name ] [ -x ] prod_code
```

Note: Certain parameters are required. Items separated with a | vertical bar denotes mutual exclusivity (for example, Kb|Mb means enter either Kb or Mb, not both.) Typically, parameters are entered on a single line at the UNIX command line.

See the *IBM Tivoli Monitoring Command Reference* for all of the parameters used with this command.

Performing the history data conversion

The **itmcmd history** script schedules the conversion of historical data to delimited flat files. Both the manual process to perform a one-time conversion and the conversion script that permits you to schedule automatic conversions are described here.

See the *IBM Tivoli Monitoring: Command Reference* for a complete description of the syntax and options.

After the conversion has taken place, the resulting delimited flat file has the same name as the input history file with an extension that is a single numerical digit. For example, if the input history file table name is KOSTABLE, the converted file is named KOSTABLE.0. The next conversion is named KOSTABLE.1, and so on.

Performing a one-time conversion:

To perform a one-time conversion process, change to the *install_dir*/bin and enter the following at the command line:

```
./itmcmd history -C prod_code
```

Scheduling basic automatic history conversions:

Use **itmcmd history** to schedule automatic conversions by the UNIX *cron* facility. To schedule a basic automatic conversion, enter the following at the command line:

```
./itmcmd history -A n prod_code
```

where *n* is a number from 1-24. This number specifies the number of times per day the data conversion program runs, rounded up to the nearest divisor of 24. The product code is also required.

For example, the following command means to run history conversion every three hours:

```
itmcmd history -A 7 ux
```

Customizing your history conversion:

You can use the **itmcmd history** script to further customize your history collection by specifying additional options. For example, you can choose to convert files that are above a particular size limit that you have set. You can also choose to perform the history conversion on particular days of the week.

See the *Command Reference* for a description of all of the history conversion parameters.

Converting history files to delimited flat files on HP NonStop Kernel Systems

If you selected the option to collect and store data to a data warehouse, that option is mutually exclusive with running the file conversion programs described in this chapter. To use these conversion procedures, you must have specified **Off** for the **Warehouse** option on the **History Collection Configuration** window of the Tivoli Enterprise Portal.

The history files collected using the rules established in the History Configuration program can be converted to delimited flat files for use in a variety of popular applications to easily manipulate the data and create reports and graphs. Use the krarloff rolloff program to manually invoke file conversion. For best results, schedule conversion to run every day.

Support is provided for IBM Tivoli Monitoring for WebSphere MQ Configuration and for IBM Tivoli Monitoring for WebSphere MQ Monitoring running on the HP NonStop Kernel operating system (formerly Tandem). For information specific to IBM Tivoli Monitoring for WebSphere MQ Monitoring relating to historical data collection, see the Customizing Monitoring Options topic found in your version of the product documentation.

Conversion process on HP NonStop Kernel Systems

When setting up the process that converts the history files you have collected to delimited flat files, schedule the process manually by running the krarloff rolloff program. Run history file conversion every 24 hours.

Using the krarloff rolloff program on HP NonStop Kernel:

The history files are kept on the DATA subvolume, under the default <$VOL>.CCMQDAT. However, the location of the history files is dependent on where you start the monitoring agent. If you started the monitoring agent using STRMQA from the CCMQDAT subvolume, the files are stored on CCMQDAT.

You can run the krarloff rolloff program from the DATA subvolume by entering the following:

RUN <$VOL>.CCMQEXE.KRARLOFF <parameters>

Note that CCMQDAT and CCMQEXE are defaults. During the installation process, you can assign your own names for these files.

Attribute formatting:

Some attributes must be formatted for display purposes. For example, floating point numbers that specify a certain number of precision digits to be printed to the left of a decimal point. These display formatting considerations are specified in product attribute files.

When you use the krarloff rolloff program to roll off historical data into a text file, any attributes that require format specifiers as indicated in the attribute file are ignored. Only the raw number is seen in the rolled off history text file. Thus, instead of displaying 45.99% or 45.99, the number 4599 displays.

Converting history files to delimited flat files on z/OS systems

You can convert the short-term history files to delimited flat files on z/OS systems with a manual archiving procedure or as part of your persistent data store maintenance procedures.

The short-term history files can be converted to delimited flat files automatically as part of your persistent data store maintenance procedures, or they can be converted manually with the MODIFY command. The delimited flat file serves as input to applications for data manipulation and report creation. For more information, see the *IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/zcommonconfig/zcommonconfig.htm)

Data that has been collected and stored cannot be extracted because this data is deleted from the persistent data store. To use these conversion procedures, you must have set the **Warehouse Interval** to **Off** in the History Collection Configuration window. For more details on the History Collection Configuration window, see the Tivoli Enterprise Portal online help or Creating a historical collection in the *Tivoli Enterprise Portal User's Guide*.

Related reference:

"Manual archiving procedure" on page 509
To manually convert historical data files on the Tivoli Enterprise Monitoring Server and on the remote managed systems, issue the following MODIFY command:

Automatic conversion and archiving process on z/OS systems

This section contains information on the automatic conversion and archiving process that takes place on z/OS systems.

When you customized your IBM Tivoli Monitoring environment, you were given the opportunity to specify the EXTRACT option for maintenance. Specification of the EXTRACT option ensures that scheduling of the process to convert and archive information stored in your history data tables is automatic. No further action on your part is required. As applications write historical data to the history data tables, the persistent data store detects when a given data set is full, launches the KPDXTRA process to copy the data set, and notifies the Tivoli Enterprise Monitoring Server that the data set can be used again to receive historical information. Additional information about the persistent data store can be found in *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

An alternative to the automatic scheduling of conversion is manually issuing the command to convert the historical data files.

Note: The KPDXTRA process currently does not support UTF8 columns.

Converting files using the KPDXTRA program:

The conversion program, KPDXTRA, is called by the persistent data store maintenance procedures when the EXTRACT option is specified for maintenance. This program reads a data set containing the collected historical data and writes out two files for every table that has data collected for it. The processing of this data does not interfere with the continuous collection being performed.

Because the process is automatic, a brief overview of the use of the KPDXTRA program is provided here. For full information about the KPDXTRA program, review the sample JCL distributed with your Tivoli Monitoring product. The sample JCL is found as part of the sample job the KPDXTRA program contained in the sample libraries RKANSAM and TKANSAM.

Attribute formatting:

Some attributes must be formatted for display purposes. For example, floating point numbers that specify a certain number of precision digits to be printed to the left of a decimal point. These display formatting considerations are specified in product attribute files.

When you use KPDXTRA to roll off historical data into a text file, any attributes that require format specifiers as indicated in the attribute file are ignored. Only the raw number is seen in the rolled off history text file. Thus, instead of displaying 45.99% or 45.99, the number 4599 displays.

About KPDXTRA:

KPDXTRA program runs in the batch environment as part of the maintenance procedures. It is capable of taking a parameter that allows the default column separator to be changed. The z/OS JCL syntax for executing this command is:

```
// EXEC PGM=KPDXTRA,PARM='PREF=dsn-prefix [DELIM=xx] [NOFF]'
```

Several files must be allocated for this job to run.

All data sets are kept in read/write state even if they are not active. This makes the data sets unavailable if the Tivoli Enterprise Monitoring Server is running. Thus, jobs cannot be run against the active data sets and the inactive data sets must be taken offline.

You can remove a data set from the Tivoli Enterprise Monitoring Server by issuing the following command:

```
F stcname,KPDCMD DELFILE FILE=DSN:datastorefile
```

Note that DELFILE drops the file only from the PDS; it does not not delete the file. The file can be added back into the PDS GROUP by issuing a RESUME command:

```
F stcname,KPDCMD RESUME FILE=DSN:datastorefile
```

If you must run a utility program against an active data store, issue a SWITCH command prior to issuing this command.

DD names required to be allocated for KPDXTRA:

The following is a summary of the DD names that must be allocated for the KPDXTRA program. Refer to the sample JCL in the Sample Libraries distributed with the product for additional information.

Table 68. DD names required

DD name	Description
RKPDOUT	KPDXTRA log messages
RKPDLOG	PDS messages
RKPDIN	Table definition commands file (input to PDS subtask) as set up by the configuration tool
RKPDIN1	PDS file from which data is to be extracted
RKPDIN2	Optional control file defined as a DUMMY DD statement

KPDXTRA parameters:

The following table specifies the parameters of KPDXTRA, along with their default values and descriptions.

Table 69. KPDXTRA parameters

Parameter	Default Value	Description
PREF=	none	Required parameter. Identifies the high level qualifier where the output files are written.

Table 69. KPDXTRA parameters (continued)

Parameter	Default Value	Description
DELIM=	tab	Specifies the separator character to use between columns in the output file. The default is a tab character X'05'. To specify some other character, specify the 2-byte hexadecimal representative for that character. For example, to use a comma, specify DELIM=6B.
QUOTE	NQUOTE	Optional parameter that puts double quotes around all character type fields. Trailing blanks are removed from the output. Makes the output format of the KPDXTRA program identical in format to the output generated by the distributed krarloff rolloff program.
NOFF	off	Causes the creation (if set to ON) or omission (if set to OFF) of a separate file (header file) that contains the format of the tables. Also controls the presence or absence of the header in the output data file that is created as a result of the extract operation. If OFF is specified, the header file is not created but the header information is included as the first line of the data file. The header information shows the format of the extracted data.

KPDXTRA program messages:

These messages can be found in the RKPDOUT sysout logs created by the execution of the maintenance procedures:

```
Persistent datastore Extract program KPDXTRA - Version V130.00
Using output file name prefix: CCCHIST.PDSGROUP
The following characters are used to delimit output file tokens:
Column values in data file............:  0x05
 Parenthesized list items in format file:  0x6b
Note: Input control file not found; all persistent data is extracted.

Table(s) defined in persistent datastore file  CCCHIST.PDSGROUP.PDS#1:
```

Application Name	Table Name	Extract Status
PDSSTATS	PDSCOMM	Excluded
PDSSTATS	PDSDEMO	Included
PDSSTATS	PDSLOG	Included
PDSSTATS	TABSTATS	Included

```
Checking availability of data in data store file:
  No data found for Appl: PDSSTATS Table: PDSDEMO  . Table excluded.
  No data found for Appl: PDSSTATS Table: TABSTATS . Table excluded.
The following 1 table(s) are extracted:
```

Application Name	Table Name	No. of Rows	Oldest Row	Newest Row
PDSSTATS	PDSLOG	431	1997/01/10 05:51:20	1997/02/04 02:17:54

```
Starting extract operation.
Starting extract of PDSSTATS.PDSLOG.
The output data file, CCCHIST.PDSGROUP.D70204.PDSLOG, does not exist; it is created.
The output format file, CCCHIST.PDSGROUP.F70204.PDSLOG, does not exist;
  it is created.
Extract completed for PDSSTATS.PDSLOG. 431 data rows retrieved, 431 written.
Extract operation completed.
```

Location of the z/OS executable files and historical data table files

The following section identifies the location of z/OS executable files and historical data table files.

The z/OS executable files are located in the *&rhilev.&rte.*RKANMOD or *&thilev.*TKANMOD library, where:

- *&rhilev* is the is the high-level qualifier for the runtime environment.
- *&rte* is the is the name of the runtime environment.
- *&thilev* is the high-level qualifier of the target libraries that were installed by SMP/E.

The z/OS historical data files created by the extraction program are located in the following library structure:

- &*hilev.&midlev.&dsnlolev.tablename.*D
- &*hilev.&midlev.&dsnlolev.tablename.*H

Manual archiving procedure

To manually convert historical data files on the Tivoli Enterprise Monitoring Server and on the remote managed systems, issue the following MODIFY command:

```
F stcname,KPDCMD SWITCH GROUP=cccccccc EXTRACT
```

where:

- *stcname* identifies the name of the started task that is running either the Tivoli Enterprise Monitoring Server or agents.
- *cccccccc* identifies the group name associated with the persistent data store allocations. The values for *cccccccc* can vary based on which products are installed. The standard group name is GENHIST.

When this command is run, only the tables associated with the group identifier are extracted. If multiple products are installed, each can be controlled by separate SWITCH commands.

This switching can be automated by using either an installation scheduling facility or an automation product.

You can also use the Tivoli Enterprise Portal's advanced automation features to run the SWITCH command. To do so, define a situation that, when it becomes true, runs the SWITCH command as the action.

Maintaining the Persistent Data Store

You have the option to run the PDS on the z/OS Tivoli Enterprise Monitoring Server or the agent. It provides the ability to record and retrieve tabular relational data 24 hours a day while maintaining indexes on the recorded data.

See “Configure the persistent data store” in *Configuring the Tivoli Enterprise Monitoring Server on z/OS* for instructions on configuring the persistent datastore.

Using your historical data for analytic usage

You can use external applications, such as an analytic engine, to be alerted about and help predict potential problems quickly.

Analytic engines need data to be available with low latency and high frequency. You must configure your Warehouse Proxy agent and Tivoli Enterprise Monitoring Agent to emit comma-separated (CSV) files that can be processed by an analytic engine.

Warehouse Proxy agent for analytics

You can enable the Warehouse Proxy agent to write incoming historical data to the Tivoli Data Warehouse and to a comma-separated (CSV) file. Best practice is to only write data to a CSV file if the data will be consumed by an external application, such as an analytic engine.

You must have a connection to the Tivoli Data Warehouse to create CSV files.

The Warehouse Proxy agent receives export requests from agents that are stored in a memory queue where they can sit for a long time depending on how busy the Warehouse Proxy agent is. This wait can affect analytics that rely on the data from the Tivoli Data Warehouse because of a latency between the data collection at the agent, the data being exported to the Warehouse Proxy agent, and the actual write to the Tivoli Data Warehouse database.

For quicker access to data, you can now configure the Warehouse Proxy agent to write to a local CSV file as the data from the agent is received, allowing faster processing and results in a reduction of the load on the Tivoli Data Warehouse.

Before creating CSV files, ensure the following conditions are met:

- You have sufficient disk space on the file system where the files will be written.
- An external application is set up to consume and delete the files. When the maximum size threshold is reached, no more CSV files are created. To reduce the load on storage resources, delete the CSV files when the files are processed by your analytic engine.
- Monitor the file system to ensure data is being consumed correctly with no time gap in the data.

Automatic maintenance is not available for the CSV files created. However, the total size of the written files is monitored, and file creation stops if the total size exceeds a configured threshold. The evaluation period and maximum size value are configurable environment variables.

Attention: No provision exists for avoiding duplication of data. The data sent might fail to be inserted into the Tivoli Data Warehouse due to any number of reasons. Events such as a timeout or database failure can cause the data to be resent. The Warehouse Proxy agent has no internal state or knowledge that the data is a resend, because the new data might have more rows or data added to it since data collection has occurred on the agent.

File format

The CSV file is created as soon as the data is completely exported by the agent. Initially, the file is created with a temporary name, and when all the data has been written the file is named as follows: *<TABLENAME>*_*<MIN TIMESTAMP>*_*<MAX TIMESTAMP>*_*<SEQUENCE NUMBER>*.csv. Data is written to the CSV files in UTF8 format.

Note: You can use the KHD_CSV_ISO_DATE_FORMAT variable to change the default proprietary timestamp in the filename to ISO format. For more information see the "Parameters for configuration" section in this topic.

The data is written in the export order as it is sent by the agent. The data includes a header line containing the column names for readability purposes, and to provide descriptive information to the data.

For example, a file named WTLOGCLDSK_110314090000000_1110314090000001_00001.csv might contain the following information:

```
"WRITETIME","Server_Name","Disk_Name","Free_Space_KB"
"1110314090000000","Primary:BRAZIL:NT","C:","163024"
"1110314090000001","Primary:BRAZIL:NT","D:","88040"
```

Note: The columns in the CSV file might be a subset of the columns in the Tivoli Data Warehouse, depending on the level of the agent sending the data.

Parameters for configuration

Warehouse Proxy agent environment variables

Use the following environment variables to configure your Warehouse Proxy agent:

KHD_CSV_EVAL_INTERVAL

Optional. The interval in seconds between evaluation of the total size of the CSV files in the output directory. The time is specified in seconds, with a minimum of 60 seconds (1 minute). The default is **900 seconds** (15 minutes).

KHD_CSV_ISO_DATE_FORMAT

Optional. Use Y to enable the conversion of proprietary format timestamps to ISO timestamps.

A proprietary format timestamp is defined as a fixed string of 16 characters in the format CYYMMDDHHMMSSmmm, where

C = Century (0 for 20th, 1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = Millisecond

For example:

Proprietary timestamp: 1130513094020000

ISO timestamp: 2013-05-13T09:40:20.000

If the column is declared as a timestamp of 16 characters but does not meet the requirements as described, then the column is not converted.

The converted timestamps do not include time zone information. The meaning of the timestamp is based on the agent producing the data. Some agents output timestamps in local time and others in Greenwich mean time (GMT).

The CSV file name is also modified to use ISO timestamps. For example, the file name for table NTMEMORY with data from 2008/09/27 12:00 to 2008/09/27 19:00 would change from:

File name with proprietary timestamp: NTMEMORY_1080927120000000_1080927190000000_00000.csv

File name with ISO timestamp: NTMEMORY_2008-09-27T12-00-00-000_2008-09-27T19-00-00-000_00000.csv

The log file with error tracing indicates which columns containing proprietary timestamps were not converted due to invalid values. The log file with state tracing indicates which columns containing proprietary timestamps were converted to ISO timestamps.

KHD_CSV_ISO_DATE_FORMAT_EXCLUDE
Optional. Specify a list of column names, separated by a comma or space, to be excluded from the timestamp conversion to ISO format. The short table name must prefix the short column name, with a period separating the two names.

KHD_CSV_OUTPUT
Required. The directory where the CSV files are written. If the directory is not specified or does not exist, a message is logged to the trace file, an audit message is generated, and the CSV output is disabled.

KHD_CSV_OUTPUT_ACTIVATE
Required. Use Y to enable CSV output and N to disable CSV output. The default is **N**.

KHD_CSV_OUTPUT_LIST
Optional. The comma or space-separated list of table names (either the short name or the object name found in the ODI or "Disk capacity planning for historical data" table in the agent user’s guide) that is written to the CSV files, for example, `KHD_CSV_OUTPUT_LIST=NTMEMORY,WTLOGCLDSK,NT_System`.

If not supplied, all tables are written to CSV files. This setting is ignored for data tagged for analytical usage. The default is empty and has no maximum length.

This variable is ignored when the `KHD_CSV_OUTPUT_TAGGED_ONLY` variable is enabled.

KHD_CSV_OUTPUT_TAGGED_ONLY
Optional. If specified, only data that has been tagged for analytic use is written to the CSV files. The `KHD_CSV_OUTPUT_LIST` variable is ignored if this variable is enabled. The default is **Y**.

Use this variable in conjunction with the private situation `USE=` parameter in the `<HISTORY>` element . For more information, see “Private situation XML specification” on page 336.

Note: The data tagged for analytical use is *not* inserted into the Tivoli Data Warehouse.

If set to N, then enterprise data and private history (tagged and non-tagged) data are written to the CSV files.

KHD_CSV_MAXSIZE

Optional. The maximum total size in MB of the CSV files.

When this threshold is exceeded, the Warehouse Proxy agent stops writing data to the CSV files at the next evaluation interval. In IBM Tivoli Monitoring V6.3 and later, a warning message is logged in the Warehouse Proxy agent logs and an audit message is displayed. Files must be deleted to resume writing.

This checking cannot be disabled. The default is **10 MB**.

Note: The Warehouse Proxy agent workspaces are *not* updated to reflect these variables.

Monitoring agent and monitoring server environment variables

Use the following environment variables to configure monitoring agents and monitoring servers that you want to export historical data to the Warehouse Proxy agent:

KHD_PURE_EVENT_UPLOAD

Optional. The interval in minutes to wait before historical data from pure events is uploaded to the Warehouse Proxy agent. This interval overrides the default setting of 30 minutes. The minimum value is 1 minute. There is no maximum value. If the historical data is stored at the monitoring agent, you must specify this environment variable in the monitoring agent's environment file. If the historical data is stored at the monitoring server, you must specify this environment variable in the monitoring servers's environment file. Default: **"30"**.

If you specify a low interval, such as 1 minute, best practice is set `KHD_HISTRETENTION=0` to minimize CPU and disk usage.

The Managed System Status > History Exports workspace is updated to reflect the Export Interval Time.

KHD_HISTRETENTION

Optional. Specifies the default retention period in hours for the short-term history files. This value can be used to reduce the amount of data that is kept on disk after a successful upload to the warehouse is performed. Default: **"24"**.

If you configured private history, the value specified for the `RETAIN=` parameter in the `<HISTORY>` element overrides the `KHD_HISTRETENTION` value. If you do not specify a value for the `RETAIN=` parameter, then the `KHD_HISTRETENTION` value is used.

Specify `KHD_HISTRETENTION=0` to delete the short term binary file after the export is complete. Deleting the short term binary file prevents space concerns and also speeds up the process of reading the files. After the short term binary file is deleted, you will not see short-term data in the Tivoli Enterprise Portal. If you set `KHD_HISTRETENTION=0`, then you must also change the variable `KFW_REPORT_TERM_BREAK_POINT` in the portal server environment file to ensure that historical queries in the portal server work correctly.

Tivoli Enterprise Monitoring Agent for analytics

You can flag your monitoring agent historical data so that is can be consumed by an external application, such as an analytic engine.

To use your monitoring agent historical data for analytics complete the following:

- Ensure the Warehouse Proxy agent environment variables required for analytics are set.

  If you want to include only specific attribute data, see "Warehouse Proxy agent for analytics" on page 510.
- Optionally, you can enable enterprise historical data collection and private historical data collection to be flagged for analytics.

  For full descriptions of the elements and parameters, see "Private situation XML specification" on page 336.

Examples of historical data collection for analytics

Examples of historical data collection to CSV files and to the Tivoli Data Warehouse are provided for your reference.

In the examples included in this topic, the Warehouse Proxy agent is installed on a UNIX computer. If your Warehouse Proxy agent is installed on a Windows computer, edit the KHDENV file.

Private history data is collected every 10 minutes and sent to a CSV file every 60 minutes

Warehouse Proxy agent configuration

In the file `hd.ini` set:

```
KHD_CSV_OUTPUT_ACTIVATE=Y
KHD_CSV_OUTPUT=/wpatest2
```

Private situation configuration

In the file <*ITMHOME*>/*localconfig*/lz/lz_situations.xml set:

```
<PRIVATECONFIGURATION>
<HISTORY Table="Linux_Disk" Interval="10" Export="60" USE="A">
    </HISTORY>
</PRIVATECONFIGURATION>
```

Result LNXDSK* files are written to the `/wpatest2` folder every 60 minutes.

Private history data is collected and sent to the CSV file every 1 minute

Warehouse Proxy agent configuration

In the file `hd.ini` set:

```
KHD_CSV_OUTPUT_ACTIVATE=Y
KHD_CSV_OUTPUT=/wpatest2
```

Private situation configuration

In the file <*ITMHOME*>/*localconfig*/lz/lz_situations.xml set:

```
<PRIVATECONFIGURATION>
<HISTORY Table="Linux_Disk" Interval="1" Export="1" USE="A">
    </HISTORY>
<HISTORY Table="Linux_CPU" Interval="1" Export="1" USE="A">
```

```
    </HISTORY>
<HISTORY Table="KLZ_CPU" Interval="1" Export="1" USE="A">
    </HISTORY>
<HISTORY Table="Linux_Process" Interval="1" Export="1" USE="A">
    </HISTORY>
</PRIVATECONFIGURATION>
```

Result LNXDSK* , LNXPROC*, LNXCPU*, and KLZCPU* are written to the /wpatest2 folder every 1 minute.

Private history data is sent to the Tivoli Data Warehouse and some data is sent to the CSV file

Warehouse Proxy agent configuration

In the file hd.ini set:

```
KHD_CSV_OUTPUT_ACTIVATE=Y

KHD_CSV_OUTPUT=/wpatest2

KHD_CSV_OUTPUT_TAGGED_ONLY=N

KHD_CSV_OUTPUT_LIST=LNXPROC
```

Private situation configuration

In the file <*ITMHOME*>/*localconfig*/lz/lz_situations.xml set:

```
<PRIVATECONFIGURATION>
<HISTORY Table="Linux_Disk" Interval="1" Export="15">
    </HISTORY>
<HISTORY Table="Linux_CPU" Interval="1" Export="15" >
  </HISTORY>
<HISTORY Table="KLZ_CPU" Interval="1" Export="15">
  </HISTORY>
<HISTORY Table="Linux_Process" Interval="1" Export="15">
  </HISTORY>
<HISTORY Table="KLZ_Disk" Interval="1" Export="15" >
  </HISTORY>
</PRIVATECONFIGURATION>
```

Result LNXPROC* files are written to the /wpatest2 folder. (If you are using private history collection with the USE="A" parameter, more data is produced.)

```
ls -latr /wpatest2
total 2148
drwxr-xr-x  52 root root 2048 Nov 14 08:40 ..
-rw-r--r--   1 root root 379967 Nov 15 11:41
  LNXPROC_1111115113805000_1111115114405072_00000.csv
-rw-r--r--   1 root root 381393 Nov 15 11:41
  LNXPROC_1111115114405073_1111115115005142_00001.csv
-rw-r--r--   1 root root 123331 Nov 15 11:42
  LNXPROC_1111115115005143_1111115115205154_00002.csv
drwxr-xr-x   2 root root 160256 Nov 15 11:42 .
```

All historical collections data is inserted into the Tivoli Data Warehouse.

Private history data is sent to only the Tivoli Data Warehouse

Warehouse Proxy agent configuration

In the file hd.ini set:

```
KHD_CSV_OUTPUT_ACTIVATE=N
```

Private situation configuration

In the file <*ITMHOME*>/*localconfig*/lz/lz_situations.xml set:

```
<PRIVATECONFIGURATION>
<HISTORY Table="Linux_Disk" Interval="1" Export="15">
  </HISTORY>
```

```
<HISTORY Table="Linux_CPU" Interval="1" Export="15" >
  </HISTORY>
<HISTORY Table="KLZ_CPU" Interval="1" Export="15">
  </HISTORY>
<HISTORY Table="Linux_Process" Interval="1" Export="15">
  </HISTORY>
<HISTORY Table="KLZ_Disk" Interval="1" Export="15" >
  </HISTORY>
</PRIVATECONFIGURATION>
```

Result Data is sent only to the Tivoli Data Warehouse.

Private history data is sent only to the Tivoli Data Warehouse for one agent, and sent only to the CSV file for another agent

Warehouse Proxy agent configuration
In the file `hd.ini` set:

```
KHD_CSV_OUTPUT_ACTIVATE=Y

KHD_CSV_OUTPUT_TAGGED_ONLY=Y
```

Private situation configuration on an agent running on HOST1
In the file <*ITMHOME*>/*localconfig*/`lz/lz_situations.xml` set:

```
<PRIVATECONFIGURATION>
<HISTORY Table="Linux_Disk" Interval="1" Export="15">
  </HISTORY>
<HISTORY Table="Linux_CPU" Interval="1" Export="15" >
  </HISTORY>
<HISTORY Table="KLZ_CPU" Interval="1" Export="15">
  </HISTORY>
<HISTORY Table="Linux_Process" Interval="1" Export="15">
    </HISTORY>
<HISTORY Table="KLZ_Disk" Interval="1" Export="15" >
  </HISTORY>
</PRIVATECONFIGURATION>
```

Private situation configuration on an agent running on HOST2
In the file <*ITMHOME*>/*localconfig*/`lz/lz_situations.xml` set:

```
<PRIVATECONFIGURATION>
<HISTORY Table="Linux_Disk" Interval="1" Export="1" USE="A">
  </HISTORY>
</PRIVATECONFIGURATION>
```

Result Data of agent on HOST1 is sent to the Tivoli Data Warehouse.

Data of agent on HOST2 is sent to the CSV file.

```
-rw-r--r--   1 root root 1570 Nov 16 08:48
  LNXDISK_1111116085832000_1111116085832007_00017.csv
-rw-r--r--   1 root root 1570 Nov 16 08:49
  LNXDISK_1111116085932000_1111116085932007_00018.csv
-rw-r--r--   1 root root 1570 Nov 16 08:50
  LNXDISK_1111116090032000_1111116090032007_00019.csv
```

Enterprise history data is sent to the CSV file and the Tivoli Data Warehouse

Warehouse Proxy agent configuration
In the file `hd.ini` set:

```
KHD_CSV_OUTPUT_ACTIVATE=Y

KHD_CSV_OUTPUT=/wpatest2

KHD_CSV_OUTPUT_TAGGED_ONLY=N
```

Note: The default value for the KHD_CSV_OUTPUT_TAGGED_ONLY variable is Y. If this variable is set to Y, data is written to the CSV files only when data is tagged in private history.

Enterprise historical collection configuration
Create the Historical Enterprise Collection in the Tivoli Enterprise Portal GUI.

Note: You do not need to edit the `lz_situations.xml` file.

Result The CSV files related to the enterprise collection set on the Tivoli Enterprise Portal GUI are created in the `/wpatest2` folder. Data is also sent to the Tivoli Data Warehouse.

Note: If the user also has private history by `USE="A"`, these files are also sent to the CSV file and the Tivoli Data Warehouse.

Enterprise history data is sent only to the Tivoli Data Warehouse

Warehouse Proxy agent configuration
Complete one of the following options:

Do not set any KHD_CSV_* variables in the `hd.ini` file.

OR

In the `hd.ini` file, ensure the `KHD_CSV_OUTPUT_TAGGED_ONLY` variable is set to the default value of `Y`.

Enterprise historical collection configuration
Create the Historical Enterprise Collection in the Tivoli Enterprise Portal GUI.

Note: You do not need to edit the `lz_situations.xml` file.

Result Collection data is sent only to the Tivoli Data Warehouse.

Chapter 17. Tivoli Common Reporting

The Tivoli Common Reporting topics have information that is unique to products that run on the Tivoli Enterprise Portal and use the Tivoli Data Warehouse as the source of historical data for generating reports. This information is intended for the administrator who sets up Tivoli Common Reporting and installs report packages for users.

To get up and running with reporting, complete the following task in the order provided:

1. "Ensure that historical reporting is enabled" on page 523.
2. Create and maintain the dimension tables.
3. Connect to the Tivoli Data Warehouse using the database client over ODBC. You only need to setup this connection once.
4. Import reports by using the report installer.
5. If you installed the OS agent reports, then you can run a prerequisite scan.

Tivoli Common Reporting overview

The Tivoli Common Reporting tool 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.

Tivoli Common Reporting consumers

- The network systems programmer who troubleshoots TCP/IP issues
- The application analyst or documentation manager
- The IT manager or service level advisor who validates service level agreements
- The capacity planner
- The service manager
- The system administrator
- The storage administrator

Tivoli Common Reporting components

Tivoli Common Reporting consists of several components:

- A *data store* for storing and organizing report designs, reports, and supporting resources. The data store is a location within the Tivoli Common Reporting infrastructure where all report-related files and reports are managed and maintained.
- A web-based user interface for specifying report parameters and other report properties, generating formatted reports, and viewing reports.
- A command-line interface for working with objects in the data store and performing additional administrative functions.
- *Report packages*, archive files containing reports, documentation, graphics, and dynamic link libraries.

- A sample set of reports is provided with the Tivoli Common Reporting product. Other sets can be downloaded and installed using the Import facility.
- A CD is available for the Cognos® version of the Tivoli Monitoring agent reports.
- BIRT report packages for some monitoring agents are included as .zip files on the Tivoli Monitoring Agent installation media in the REPORTS/k*pc* directory, where *pc* is the two-character product code. Report packages are available on the IBM Integrated Service Management Library (http://www.ibm.com/software/brandcatalog/ismlibrary) for a number of Tivoli Monitoring products. You can search "Tivoli Common Reporting" to find report packages in the IBM Integrated Service Management Library (http://www.ibm.com/software/brandcatalog/ismlibrary).

You can find additional report packages generated by other non-IBM users, business report templates, and the *Tivoli Common Reporting: Development and Style Guide* at the IBM developerWorks® Tivoli Common Reporting community.

- The open-source Eclipse BIRT Report Designer that you can use to modify reports or create your own. This tool can be downloaded from the IBM developerWorks Tivoli Common Reporting community.

Note: Tivoli Common Reporting provides enhanced security that enables you to assign a security string to hypertext links in a report. The *Tivoli Common Reporting User's Guide* provides instructions for entering a security set.

For more information about Tivoli Common Reporting, including information about installing and administering Tivoli Common Reporting and creating reports, refer to the following information centers:

"Installing Tivoli Common Reporting" topic in the Tivoli Common Reporting Information Center for Tivoli Common Reporting Version 2.1.1

"Installing Tivoli Common Reporting" topic in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/install/tcr_t_install.html) for Tivoli Common Reporting Version 3.1 or higher

Prerequisites for Tivoli Common Reporting

These are the prerequisite components for installing and running Tivoli Common Reporting packages in Tivoli Monitoring products.

To use the reports, you need the following components:

- IBM Tivoli Monitoring
- Tivoli Common Reporting

  IBM Tivoli Monitoring Version 6.3 Fix Pack 2 includes Tivoli Common Reporting Version 3.1.0.1, which is a component of Jazz for Service Management.

  The reports for the IBM Tivoli Monitoring Version 6.3 operating system agents, can be installed with Tivoli Common Reporting Version 3.1 or 2.1.1. (Tivoli Common Reporting Version 2.1.1 is provided with IBM Tivoli Monitoring Version 6.2.3.) For other monitoring agents, see the agent's user guide to determine the supported versions of Tivoli Common Reporting.

  If you want to upgrade an existing version of Tivoli Common Reporting Version 2.x to Version 3.1 or higher, the upgrade is a side-by-side process that results in two instances of Tivoli Common Reporting in your environment. You can remove the earlier version after you finish setting up Tivoli Common Reporting Version 3.1 or higher. If you want to migrate reports, including any custom reports or report scheduling, from the earlier version of Tivoli Common Reporting into the latest version, you must migrate the reporting data before installing the latest monitoring agent reports into Tivoli Common Reporting Version 3.1 or higher.

If you have not done so already, install and configure Tivoli Common Reporting, using the information found in one of the following information centers:

"Installing Tivoli Common Reporting" topic in the Tivoli Common Reporting Information Center for Tivoli Common Reporting Version 2.1.1

"Installing Tivoli Common Reporting" topic in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/install/tcr_t_install.html) for Tivoli Common Reporting Version 3.1 or higher

To ensure that Tivoli Common Reporting is running, go to `http://`*`computer_name`*`:`*`port_number`*`/ibm/console/`. The default port number for http is 16310 and for https is 16311. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.

If you already have an earlier version of Tivoli Common Reporting and you install a newer version in a different directory on the same computer system, the port assignment is different to avoid a conflict.

- Report packages

  Your product might have a separate reports package that must be extracted. See your product user's guide for instructions. This does not apply to the OS Agent reports, which are extracted as part of their installation.

  Note: If you upgraded Tivoli Common Reporting to Version 3.1 or higher and you want to migrate reports, including any custom reports or report scheduling, from the earlier version of Tivoli Common Reporting into the latest version, you must migrate the reporting data before installing the latest monitoring agent reports into Tivoli Common Reporting Version 3.1 or higher. For instructions on how to export report packages from the earlier version of Tivoli Common Reporting and import the packages into the latest version of Tivoli Common Reporting, see the "Installing Jazz for Service Management" topic in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/install/psc_c_install.html).
- Historical data stored in a database manager product supported by IBM Tivoli Monitoring. For more information about supported databases, refer to "Supported databases for the Tivoli Data Warehouse" in the *IBM Tivoli Monitoring Installation and Setup Guide*.
- For the IBM Tivoli Monitoring agent reports Cognos package, refer your monitoring agent user's guide for report details. For more information on the operating system agent reports, see Operating System Agent Documentation http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_osagents.htm. The documentation contains the required views for each report. If these views are not present, the report might not work. In order to ensure that the required views are present, run the following query against the Tivoli Data Warehouse:
  - DB2

    ```
    select distinct "VIEWNAME" from SYSCAT.VIEWS where "VIEWNAME" like '%V'
    ```
  - Oracle

```
select distinct "VIEW_NAME" from USER_VIEWS where "VIEW_NAME" like '%V'
```

- MS SQL Server

  ```
  select distinct "NAME" from SYS.VIEWS where "NAME" like '%V'
  ```

Note: In Tivoli Common Reporting, both BIRT and Cognos report engines can co-exist.

Upgrading from a previous version of monitoring agent reports

You can upgrade your monitoring agent reports from a previous version of a COGNOS report package.

The report packages for the operating system agents that are provided in IBM Tivoli Monitoring Version 6.3, can be installed with Tivoli Common Reporting 3.1 or 2.1.1. Each time an OS agent report package is installed it replaces the previous package if applicable. Best practice is to backup any report data before upgrading. For more information, see "Working with reports" (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/tcr_original/ttcr_working.html).

Upgrading monitoring agent reports and upgrading the monitoring infrastructure are independent tasks. To use some new features and capabilities, your infrastructure might need to be upgraded. An example of this upgrade difference is the ability to use the Managed System Lists for reporting with the OS monitoring agents. To use this feature, you must upgrade your Tivoli Data Warehouse in addition to your monitoring agent reports.

Limitations

The limitations of the reports produced by Tivoli Common Reporting are described in this section.

- The Tivoli Monitoring agent Cognos reports are coded to connect to a data connection in Tivoli Common Reporting with the name `TDW`.
- Tivoli Monitoring agent reports run against the Tivoli Data Warehouse. DB2 limits the length of columns to 30 characters. Because the Tivoli Data Warehouse uses attribute group names as the column headers, attribute names longer than 30 characters in a DB2 Tivoli Data Warehouse are replaced with the internal column name, abbreviated database name for the attribute (for example, CPU_UTIL or DISK_UTIL rather than CPU Utilization or Disk Utilization).
- Reports that cover a long time period or a processing-intensive attribute might cause SQL arithmetic overflow.
- Some of the reports do not support the Tivoli Data Warehouse Summarization and Pruning agent optional definition of shift hours. Customers can use shift hour support to flag collected data as being either Peak or Off-Peak periods. However, some reports include all data collected between the customer-selected report start and end times, whether that data was collected during Peak or Off-Peak periods. See "About the Summarization and Pruning agent" on page 481 and "Changing global configuration settings" on page 490.
- If the Summarization and Pruning agent shift hours configuration is changed, the most recent configuration is used for reporting. If you specify a date range for an availability report that crosses multiple configurations, the availability metrics might be incorrect. For example, if you edited the peak hours to add one hour, the summarization for peak hours and off-peak hours are different before

and after the time when the agent was reconfigured. Only time ranges before or after the reconfiguration are valid and you should avoid specifying a time range that crosses two configurations.

Ensure that historical reporting is enabled

The first step in preparing to run reports is to ensure that historical reporting is enabled.

About this task

Reports run against long-term historical data that is stored in the Tivoli Data Warehouse. Before you can run reports, ensure that you have installed the required components and configured historical data collection:

Procedure

1. Install and configure the Tivoli Data Warehouse and warehouse agents: Warehouse Proxy agent and Summarization and Pruning agent, see the *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm).
2. Set up historical collection using the Historical Collection Configuration feature in the Tivoli Enterprise Portal, see Historical collection configuration.

   For z/OS-based monitoring agents, configure the persistent data store using the Configuration Tool, see “Configure the persistent data store” in the *Configuring the Tivoli Enterprise Monitoring Server on z/OS* and also refer to the *IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/zcommonconfig/zcommonconfig.htm).
3. Optionally, enable access to summarized data in the Tivoli Data Warehouse. The use of summarized data in reports can simplify analysis of displayed reports and improve the performance of generating the reports.

What to do next

After starting the Tivoli Data Warehouse, the warehouse agents, and data collection, allow enough time for the Tivoli Data Warehouse to save historical data for your requested report time period or the appropriate amount of data for a summarized report. For example, if you want a monthly report, you need at least a month's worth of stored data.

Creating and maintaining the dimension tables

Preparing the Tivoli Data Warehouse for Tivoli Common Reporting includes creating the dimension tables, which are required for running the Cognos reports and using the data models.

There are two groups of dimension tables:

Time dimension tables

The time dimension tables, TIME_DIMENSION, MONTH_LOOKUP, and WEEKDAY_LOOKUP, are required by Tivoli Common Reporting. The time dimension tables belong to the IBM_TRAM schema. TRAM stands for Tivoli Reporting and Analytics Model, and is the common data model used by Tivoli products.

The OS Agents Report Prerequisites Scanner report refers to these tables as the Tivoli Common Reporting Shared Dimensions.

The following descriptions apply:

IBM_TRAM Schema
TRAM stands for Tivoli Reporting and Analytics Model, is the common data model used by Tivoli products.

Warning: When your site changes either the IBM Tivoli Monitoring data model or the operating system agent reports, your updated data model and reports are no longer supported. In addition, these updates can be lost when you apply maintenance to Tivoli Monitoring or move to a subsequent release.

TIME_DIMENSION table
Includes years of time dimensional data and granularity to a specified number of minutes. Each row of this table is a unique minute key with various dimensions related to it, such as hour, weekday, day of month, and quarter.

MONTH_LOOKUP table
Globalizes the month names for Time Dimension.

WEEKDAY_LOOKUP table
Globalizes the weekday names for Time Dimension.

Resource dimension tables
The resource dimension tables are MANAGEDSYSTEM, MANAGEDSYSTEMLIST, and MANAGEDSYSTEMLISTMEMBERS.

The OS Agents Report Prerequisites Scanner report refers to these tables as the IBM Tivoli Monitoring Shared Dimensions.

The following descriptions apply:

MANAGEDSYSTEMLIST table
Contains the name, product, and description (which is blank by default and can be customized) for the managed system group on the monitoring server.

MANAGEDSYSTEMLISTMEMBERS table
Contains the managed systems that are a member of a given managed system group as defined in the MANAGEDSYSTEMLIST table.

Attention: If you are running your Summarization and Pruning agent in autonomous mode (`KSY_AUTONOMOUS=Y`), the MANAGEDSYSTEMLIST and MANAGEDSYSTEMLISTMEMBERS tables cannot be maintained or created by the Summarization and Pruning agent.

There are three ways to create the dimension tables:

- "Configuring the Summarization and Pruning agent to create and maintain the dimension tables" on page 525

  In IBM Tivoli Monitoring V6.3 or later, you can create your tables by using the Summarization and Pruning agent. Following the steps in this topic also configures the Summarization and Pruning agent to automatically maintain the dimension tables.
- "Creating the dimension tables using the Schema Publication Tool" on page 528

You can execute the SQL scripts created by the Schema Publication Tool to create your dimension tables. If you want your dimension tables to be maintained automatically, you must configure the Summarization and Pruning agent to do so after you have used the SQL scripts to create the dimension tables.

- "Manually creating and maintaining the dimension tables" on page 531

  Prior to IBM Tivoli Monitoring V6.3, you had to manually create and maintain the dimension tables. If you want to continue to manually update the dimension tables, this option is still available.

Configuring historical data collection before you begin

You must first configure historical data collection.

In order to build the resource dimension table, configure historical data collection for one or more of the following attribute groups, depending on the operating system you are getting reports for:

Type	Attribute group	Table	Summarize
Linux	Linux IP Address	Linux_IP_Address	Daily
UNIX	UNIX IP Address	UNIX_IP_Address	Daily
Windows	Computer Information	NT_Computer_Information	Daily
Warehouse Summarization and Pruning Agent	KSY Summarization Config	KSY_Summarization_Config_DV	Daily
IBM i	Miscellaneous	i5OS_Miscellaneous	Daily

You can configure historical data collection in the Tivoli Enterprise Portal or in the Command Line Interface. The following example shows how a local historical collection for NT Computer Information was created and distributed from the CLI:

```
tacmd login -s MyComputer -u MyUser -p MyPassword
tacmd tepslogin -s localhost -u sysadmin
tacmd histconfiguregroups -t knt -o "NT Computer Information" -m -d YQMWDH
  -p Y=2y,Q=2y,M=1y,W=1y,D=6m,H=14d,R=7d
tacmd histcreatecollection -t knt -o "NT Computer Information"
  -a "ComputerInformation" -c 15m -i 15m -l TEMA -e
 "Needed for resource dimension table for TCR."
tacmd histstartcollection -n TEMS_NAME -t "knt" -o "NT Computer Information"
```

Warning: When your site changes either the IBM Tivoli Monitoring data model or the operating system agent reports, your updated data model and reports are no longer supported. In addition, these updates might be lost when you apply maintenance to Tivoli Monitoring or move to a subsequent release.

Configuring the Summarization and Pruning agent to create and maintain the dimension tables

Configure the Summarization and Pruning agent to create and maintain the dimension tables required by Tivoli Common Reporting and IBM Tivoli Monitoring. The time dimension tables, TIME_DIMENSION, MONTH_LOOKUP, and WEEKDAY_LOOKUP, and the resource dimension tables, MANAGEDSYSTEM, MANAGEDSYSTEMLIST, MANAGEDSYSTEMLISTMEMBERS, are all created and maintained using this procedure.

Before you begin

- You must have IBM Tivoli Monitoring V6.3 Fix Pack 2 or later.
- Ensure that any historical collections and summarization needed for the MANAGEDSYSTEM table are enabled for each agent. For more information, see "Configuring historical data collection before you begin" on page 525 and your agent documentation.
- During this procedure the MANAGEDSYSTEM table is populated using information from the WAREHOUSETCRCONTROL table. The WAREHOUSETCRCONTROL table is created at the first successful start of the Summarization and Pruning agent or by running the scripts created when using the schema publication tool to generate the DDL.

  It is the responsibility of each monitoring agent to put entries into the WAREHOUSETCRCONTROL table. Your monitoring agent might use scripts or manual steps to add their entries. For detailed information see your agent's user's guide.

  For example, the OS agents reports package populates the WAREHOUSETCRCONTROL table during an installation step. This means if you want to use Tivoli Common Reporting with your OS agent reports, you must install the OS agents reports package before completing this task to ensure the required WAREHOUSETCRCONTROL entries are picked up. Note: You do not need to install the OS agent report package to enable historical collections.
- For DB2 LUW and DB2 z/OS, you must create a user in the Tivoli Data Warehouse database that has privileges to create tables and execute the stored procedure with the IBM_TRAM schema. The user must have SYSADM or DBADM authority to create an IBM_TRAM schema.
- For Oracle, you must create the IBM_TRAM user by using the IBM_TRAM schema in the Tivoli Data Warehouse database, or you can create a user that has privileges to create tables and execute the stored procedure with the IBM_TRAM schema.
- For Microsoft SQL Server, you must create a user in the Tivoli Data Warehouse database that has privileges to create tables and execute the stored procedure with the IBM_TRAM schema.

About this task

This task configures the Summarization and Pruning agent to create and maintain the dimension tables. If the tables already exist, the Summarization and Pruning agent ensures the tables contain all the needed columns and then adds any missing columns. If a table already exists, but the index does not exist, the Summarization and Pruning agent will not create the missing index. If you have already customized your table data, your customized data is preserved.

Procedure

1. Edit the Summarization and Pruning agent environment variable file.

   Windows

   On the computer where the Summarization and Pruning agent is installed, in the Manage Tivoli Enterprise Monitoring Services application, right-click the agent and select **Advanced → Edit ENV file**.

   Linux UNIX

   On the computer where the Summarization and Pruning agent is installed, change to the *install_dir*/`config` directory.

   Open the `sy.ini` file in a text editor.

Configure the following environment variables, then save the file:

- `KSY_TRAM_ENABLE=Y`

  Indicates whether reporting integration is enabled. If enabled, the time dimension tables are created if they do not already exist. The default value is N.

  If this variable is set to Y, and the time dimension tables do not exist, but the KSY_TRAM_USER and KSY_TRAM_PASSWORD are not set, then a warning message in the trace file and in the Summarization and Pruning agent **Configuration** workspace is returned indicating that the time dimension tables were not created.
- `KSY_TRAM_USER=`*`username`*

  The reporting user that creates the time dimension tables. For all databases except DB2 LUW and DB2 z/OS the reporting user is set by default to IBM_TRAM.

  For DB2 LUW and DB2 z/OS, because IBM_TRAM does not follow the user naming conventions for these databases, a different user name must be used. (In DB2 LUW a user name cannot start with IBM and in DB2 z/OS a user name cannot contain an underscore.) The user you create must have the necessary privileges to create an IBM_TRAM schema.

  This variable is required only if the time dimension objects need to be created in the database.
- `KSY_TRAM_PASSWORD=`*`password`*

  The password for the reporting user.

  This variable is required only if the time dimension objects need to be created in the database.
- `KSY_TRAM_TD_GRANULARITY=`*`minutes`*

  The number of minutes that the data is inserted into the TIME_DIMENSION table. The minimum value is 1. The default value is 5.

  If the value specified is below the minimum value, the value is reset by the Summarization and Pruning agent at start up to the minimum value.
- `KSY_TRAM_TD_INITIAL_LOAD=`*`months`*

  The amount of data in months to be loaded into the TIME_DIMENSION table when it is empty or first created. The minimum value is 1. The default value is 24 months.

  If the value specified is below the minimum value, the value is reset by the Summarization and Pruning agent at start up to the minimum value.

2. Reconfigure the Summarization and Pruning agent to enable reporting integration. You can use one of the following options:
   - Reconfigure the Summarization and Pruning agent through Manage Tivoli Enterprise Monitoring Services enabling reporting integration.

     For detailed steps, see the *IBM Tivoli Monitoring Installation and Setup Guide*.
   - Reconfigure the Summarization and Pruning agent using the command-line interface enabling reporting integration.

     For detailed steps, see the *IBM Tivoli Monitoring Installation and Setup Guide*.
   - Reconfigure the Summarization and Pruning agent using remote deploy enabling reporting integration.

     For detailed steps, see the *IBM Tivoli Warehouse Summarization and Pruning Agent Reference*.
3. Restart the Summarization and Pruning agent.

Results

The dimension tables have been created and are now maintained by the Summarization and Pruning agent.

The Summarization and Pruning agent only adds new systems to the MANAGEDSYSTEM table. This means if you have customized data, it is preserved.

What to do next

Verify the contents of the tables and ensure they have data for the upcoming month. The Summarization and Pruning agent maintains data one month ahead of the current month.

You can view the Summarization and Pruning agent **Configuration** workspace for an overview of your configurations. Any errors encountered are also reported in this workspace. For detailed information about this workspace, see the *IBM Tivoli Warehouse Summarization and Pruning Agent Reference*.

You can create reports and queries based on your managed system groups.

Creating the dimension tables using the Schema Publication Tool

Use the schema publication tool to create the dimension tables required by Tivoli Common Reporting and IBM Tivoli Monitoring.

The following time dimension tables are created by this task: TIME_DIMENSION, MONTH_LOOKUP, and WEEKDAY_LOOKUP. The following resource dimension tables are also created by this task: MANAGEDSYSTEM, MANAGEDSYSTEMLIST, MANAGEDSYSTEMLISTMEMBERS.

Before you begin

- You must have IBM Tivoli Monitoring V6.3 Fix Pack 2 or later.
- Ensure that any historical collections and summarization needed for the MANAGEDSYSTEM table are enabled for each agent. For more information, see “Configuring historical data collection before you begin” on page 525 and your agent documentation.
- Complete this task before any reports are executed.
- For DB2 LUW and DB2 z/OS, you must create a user in the Tivoli Data Warehouse database that has privileges to create tables and execute the stored procedure with the IBM_TRAM schema. The user must have SYSADM or DBADM authority to create an IBM_TRAM schema.
- For Oracle, you must create the IBM_TRAM user by using the IBM_TRAM schema in the Tivoli Data Warehouse database, or you can create a user that has privileges to create tables and execute the stored procedure with the IBM_TRAM schema.
- For Microsoft SQL Server, you must create a user in the Tivoli Data Warehouse database that has privileges to create tables and execute the stored procedure with the IBM_TRAM schema.

About this task

You can use this task to create both the time dimension tables and the resource dimension tables, or you can create just the time dimension tables, or just the resource dimension tables. Each set of tables can be created using a different connection to the Tivoli Data Warehouse.

The following schema publication tool modes are supported:

- Installed: Generates the DDL for all the monitoring agents installed in your environment. This mode also generates the DDL for the resource dimension tables and the DDL for the time dimension tables if KSY_TRAM_ENABLE is set to Y. This mode also provides the insert statement into the WAREHOUSEID table and the create statements for the TDW functions.
- Configured: Generates the DDL only for the attribute groups that have historical collections configured. The DDL for the summarized tables only exists if the summarization for those attribute groups is configured. This mode also generates the DDL for the resource dimension tables and the DDL for the time dimension tables if KSY_TRAM_ENABLE is set to Y. This mode also provides the insert statement into the WAREHOUSEID table and the create statements for the TDW functions.
- Updated: This mode is the best practice. Generates the DDL only for the attribute groups that have historical collections configured, but that do not exists in the Tivoli Data Warehouse database. If the DDLs do not already exist, this mode also generates the DDL for the resource dimension tables and the DDL for the time dimension tables if KSY_TRAM_ENABLE is set to Y. This mode also provides the insert statement into the WAREHOUSEID table and the create statements for the TDW functions.

For additional information on running the schema publication tool see “Generating SQL for data warehouse tables” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Attention: The procedure below uses the **updated mode**.

Procedure

1. If you want to create the resource dimension tables used by the warehouse, you must edit the response file.

   Windows

   install_dir\TMAITM6\tdwschema.rsp for 32-bit Windows systems

   install_dir\TMAITM6_x64\tdwschema.rsp for 64-bit Windows systems

   Linux UNIX

   install_dir/*arch*/bin/tdwschema.rsp

   Configure the following environment variables:

   ```
   KSY_PRODUCT_SELECT = updated
   KSY_SQL_OUTPUT_FILE_PATH = optional file path for SQL output
   ```

   Save and close the file.
2. If you want to create the time dimension tables used for reporting, you must edit the Summarization and Pruning agent environment variable file and set `KSY_TRAM_ENABLE=Y`.

   Windows

   On the computer where the Summarization and Pruning agent is installed, in the Manage Tivoli Enterprise Monitoring Services application, right-click the agent and select **Advanced** → **Edit ENV file**.

Linux UNIX

On the computer where the Summarization and Pruning agent is installed, change to the `*install_dir*/config` directory.

Open the `sy.ini` file in a text editor.

Save and close the file.

Note: Scripts to create the time dimension tables will not be generated if KSY_TRAM_ENABLE=Y and you also use the KSY_TABLE_FILTER variable in the tdwschema.rsp file but do not include the time dimension tables and resource dimension tables in the list of tables to filter by.

3. Ensure the Tivoli Enterprise Portal Server is started.
4. Run the schema publication tool script:

   Windows

   install_dir`\TMAITM6\tdwschema.bat` for 32-bit Windows systems

   install_dir`\TMAITM6_x64\tdwschema.bat` for 64-bit Windows systems

   Linux UNIX

   install_dir/*arch*`/bin/tdwschema.sh`

   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. Connect to the Tivoli Data Warehouse with the authorized user ID and execute the following scripts in the order listed.

   To create the resource dimension tables and any other attribute group tables, execute the following SQL scripts in the order listed with the Tivoli Data Warehouse user ID:

   ```
   tdw_schema_table.sql
   tdw_schema_index.sql
   tdw_schema_view.sql
   tdw_schema_function.sql
   tdw_schema_insert.sql
   ```

   To create the time dimension tables execute the following SQL scripts in the order that is listed while connected to the Tivoli Data Warehouse database with the user ID authorized to create tables with the IBM_TRAM schema:

   Note:

   If using Oracle or Microsoft SQL Server, run the scripts as the TRAM user IBM_TRAM.

   If using DB2, run the scripts as a user that has privileges to create tables with the IBM_TRAM schema.

   ```
   tdw_tram_table.sql
   tdw_tram_index.sql
   tdw_tram_function.sql
   tdw_tram_insert.sql
   ```

Attention: You must run the scripts in the order that is listed or they will fail. The tdw_tram_insert.sql calls the stored procedure in the tdw_tram_function.sql script to populate the IBM_TRAM.TIME_DIMENSION table after the tdw_tram_table.sql script is executed.

Example on DB2:

```
db2 connect to <TDW_database> user <TDW_user> using <password>
db2 -tvf tdw_schema_table.sql
db2 -tvf tdw_schema_index.sql
db2 -tvf tdw_schema_view.sql
db2 -td# -f tdw_schema_function.sql
db2 -tvf tdw_schema_insert.sql

db2 connect to <TDW_database> user <TRAM_user> using <TRAM_password>
db2 -tvf tdw_tram_table.sql
db2 -tvf tdw_tram_index.sql
db2 -td# -f tdw_tram_function.sql
db2 -tvf tdw_tram_insert.sql
```

Example on MSSQL:

```
osql -S <SQL server> -U <user ID> -P <password> -d <database name>
 -I -i tdw_schema_table.sql
osql -S <SQL server> -U <user ID> -P <password> -d <database name>
 -I -i tdw_schema_index.sql
osql -S <SQL server> -U <user ID> -P <password> -d <database name>
 -I -i tdw_schema_view.sql
osql -S <SQL server> -U <user ID> -P <password> -d <database name>
 -I -i tdw_schema_function.sql
osql -S <SQL server> -U <user ID> -P <password> -d <database name>
 -I -i tdw_schema_insert.sql

osql -S <SQL server> -U <IBM_TRAM> -P <IBM_TRAM_password> -d <database name>
 -I -i tdw_tram_table.sql
osql -S <SQL server> -U <IBM_TRAM> -P <IBM_TRAM_password> -d <database name>
 -I -i tdw_tram_index.sql
osql -S <SQL server> -U <IBM_TRAM> -P <IBM_TRAM_password> -d <database name>
 -I -i tdw_tram_function.sql
osql -S <SQL server> -U <IBM_TRAM> -P <IBM_TRAM_password> -d <database name>
 -I -i tdw_tram_insert.sql
```

Example on Oracle:

```
sqlplus <user ID>/<password>@<TDW Database SID> @tdw_schema_table.sql
sqlplus <user ID>/<password>@<TDW Database SID> @tdw_schema_index.sql
sqlplus <user ID>/<password>@<TDW Database SID> @tdw_schema_function.sql
sqlplus <user ID>/<password>@<TDW Database SID> @tdw_schema_insert.sql

sqlplus <IBM_TRAM>/<IBM_TRAM_password>@<TDW Database SID> @tdw_tram_table.sql
sqlplus <IBM_TRAM>/<IBM_TRAM_password>@<TDW Database SID> @tdw_tram_index.sql
sqlplus <IBM_TRAM>/<IBM_TRAM_password>@<TDW Database SID> @tdw_tram_function.sql
sqlplus <IBM_TRAM>/<IBM_TRAM_password>@<TDW Database SID> @tdw_tram_insert.sql
```

Results

The dimension tables are now created.

What to do next

If you received errors while executing the `tdw_schema_insert.sql` script, see the *IBM Tivoli Monitoring Troubleshooting Guide*.

Manually creating and maintaining the dimension tables

You can create the dimension tables by creating and maintaining them manually using database scripts.

Creating and populating the time dimensions tables

Preparing the Tivoli Data Warehouse for Tivoli Common Reporting includes creating the IBM_TRAM dimensions, which are required for running the Cognos reports and using the data models.

About this task

The following dimensions tables are created by this procedure:

- IBM_TRAM Schema
- TIME_DIMENSION table
- MONTH_LOOKUP table
- WEEKDAY_LOOKUP table
- Other dimensions

  Other dimensions conforming to the Tivoli Common Data Model, such as ComputerSystem, BusinessService, and SiteInfo.

You will need the database scripts included in the extracted reports package under the `db_scripts` directory.

If reports are distributed with an installer, the following manual procedures can be handled automatically by the report installer. See your agent-specific user's guide for information on automated TRAM creation.

When installing multiple report packages, the following steps need only be completed once. When installing multiple report packages the same TIME_DIMENSION tables are used. If you want to reset granularity or begin and end times, you can repeat the procedure.

Procedure

- **IBM DB2**
  1. Copy the database scripts (.db2 files) from the reports package to a location where they can be run against the Tivoli Data Warehouse. The scripts are in the db_scripts branch of the directory where the reports package was extracted to.
  2. Log in as db2admin. Your user ID must have administrator access to create the IBM_TRAM schema.
  3. Connect to the database that you want to create the dimension tables for. This is your Tivoli Data Warehouse.

     ```
     db2 connect to WAREHOUS
     ```
  4. If you have an older version of the database scripts already installed clean up the database:

     ```
     db2 -tf clean.db2
     ```
  5. Create the schema and tables:

     ```
     db2 -tf create_schema_IBM_TRAM.db2
     ```

     After the command completes successfully, several tables are shown under IBM_TRAM: TIME_DIMENSION, MONTH_LOOKUP, WEEKDAY_LOOKUP, ComputerSystem, BusinessService, SiteInfo, and so on.
  6. Create the stored procedure for generating the time dimension:

     ```
     db2 -td@ -vf gen_time_dim_granularity_min.db2
     ```

7. To populate TIME_DIMENSION table, call the time dimension stored procedure with dates and granularity to generate the timestamps. You can generate up to five years at a time or have the data regenerated every day.

   ```
   db2 "call IBM_TRAM.CREATE_TIME_DIMENSION('start_date',
   'end_date', granularity_of_data)"
   ```

   where `start_date` and `end_date` are in this format YYYY-MM-DD-HH.MM.SS.MILSEC and `granularity_of_data` is the frequency in minutes. For example, the following command extracts data from 1/1/2010 to 1/1/2015 with 60-minute granularity.

   ```
   db2 "call IBM_TRAM.CREATE_TIME_DIMENSION('2010-01-01-00.00.00.000000',
   '2015-01-01-00.00.00.000000', 60)"
   ```

   Tip: When populating the time dimension use the following guidelines:

   - To view yearly data, you must provide the first day of the year, as seen in the preceding example.
   - Specify the end date far enough into the future so that new incoming data can map to and be displayed correctly in the reports.
   - Best practice is to use a value of 60-minute granularity.

- **Microsoft SQL Server**
  1. Copy the database scripts (.sql files) from the reports package to a location where they can be run against the Tivoli Data Warehouse. The scripts are in the db_scripts branch of the directory where the reports package was extracted.
  2. Customize the provided scripts by changing the default database name in the use statement (replace USE IBM_TRAM) if it is different from the default. If the name of your Tivoli Data Warehouse is "warehouse," the statement is `USE warehouse`:
     a. If you have an older version of the database scripts already installed, clean up the database using the **clean.sql** command..
     b. Run the **createSchema.sql** command.
     c. Run the **createProcedure.sql** command.
     d. Run the **populateTimeDimension.sql** command. Also, modify the boundary parameters for the time dimension and granularity, for example,

     ```
     @startDate    = '2010-01-01 00:00:00',
     @endDate      = '2012-12-31 00:00:00',
     @granularity  = 60,
     ```

     If Monday must be the first day of the week, add the fourth parameter equal to 1; otherwise, release three parameters.

     ```
     @weekday      = 7
     ```

  3. If you have an older version of the database scripts already installed, clean up the database.

     ```
     sqlcmd -i clean.sql [-U username -P password] [-S hostname]
     ```

  4. Run the scripts at the MS SQL command line in this order:

     ```
     sqlcmd -i createSchema.sql [-U username -P password] [-S host]
     ```

     ```
     sqlcmd -i createProcedure.sql [-U username -P password] [-S host]
     ```

     ```
     sqlcmd -i populateTimeDimension.sql [-U username -P password] [-S host]
     ```

- **Oracle** manual installation

1. Copy the database scripts (.sql files) from the reports package to a location where they can be run against the Tivoli Data Warehouse. The scripts are in the db_scripts branch of the directory where the reports package was extracted to.
2. Start an SQL *Plus session if it is not already running.
3. Check that you can access remotely as sys user.
4. If you have an older version of the database scripts already installed clean up the database (the procedure must be called by the sys user):

   ```
   clean.sql
   ```
5. Take one of the following steps:
   - If you can access remotely as sys user, run this command and provide all the information that the script requires:

     ```
     @MY_PATH\setup_IBM_TRAM.sql
     ```
   - If you cannot access remotely as sys user, run this command locally at the Oracle server and provide all the information that the script requires:

     ```
     @MY_PATH\local_setup_IBM_TRAM.sql
     ```

- **Oracle** batch installation
  1. Copy the database scripts (.sql files) from the reports package to a location where they can be run against the Tivoli Data Warehouse. The scripts are in the db_scripts branch of the directory where the reports package was extracted to.
  2. Start an SQL *Plus session if it is not already running.
  3. If you have an older version of the database scripts already installed clean up the database (the procedure must be called by the sys user):

     ```
     clean.sql
     ```
  4. Create user IBM_TRAM (the script must be called by a user with system rights, such as SYS/SYSTEM):

     ```
     @MY_PATH\create_IBM_TRAM.sql TCR_PASS USER_TBSPC TEMPORARY_TBSPC
     ```

     where TCR_PASS is the password for the new user, *USER_TBSPC* is the default user tablespaces name (must exist), and *TEMPORARY_TBSPC* is the default temporary tablespaces name (must exist)
  5. Create the IBM_TRAM tables (the script must be called by the IBM_TRAM user created in the previous step):

     ```
     @MY_PATH\create_schema.sql USER_TBSPC
     ```

     where *USER_TBSPC* is the default user tablespaces name (must exist)
  6. Grant privileges to the user, such as ITMUSER (the script must be called by the IBM_TRAM user):

     ```
     @MY_PATH\grant_IBM_TRAM.sql USER
     ```

     where *USER* is the name of the user to grant privileges to.
  7. Create the procedure (the script must be called by the IBM_TRAM user):

     ```
     @MY_PATH\gen_time_dim_granularity_hr.sql
     ```
  8. Load the lookup data (the script must be called by the IBM_TRAM user):

     ```
     @MY_PATH\populateLookup.sql
     ```
  9. Generate the time dimension (the procedure must be called by the IBM_TRAM user):

     ```
     @MY_PATH\populateTimeDimension.sql StartDate EndDate Granularity
     ```

where `StartDate` is the start date in the format 'yyyy-mm-dd HH:MM', *`EndDate`* is the end date in the format 'yyyy-mm-dd HH:MM', and *`Granularity`* is the number of minutes. Example:

```
@MY_PATH\populateTimeDimension.sql '2010-01-01 00:00' '2012-
          12-31 00:00' '60'
```

Results

The time dimension tables are complete.

Troubleshooting

If the DB2 commands are failing with the following error:

```
UDA-SQL-0107 A general exception has occurred during the operation
"prepare".[IBM][CLI Driver][DB2/NT64] SQL0551N "ITMUSER" does not have
the required authorization or privilege to perform operation "SELECT" on
object "IBM_TRAM......
```

Execute the following commands to resolve the issue:

1. Connect to the Tivoli Data Warehouse as a user with DB2 privileges.
2. Issue the following grants:

   Grant select on `IBM_TRAM."ComputerSystem"` to `ITMUSER`

   Grant select on `IBM_TRAM.MONTH _LOOKUP` to `ITMUSER`

   Grant select on `IBM_TRAM.TIMEZONE_ DIMENSION` to `ITMUSER`

   Grant select on `IBM_TRAM.TIME_DIMENSION` to `ITMUSER`

   Grant select on `IBM_TRAM.WEEKDAY_LOOKUP` to `ITMUSER`

   Grant execute on procedure `IBM_TRAM.CREATE_TIME_DIMENSION` to `ITMUSER`

What to do next

Create and populate the resource dimension table.

Creating and populating the resource dimension table

Preparing the Tivoli Data Warehouse for Tivoli Common Reporting includes creating and populating the resource dimension table “ManagedSystem”, which is required for running the Cognos reports and using the data models.

Before you begin

You must first configure historical data collection. For more information, see “Configuring historical data collection before you begin” on page 525.

About this task

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.

Important: The following scripts use hardcoded user schemas. If you use a different schema, you must replace every instance of the hardcoded schema with the user you specified.

Procedure

- **IBM DB2**

1. Log in as **db2admin**. Your user ID must have administrator access to create the resource dimension.
2. Connect to the database that you want to create the resource dimension table for. This is your Tivoli Data Warehouse.

   ```
   db2 connect to WAREHOUS
   ```
3. If you specified a different user from the default of ITMUSER for connecting to your warehouse, customize the provided scripts `gen_resources.db2`, `populate_resources.db2`, `populate_resources_zos.db2`, replacing every instance of the hardcoded schema "ITMUSER" with the user you specified.
4. Create the tables:

   ```
   db2 -tf gen_resources.db2
   ```

   After the command completes successfully, a new table is shown under the ITMUSER schema: ManagedSystem.
   Note: If the table "ITMUSER.ManagedSystem" has already been created the following message is displayed when executing the `gen_resources.db2` script and can be ignored: `DB21034E  The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned: SQL0601N  The name of the object to be created is identical to the existing name "ITMUSER.MANAGEDSYSTEM" of type "TABLE". SQLSTATE=42710"`
5. Create the stored procedure to populate the ManagedSystem table:

   ```
   db2 -td@ -vf populate_resources.db2
   ```
6. To populate ManagedSystem table, call the stored procedure:

   ```
   db2 "call ITMUSER.POPULATE_OSAGENTS()"
   ```

   Note: If you specified a different user from the default, replace `ITMUSER` with the user specified during your warehouse configuration.

- **Microsoft SQL Server**
  1. Customize the provided scripts:
     a. In **create_table.sql**, change the default database name in the use statement (replace USE WAREHOUS) if it is different from the default.
     b. In **create_procedure.sql**, change the default database name in the use statement (replace USE WAREHOUS) if it is different from the default.
     c. In **populate_agents.sql**, change the default database name in the use statement (replace USE WAREHOUS) if it is different from the default.
  2. Run the scripts at the MS SQL command line in this order:

     ```
     sqlcmd -i create_table.sql [-U myusername -P mypassword] [-H myhost])
     sqlcmd -i create_procedure.sql [-U myusername -P mypassword] [-H myhost])
     sqlcmd -i populate_agents.sql [-U myusername -P mypassword] [-H my_host])
     ```
- **Oracle** manual installation
  1. Start a SQL *Plus session if it is not already running.
  2. Run this command (path with no spaces) and provide all the information that the script requires:

     ```
     @MY_PATH\setup_populate_agents.sql
     ```
- **Oracle** batch installation
  1. Start a SQL *Plus session if it is not already running.
  2. Create the ITMUSER.ManagedSystem table. The script must be called by the Tivoli Data Warehouse user, which is ITMUSER by default. If you used a different user name, modify the script for the correct name.

```
@MY_PATH\create_table.sql USER_TBSPC
```

3. Create the procedure to populate the table:

```
@MY_PATH\create_procedure.sql
```

4. Start the procedure to populate the ManagedSystem table:

```
begin
POPULATE_OSAGENTS('ITMUSER');
end;
/
```

Results

The resource dimension table is complete.

What to do next

Install and run IBM Cognos reports.

Importing reports by using the report installer

The report installer is available only for specific agents that include reports that are bundled with the installer. This method of installation can only be used when available.

The report installer supports Tivoli Common Reporting versions 2.1 and later.

Before you begin

Set the PATH environment variable to include the bin directory of Java Runtime Environment (JRE) 1.5 or later before using the report installer.

See your agent-specific user's guide for additional installation steps and possible troubleshooting items for your agent.

About this task

Use the following procedure to import reports that are bundled by using the report installer. This procedure can be used on the 6.2.3 or later versions of the OS agents and the TivoliPerformance Analyzer reports.

For additional Tivoli Common Reporting information, see the “Installing Tivoli Common Reporting” topic in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/install/tcr_t_install.html).

Procedure

- Using the GUI:
  1. From the `\osreports` directory on the product CD, run the command appropriate for your operating system:

     Windows `setup_platform.exe`

     Linux UNIX `setup_platform.bin`

     The installer window opens.
  2. Choose your language and click **OK**.
  3. On the Welcome page click **Next**.

4. Specify the Tivoli Common Reporting installation directory, such as `<JazzSM_Home>/reporting` if you are using Tivoli Common Reporting 3.1 or later with Jazz for Service Management and `<TIP_Home>/TCRComponents` if you are using Tivoli Common Reporting 2.1.1 with Tivoli Integrated Portal. You can use the default folder or use the **Choose** button to provide the path. Click **Next**.
5. Choose the reports you want to install. Click **Next**.
6. Enter the Tivoli Common Reporting user name and password. Click **Next**.
7. For *each* data source and data script, enter the data source configuration information:
   - If applicable, on the **Cognos Data Sources** configuration window, enter the Tivoli Data Warehouse database user name, password, database type, and database name. For DB2 or Oracle, enter the database name. For SQL, enter the ODBC Data Source Name.
   - If applicable, on the **BIRT Data Source** window and the **Configure Data Script** window enter the configuration information:

     In the **JDBC User Credentials** tab, enter the Tivoli Data Warehouse database user name and password to be used during the installation. Check the box to skip defining the data source now.

     In the **JDBC Database Credentials** tab, choose the database type from the list; then, enter the database JDBC URL and specify the JDBC driver files. Use the **Browse** button to search for the JAR files, or type in the file names separated by a semicolon (;). Enter the driver JDBC class. Check the box to skip defining the data source now.

     Note: More than one JAR file might be required for a data source. For JDBC, given driver files are automatically copied into a Tivoli Common Reporting directory from which they are leveraged in creating database connections to collect data for reports.

     Click **Next**.
8. The pre-installation summary is displayed. Read it carefully to check if the information is correct. If it is, click **Install** or use the **Previous** button to change any of the previously specified parameters.
9. A window is displayed that shows the progress of your installation.
10. The post-installation report is displayed. Check if the installation was successfully finished and click **Finish**.

- Using the command line:
  1. Run the `setup_<platform>.exe/.bin -i console` command.
  2. Choose your installation language.
  3. Enter the location of the Tivoli Common Reporting directory such as `<JazzSM_Home>/reporting` if you are using Tivoli Common Reporting 3.1 or later with Jazz for Service Management and `<TIP_Home>/TCRComponents` if you are using Tivoli Common Reporting 2.1.1 with Tivoli Integrated Portal.
  4. Choose the type of reports to be installed.
  5. Enter your Tivoli Common Reporting user name and password.
  6. Configure your Tivoli Data Warehouse and data scripts. Some report packages might not have data scripts.
  7. An installation summary is provided, then press **Enter** to being installation.
- Using silent mode:
  1. Create the silent installer response file and name it `silent_installer.properties`.

2. Run the `setup_<platform>.exe/.bin -i silent -f <path_to_response_file>` command.

Results

Agent reports are now installed on your Tivoli Common Reporting server.

What to do next

You can now use the reports to display monitoring data gathered by the monitoring agents. To learn more on how to run, administer, and edit reports in Tivoli Common Reporting, see the "Working with reports" (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/tcr_original/ttcr_working.html) topics.

See the agent-specific user's guide for additional reporting information.

Importing and running IBM Cognos reports

You must create and populate the Cognos dimensions tables and then import the report package to enable Tivoli Common Reporting for the monitoring agents.

About this task

If your reports are included with the report installer, see "Importing reports by using the report installer" on page 537.

If your reports are not included with the report installer, see "Importing reports through the Dashboard Application Services Hub" on page 541.

For detailed report installation information, see the "Installing Tivoli Common Reporting" topic in the Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/install/tcr_t_install.html).

Running a prerequisites scan

The OS Agents Report Prerequisites Scanner report provided by the OS agents report package, allows you to check your system's prerequisites for using the IBM Tivoli Monitoring OS Agent reporting solution and Tivoli Common Reporting. Run this health check report for an overview of your prerequisites and to verify that your dimensions tables are available.

Before you begin

A DataSource connection must be defined in Tivoli Common Reporting to connect to the Tivoli Data Warehouse for this report to work.

For DB2 on z/OS, you must establish a connection and grant the required privileges. Run the following commands from the `*DB2_install_dir*/*version*/bnd` directory:

```
db2 bind @ddcsmvs.lst SQLERROR CONTINUE BLOCKING ALL GRANT PUBLIC
db2 grant select on SYSIBM.SYSVIEWS to USER1
```

About this task

The OS Agents Report Prerequisites Scanner report returns data on the following areas:

- Prerequisites for Tivoli Common Reporting Shared Dimensions
- Prerequisites for IBM Tivoli Monitoring Shared Dimensions
- Prerequisites by report

If your system is configured correctly, the time dimensions will report as satisfied. If a report set returns as unsatisfied, you can dismiss this if it pertains to a report set you do not wish to collect.

Limitation: When using reporting in your national language, some text strings display in only English.

Procedure

1. From the Tivoli Common Reporting Navigator tree, select Public Folders → IBM Tivoli Monitoring OS Agents Reports → Prerequisites Validation → **OS Agents Report Prerequisites Scanner report**.
2. Select between the two available report types:
   - Show all sections: This option shows all sections, including failing (unsatisfied prerequisites) and success (satisfied prerequisites).
   - Show failing sections only: This option shows only sections with failing (unsatisfied prerequisites).
3. Review the returned report information. Take any actions needed to resolve any unsatisfied prerequisites. Read the returned information in the **Legend** tables and look in the **Status** and **Details** columns for summary information.

What to do next

When viewing your report, in the **Details** column, you can click on ⓘ to view the **Table Details** report for additional information.

Note: The **Table Details** report can only be accessed from clicking the icon in the **Details** column. If you try to run the report directly from Public Folders → IBM Tivoli Monitoring OS Agents Reports → Prerequisites Validation → **Table Details**, you will receive an error.

Connecting to the Tivoli Data Warehouse using the database client over ODBC

Cognos uses ODBC to connect to the database. It is important to first install a database client on the Tivoli Common Reporting server and have it connect to the Tivoli Data Warehouse. If your database and the Tivoli Common Reporting server are on the same computer, no database client is needed.

Procedure

- **IBM DB2**
  1. Make sure you have deployed a DB2 database client on the computer where the Cognos-based Tivoli Common Reporting engine is installed. The client should be of the same version as the database that the Tivoli Data Warehouse is using.
     Linux UNIX If a DB2 server is installed where the Cognos-based Tivoli Common Reporting engine is installed, the DB2 client files are already

available. However, you will need to copy the DB2 library file (libdb2.a) to the `Cognos_8_Install_dir/bin` directory to allow Cognos to successfully connect to the database server where the Tivoli Data Warehouse resides.
2. Connect the DB2 database client to the database server by running the DB2 Configuration Assistant, configuring the local net service name configuration, and restarting your system.
3. Note the name of the connection you have created because it is used in Tivoli Common Reporting by the report installer.

- **Microsoft SQL Server**
  1. Make sure you have deployed the MS SQL database client on the computer where the Cognos-based Tivoli Common Reporting engine is installed.
  2. Connect the MS SQL client to the database server by running the MS SQL Management Studio Express, configuring the local net service name configuration, and restarting your system.
  3. Note the name of the connection you have created because it is used in Tivoli Common Reporting by the report installer.
- **Oracle**
  1. Make sure you have deployed the Oracle database client on the computer where the Cognos-based Tivoli Common Reporting engine is installed.
  2. Connect the Oracle database client to the database server by running the Oracle Net Configuration Assistant, configuring the local net service name configuration, and restarting your system.
  3. Note the name of the connection you have created because it is used in Tivoli Common Reporting by the report installer.

Results

You can now import and run reports.

Importing reports through the Dashboard Application Services Hub

Use the Dashboard Application Services Hub user interface to import reports for the Tivoli Monitoring OS agents.

Before you begin

The monitoring agent report models are based on IBM Cognos. You must have Tivoli Common Reporting V3.1 (with Cognos engine) installed and running on the computer on which you install the reports. You must also have created and populated the dimensions tables as instructed in “Creating and populating the time dimensions tables” on page 532 and “Creating and populating the resource dimension table” on page 535, and connected to the data warehouse as described in “Connecting to the Tivoli Data Warehouse using the database client over ODBC” on page 540.

Restriction: This import method can be used for Cognos reports only.

About this task

You must obtain a report package you want to work with. You can download packages from the IBM Integrated Service Management Library (http://www.ibm.com/software/brandcatalog/ismlibrary), or you can create one by using the Content Administrator interface. All the packages you want to import must be stored in the `TCR_component_dir\cognos\deployment` directory.

Procedure

1. Launch the Dashboard Application Services Hub administrative console and log in.
2. Go to **Common Reporting**.
3. In the **Work with reports** window on the right, click **Administration** from the **Launch** drop-down list.
4. Go to **Configuration** tab, and open the **Content Administration** section.
5. Create a new package import by clicking **Import**, which opens a **New Import wizard**.
6. Follow the wizard to import a new package.

Results

OS Agent Reports are now installed on your Tivoli Common Reporting server.

What to do next

You can now use the reports to display monitoring data gathered by the OS Monitoring Agents. To learn more on how to run, administer, and edit Cognos reports in Tivoli Common Reporting, see the "Working with reports" (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/tcr_original/ttcr_working.html) topics.

For additional reporting information, see the operating system agent user's reference at Operating System Agent Documentation http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/ic/landing_osagents.htm.

If you encounter any problems installing the report package, see the *IBM Tivoli Monitoring Troubleshooting Guide*.

Importing and running BIRT reports

Use the Eclipse BIRT (Business Intelligence and Reporting Tools) Report Designer to develop your own reports or edit exisitng reports. The Eclipse BIRT Report Designer is not required to download or install reports.

Import a BIRT report package

Import the report package for a monitored application to get the required files for defining BIRT reports.

Before you begin

A *report package* is a .zip file containing all of the data required for defining one or more reports, including the required designs and resources and the hierarchy of report sets to contain the reports. The monitoring agent reports are included as .zip files on the agent image in the REPORTS directory. For example, on a Windows computer, if the image drive is labelled D:, reports are in directories such as: D:\REPORTS\kqb. See the agent reporting chapter or *Product reporting guide* for the location of the reports.

About this task

The **-import** command flag for the **trcmd** command imports BIRT and Cognos report packages and report designs. The type of a package is recognized automatically. This command can be used for single-box installation and on the reporting engine. It is not supported for other scenarios.

For more information, see trcmd -import in the Tivoli Common Reporting Information Center.

For Tivoli Common Reporting V2.1 use the following procedures:

Procedure

1. Use this syntax to import a report package:
   trcmd -import -bulk *pkgFile* [**-reportSetBase** *rsBase*] [**-resourceBase** *resourceBase*] [**-designBase** *designBase*] [**-help**]

   This example imports a BIRT package named avail_skills.zip with its resource directory imported from C:\download:`trcmd -import -bulk C:\download\sth\report\avail_skills.zip -reportSetBase myReportSetBase -resourceBase myResourceBase -designBase myDesignBase -user tipadmin -password admin`
2. Use this syntax to import a report design and also create a new report associated with the design:
   trcmd -import -design *designPath* [**-resourceDir** *resourcePath*] -**reportSetBase** *rsBase*

   Usage notes:
   - During Cognos reports import, the **-resourceBase**, **-designBase**, and **-resourceDir** parameters are ignored.
   - You can import a single Cognos report from an .xml file using the **-design** parameter.

Results

The Navigation tree shows an item for the reports and items for subsets of the reports.

What to do next

Changing the data source in a report will change the data sources for all reports. You do not need to repeat the change for all reports.

For more information, see the “Working with reports” (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/tcr_original/ttcr_working.html).

Configure the data source

All reports in a BIRT report package must point to the same data source. The data source pointer needs to be modified to point to your Tivoli Data Warehouse.

About this task

After you have installed Tivoli Common Reporting and imported your first set of reports, you or a user with Administrator authority must copy JDBC drivers from the local or remote database manager that you are using to run the Tivoli Data

Warehouse into the Tivoli Common Reporting server directory. You specify these files in the **Edit Data Source** window. Perform the following steps to install these drivers.

Procedure

1. Locate the JDBC driver files:

   IBM DB2 `db2jcc.jar` and `db2jcc.license_cu.jar`

   Windows `C:\Program Files\IBM\SQLLIB\java`

   Linux UNIX *`db2_installdir`*`/java`
   For example, the default DB2 on the workstation Version 9 installation directory is `/usr/opt/db2_09_01` on AIX and `/opt/IBM/db2/V9.1` on Linux and Solaris.

   The JDBC drivers are typically found in this default DB2 installation path or in the `java` directory of whatever alternate path you specified for DB2 installation.

   You can also download the IBM DB2 Driver for JDBC and SQLJ from the IBM website.

   Microsoft SQL Server `sqljdbc.jar`

   Download the Microsoft SQL Server JDBC Driver from the Microsoft website. The SQL Server 2005 JAR file name and location after installation is *`mssql2005installdir`*`/sqljdbc_1.1/enu/sqljdbc.jar`.

   Oracle `oraclethin.jar`

   Obtain the JDBC Type 4 driver from the Oracle website. The Oracle JDBC driver JAR file name and location after installation is *`oracleinstalldir`*`/jdbc/lib/oraclethin.jar`.

2. Copy the JDBC driver to your Tivoli Common Reporting installation directory:
   - *`tcr_install_dir`*`TCR_component_dir\lib\birt-runtime-2_2_2\ReportEngine\plugins\org.eclipse.birt.report.data.oda.jdbc_2.2.2.r22x_v20071206\drivers`
   - For a DB2 data source, copy the DB2 JDBC drivers and the license JAR file to the same location. You can copy `db2jcc.jar` and th `db2jcc_licence_cu.jar` file on the DB2 server system from the *`db2_installdir`*`/java` location (for example, `C:\Program Files\IBM\SQLLIB\java`).

3. Run the **trcmd -modify** command to configure the data source. See the trcmd-modify command topic in the Tivoli Common Reporting Information Center for complete instructions.

What to do next

For additional information, refer to the JDBC driver section of the *IBM Tivoli Common Reporting: Development and Style Guide* on IBM developerWorks Tivoli Common Reporting community.

For more information about Tivoli Data Warehouse connectivity issues, refer to "Part 5. Setting up data warehousing" in *IBM Tivoli Monitoring Installation and Setup Guide* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc_6.3fp2/install/itm_install.htm).

Generate a sample BIRT report

Tivoli Common Reporting BIRT report packages are organized by product. Select a report set to generate a report.

Procedure

1. Launch a browser that is compatible with Tivoli Common Reporting, and enter this address: `https://<address>:16311/ibm/console/logon.jsp`, where <address> is the IP address or host name of the system where Tivoli Common Reporting is installed. If your environment was configured with a port number other than the default, enter that number instead. The default path to the server is `/ibm/console`. However, this path is configurable, and might differ from the default in your environment.
2. In the navigation tree on the left, expand the **Reporting** item.
3. Click **Common Reporting**. All the reports available (all the report packages that you have imported) are displayed in the text area of the screen.
4. On the **Navigation** tab, expand the **Tivoli Products** item.
5. Select the Tivoli product whose reports you want to use from the list of available products.
6. If this is the first time you have run reports based on data from the Tivoli Data Warehouse, perform the following steps:
   a. Define the Tivoli Data Warehouse as the data source for your reports; for more information, see "Configure the data source" on page 543. For information about data sources, see the *IBM Tivoli Common Reporting User's Guide* or the online help for Tivoli Common Reporting.
   b. Copy the required JDBC drivers from the local or remote database manager that you are using to run the Tivoli Data Warehouse into the Tivoli Common Reporting server directory.

   You might need to increase the default heap size for the Java Virtual Machine (JVM) on the Java command to start the Tivoli Common Reporting server. If you see these messages displayed when you create a report, you might have to increase the default heap size:

   ```
   Processing has ended because of an unexpected error.
   See the Tivoli Common Reporting log files for more information.
   ```

   See *OMEGAMON XE and Tivoli Management Services on z/OS: Reports for Tivoli Common Reporting* for information about how to increase the default heap size.
7. In the Actions column, click the run ▶ icon for the report you want to launch, and then select the type of report format: HTML (the default), PDF, Microsoft Word, Microsoft Excel, or Adobe Postscript.

   When you select a report from Tivoli Common Reporting, you are presented with a **Report Parameters** window that prompts you for information that will be used to generate the report. The title of the parameters window indicates the type of report that will be generated. For a description of the types of reports, see the agent user's guide or product reporting guide for the agent or product with which you are working.

   A Report Parameters window contains some fields common to all reports (for example, timeframe). Other fields are specific to the agent running the reports. For most reports, you select a timeframe, resources, the summarization level of the data, and the attributes to graph, or click ▶ to accept all displayed defaults.
8. Click **Run** to generate a report matching your parameter definitions.

Results

An hourglass is displayed while Tivoli Common Reporting gathers report data and creates formatted output. After processing finishes, the report is displayed in the Dashboard Application Services Hub.

What to do next

If no report is generated or you see a message indicating that the requested data is unavailable, see the *IBM Tivoli Common Reporting User's Guide* for information about defining a data source.

If you are viewing an HTML or PDF report, you can also click any embedded links to open drill-through reports. Clicking a drill-through embedded link causes the report to link back to itself with the newly passed parameters or to a secondary (drill-down or summarized) report. Examples of drill-through links include clicking on a bar or line chart or on a table heading.

Chapter 18. Replicating the Tivoli Enterprise Portal Server database

Use the utilities provided with the Tivoli Enterprise Portal Server to migrate the Tivoli Enterprise Portal customizations that are stored in the TEPS database. This includes user IDs, Navigator views, custom queries, custom workspaces, and local terminal scripts.

You can also use the migrate-import and migrate-export scripts to switch from a 32-bit Tivoli Enterprise Portal Server to a 64-bit Tivoli Enterprise Portal Server on the same operating system.

Understanding the Tivoli Enterprise Portal Server database

Before replicating the Tivoli Enterprise Portal Server database, be aware of what the database contains and what is required before you begin.

Tivoli Enterprise Portal customizations

Tivoli Enterprise Portal customizations are stored at the portal server in the TEPS database. This includes user IDs, Navigator views, custom queries, custom workspaces, and local terminal scripts. It does not store situations, policies, and managed system groups.

During an installation upgrade to a new version of the portal server, the TEPS database is updated with any new or changed predefined Navigator views, queries and workspaces. Custom Navigator views, queries, and workspaces that you created are not affected.

The TEPS database replication is required for moving from a test environment to a production environment. You can also use the procedure for backing up the database as a precautionary measure before applying a fix pack or upgrading to a new version.

All workspaces previously created in the destination environment are replaced with those that were created in the source environment. Any existing user changes in the destination environment are also replaced.

Requirements for replication

Before migrating the Tivoli Enterprise Portal Server, make sure your environment fulfills these requirements:

- The portal servers on the source and target computers must be configured to connect to the same hub monitoring server.
- The portal servers on the source and target computers must be at Version 6.2.1 or later and, ideally, both have been installed from the same Tivoli Monitoring Base DVD.
- The portal servers on the source and target computers must have been installed the same way:

- The selected applications are the same. For example, if the source portal server has support for the UNIX, Windows Servers, and MQ Series applications installed, then the target portal server must have the same application support.
- The same database program is used for the portal server database. For example, IBM DB2 UDB.

CLI tacmds for selective replication

The command line interface has tacmds for exporting and importing specific IBM Tivoli Monitoring objects. See the *IBM Tivoli Monitoring Command Reference* for a description of each of these commands and their syntax.

tacmd exportworkspaces
tacmd importworkspaces
Selectively copy workspaces from one portal server to another.

tacmd exportQueries
tacmd importQueries
Export custom queries to an XML file; then import them into a portal server.

tacmd bulkExportSit
tacmd bulkImportSit
Export all Tivoli Monitoring enterprise situations from one hub monitoring server and importing into another.

tacmd bulkExportPcy
tacmd bulkImportPcy
Export all Tivoli Monitoring policies from one hub monitoring server and importing into another.

tacmd exportNavigator
tacmd importNavigator
Export custom Navigator views and their assigned workspaces, queries, and situation associations to an XML file; then import them into a portal server.

tacmd exportSitAssociations
tacmd importSitAssociations
Export all the situation associations for a Navigator view or a particular Navigator item to an XML file; then import them into a portal server.

tacmd exportSysAssignments
tacmd importSysAssignments
Export all managed system assignments for a Navigator view or a particular Navigator item to an XML file; then import them into a portal server.

Running the migrate-export script

Export the Tivoli Enterprise Portal Server to create a copy of the TEPS data base for applying to another computer or to keep as a backup.

Before you begin

The portal server can be running or stopped when you initiate the migrate-export script. If the server is stopped, the script starts it temporarily in a limited mode to accomplish the export. Do not start the portal server manually until the migrate-export has completed.

About this task

On the computer where the source Tivoli Enterprise Portal Server is installed, take these steps to create a copy of the TEPS database

Procedure

- Windows
  1. Open a command prompt window: **Start→ Run**, enter `CMD`.
  2. Change to the *`install_dir`*`\CNPS` directory.
  3. Enter: `migrate-export`

  The migrate-export script generates a file named **saveexport.sql** in the *`install_dir`*`\CNPS\sqllib` subdirectory. It contains all the Tivoli Enterprise Portal Server data.
- Linux UNIX
  1. On the source system, open a terminal window.
  2. Change to the bin subdirectory of your IBM Tivoli Monitoring installation, such as: `cd /opt/IBM/ITM/bin`
  3. Enter: `./itmcmd execute cq "runscript.sh migrate-export.sh"` Be sure to use the " double-quote symbol and not ' single-quote.

  The migrate-export script generates a file named **saveexport.sql** in the *`install_dir`*`/$platform/cq/sqllib` subdirectory. It contains all the Tivoli Enterprise Portal Server data.

Running the migrate-import script

When you have a copy of the Tivoli Enterprise Portal Server database, named `saveexport.sql`, import it to a any portal server installation of the same version where you want duplicate settings.

Depending on the contents of the saveexport.sql, this process can completely replace the existing TEPS database.

Some of the tables included in the import script are applicable only to the CandleNet Portal Server, the predecessor to Tivoli Enterprise Portal Server. Unless you are not importing a Tivoli Enterprise Portal Server database, the migrate-import log file will contain SQL errors about an undefined name, such as `SQLExecDirect rc=-1: SQL_ERROR SQLSTATE: 42S02, ERR: -204, MSG: [IBM][CLI Driver][DB2/LINUX] SQL0204N "ITMUSER.TAGGROBJ" is an undefined name. SQLSTATE=42704 RC = -1` (also `ITMUSER.TMANOBJS`, `ITMUSER.TMANTMPL`, `ITMUSER.TTMPLSIT`, `ITMUSER.TTMPLSTA`, `ITMUSER.TSTUSERA`). Ignore these errors.

Running migrate-import from source Windows to target Windows

Run the migrate-import script to import a copy of the Tivoli Enterprise Portal Server database from a Windows computer to another Windows computer.

Before you begin

This procedure overwrites the TEPS database on the target computer.

About this task

On the Windows computer where the target portal server is installed, take these steps to import the TEPS database that was copied from another Window computer using migrate-export.

Procedure

1. Stop the portal server on the target system.
2. On the source system, open a command prompt: Click **Start** → **Run**, and enter `CMD`.
3. Copy file **saveexport.sql** that was generated by the migrate-export.bat script from the source system to *`install_dir`*`\CNPS\sqllib` on the destination system, where <*mapped drive on destination system*> is the disk drive on the source system where this file resides. Example:

   ```
   copy <mapped drive on destination system>:\IBM\ITM\CNPS\sqllib
        \saveexport.sql c:\ibm\itm\cnps\sqllib
   ```

   If a drive is not already defined, you must map a drive to the source system from the destination system with the **`net use`** command.
4. On the target system, change to the *`install_dir`*`\CNPS` directory and enter: `migrate-import`. Running the migrate-import process stops the portal server if it is currently running.
5. If you are using the migrate-import function to move the TEPS database from one release to another, perform this task after migrating the database to add application support:
   a. Open `<`*`install_dir`*`>\CNPS\kfwalone` in a text editor.
   b. Set `KFW_MIGRATE_FORCE=Y`, then save and close the file.
   c. Invoke this script to apply the current portal server application support to the newly migrated TEPS database: `<`*`install_dir`*`>\CNPS\buildpresentation.bat`
6. Restart the portal server.

Running migrate-import from source Windows to target Linux or UNIX

Run the migrate-import script to import a copy of the Tivoli Enterprise Portal Server database from a Windows computer to a Linux or UNIX computer.

Before you begin

This procedure overwrites the TEPS database on the target computer.

About this task

On the Linux or UNIX computer where the target portal server is installed, take these steps to import the TEPS database that was copied from a Window computer using migrate-export.

Procedure

1. Stop the portal server on the target system.
2. On the source system, open a command prompt: Click **Start** → **Run**, and enter `CMD`.

3. Copy **saveexport.sql** that was generated by the migrate-export.bat script from the source Windows system to the target system's *install_dir*/*$platform*/cq/sqllib directory, where *$platform* is li6243 for Intel Linux or ls3263 for zSeries Linux on the destination system.
4. Open a terminal window on the target system.
5. Change to the bin subdirectory of the Tivoli Monitoring installation: *Install_dir*/bin. For example,

   ```
   cd /opt/IBM/ITM/bin
   ```
6. In the terminal window, enter `./itmcmd execute cq "runscript.sh migrate-import.sh"`. Be sure to use the " double-quote symbol and not ' single-quote. The script processes a file named **saveexport.sql** in the *install_dir*/*$platform*/cq/sqllib directory. Depending on the contents of the saveexport.sql file, this process can completely replace the existing portal server data.
7. If you are using the migrate-import function to move the TEPS database from one release to another, perform this task after migrating the database to add application support:
   a. Open *Install_dir*/cq/bin/lnxnocmsenv in a text editor.
   b. Set KFW_MIGRATE_FORCE=Y, then save and close the file.
   c. Invoke the following script to apply the current portal server application support to the newly migrated TEPS database: *Install_dir*/bin/itmcmd execute cq InstallPresentation.sh. For example,

      ```
      /opt/IBM/ITM/bin/itmcmd execute cq InstallPresentation.sh
      ```
8. Restart the portal server from the *Install_dir*/bin directory with the following command:

   ```
   ./itmcmd agent start cq
   ```

Running migrate-import from source Linux or UNIX to target Windows

Run the migrate-import script to import a copy of the Tivoli Enterprise Portal Server database from a Linux or UNIX computer to a Windows computer.

Before you begin

This procedure overwrites the TEPS database on the target computer.

About this task

On the Windows computer where the target portal server is installed, take these steps to import the TEPS database that was copied from a Linux or UNIX computer using migrate-export.

Procedure

1. Stop the portal server on the target system.
2. Copy file **saveexport.sql** that was generated by the migrate-export script from the source Linux or UNIX system (/opt/IBM/ITM/$platform/cq/sqllib) to *install_dir*\CNPS\sqllib on the target system.
3. On the target system, change to the *install_dir*\CNPS directory and enter: `migrate-import`. Running the migrate-import process stops the portal server if it is currently running.

4. If you are using the migrate-import function to move the TEPS database from one release to another, perform this task after migrating the database to add application support:
   a. Open <*install_dir*>\CNPS\kfwalone in a text editor.
   b. Set KFW_MIGRATE_FORCE=Y, then save and close the file.
   c. Invoke this script to apply the current portal server application support to the newly migrated TEPS database: <*install_dir*>\CNPS\buildpresentation.bat
5. Restart the portal server.
6. Restart the Tivoli Enterprise Portal Server.

Running migrate-import from source Linux or UNIX to target Linux or UNIX

Run the migrate-import script to import a copy of the Tivoli Enterprise Portal Server database from a Linux or UNIX computer to another Linux or UNIX computer.

Before you begin

This procedure overwrites the TEPS database on the target computer.

About this task

On the Linux or UNIX computer where the target portal server is installed, take these steps to import the TEPS database that was copied from another Linux or UNIX computer using migrate-export.

Procedure

1. Stop the portal server on the target system.
2. Copy file **saveexport.sql** that was generated by the migrate-export script from the source Linux or UNIX system *install_dir*/*$platform*/cq/sqllib directory to the same directory on the target system, where *install_dir* is the installation directory on the destination system, such as /opt/IBM/ITM/, and *$platform* is the operating system, such as li6243 for Intel Linux or ls3263 for zSeries Linux.
3. Open a terminal window on the target system.
4. Change to the bin subdirectory of the Tivoli Monitoring installation: *Install_dir*/bin. For example,
   ```
   cd /opt/IBM/ITM/bin
   ```
5. In the terminal window, enter the following command.
   ```
   ./itmcmd execute cq "runscript.sh migrate-import.sh"
   ```

   Be sure to use the " double-quote symbol and not ' single-quote. The script processes a file named saveexport.sql in the IBM/ITM/$platform/cq/sqllib subdirectory. Depending on the contents of the saveexport.sql file, this process can completely replace the existing portal server data.
6. If you are using the migrate-import function to move the TEPS database from one release to another, perform this task after migrating the database to add application support:
   a. Open *Install_dir*/cq/bin/lnxnocmsenv in a text editor.
   b. Set KFW_MIGRATE_FORCE=Y, then save and close the file.

   c. Invoke the following script to apply the current portal server application support to the newly migrated TEPS database: *Install_dir*/bin/itmcmd execute cq InstallPresentation.sh. For example,

      ```
      /opt/IBM/ITM/bin/itmcmd execute cq InstallPresentation.sh
      ```

7. Restart the portal server from the *Install_dir*/bin directory with the following command:

   ```
   ./itmcmd agent start cq
   ```

Appendix A. IBM Tivoli Monitoring Web Services for the SOAP server

This appendix describes the IBM Tivoli Monitoring Web Services feature of the SOAP server. The IBM Tivoli Monitoring Web Services solution provides you with an industry-standard open interface into IBM Tivoli Monitoring solutions. This open interface provides easy access to Tivoli performance and availability data, allowing you to use this information for advanced automation and integration capabilities.

IBM Tivoli Monitoring Web Services implements a client/server architecture. The client sends Simple Object Access Protocol (SOAP) requests to the SOAP server which is installed with the hub monitoring server. The server receives and processes the SOAP requests from the client.

Predefined SOAP methods let you perform many functions within the monitored environment. You can begin to use the SOAP methods immediately. You can also use these SOAP methods as templates in creating your own advanced methods.

SOAP works with any programming or scripting language, any object model and any Internet wire protocol. Tivoli SOAP methods can be invoked by PERL, Javascript, VBSCRIPT, JSCRIPT, C++, Java, and through a browser.

Note: Web Services does not support situation creation. Use the Tivoli Enterprise Portal Situation editor or the CLI **tacmd createSit** function for situation creation. The SOAP server can query only agent and managed system attributes.

About the SOAP client

Simple Object Access Protocol (SOAP) is a communications XML-based protocol that lets applications exchange information through the Internet.

SOAP is platform independent and language independent. SOAP uses XML to specify a request and reply structure. It uses HTTP as the transport mechanism to drive the request and to receive a reply.

Important: Prior to using IBM's solution, you must have a basic understanding of SOAP, of Extensible Markup Language (XML) and XML Namespaces, and of the Web Services Description Language (WSDL).

Configuring Tivoli Monitoring Web Services (SOAP Server)

By default, the SOAP server is installed on the Hub Tivoli Enterprise Monitoring Server. Use the configuration topics to establish SOAP server communication between hub monitoring servers and to establish security on the SOAP server.

The instructions in this chapter assume that you have a basic understanding of SOAP, XML and XML Namespaces, and the Web Services Description Language (WSDL). These steps are required to configure SOAP:

- Define the hubs with which your SOAP Server communicates.
- Create users and grant them access.
- Verify that you have successfully configured SOAP.

Note: You cannot make SOAP requests to earlier version SOAP servers.

Defining hubs

The procedure below describes how you can use the Manage Tivoli Enterprise Monitoring Services to activate the SOAP server and define the hubs with which the SOAP server communicates.

About this task

Use the following steps to define SOAP hubs:

Procedure

1. On the computer where the hub monitoring server is installed, start Manage Tivoli Enterprise Monitoring Services:
   a. Windows Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.
   b. Linux or UNIX Change directory to *install_dir*/`bin` and enter: `./itmcmd manage`

      To use the command-line interface, see "Defining hubs on Linux and UNIX using the CLI" on page 557.
2. Right-click Tivoli Enterprise Monitoring Server and click **Reconfigure**.
3. Select or clear the □ **Security: Validate User** field.
4. Open Manage Tivoli Enterprise Monitoring Services.
5. Right-click **Tivoli Enterprise Monitoring Server**.
6. Click **Advanced** → **Configure SOAP Server Hubs**.
7. Click **Add Hub**. The Hub Specification window is displayed.
8. Select the communications protocol to be used with the from the **Protocol** menu.
9. Specify an alias name in the **Alias** field (for example: `HUB2`). Alias names can be a minimum of 3 characters and a maximum of 8 characters.
10. Perform one of the following steps:
    a. If you are using TCP/IP or TCP/IP Pipe communications, complete the following fields:

Table 70. 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.

    b. If you are using SNA communications, complete the following fields:

Table 71. 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.

Table 71. SNA Fields in Hub Specification Dialog (continued)

Field	Description
LU6.2 LOGMODE	The name of the LU6.2 logmode. Default: CANCTDCS.
TP Name	The Transaction Program name for the monitoring server.

Note: If you are connecting to a remote monitoring server, the protocol information must be identical to that used for the hub monitoring server.

11. Click **OK**. The server tree is displayed.

Defining hubs on Linux and UNIX using the CLI

Define the SOAP server hubs on a Linux/UNIX system using the CLI procedure.

About this task

Complete the following steps to configure the SOAP server:

Procedure

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 72. SOAP hub configuration values

Configuration	Value
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.

Table 72. SOAP hub configuration values (continued)

Configuration	Value
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

Define users on each hub and specify access rights for each user (query or update) by following the procedure below.

About this task

Complete the following procedure to define users and specify access rights:

Procedure

1. Select the server (click anywhere within the server tree displayed), if required.
2. Under Add User Data, type the user name. 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.

   Note: If you do not supply a user ID, all users are given permission to update data.
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**.

Attention: When SOAP security is enabled, the user **Query** and **Update** permissions control authorization of the **tacmd** commands that send requests to the hub monitoring server and requests from other SOAP clients. The SOAP_IS_SECURE environment variable of the hub monitoring server must also be set to YES if you want to control which users can issue SOAP CT_EMail and CT_Export requests.

The query permission prevents the user from performing the create, update, and delete operations, and also the **tacmd** commands for remote deploy and execution of commands, such as **executeaction** and **executecommand**.

The update permission grants permission to run all SOAP operation and applies to all **tacmd** commands that send requests to the hub monitoring server, except for the **tacmd getFile** and **tacmd putFile** commands. Permission to run the **tacmd getFile** and **tacmd putFile** commands is controlled by the KT1_TEMS_SECURE environment variable of the hub monitoring server. For more details on how to enable permission to run these two commands, see the *IBM Tivoli Monitoring Command Reference*.

Configuring IBM Tivoli Monitoring Web Services (SOAP Server) on UNIX and Linux systems

Configure the SOAP Server on UNIX and Linux computers.

About this task

Use the following steps to define SOAP hubs on UNIX or Linux using Manage Tivoli Enterprise Monitoring Services:

Procedure

1. Change to the *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 is displayed. If the current host is not displayed in the Hubs tree, define it before defining the hubs with which it communicates.
4. Confirm that the host name or IP address, port number, and protocol 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 port number of the host in the appropriate fields.
   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.

Tuning SOAP transaction performance on AIX systems

SOAP transaction performance can be modified on AIX by deciding whether or not to allow delayed acknowledgments. Tune the performance by following the procedure below.

About this task

The default behavior on AIX systems for Transmission Control Protocol (TCP) connections is to allow delayed acknowledgments (*Ack* packets) by up to 200 ms, and is controlled by the **tcp_nodelayack** network option. This delay allows the packet to be combined with a response and it minimizes system overhead. If you set **tcp_nodelayack** to **1**, the acknowledgment is immediately returned to the sender. With this setting, slightly more system overhead is generated but results in much higher network transfer performance when the sender is waiting for acknowledgment from the receiver. To find out more about the **tcp_nodelayack** option, refer to the IBM System p® and AIX Information Center.

To set this parameter, complete the following procedure:

Procedure

Access a user account that has **root** privileges and issue the following command:

```
no -p -o tcp_nodelayack=1
```

Results

The following output is typical:

```
Setting tcp_nodelayack to 1
Setting tcp_nodelayack to 1 in nextboot file
```

This is a dynamic change that takes effect immediately. The **-p** flag makes the change persistent, so that it is still in effect the next time you start the operating system.

Enabling SOAP security

When you enable the **Security: Validate Users** option when configuring the hub monitoring server, all SOAP requests are authenticated except for CT_EMail and CT_Export requests.

For information on how to configure the hub monitoring server to validate SOAP users, see the *IBM Tivoli Monitoring Installation and Setup Guide*. If you also want to authenticate SOAP users who are sending CT_EMail or CT_Export SOAP requests, you must enable the `SOAP_IS_SECURE` environment variable on the hub monitoring server.

About this task

By default the `SOAP_IS_SECURE` environment variable is disabled. Enabling this variable requires all users who submit CT_EMail or CT_Export requests to know a hub monitoring server credential. Setting `SOAP_IS_SECURE=YES` only works if user validation is turned on in the hub monitoring server.

Procedure

1. On the computer where the hub monitoring server is installed, open the KBBENV or ms.ini file:

Windows

Use Manage Tivoli Enterprise Monitoring Services (Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Enterprise Monitoring Services) to edit environment files. Right-click the component you want to modify and click **Advanced** → **Edit ENV File**. You must recycle the component to implement the changes.

Linux UNIX

Edit the environment file directly. Edit environment variables in the `<`*`install_dir`*`>/config/ms.ini` file.

2. Locate and uncomment the line: `# SOAP_IS_SECURE=YES`. For example: `SOAP_IS_SECURE=YES`
3. Save and close the monitoring server environment file.
4. For Windows, you must recycle the component to implement the changes. For Linux or UNIX, you must reconfigure and recycle the monitoring server to implement the changes.

Verifying the configuration

Verify that you have configured the SOAP server correctly.

About this task

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.

Procedure

1. Verify that the monitoring server that you used to enable SOAP is running. If not, start it.
2. Open a command window.
3. Windows Change to the *install_dir*\cms directory or Linux UNIX The *install_dir*/TBD.
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.

Results

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.

Using IBM Tivoli Monitoring web services

Numerous SOAP methods are included with IBM Tivoli Monitoring web services. These methods allow you to dynamically query and control IBM Tivoli Monitoring environments.

Using these SOAP methods, you can:

- Stop or start policies and situations
- Forward trapped messages from System Automation for Integrated Operations Management and display them on a Universal Message console
- Retrieve attribute data that you can display in charts or reports
- Open and close events
- Make real-time requests for data
- Issue SOAP requests as system commands in Tivoli Enterprise Portal

You can also use this product to test a request to ensure it works correctly. You can then create a policy that submits multiple requests for processing. In addition, you can generate daily operation summaries.

You can store retrieved data in the Tivoli Data Warehouse, as described in the historical data collection guide.

Note: IBM Tivoli Monitoring web Services provides XML data rows. Use IBM's SOAP methods in combination with your own scripts to display the data in charts and tables.

SOAP query responses might appear to be ordered alphabetically, but some attributes do not follow alphabetical order. Automation tasks must examine content without regard to order.

User IDs

At installation and configuration time, you are asked to supply user IDs for those who need access to monitoring server data. If no user IDs are supplied, all users are given permission to update data.

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.

You can also make changes at a later time to add or to remove users' access to monitoring server data. See the *IBM Tivoli Monitoring: Installation and Setup Guide* for details.

Starting the SOAP client and making a request

Start the SOAP client either by using Internet Explorer or using the SOAP client command-line utility (not available on z/OS systems).

About this task

When you use the SOAP client in conjunction with Internet Explorer to issue SOAP requests, you can modify, if needed, the tags or the text. In contrast, the command-line utility simply displays the output of the request at the command line.

Using your browser

Use Windows Internet Explorer or Mozilla Firefox to enter the URL for the SOAP service console.

About this task

After installing the Tivoli Monitoring Web Services SOAP client, perform these actions:

Procedure

1. Start Internet Explorer version 5 or later, or Mozilla Firefox . Be sure to enable the **Access data sources across domains** option in Internet Explorer's security settings.
2. In the Address field, type the URL for the SOAP client, where `localhost` can be used literally when accessing the SOAP server running on the same system or changed to the proper host name or network address of a SOAP server running on a different system:

   ```
   http://localhost:1920///cms/soap/kshsoap.htm
   ```

   The port number for the HTTP service is 1920.

   Note: You can also route requests to a remote hub by replacing **soap** in the Address field with the alias name of the hub you want to access (**HUB_localhost** in the example below). The alias must have been previously defined to the SOAP server (for information about defining hub aliases, see the installation documentation). For example: `http://localhost:1920///cms/HUB_localhost/kshsoap.htm`
   The SOAP client HTML page is displayed.
3. Select a SOAP method from the list in the first field. After you select a method, the other fields are updated automatically.
4. Modify, if needed, the tags or the text in the “Edit Payload (XML)” area.
5. Click **Make SOAP Request**. The output of the request displays in the Your SOAP Response Payload area.

What to do next

When issuing a CT_Get request against a particular agent type, the monitoring server where the SOAP server is running must be configured and have the application support for that agent type. For example, when issuing a CT_Get request for a z/OS agent connected to an z/OS monitoring server, the monitoring server running the SOAP server must be configured and have the application support for that z/OS agent.

Using the SOAP client command-line utility (kshsoap)

The SOAP client command-line utility, kshsoap, is an http client. It issues direct SOAP requests and displays the output at the command line.

About this task

Complete these steps to create a SOAP request file and a SOAP URL receiver file and send the request.

Procedure

- Windows
  1. On the Tivoli Enterprise Monitoring Server system where the SOAP server is installed, change to the *install_dir*\cms directory.
  2. Create a text file named SOAPREQ.txtand type 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>
     ```
  3. Create another text file named URLS.txt containing the URLs to receive the SOAP request. In this example, affiliatecompanylocalhost is the name of the receiving system and where the hub monitoring server is installed: http://affiliatecompanylocalhost:1920///cms/soap
  4. At the command line, enter **kshsoap SOAPREQ.txt URLS.txt**
- Linux UNIX Run the kshsoap script located in the *install_dir*/*interp*/ms/bin directory. The monitoring server configuration settings must be incorporated into your current shell before you invoke the kshsoap client. Incorporate the settings by entering the following command: . *install_dir*/config/*hostname*_ms_*temsname*.config. To verify this step, you can use the ***env*** command to show your environment variables and compare the entries to those in the .config file.
- When running the **kshsoap** command on systems that have APPN installed, you might encounter an error message stating that an APPN file needs to be configured. To resolve this situation, modify the environment variable **KDE_WAPPC32** from the command line window that you are going to run the **kshsoap** command in:

  ```
  SET KDE_WAPPC32=none
  ```

Results

The kshsoap utility processes the SOAPREQ file and displays the URL destination and request. It sends the SOAP request to each URL listed in the URLS file, then displays the URL and the response message received.

Issuing SOAP requests as system commands

You can use the Take Action feature in the Tivoli Enterprise Portal to issue SOAP requests as system commands in policies or in situations.

The SOAP requests are stored in a text file. For details, see the Specifying an action and Action Settings topics in the *Tivoli Enterprise Portal User's Guide*.

The soap command is:

```
soap:CT_Execute,filename=SOAPREQ
```

where:

CT_Execute is the name of the SOAP method that allows you to run a SOAP request that is stored in a file.

SOAPREQ is the name of the file you created that contains the CT_EMail SOAP request

For example, SOAPREQ might contain:

```
<CT_EMail><server>n-smtpmta</server>
<sender>soap@ibm.com</sender>
<receiver>jane_brown@ibm.com</receiver>
<subject>AFDATA untouched by human hands</subject>
<attachmenttitle>AFData.htm</attachmenttitle>
<request><attach>res.pfx</attach></request>
<request id="XMLID">
<CT_Redirect endpoint="http://sp22.ibm.com:18882">
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
  "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" >
<SOAP-ENV:Body><AF_Execute><Exec>SOAP0002</Exec></AF_Execute></SOAP-ENV:Body>
  </SOAP-ENV:Envelope></CT_Redirect></request>
<request><attach>res.sfx</attach></request></CT_EMail>
```

SOAP methods

Use the predefined SOAP methods to compose requests for invocation by PERL, Javascript, VBSCRIPT, JSCRIPT, C++, Java, and through a browser. With each method is a list of supported tags and usage examples. Each SOAP method provided by IBM and its supported tags is described here.

CT_Acknowledge

Send an event acknowledgement into the IBM Tivoli Monitoring platform.

<name>
The name of the situation. This is required.

<source>
The source of the event (agent name or monitoring server name). The acknowledge applies to all the active sources of the named alert if the source is not supplied.

<data>
"No data was provided" is inserted if not provided.

<item>
Display item.

<userid>
Optional. The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
Optional. The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<type>
Optional. Specifies the event type. The value can be "sampled" or "0", "pure" or "1", and "meta" or "2". Default: “**sampled**”

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

<expire>
Optional. Expires the acknowledgement after the number of minutes entered here.

Example:

```
<CT_Acknowledge>
  <hub>z/OSPROD</hub>
  <name>situation_from_CT</name>
  <source>CT_supported_system</source>
  <data>Jack is taking care of this failure</data>
  <item>subsystem</item>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
  <type>pure</type>
  <expire>60</expire>
</CT_Acknowledge>
```

CT_Activate

Start a situation or a policy running on the IBM Tivoli Monitoring platform.

Note: Situations for agents connecting to a remote Tivoli Enterprise Monitoring Server cannot be started using this method.

<name>
The name of the situation. This is required.

<type>
The type of object being activated. This is required.

<userid>
Optional. The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

Example:

```
<CT_Activate>
  <hub>z/OSPROD</hub>
  <name>name_of_situation_or_policy</name>
  <type> situation</type>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
</CT_Activate>
```

CT_Alert

Send an event into the IBM Tivoli Monitoring platform.

<name>
The name of the situation. This is required.

<source>
The source of the event (agent name or monitoring server name). This is required

<data>
"No data was provided" is inserted if not provided or if no optional object.attribute tag provided..

<item>
Display item.

<userid>
Optional. The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
Optional. The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<type>
Optional. Specifies the event type. The value can be "sampled" or "0", "pure" or "1", and "meta" or "2". Default: "**sampled**"

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

<data><object.attribute>
Returns the value of the attribute (or attributes) specified to the Initial Attributes view of the Event results workspace.

Example:

```
<CT_Alert>
  <hub>z/OSPROD</hub>
  <name>situation_from_XXX</name>
  <source>XXX_supported_system</source>
  <data><NT_Logical_Disk.Disk_Name>
C:</NT_Logical_Disk.Disk_Name></data>
  <item>subsystem</item>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
</CT_Alert>
```

Note: When you specify object.attribute in the data tag, leave out any non-alphanumber characters other than the underscore (_). For example, NT_System.%_Total_Processor_Time is entered as `NT_System.Total_Processor_Time`.

CT_Deactivate

Stop a situation or policy on the IBM Tivoli Monitoring platform.

Note: Situations for agents connecting to a remote Tivoli Enterprise Monitoring Server cannot be stopped with this method.

<name>
The name of the situation or policy. This is required.

<type>
The type of object (situation or policy). This is required.

<userid>
The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

Example:

```
<CT_Deactivate>
  <hub>z/OSPROD</hub>
  <name>name_of_situation_or_policy</name>
  <type>situation</type>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
</CT_Deactivate>
```

CT_EMail

Send the output from another CT SOAP method, such as CT_Get, using e-mail through an SMTP server to a defined e-mail address (not available on z/OS).

<server>
The smtp server name/network address is required.

<sender>
Sender's e-mail address is required.

<receiver>
Receiver's e-mail address is required.

<subject>
Optional. E-mail subject.

<message>
Optional. E-mail message.

<attachmenttitle>
Optional. Title of an attachment.

<request>
When specifying a second-level request, such as CT_Get, each sub-request must be included within a <request> </request> tag.

Optional: An id=" " element added to the <request> tag generates a <request id="XMLID"> element enclosing the corresponding response for that sub-request.

If additional security is enabled (SOAP_IS_SECURE=YES in the monitoring server environment variables) the following tags are also required:

<userid>
The user ID to access the hub monitoring server.

<password>
The password to access the hub monitoring server. Required for monitoring server logon validation.

Example:

```
<CT_EMail>
  <server>smtp.server</server>
  <sender>myemail@something.com </sender>
  <receiver>youremail@whatever.com </receiver>
  <subject>Here's your data.</subject>
  <message>Table data supplied as attachment below. It is
  presented in csv format to be used by MS/Excel.</message>
  <attachmenttitle>tabledata.csv</attachmenttitle>
  <request id="XMLID">
    <CT_Get>
      <object>NT_Process </object>
      <target>TlPrimary:DCSQLSERVER:NT</target>
      <userid>sysadmin</userid>
      <password>xxxxxxxx</password>
    </CT_Get>
  </request>
</CT_EMail>
```

Example with additional security:

```
<CT_EMail>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
  <server>smtp.server</server>
  <sender>myemail@something.com </sender>
  <receiver>youremail@whatever.com </receiver>
  <subject>Here's your data.</subject>
  <message>Table data supplied as attachment below. It is
  presented in csv format to be used by MS/Excel.</message>
  <attachmenttitle>tabledata.csv</attachmenttitle>
  <request id="XMLID">
    <CT_Get>
   <userid>sysadmin</userid>
   <password>xxxxxxxx</password>
       <object>NT_Process </object>
       <target>TlPrimary:DCSQLSERVER:NT</target>
    </CT_Get>
  </request>
</CT_EMail>
```

With the additional security, the user ID and password are requested by CT_EMail in order to be authorized. If a CT_Get is specified the same credentials are used to issue the CT_Get.

CT_Execute

Runs the SOAP request that is stored in a file.

<filename>
: Specifies the file name that contains the SOAP request to be run. The file must reside in the \html directory. On z/OS, it must reside in RKANDATV. This is required.

Example:

```
<CT_Execute>
  <filename>execute1.xml</filename>
</CT_Execute>
```

CT_Export

Send the output from another CT SOAP method, such as CT_Get, to a defined file (not available on z/OS).

<filename>
: The name of the file to contain the exported data. This is required.

Note: When inserting the file name tag into a quoted string literal of certain programming languages, such as C++, back slashes must be doubled.

To the <filename> tag, you can add an optional date/time stamp variable. The variable is enclosed in dollar signs ($) and can contain a combination of yy/mm/dd/hh/mm/ss (for year/month/day/hours/minutes/seconds).

The date/time stamp attributes can be specified in any order, except mm must be preceded by yy or hh to identify it as either month (after year) or minutes (after hours). For example:

```
<filename>g:\exchange\excel\ntprocess$yymmdd$.htm</filename>
```

<warehouse/>
Specifies that data is to be exported to the Tivoli Enterprise Portal data warehouse through ODBC. <filename> and <warehouse/> are mutually exclusive, but one must be supplied.

<request>
When specifying a second-level request, such as CT_Get, each sub-request must be included within a <request> </request> tag.

Optional: An id=" " element added to the <request> tag generates a <request id="XMLID"> element enclosing the corresponding response for that sub-request.

If additional security is enabled (SOAP_IS_SECURE=YES in the monitoring server environment variables) the following tags are also required:

<userid>
The user ID to access the hub monitoring server.

<password>
The password to access the hub monitoring server. Required for monitoring server logon validation.

Example:

```
<CT_Export>
  <filename>g:\exchange\excel\ntprocess$yymmddhhmmss$.htm</filename>
  <request>
    <attach>prefix.xsl</attach>
  </request>
  <request id="XMLID">
    <CT_Get>
      <object>NT_Process</object>
      <target>Primary:DCSQLSE RVER:NT</target>
      <userid>sysadmin</userid>
      <password>xxxxxxxx</password>
    </CT_Get>
  </request>
  <request>
    <attach>suffix.xsl</attach>
  </request>
</CT_Export>
```

Example with additional security:

```
<CT_Export>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
  <filename>g:\exchange\excel\ntprocess$yymmddhhmmss$.htm</filename>
  <request>
    <attach>prefix.xsl</attach>
  </request>
  <request id="XMLID">
    <CT_Get>
   <userid>sysadmin</userid>
   <password>xxxxxxxx</password>
       <object>NT_Process</object>
       <target>Primary:DCSQLSE RVER:NT</target>
    </CT_Get>
  </request>
  <request>
    <attach>suffix.xsl</attach>
  </request>
</CT_Export>
```

With the additional security, the user ID and password are requested by CT_Export in order to be authorized. If a CT_Get is specified the same credentials are used to issue the CT_Get.

CT_Get

Receive a group of XML objects or individual XML objects from any IBM Tivoli Monitoring platform agent. You can use this to obtain real time data.

Important: When issuing a CT_Get request against a particular agent type, the monitoring server where the SOAP server is running must be configured and seeded for that agent type.

<object>
The name of the object to be retrieved. Required (by default, retrieves all the public elements of an object).

<userid>
The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<target>
Name of the agent.

Caution: Defaults to "*ALL". Retrieves all available targets.

<history>
Y retrieves historical data if available.

<results>
PARSE retrieves status history event attributes. Only valid for Status_History object.Multiple: more than one can be specified.

<attribute>
Attribute name of object. This tag can be specified multiple times.

<hub>
Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

<afilter>
Returns rows meeting filter criteria, such as attribute; operator; value operators: EQ, NE, GE, GT, LE, LT, LIKE. Like pattern characters: '*' matches one or more characters. Only supported for character attributes. Multiple afilters are only supported as conjuncts, for example, using AND to join together.

Example:

```
<CT_Get>
  <hub>z/OSPROD</hub>
  <object>NT_System</object>
  <target>Primary:DCSQLSERVER:NT</target>
  <userid>sysadmin</userid>
  <password></password>
  <history>Y</history>
  <attribute>Server_Name</attribute>
  <attribute>Processor_Queue_Length</attribute>
  <afilter>Write_Time;GT;1020804</afilter>
  <afilter>Write_Time;LT;1020805</afilter>
</CT_Get>
```

Note: When you specify an attribute in the attribute tags, leave out any non-alphanumeric characters other than the underscore (_). For example, %_Total_User_Time is entered as `Total_User_Time`.

CT_Redirect

Reroute a SOAP request to another registered SOAP method outside of the domain of the IBM Tivoli Monitoring platform.

<request endpoint=" ">
The <request endpoint= " "> value must specify the target of the redirected SOAP request. The entire XML supplied as the value of the request element is sent to that endpoint. When CT_Redirect is specified within a second- level request, such as, CT_Export, the <endpoint=" "> attribute is specified *only* within the CT_Redirect method. This is required.

Example:

```
<CT_Redirect>
  <request endpoint= \"http://services.xmethods.net:80/soap/servlet/rpcrouter\">
    <SOAP-ENV:Envelope xmlns:SOAP-ENV=\"http://schemas.xmlsoap.org/soap/envelope/\">
      <SOAP-ENV:Body>
        <ns1:getTemp xmlns:ns1=\"urn:xmethods-Temperature\"SOAP-ENV:
        encodingStyle=\"http://schemas.xmlsoap.org/soap/encoding/\">
          <zipcode>93117</zipcode>
        </ns1:getTemp>
      </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
  </request>
</CT_Redirect>
```

CT_Reset

Send an event reset (close event) to the IBM Tivoli Monitoring platform.

<name>
The name of the situation. This is required.

<source>
The source of the event (agent name or monitoring server name). The reset applies to all the active sources of the named alert if the source is not supplied.

<item>
Display item.

<userid>
Optional. The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
Optional. The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

<type>
Optional. Specifies the event type. The value can be "sampled" or "0", "pure" or "1", and "meta" or "2". Default: “**sampled**”

Example:

```
<CT_Reset>
  <hub>z/OSPROD</hub>
  <name>situation_from_CT</name>
  <source>CT_supported_system</source>
  <item>subsystem</item>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
</CT_Reset>
```

Note: Sampled events can be closed only if the situation has been stopped or deleted. Use the <type> tag if CT_Reset will be closing a pure event.

CT_Resurface

Resurface an acknowledged event in the IBM Tivoli Monitoring platform.

<name>
The name of the situation. This is required.

<source>
The source of the event (agent name or monitoring server name). The resurface applies to all the active sources of the named alert if the source is not supplied.

<item>
Display item.

<userid>
Optional. The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
Optional. The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

<type>
Optional. Specifies the event type. The value can be "sampled" or "0", "pure" or "1", and "meta" or "2". Default: “**sampled**”

Example:

```
<CT_Resurface>
  <hub>z/OSPROD</hub>
  <name>situation_from_CT</name>
  <source>CT_supported_system</source>
  <item>subsystem</item>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
</CT_Resurface>
```

CT_WTO

Send a Universal Message into the IBM Tivoli Monitoring Platform.

<data>
The message to be sent. This is required.

<category>
Optional. Blank is the default.

<severity>
Optional. Blank is the default.

<userid>
Optional. The user ID to access the hub monitoring server. "nnn.nnn.nnn.nnn" is inserted if not provided.

<password>
Optional. The password to access the hub monitoring server. Required for monitoring server/hub logon validation.

<hub>
Optional. Specifies the alias name of a remote hub that has been configured in the hubs list. The SOAP request is routed to this hub.

Example:

```
<CT_WTO>
  <hub>z/OSPROD</hub>
  <data>This is Universal Message</data>
  <category>Critical Messages</category>
  <severity>High Severity</severity>
  <userid>sysadmin</userid>
  <password>xxxxxxxx</password>
</CT_WTO>
```

Issuing second-level SOAP requests

Some second-level SOAP methods perform a particular function with the data retrieved, using embedded lower-level methods. CT_EMail and CT_Export are second-level methods that perform this function.

These are the lower-level methods:

- <CT_Get>
- <CT_Redirect>
- <attach>
- <insert>

The **<CT_Get>** and **<CT_Redirect>** tags are used as described in "SOAP methods" on page 565. The **<attach>** tag is used to load a file. The file must be located in the <*install_dir*>\cms\html directory on Windows or <*install_dir*>/tables/HUB_Name/HTML on Linux and UNIX. You cannot use relative paths in the <attach> tag. The **<insert>** tag allows you to load the imbedded text into the retrieved (output) data stream at a point corresponding to its position in the XML request.

The following example shows how a second-level request might be used. This XML creates the file tabledata.htm, which is written with the data from prefix.xls. Next, embedded data is entered by using the **<insert>** tag and a request using the **<CT_Get>** command is made. Note that this request has an ID value of "NTDATA", which will result in the data tag **<XML id="NTDATA">**

being wrapped around that particular request data. The **<CT_Redirect>** command is used to reroute the request to `http://services.xmethods.net:80/soap/servlet/rpcrouter`, and a final request is made to insert the data from `suffix.xls` into `tabledata.htm`.

```
<CT_Export>
  <filename>tabledata.htm</filename>
  <request>
    <attach>prefix.xls</attach>
  </request>
  <request>
    <insert>
    <insertelement>
      <insertdata>
        This data has been inserted compliments of CT SOAP server.
      </insertdata>
    </insertelement>
    </insert>
  </request>
  <request id="NTDATA">
    <CT_Get>
      <userid>sysadmin</userid>
      <password></password>
      <object>NT_System</object>
      <target>*ALL</target>
    </CT_Get>
  </request>
  <request>
    <CT_Redirect endpoint="http://services.xmethods.net:80/soap/servlet/rpcrouter">
      <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Body>
          <ns1:getTemp xmlns:ns1="urn:xmethods-Temperature" SOAP-
          ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <zipcode>93117</zipcode>
          </ns1:getTemp>
        </SOAP-ENV:Body>
      </SOAP-ENV:Envelope>
    </CT_Redirect>
  </request>
  <request>
    <attach>suffix.xls</attach>
  </request>
</CT_Export>
```

Sample CT_Get SOAP request

Here is a sample **CT_Get** SOAP request submitted and the response received.

SOAP Request sent to SOAP Endpoint, http://esada.ibm.com:19221/SOAP

```
<?xml.version="1.0" encoding="UTF-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
   <CT_Get>
    <Object>NT_System</Object>
    <Source>Primary:ESADA:NT</Source>
   </CT_Get>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response from SOAP Endpoint, http://esada.ibm.com:19221/SOAP

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
 <SOAP-CHK:Success xmlns:SOAP-CHK = "http://soaptest1/soaptest/">
 <PARMS> </PARMS>
 <TABLE name="KNT.WTSYSTEM">
  <OBJECT>NT_System</OBJECT>
  <DATA>
   <ROW>
     <Server_Name>Primary:ESADA:NT</Server_Name>
     <Timestamp >1011127123323391</Timestamp>
     <User_Name>SYSTEM</User_Name>
     <Operating_System_Type>Windows_NT</Operating_System_Type>
     <Operating_System_Version>4.0</Operating_System_Version>
     <Network_Address>10.21.2.154</Network_Address>
     <Number_of_Processors  dt:dt="number">1</Number_of_Processors>
     <Processor_Type dt:dt="number">586</Processor_Type>
     <Page_Size dt:dt="number">4096</Page_Size>
     <_Total_Privileged_Time dt:dt="number">1</_Total_Privileged_Time>
     <_Total_Processor_Time dt:dt="number">7</_Total_Processor_Time>
     <_Total_User_Time dt:dt="number">6</_Total_User_Time>
     <Context_Switches_Sec dt:dt="number">1745</Context_Switches_Sec>
     <File_Control_Bytes_Sec dt:dt="number">4500</File_Control_Bytes_Sec>
     <File_Control_Operations_Sec dt:dt="number">98
      </File_Control_Operations_Sec>
     <File_Data_Operations_Sec dt:dt="number">28
      </File_Data_Operations_Sec>
     <File_Read_Bytes_Sec dt:dt="number">800</File_Read_Bytes_Sec>
     <File_Read_Operations_Sec dt:dt="number">27
      </File_Read_Operations_Sec>
     <File_Write_Bytes_Sec dt:dt="number">9772</File_Write_Bytes_Sec>
     <File_Write_Operations_Sec dt:dt="number">1
      </File_Write_Operations_Sec>
     <Processor_Queue_Length dt:dt="number">0</Processor_Queue_Length>
     <System_Calls_Sec dt:dt="number">2368</System_Calls_Sec>
     <System_Up_Time dt:dt="number">956388</System_Up_Time>
     <Total_Interrupts_Sec dt:dt="number">1076</Total_Interrupts_Sec>
    </ROW>
   </DATA>
  </TABLE>
 </SOAP-CHK:Success>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

IBM Tivoli Monitoring web services scenarios

Here are a few examples of how you might use IBM Tivoli Monitoring web services. You can use these examples as suggestions for creating your own applications.

Note: These scenarios do not describe the actual code that was used to develop them. To produce the charts and tables shown in these examples, you must develop your own scripts.

Generating daily logical operation summaries and charts

You can retrieve data from multiple agents, using the SOAP server against a live hub, to generate daily logical operation summaries. You can use the **CT_EMail** SOAP method to e-mail these summaries to management.

You might want to add an **<insert>** tag into **CT_EMail**. This tag contains instructions for the preferred format for the summaries. Management can view these summaries at their desktops using Internet Explorer. Summaries provide an efficient and speedy look at problems that might have occurred during the night.

In addition to the general features, you might add to tables and charts:

- Transaction volumes/response times and whether they are meeting service levels can be plotted with respect to resource trends and error conditions.
- Charts can be plotted over multiple segments, making them easier to view and to print.
- The X-axis can use a variable scale to show the prime shift in greater detail.
- Multiple objects/attributes can be plotted from multiple sources and exceptions can be correlated by time, providing focus on problem areas.
- A Status map can show the status of situations.

Obtaining data snapshots and offline table and charts

Using SOAP method **CT_Get** against a live hub, you can obtain a data snapshot from multiple agents to produce charts and reports. You can also create an AF REXX script that requests a snapshot of its data.

In addition to the general features you might add to tables and charts. This type of request might contain these features:

- The chart can be plotted over multiple segments, making it easier to view and print.
- Clicking the attribute name in the legend box might display that attribute in the Y-axis and show its threshold value.
- The threshold value, when changed, can be used as the new threshold value.

The graphics that follow depict sample Daily Business Operation summaries.

The graphics that follow show sample charts/reports generated for this type of request.

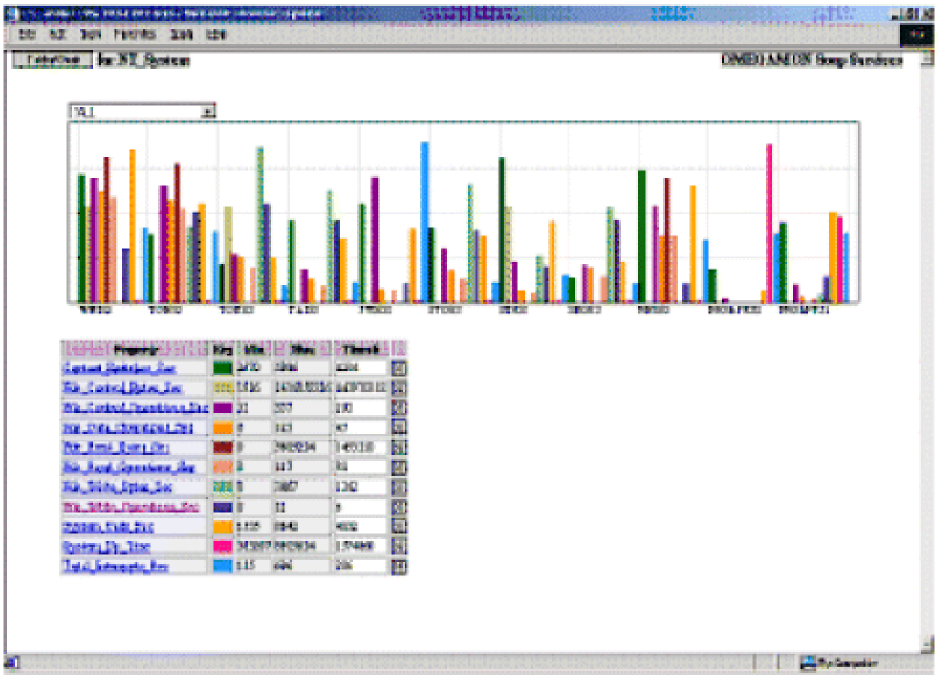
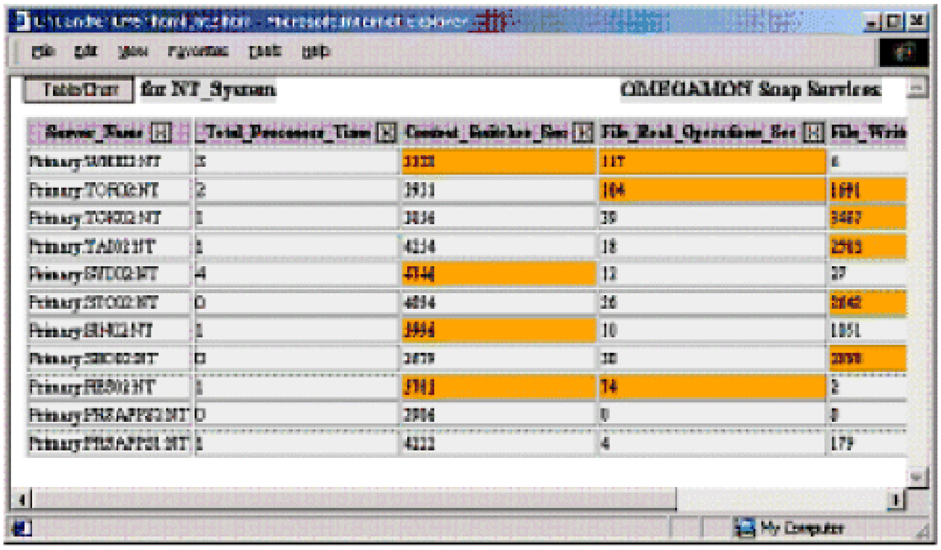


Figure 33. Data snapshot chart and table



Table/Chart for NT_System OMEGAMON Soap Services

Server_Name	_Total_Processor_Time	Context_Switches_Sec	File_Read_Operations_Sec	File_Writ
Primary:[illegible]:NT	3	3333	117	6
Primary:TOR02:NT	2	3931	104	1691
Primary:TOK02:NT	1	3836	39	3467
Primary:TAI02:NT	1	4234	18	2583
Primary:SYD02:NT	4	4344	13	27
Primary:STO02:NT	0	4694	26	3642
Primary:SIN02:NT	1	3994	10	1851
Primary:SEO02:NT	0	3679	38	3898
Primary:RES02:NT	1	3781	74	2
Primary:PRSAPPS2:NT	0	2906	0	8
Primary:PRSAPPS1:NT	1	4222	4	179

Figure 34. Data snapshot table

Sending alerts into an IBM Tivoli Monitoring platform

Using SOAP method **CT_Alert**, you can send a new alert into an IBM Tivoli Monitoring platform.

For example, System Automation for Integrated Operations Management detects a problem on a HP NonStop Kernel system and generates an alert within an IBM Tivoli Monitoring platform. The IBM Tivoli Monitoring platform then displays alert information from the HP NonStop Kernel platform.

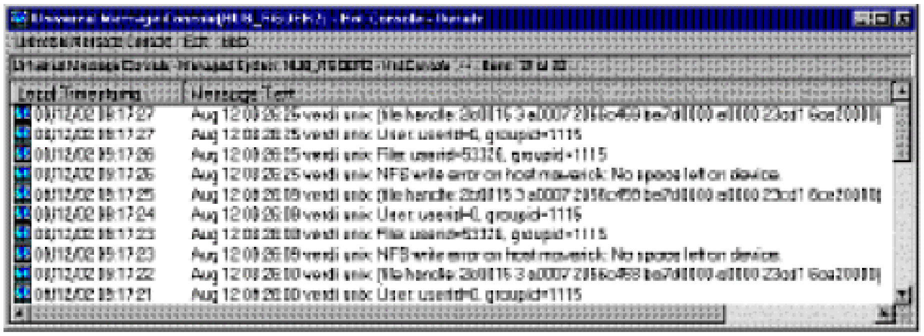
Creating collaborative automation using SA IO

You can create a System Automation for Integrated Operations Management REXX application that calls JSCRIPT SOAP functions to forward any SA IO trapped message and display it on a Universal Message console. You can use SA IO scripts to trap and send any log messages, console messages, and so on, to IBM Tivoli Monitoring using SOAP methods.

You can create an application that provides these benefits:

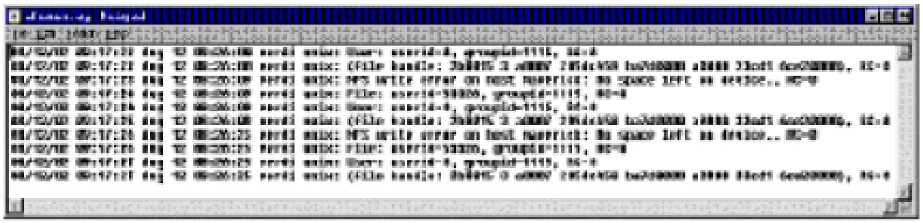
- You can monitor devices, such as HP NonStop Kernel, by trapping VT100 messages and raising Universal Messages.
- You can send commands to SA IO monitored Telnet sessions and send replies back to those commands.
- Source messages can be either excluded or included, based on any criteria using powerful regular expressions.
- A local log can keep audit information about the status of messages received and messages sent.
- A local log can keep information about the source hub connection/retry status.

The graphics that follow show a sample Telnet session, a Universal Message console showing messages received, and a sample message log.



Local Timestamp	Message Text
08/12/02 09:17:27	Aug 12 08:26:25 verdi unix: [file handle: 2c0115 3 a0007 2f56c459 be7d0000 a0000 23cd1 6ce20000]
08/12/02 09:17:27	Aug 12 08:26:25 verdi unix: User: userid=0, groupid=1115
08/12/02 09:17:26	Aug 12 08:26:25 verdi unix: File: userid=53326, groupid=1115
08/12/02 09:17:26	Aug 12 08:26:25 verdi unix: NFS write error on host maverick: No space left on device.
08/12/02 09:17:25	Aug 12 08:26:09 verdi unix: [file handle: 2c0115 3 a0007 2f56c458 be7d0000 a0000 23cd1 6ce20000]
08/12/02 09:17:24	Aug 12 08:26:09 verdi unix: User: userid=0, groupid=1115
08/12/02 09:17:23	Aug 12 08:26:09 verdi unix: File: userid=53326, groupid=1115
08/12/02 09:17:23	Aug 12 08:26:09 verdi unix: NFS write error on host maverick: No space left on device.
08/12/02 09:17:22	Aug 12 08:26:00 verdi unix: [file handle: 2c0115 3 a0007 2f56c458 be7d0000 a0000 23cd1 6ce20000]
08/12/02 09:17:21	Aug 12 08:26:00 verdi unix: User: userid=0, groupid=1115

Figure 35. Universal Message Console Showing Messages Received



```
08/12/02 09:17:21 Aug 12 08:26:00 verdi unix: User: userid=0, groupid=1115, Id=0
08/12/02 09:17:22 Aug 12 08:26:00 verdi unix: (file handle: 2c0115 3 a0007 2f56c458 be7d0000 a0000 23cd1 6ce20000), Id=0
08/12/02 09:17:23 Aug 12 08:26:09 verdi unix: NFS write error on host maverick: No space left on device., Id=0
08/12/02 09:17:23 Aug 12 08:26:09 verdi unix: File: userid=53326, groupid=1115, Id=0
08/12/02 09:17:24 Aug 12 08:26:09 verdi unix: User: userid=0, groupid=1115, Id=0
08/12/02 09:17:25 Aug 12 08:26:09 verdi unix: (file handle: 2c0115 3 a0007 2f56c458 be7d0000 a0000 23cd1 6ce20000), Id=0
08/12/02 09:17:26 Aug 12 08:26:25 verdi unix: NFS write error on host maverick: No space left on device., Id=0
08/12/02 09:17:26 Aug 12 08:26:25 verdi unix: File: userid=53326, groupid=1115, Id=0
08/12/02 09:17:27 Aug 12 08:26:25 verdi unix: User: userid=0, groupid=1115, Id=0
08/12/02 09:17:27 Aug 12 08:26:25 verdi unix: (file handle: 2c0115 3 a0007 2f56c459 be7d0000 a0000 23cd1 6ce20000), Id=0
```

Figure 36. Message Log Details

Acknowledging an event within an IBM Tivoli Monitoring platform

You can acknowledge an event within the IBM Tivoli Monitoring platform.

For example, in AF/Operator or System Automation for z/OS V3.2 or later:

1. A situation event is received from the hub Tivoli Enterprise Monitoring Server
2. A responsible party is paged who, in turn, sends back an acknowledgment
3. The alert acknowledgment is forwarded to the monitoring server

To accomplish this task, use the **CT_Acknowledge** SOAP method. This method enables you to control events in the IBM Tivoli Monitoring environment based upon information obtained and detected by IBM's automation solutions.

Report contents

You can design a report to contain both a table and a chart view. You might want to add a **Table/Chart** button that allows you to toggle between the chart and the table view.

Chart view features

Charts can have specific features to enable you to:

- View different types of charts, depending upon the data retrieved
- Choose the Y-axis by selecting additional attributes from the drop-down attribute list
- Change the title and instructions for the chart
- View the flyover text containing the name and value of the attribute plotted by placing your mouse over each plotted item

Table view features

Tables can have specific features. For example, you can design tables that allow you to:

- View the flyover text containing the name and value of the attribute plotted by placing your mouse over each plotted item
- Modify the table by filtering the attributes that display
- Remove attributes from a table by clicking the X button next to the attribute name

Appendix B. Enabling the IBM Tivoli Monitoring Charting Web Service

If you have a product based on the Tivoli Integrated Portal V2.2 platform, you can use the charting feature to display charts with values queried from your Tivoli Monitoring environment. You must enable the IBM Tivoli Monitoring Charting Web Service (ITMWebService) in the Tivoli Enterprise Portal Server. The charting web service is also used with the Tivoli Business Service Manager policy-based data fetcher function.

Note: The IBM Tivoli Monitoring dashboard data provider is used instead of the charting web service when you display monitoring data in IBM Dashboard Application Services Hub.

Before you begin

Single sign-on must be enabled for users of the Tivoli Integrated Portal console and Tivoli Enterprise Portal Server. The users who display charts containing data from the charting web service must have a Tivoli Enterprise Portal user ID that is assigned the monitoring application whose data is displayed in the chart.

About this task

Complete these steps on the computer where the Tivoli Enterprise Portal Server is installed and running to enable the Tivoli Monitoring Charting Web Service.

Procedure

1. Copy the **kfwtipewas.properties** file to the portal server directory:

   Windows From *install_dir*`\CNPS\SQLLIB\` to *install_dir*`\CNPS`.

   Linux UNIX From *install_dir*`/`*platform*`/cq/sqllib/` to *install_dir*`/`*platform*`/cq/`.
2. Reconfigure the Tivoli Enterprise Portal Server.

What to do next

As a Tivoli Enterprise Portal user you are entitled to see certain workspaces that belong to a particular monitored application based on your permissions. If you are entitled to see, say, Linux workspaces, then those workspace queries are available in the Tivoli Integrated Portal for charting.

If you want to use HTTPS to secure communication between the charting web service, see “Tivoli Business Service Manager and Tivoli Enterprise Portal Server integration over SSL” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

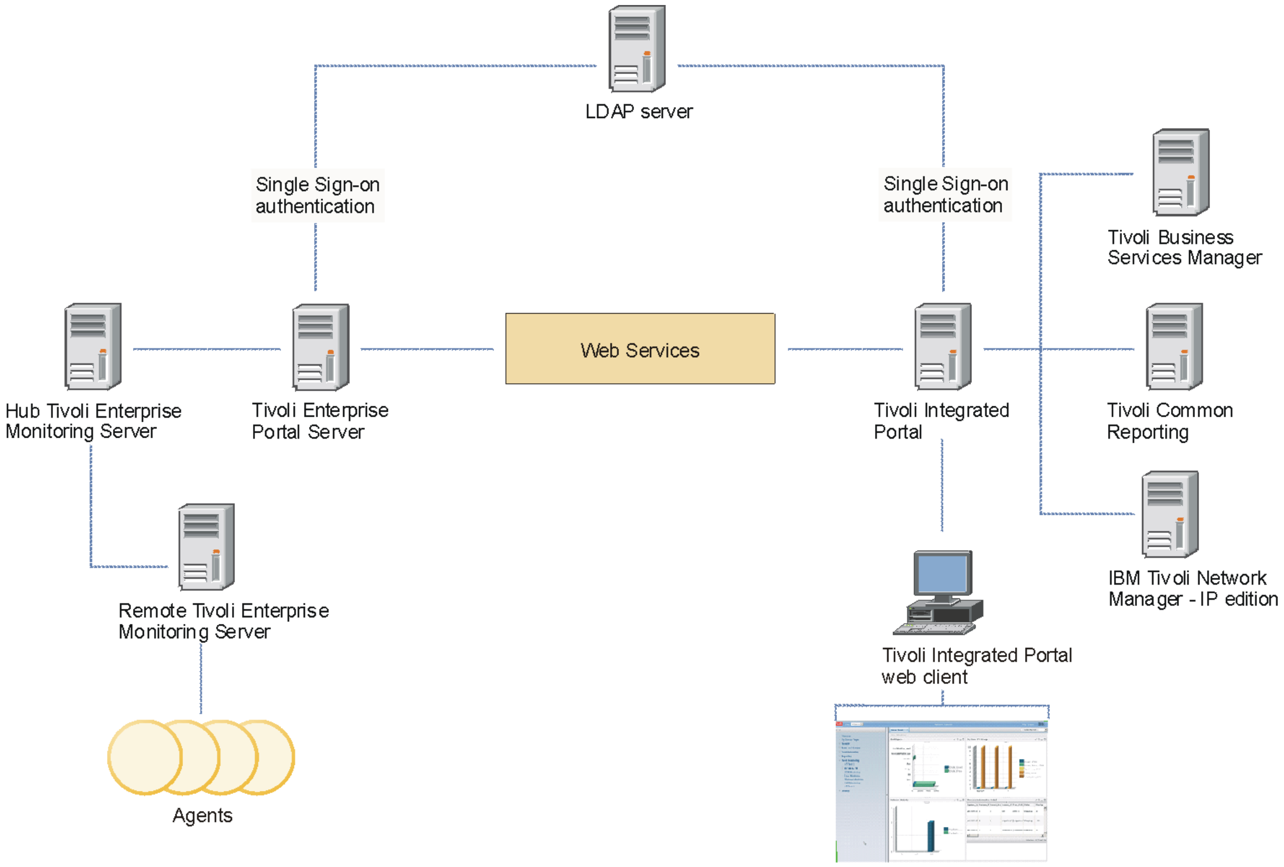

LDAP server
Single Sign-on authentication
Single Sign-on authentication
Tivoli Business Services Manager
Hub Tivoli Enterprise Monitoring Server
Tivoli Enterprise Portal Server
Web Services
Tivoli Integrated Portal
Tivoli Common Reporting
Remote Tivoli Enterprise Monitoring Server
IBM Tivoli Network Manager - IP edition
Tivoli Integrated Portal web client
Agents

Figure 37. Cross-product connections for the charting web service

Appendix C. Using the Tivoli Management Services Discovery Library Adapter

Use the Tivoli Management Services Discovery Library Adapter (TMS DLA) program for scanning your monitored environment to identify the managed systems. You can then feed this information (an XML output file) into the Change and Configuration Management Database (CCMDB), Tivoli Application Dependency Discovery Manager (TADDM), or Tivoli Business Service Manager (TBSM).

The TMS DLA identifies all distributed and z/OS managed systems registered to the Tivoli Management Services.

Before you begin

When the **tmsdla** script is launched, the TMS DLA gathers information by querying the hub monitoring server for all managed systems and mapping them to Common Data Model resources based on the agent product code and managed system name format. The queries specified in the XML input file provided by each product are run and the results saved to a single output file.

See the agent-specific user guide's to determine if the agent supplies an input XML file for the TMS DLA and possible mapping information between the agent's monitored resources and Common Data Model resources.

The monitoring servers and the Tivoli Enterprise Portal Server must be running for these queries. Also, any managed systems that are not online will be ignored.

About this task

Run the following TMS DLA script from the command line on the computer where the portal server is installed:

Procedure

- Windows To make a create-type IDML book, enter the following command:

  ```
  install_dir\CNPS\tmsdla.bat
  ```

  Alternatively, to make a refresh-type IDML book, enter the following command. After you import into TADDM, any systems that are offline (such as for maintenance operations) are removed from TADDM. The same is true for Tivoli Business Service Manager (TBSM).

  ```
  install_dir\CNPS\tmsdla.bat -r
  ```

- Linux UNIX To make a create-type IDML book, enter the following command:

  ```
  install_dir/bin/itmcmd execute cq "tmsdla.sh"
  ```

  Alternatively, to make a refresh-type IDML book, enter the following command. After you import into TADDM, any systems that are offline (such as for maintenance operations) are removed from TADDM. The same is true for TBSM.

  ```
  install_dir/bin/itmcmd execute cq "tmsdla.sh -r"
  ```

Results

The TMS DLA generates the XML output file to the same directory on the portal server. The name of this file follows the standard Discovery Library file name format. To use this information in CCMDB, TADDM, or TBSM you must transfer the XML file to the Discovery Library File Store and then use the Discovery Library Bulk Loader.

The TMS DLA also creates an output file with the `.xml.original` extension which contains the TMS DLA output before any relationships are removed. Removed relationships are written to `tmsdla.log`. See "OS agent dependency" on page 585 for examples of scenarios where relationships might be removed from the TMS DLA XML output file.

Usage

Use the following usage notes for your reference:

```
tmsdla [ [-? | -h] [-d directory] [-f outputFilename] [-l] [-m listOfManagedSystems]
        [-o] [-p portNumber] [-r] [-s] [-t threadCount] [-w waitTime] ]
```

where:

-? | -h Displays the syntax help information.

-d Specify the template directory location.

-f Specify the resulting output file name.

-l Discovers logical views.

-m Specify the list of managed systems.

The list is double quote delimited and follows this syntax:

```
"os_msys1, os_apptype1, [msys1, apptype1] ~ [os_msys2, os_apptype2,
[msys2, apptype2]] ~ .. ~ [os_msysN, os_apptypeN, [msysN, apptypeN]]"
```

-o Force processing of offline managed systems.

-p Specify the portal server's port number if it is not the default value of 1920. The port number is included in the output book and used by TADDM or TBSM to generate the URL to launch to the Tivoli Enterprise Portal.

-r Generate a refresh-type output XML file. After you import a refresh-type output file into TADDM, the objects for any managed systems that are offline, such as for maintenance operations, and their monitored resources are removed from the TADDM database. The same is true when you import a refresh-type output file into TBSM or CCMDB. If you do not specify this option, a create-type output XML file is generated that only contains online managed systems and the resources that they are monitoring. When you import a create-type output XML file into TADDM, TBSM, or CCMDB, managed systems and monitored resources are added or updated but no deletions occur.

-s Suppress generation of the `.original` file from cleanup process.

-t Specify the number of threads to use.

-w Specify the number of seconds to wait for query to be serviced by agent before timing out. Use this option if monitoring agents might not be able to service queries in a reasonable time due to a heavy load on the queried system. **Default value**: 120 seconds.

Minimum value: 50 seconds. Values lower than 50 are ignored and the default value is used.

Maximum value: 600 seconds.

Note: No return codes are provided on the completion of the book to alert you if there was a timeout or missing agent data. To determine if you need to set a higher value than the default, analyze the book to ensure that all agents have responded.

Related reference:

Tivoli Change and Configuration Management Database
Search for "Discovery Library File Store" and "Discovery Library Bulk Loader" in the CCMDB Information Center

Problems with the Tivoli Monitoring DLA
Solutions for common problems that occur with the Tivoli Monitoring DLA

Tivoli Monitoring Command Reference
The itmcmd commands, available only on Linux and UNIX Tivoli Enterprise Monitoring Servers, are described here

OS agent dependency

The Tivoli Management Services Discovery Library Adapter (TMS DLA) requires operating system (OS) agents to monitor the same systems as application agents that provide DLA templates.

Some application agents (such as the DB2 agent) rely on the OS agents to create the elements in the DLA book that describe the computer system and operating systems that the agents are running on. In their DLA templates, the application agents reference the computer system and operating system elements that will be created by the OS agents using a relationship element. For example, a Db2System runsOn a ComputerSystem, where runsOn is the relationship type, Db2System is the source element, and ComputerSystem is the target element.

If an OS agent is not monitoring the same system as an application agent or the OS agent and application agent are connected to different IBM Tivoli Monitoring environments then the source or target of a relationship element might not exist in a DLA book. The source and target of a relationship to a computer system or operating system might also not exist if:

- The Windows OS agent is monitoring a Windows 2000 system. In this case, the OS agent cannot discover the IP address of the Windows 2000 system and, therefore, does not create computer system elements in the DLA book.
- You are using an agentless OS agent to monitor a system that an application agent is also monitoring since agentless OS agents do not provide input to the DLA books.

If either the source or target of a relationship do not exist in the DLA book, the book will not load successfully into Tivoli Application Dependency Discovery Manager (TADDM) or Tivoli Business Service Manager (TBSM). Therefore, the DLA removes a relationship from an DLA book if the source or target of a relationship do not exist in the book. This allows the DLA books to load successfully. However, if a relationship is removed, the DLA book will not contain the information to map the affected resource (such as a database system) to a computer system or operating system in TADDM or TBSM.

The DLA creates two XML files each time the **`tmsdla`** command is run:

- A book with the `.xml` extension
- A book with the `.xml.original` extension

The file with the `.xml.original` extension contains the contents of the DLA book before any relationships are removed. Removed relationships are written in `tmsdla.log`.

If you want your resources to be mapped to computer systems and operating systems in the DLA books that are loaded into TADDM and TBSM, install an OS agent on the systems that are being monitored by your application agents.

Private network address filtering

In environments where private network addresses are not duplicated, you can change this behavior so that the Tivoli Management Services Discovery Library Adapter (TMS DLA) populates these computer systems.

Before you begin

The TMS DLA does not populate computer systems with data from private network interfaces configured according to Internet Engineering Task Force (IETF) RFC 1918 and IETF RFC 4193. For details about RFCs, see the RCF Index (http://tools.ietf.org/rfc/index). This behavior prevents the incorrect merging of computer systems when multiple private networks use overlapping address ranges.

About this task

To enable discovery of computer systems with private network interfaces, edit the IP address filters in the XML template files that control the TMS DLA behavior.

Procedure

1. Back up the template files before editing.
   - Linux UNIX The template files are stored in `$ITM_HOME/`*`arch`*`/cq/tmsdla` on the Tivoli Enterprise Portal Server.
   - Windows The template files are stored in `%ITM_HOME%\CNPS\tmsdla` on the Tivoli Enterprise Portal Server.
2. A template for each monitoring agent provides discovery data.
   a. Check each of the template files to determine whether it has one or more `<tmsdla:filter>` sections. For example, the template file names for the operating system agents are:

      `knt_tmsdla.xml` for the Windows OS agent

      `kux_tmsdla.xml` for the UNIX OS agent

      `klz_tmsdla.xml` for the Linux OS agent

   b. Update the multiple `<tmsdla:filter>` sections in each of the template files to contain only the filters for loopback addresses (127.0.0.1 for IPv4 and ::1 for IPv6), as shown in the following example:

```
<tmsdla:filters>
    <tmsdla:filter name="IF_IP_ADDR" exclude="127\.0\.0\.1"/>
    <tmsdla:filter name="IF_IP_ADDR" exclude="::1"/>
</tmsdla:filters>
```

What to do next

When an agent's application support is updated on the Tivoli Enterprise Portal Server, the agent's current DLA template file with your modifications is renamed to use the `.bak` extension and the latest version of the template file is installed. After the application support installation is complete, you must update the new version of the agent's DLA template file to contain the edits that you made to the IP address filtering in the `*.bak` version of the template file.

Appendix D. Using the z/OS Tivoli Management Services Discovery Library Adapter

The z/OS Tivoli Management Services Discovery Library Adapter (zTMS DLA), available as of V6.2.2 Fix Pack 7 or later and IBM Tivoli Monitoring V6.2.3 Fix Pack 1 or later, scans the IBM Tivoli Monitoring environment and discovers resources monitored by the OMEGAMON agents.

The zTMS DLA identifies resources only on z/OS operating systems. To discover resources on distributed systems, you must also run the Tivoli Monitoring Services Discovery Library Adapter.

Before you begin

The data in the IDML book generated by the zTMS DLA can be used to supplement data gathered by the z/OS DLA, enabling context sensitive launching into the Tivoli Enterprise Portal from z/OS objects or events in Tivoli Business Service Manager (TBSM) or Tivoli Application Dependency Discovery Manager (TADDM).

The zTMS DLA creates the following Common Data Model (CDM) objects:

- sys.zOS.Sysplex
- sys.zOS.SysplexGroup
- sys.zOS.ZSeriesComputerSystem
- sys.zOS.ZOS
- sys.zOS.CICSRegion
- sys.zOS.DB2Subsystem
- sys.zOS.IMSSubsystem
- sys.zOS.MQSubsystem
- sys.zOS.AddressSpace

Important: To reconcile with the zTMS DLA book, the version of the z/OS DLA required is V3.1, with PTF UA61720. The IDML book generated by the zTMS DLA does not contain all required attributes for the various z/OS CDM objects. Before importing the zTMS DLA book into a consuming application, you must first run the z/OS DLA on all the systems monitored by the IBM Tivoli Monitoring OMEGAMON agents and import the resulting z/OS DLA books into the consuming application. Then you can import the zTMS DLA book, and all objects will be reconciled with existing objects found in the z/OS DLA books. If the zTMS DLA book is imported without having imported the z/OS DLA books first, objects might be displayed in the TBSM UI as NO_LABEL_SUPPLIED.

User scenarios

1. On z/OS LPARS, execute the z/OS DLA on all z/OS LPARs of interest.
2. On the Tivoli Enterprise Portal Server, execute the zTMS DLA on portal servers with z/OS agents.
3. If you are using TBSM, complete the following steps:
   a. On the TBSM data server, import z/OS DLA books into TBSM by using the TBSM discovery library toolkit.

   b. On the TBSM data server, import the zTMS DLA book into TBSM by using the TBSM discovery library toolkit.

   c. On the TBSM server graphical user interface, right-click on the TBSM object to display the IBM Tivoli Monitoring launch-in-context menu items.

4. If you are using TADDM, complete the following steps:

   a. On the TADDM server, import the z/OS DLA books into TADDM using the bulk load utility.

   b. On the TADDM server, import the zTMS DLA books into TADDM using the bulk load utility.

About this task

Run the following DLA script from the command line on the computer where the Tivoli Enterprise Portal Server is installed:

Procedure

- Windows To make an IDML book, enter the following command:

  ```
  install_dir\CNPS\ztmsdla.exe
  ```

  The output file is written to the *install_dir*\CNPS\tmsdla directory.

- Linux UNIX To make an IDML book, enter the following command:

  ```
  install_dir/bin/itmcmd execute cq "ztmsdla"
  ```

  The output file is written to the *install_dir*/*arch*/cq/bin/tmsdla directory.

Results

The DLA generates the XML output file in the directory identified above. The name of this file begins with the string ZTMSDISC100-B and follows the standard Discovery Library file name format, for example: ZTMSDISC100-B.<*hostname*>.<*timestamp*>.refresh.xml.

To import this book into TADDM or TBSM, you must transfer the XML file to the Discovery Library File Store and then use the Discovery Library Bulk Loader.

Usage

Use the following usage notes for your reference:

```
ztmsdla [/?] [/b] [/d] [/o orgname] [/s] [/p port] [/x outputfile]
```

where:

/? Displays the syntax help information.

/b Opens a browser to view the output file of the Discovery Library Adapter (on Windows only).

/d Creates a diagnostic file during the discovery process. You can use this file for debugging. The file is located in the same directory as the DLA IDML book. The file name is the same as the DLA IDML book, with an extension of .log at the end of the file name, for example: ZTMSDISC100-B.<*hostname*>.<*timestamp*>.refresh.log.

/o orgname
Sets the Organization GlobalName value. If this argument is not specified, the GlobalName defaults to <defaultOrg>.

/s When specified, the URL created for the sourceContactInfo attribute of the ManagementSoftwareSystem class is created by using the HTTPS protocol. This URL is used by TBSM and TADDM when performing a launch in-context to the Tivoli Enterprise Portal.

/p port
Set the port used for the URL created for the sourceContactInfo attribute. The default port is 1920 for HTTP. If the /s option is specified, the default port is 3661 for HTTPS. Use this option if the IBM Tivoli Monitoring Administrator has changed the default ports of the web server used to contact the portal server.

/x outputfile
Indicates the name of the XML output file.

Known limitations

Limitations exist when using SNA instead of IP.PIPE communication between a z/OS Tivoli Enterprise Monitoring Server and OMEGAMON agents. Two CICS® objects representing the same CICS Region are displayed in TBSM, but only one can be right-clicked to launch one of the TBSM CICS objects to the OMEGAMON CICS Tivoli Enterprise Portal workspace.

Appendix E. MIB SNMP agent event descriptions

Tivoli monitoring agents emit three types of SNMP alerts to convey agent operational status, sampled situation events, and pure situation events. The alert types are defined in the canbase.mib and cansyssg.mib files, which are available on the IBM Tivoli Monitoring- and IBM Tivoli Monitoring Agents installation media.

Agent situation state SNMP traps are sent using enterprise 1.3.6.1.4.1.1667.1.3 (Candle-BASE-MIB::candle-Alert-MIB).

agentStatusEvent

The agentStatusEvent is a monitoring agent operational status information trap generated by the Tivoli Autonomous Agent SNMP Event Exporter to inform and notify about a specific agent operational event.

Specific trap: 20

Access: read-only

Status: mandatory

Table 73. SNMP trap variables for agentStatusEvent

Variable	Description	OID
agentSit-Name	The situation name, up to 32 bytes, identifies the name and nature of the status event.	1.3.6.1.4.1.1667.1.2.1.10.1.3
agentSit-OriginNode	The name of the managed system where the situation was evaluated, up to 32 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.4

agentSit-LocalTimeStamp	The timestamp when the situation state changed. The format is CYYMMDDHHMMSSmmm (such as 1090415094501000 for April 15, 2009 at 09:45:01) where:
	C = Century (1 for 21st)
	Y = Year
	M = Month
	D = Day
	H = Hour
	M = Minute
	S = Second
	m = millisecond
		1.3.6.1.4.1.1667.1.2.1.10.1.5

autoSit-Category	Assigned situation category. Valid values are: 0 - Threshold 1 - Network Topology 2 - Error 3 - Status 4 - Node Configuration 5 - Application Alert 6 - All Category 7 - Log Only 8 - Map 9 - Ignore	1.3.6.1.4.1.1667.1.2.1.6

Table 73. SNMP trap variables for agentStatusEvent (continued)

Variable	Description	OID
autoSit-Severity	Assigned situation severity. Valid values are: 0 - Cleared 1 - Indeterminate 2 - Warning 3 - Minor 4 - Major 5 - Critical	1.3.6.1.4.1.1667.1.2.1.7
autoSit-StatusText	The agent status trap description message text, from 0 to 256 bytes.	1.3.6.1.4.1.1667.1.2.1.9
autoSit-Interval	The agent status trap interval; typically used for the heartbeat interval. See the "Sample trap configuration file" in "SNMP alert configuration" on page 363 for an example of setting the heartbeat interval: `<StatTrap name="EE_HEARTBEAT" sev="1" interval="15" cat="3" />`	1.3.6.1.4.1.1667.1.2.1.11

agentSitSampledEvent

A sampled situation event was detected. This trap was generated by the Tivoli Autonomous Agent SNMP Event Exporter in response to a situation threshold being exceeded at the time of the data sampling.

Specific trap: 21

Access: read-only

Status: mandatory

Table 74. SNMP trap variables for agentSitSampledEvent

Attribute	Description	OID
agentSit-Application	This is the product application name, from 1 to 8 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.1
agentSit-Table	This is the name of the product application table (attribute group), from 1 to 12 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.2
agentSit-Name	The situation name, up to 32 bytes, identifies the name and nature of the status event.	1.3.6.1.4.1.1667.1.2.1.10.1.3
agentSit-OriginNode	The name of the managed system where the situation was evaluated, up to 32 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.4

Table 74. SNMP trap variables for agentSitSampledEvent (continued)

Attribute	Description	OID

agentSit-LocalTimeStamp	The timestamp when the situation state changed. The format is CYYMMDDHHMMSSmmm (such as 1091031183005000 for October 31, 2009 at 18:30:05) where:
	C = Century (1 for 21st)
	Y = Year
	M = Month
	D = Day
	H = Hour
	M = Minute
	S = Second
	m = millisecond
		1.3.6.1.4.1.1667.1.2.1.10.1.5

agentSit-Context	Unique situation context identifier, expressed as an integer (-2147483647 to 2147483647). This is the handle number identifying an agent running request. In an SNMP environment, trap-direct polling is typically used whereby a trap is received and the network manager polls the originating agent for additional detailed information. This identifier is used to supply context for the targeted agent to correlate the request to the problem source. Although agentSit-Context is sent, it is not used in this release.	1.3.6.1.4.1.1667.1.2.1.10.1.6
agentSit-SampleInterval	Sampled situation interval in seconds, from 0 to 86400.	1.3.6.1.4.1.1667.1.2.1.10.1.7
agentSit-Source	Situation current status. The valid values are: 0 - Undefined 1 - Enterprise, meaning the situation was defined on the Tivoli Enterprise Monitoring Server 2 - Private, meaning the situation was defined by the local private situation configuration file.	1.3.6.1.4.1.1667.1.2.1.10.1.20
autoSit-Category	Assigned situation category. Valid values are: 0 - Threshold 1 - Network Topology 2 - Error 3 - Status 4 - Node Configuration 5 - Application Alert 6 - All Category 7 - Log Only 8 - Map 9 - Ignore	1.3.6.1.4.1.1667.1.2.1.6

Table 74. SNMP trap variables for agentSitSampledEvent (continued)

Attribute	Description	OID
autoSit-Severity	Assigned situation severity. Valid values are: 0 - Cleared 1 - Indeterminate 2 - Warning 3 - Minor 4 - Major 5 - Critical	1.3.6.1.4.1.1667.1.2.1.7
autoSit-Predicates	This is the situation formula, up to 3210 bytes, in the form attributeName Operator CompareValue. When the formula has multiple expressions, their Boolean AND or OR connectors are shown.	1.3.6.1.4.1.1667.1.2.1.8
sitAttributeList	The attribute values for the situation that is assigned to the monitoring agent, from 0 to 3200 bytes.	1.3.6.1.4.1.1667.1.2.1.5

agentSitPureEvent

A pure situation event was detected. This trap was generated by the Tivoli Autonomous Agent SNMP Event Exporter in response to a situation threshold being exceeded. The variables in a pure event trap are identical to those for a sampled event trap except there is no agentSit-SampleInterval because pure events are not sampled; rather the arrival of unsolicited data from the monitored attribute group causes the situation to become true. A situation created with an attribute group for a system log, for example, opens a pure event when a log entry arrives.

Specific trap: 22

Access: read-only

Status: mandatory

Table 75. SNMP trap variables for agentSitPureEvent

Attribute	Description	OID
agentSit-Application	This is the product application name, from 1 to 8 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.1
agentSit-Table	This is the name of the product application table (attribute group), from 1 to 12 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.2
agentSit-Name	The situation name, up to 32 bytes, identifies the name and nature of the status event.	1.3.6.1.4.1.1667.1.2.1.10.1.3
agentSit-OriginNode	The name of the managed system where the situation was evaluated, up to 32 bytes.	1.3.6.1.4.1.1667.1.2.1.10.1.4

Table 75. SNMP trap variables for agentSitPureEvent (continued)

Attribute	Description	OID

agentSit-LocalTimeStamp	The timestamp when the situation state changed. The format is CYYMMDDHHMMSSmmm (such as 1091031183005000 for October 31, 2009 at 18:30:05) where:
	C = Century (1 for 21st)
	Y = Year
	M = Month
	D = Day
	H = Hour
	M = Minute
	S = Second
	m = millisecond
		1.3.6.1.4.1.1667.1.2.1.10.1.5

agentSit-Context	Unique situation context identifier, expressed as an integer (-2147483647 to 2147483647). This is the handle number identifying an agent running request. In an SNMP environment, trap-direct polling is typically used whereby a trap is received and the network manager polls the originating agent for additional detailed information. This identifier is used to supply context for the targeted agent to correlate the request to the problem source. Although agentSit-Context is sent, it is not used in this release.	1.3.6.1.4.1.1667.1.2.1.10.1.6
agentSit-Source	Situation current status. The valid values are: 0 - Undefined 1 - Enterprise, meaning the situation was defined on the Tivoli Enterprise Monitoring Server 2 - Private, meaning the situation was defined by the local private situation configuration file.	1.3.6.1.4.1.1667.1.2.1.10.1.20
autoSit-Category	Assigned situation category. Valid values are: 0 - Threshold 1 - Network Topology 2 - Error 3 - Status 4 - Node Configuration 5 - Application Alert 6 - All Category 7 - Log Only 8 - Map 9 - Ignore	1.3.6.1.4.1.1667.1.2.1.6
autoSit-Severity	Assigned situation severity. Valid values are: 0 - Cleared 1 - Indeterminate 2 - Warning 3 - Minor 4 - Major 5 - Critical	1.3.6.1.4.1.1667.1.2.1.7

Table 75. SNMP trap variables for agentSitPureEvent (continued)

Attribute	Description	OID
autoSit-Predicates	This is the situation formula, up to 3210 bytes, in the form attributeName Operator CompareValue. When the formula has multiple expressions, their Boolean AND or OR connectors are shown.	1.3.6.1.4.1.1667.1.2.1.8
sitAttributeList	The attribute values for the situation that is assigned to the monitoring agent, from 0 to 3200 bytes.	1.3.6.1.4.1.1667.1.2.1.5

Appendix F. Agent operation log

A Tivoli Enterprise Monitoring Agent can run autonomously for an undetermined period of time, taking data samples and saving events. Review the audit trail log to examine and review the agent activities, including while it was running autonomously.

When an agent runs autonomously, audit trail records for all events and true sampled application data rows are written to the operations log. The agent leverages the existing Agent Operation Log facility and outputs audit trail records to it. The Agent Operation Log can be viewed on the Tivoli Enterprise Portal while the agent is online.

- On distributed systems, the agent creates the Operation Log file automatically in the agent installation directory, names it `ComputerName_product.LG0` for the current running log file, and renames the previous log file `ComputerName_product.LG1` (the backup file).
- On z/OS systems, the agent writes the Agent Operation log records to a `SYSOUT` class, saving portions of records in memory cache.

The agent operations log also shows the activity of private situations.

The autonomous activity log record contains these fields:

- Agent system name
- Message ID: KRAIRA005
- Global timestamps, showing the actual local time of the event activity
- The message, which shows the situation name, application table name, system name, filter column name, filter value, and actual sampled value or event value. If the situation filter criteria specify several threshold name and value pairs and thus the output exceeds the operation log's record size, then the agent outputs multiple log records.

To obtain an agent autonomous operation activity report, create an Agent Operation Log custom query in the Tivoli Enterprise Portal that filters on message KRAIRA005, and then assign the query to a table view in a workspace at the agent level of the Navigator Physical view. Alternatively, you can assign the predefined query named *Agent Operations Log* to a table view and apply a post-filter through the Properties editor Filters tab filters out all rows except those with message KRAIRA005. shows a possible autonomous activity log that might result from such a query.

This is the result of a table view of the Agent Operations Log filtered to include only the agent autonomy messages: ▾ == KRAIRA005

Server Name	Message Number	Global Timestamp	Managed System Type
Primary:East:NT	KRAIRA005	02/16/2009 12:35:42	Situation NT_Process_CPU_Critical for KNT.WTPROCESS reset
Primary:East:NT	KRAIRA005	02/16/2009 12:34:43	Situation NT_Process_CPU_Critical for KNT.WTPROCESS triggered (03) Process_Name [_Total] value <kdsmain>
Primary:East:NT	KRAIRA005	02/16/2009 12:34:42	Situation NT_Process_CPU_Critical for KNT.WTPROCESS triggered (02) Priority_Base [0] value <8>

Server Name	Message Number	Global Timestamp	Managed System Type
Primary:East:NT	KRAIRA005	02/16/2009 12:34:42	Situation NT_Process_CPU_Critical for KNT.WTPROCESS triggered (01) %_Processor_Time [65] value <66>
Primary:East:NT	KRAIRA005	02/16/2009 12:34:21	Situation NT_Log_Space_Low for KNT.WTPROCESS triggered %_Usage [95] value <100>
Primary:East:NT	KRAIRA005	02/16/2009 12:32:42	Situation NT_Process_Memory_Critical for KNT>WTPROCESS triggered (02) Working_Set [40000000] value <48832512>
Primary:East:NT	KRAIRA005	02/16/2009 12:32:41	Situation NT_Process_Memory_Critical for KNT>WTPROCESS triggered (01) Process_Name [_Total] value <Rtvscan>
Primary:East:NT	KRAIRA005	02/16/2009 12:31:21	Situation NT_System_CPU_Critical for KNT.WTSYSTEM triggered Operating_System_Version [5.0] value <5.1>
Primary:East:NT	KRAIRA005	02/16/2009 12:29:41	Situation CHECK_NETWORK_STAT for KNT.IPSTATS triggered (06) Datagrams_Received_Header_Errors [0] value <0>
Primary:East:NT	KRAIRA005	02/16/2009 12:29:41	Situation CHECK_NETWORK_STAT for KNT.IPSTATS triggered (05) Datagrams_Outbound_Header_Errors [0] value <0>

Note: CTIRA_LOG_PATH agent environment variable for distributed enterprise monitoring agents specifies the directory where the agent's Operations Log file is stored (Windows <*install_dir*>\TMAITM6\logs; Linux and UNIX <*install_dir*>/config/logs.) The file names use the suffixes .LG0and .LG1.

Documentation library

Various publications are relevant to the use of 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

Documentation is delivered in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp and also in the **Files** section of the Application Performance Management community.

For information about accessing and using the publications, select IBM Tivoli Monitoring → **Using the publications** in the **Contents** pane of the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp.

To find a list of new and changed publications, click the **New in this release** topic on the IBM Tivoli Monitoring welcome page. 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 IBM Tivoli Monitoring library provides information about the commonly shared components of Tivoli Management Services.

- *Quick Start Guide*

  Introduces the components of IBM Tivoli Monitoring.
- *Installation and Setup Guide*, SC22-5445

  Provides instructions for installing and configuring IBM Tivoli Monitoring components on Windows, Linux, and UNIX systems.
- *Installation Roadmap* available on Service Management Connect

  Provides a roadmap that covers the installation of IBM Tivoli Monitoring.
- *High Availability Guide for Distributed Systems*, SC22-5455

  Gives instructions for several methods of ensuring the availability of the IBM Tivoli Monitoring components.
- *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.
- *Administrator's Guide*, SC22-5446

  Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.
- *Command Reference* available on Service Management Connect

  Provides detailed syntax and parameter information, as well as examples, for the commands you can use in IBM Tivoli Monitoring.
- *Messages* available on Service Management Connect

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).

- *Troubleshooting Guide* available on Service Management Connect

  Provides information to help you troubleshoot problems with the software.
- *Tivoli Enterprise Portal User's Guide* available on Service Management Connect

  Complements the Tivoli Enterprise Portal online help. The guide provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.
- 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.

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.

- 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
- OS agent documentation is delivered in the following locations:

  Agent Installation and Configuration Guide

  Available in the Information Center:
  - *IBM i OS Agent Installation and Configuration Guide*, SC27-5653
  - *Linux OS Agent Installation and Configuration Guide*, SC27-5652
  - *UNIX OS Agent Installation and Configuration Guide*, SC27-5651
  - *Windows OS Agent Installation and Configuration Guide*, SC27-5650

  Agent Reference

  Available on Service Management Connect

  Agent Troubleshooting Guide

  Available on Service Management Connect

  Infrastructure Management Dashboards for Servers Reference

  Available on Service Management Connect
- Warehouse agent documentation is delivered in the following locations:

  Agent Installation and Configuration Guide

  Available in the Information Center:
  - *Warehouse Proxy Agent Installation and Configuration Guide*, SC27-5655
  - *Warehouse Summarization and Pruning Agent Installation and Configuration Guide*, SC27-5654

Agent Reference
Available on Service Management Connect

Agent Troubleshooting Guide
Available on Service Management Connect

- 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
  - *Agent Builder User's Guide*, SC32-1921
  - *Performance Analyzer User's Guide*, SC27-4004
  - *Systems Director base Agent User's Guide*, SC27-2872
  - *Tivoli Log File Agent User's Guide*, SC14-7484
  - *Tivoli zEnterprise Monitoring Agent User's Guide*, SC14-7359 and the *Tivoli zEnterprise Monitoring Agent Installation and Configuration Guide*, SC14-7358

Related publications

For information about related products and publications select **OMEGAMON XE shared publications** or other entries in the **Contents** pane of the IBM Tivoli Monitoring and OMEGAMON XE Information Center.

You can access the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp .

You can also access other information centers at IBM Tivoli Documentation Central (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central).

Tivoli Monitoring community on Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.

For information about Tivoli products, see the Application Performance Management community on SMC at IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm).

For introductory information, see IBM Service Management Connect (http://www.ibm.com/developerworks/servicemanagement).

Use Service Management Connect in the following ways:

- Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the (enter your community name here) community.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Other sources of documentation

You can obtain additional technical documentation about monitoring products from other sources.

- Tivoli wikis

  IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm) includes a list of relevant Tivoli wikis that offer best practices and scenarios for using Tivoli products, 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:

  – The IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring) provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Management products.

  – The Tivoli System z® Monitoring and Application Management Wiki 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.
- IBM Integrated Service Management Library

  http://www.ibm.com/software/brandcatalog/ismlibrary/

  IBM 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/.

Support information

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

Online
The following sites contain troubleshooting information:

- Go to the IBM Support Portal (http://www.ibm.com/support/entry/portal/software) and follow the instructions.
- Go to IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm) and select the appropriate wiki.

IBM Support Assistant
The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to IBM Support Assistant (http://www-01.ibm.com/software/support/isa).

Troubleshooting Guide
For more information about resolving problems, see the product's Troubleshooting Guide.

Using IBM Support Assistant

The IBM Support Assistant is a free, stand-alone application that you can install on any workstation. You can then enhance the application by installing product-specific plug-in modules for the IBM products you use.

The IBM Support Assistant saves you the time it takes to search the product, support, and educational resources. The IBM Support Assistant helps you gather support information when you need to open a problem management record (PMR), which you can then use to track the problem.

The product-specific plug-in modules provide you with the following resources:

- Support links
- Education links
- Ability to submit problem management reports

For more information, and to download the IBM Support Assistant, see http://www.ibm.com/software/support/isa. After you download and install the IBM Support Assistant, follow these steps to install the plug-in for your Tivoli product:

1. Start the IBM Support Assistant application.
2. Select **Updater** on the Welcome page.
3. Select **New Properties and Tools** or select the **New Plug-ins** tab (depending on the version of IBM Support Assistant installed).
4. Under **Tivoli**, select your product, and then click **Install**. Be sure to read the license and description.

If your product is not included on the list under **Tivoli**, no plug-in is available yet for the product.

5. Read the license and description, and click **I agree**.
6. Restart the IBM Support Assistant.

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 website at http://www.ibm.com/software/support.
2. Under **Select a brand and/or product**, select **Tivoli**.

   If you click **Go**, the **Search within all of Tivoli support** section is displayed. If you don't click **Go**, you see the **Select a product** section.
3. Select your product and click **Go**.
4. 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 **Search**.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.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 website 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 ETR directly from the IBM Support Assistant.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online
  Go to the Passport Advantage website at http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm .

  By telephone
  For the telephone number to call in your country, go to the IBM Software Support website 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 website at https://techsupport.services.ibm.com/ssr/login.
- For customers with Linux, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line website 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 website 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://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html and click the name of your geographic region for telephone numbers of people who provide support for your location.

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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. 2013. 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.

Adobe and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, and Windows 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.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name for purposes of session management, authentication, and single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

Printed in USA

SC22-5446-01