Tivoli® IBM Tivoli Monitoring
Version 6.2.2 Fix Pack 2 (Revised June 2010)

Administrator's Guide

SC32-9408-03

Tivoli® IBM Tivoli Monitoring
Version 6.2.2 Fix Pack 2 (Revised June 2010)

Administrator's Guide

SC32-9408-03

Note
Before using this information and the product it supports, read the information in "Notices" on page 389.

June 2010

This edition applies to version 6, release 2, modification 2, 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

Figures

Tables

About this guide

This 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 Tivoli Enterprise 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 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 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
- 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 IBM Tivoli Data Warehouse, you must be familiar with the operating system that hosts the warehouse. To learn more about this family of products, see http://www-306.ibm.com/software/tivoli/solutions/availability/products.html.

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 6.2.2 Tivoli Enterprise Portal features, please consult the integrated help (Help → Contents and Index) or the *IBM Tivoli Monitoring 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 *Administrator's Guide*.

New in Version 6.2.2 Fix Pack 2

This topic describes enhancements to the Tivoli Management Services components that affect the system administrator since the release of Version 6.2.2 Fix Pack 1.

Centralized Configuration to maintain local configuration files from a central location

You can implement a configuration load list XML file to pull local configuration file updates from one or more central locations that serve agents throughout the monitored enterprise. Using the repository, you can install or maintain monitoring agents (or both) and download all required files to become fully operational. See Chapter 13, "Centralized Configuration," on page 255.

The new **tacmd getfile** and **tacmd putfile** CLI commands are useful for implementing Centralized Configuration and agent autonomy. See *IBM Tivoli Monitoring Command Reference* for details.

Send agent heartbeat as an EIF event or SNMP alert to a receiver

You can now monitor the online status of a monitoring agent by configuring the heartbeat interval that sends a heartbeat event to a receiver after each interval. See "EIF heartbeat event" on page 225

Agent Management Services enables instance-specific CAP files

In previous releases only one executable or 'family' of executable files was supported per CAP (common agent package) file, thus multi-instance monitoring agent instances were governed by the same CAP file. Now you have the ability to configure CAP files for specific instances of monitoring agents.

Agent Management Services disabled on zLinux OS agent

In previous versions Agent Management Services was enabled on the zLinux OS agent. Now it is disabled, even if it was enabled before upgrading to V6.2.2 Fix Pack 2. The watchdog must be enabled manually. See "Tivoli Agent Management Services installation and configuration" on page 151.

Autonomous operation of the Tivoli Data Warehouse agents

To enable autonomous agents to write out their accumulated historical data to the Tivoli Data Warehouse without the intervention of the Tivoli

Enterprise Monitoring Server and thereby prevent loss of such data, the warehouse agents have been enhanced to allow them to run autonomously.

Run Warehouse Proxy agent without requiring registration with the Tivoli Enterprise Monitoring Server
The Warehouse Proxy can be configured to run with no connection to the hub monitoring server.

Monitoring agents can be configured to specify the location of their Warehouse Proxy agent instance, bypassing any other methods of obtaining its address. See the KHD-prefixed environment variables in "Control autonomy in Tivoli Enterprise Monitoring Agents" on page 161.

Run Summarization and Pruning agent without requiring connection to the Tivoli Enterprise Portal Server
In previous releases the summarization and pruning configuration settings for attribute groups that have been configured for historical data collection were stored on the Tivoli Enterprise Portal Server database in a table named KDWHISTDATA. The summarization and pruning configuration has been moved from the portal server to the Tivoli Data Warehouse database in a table named WAREHOUSESUMPRUNE. After the portal server and summarization and pruning agent have been upgraded to Tivoli Monitoring V6.2.2 Fix Pack 2, the next time summarization and pruning takes place and the portal server is restarted, the migration of the summarization and pruning configuration occurs automatically.

See "Configuring a Summarization and Pruning agent to run autonomously" in *IBM Tivoli Monitoring Installation and Setup* for a complete description.

Tivoli Common Reporting V1.3
The current version of Tivoli Common Reporting is Version 1.3, which introduces IBM Cognos Business Intelligence and Reporting Version 8.4 reports for the Tivoli Monitoring OS Agents and other monitoring products. A set of predefined reports is provided on a separate eAssembly media package for monitoring individual, multiple, and enterprise resources. See Chapter 15, "Tivoli Common Reporting," on page 327 and the user's guide for your Tivoli Enterprise Monitoring Agent.

Filtered historical data collection
The Tivoli Monitoring Version 6.2.2 release included the added capability of configuring multiple data collections for the same attribute groups with different settings and managed system distributions. Also added was the ability to create historical groups of historical collections that share the same managed system distribution.

Now you can write filter criteria to specify the data to collect. The new **Filter** tab in the Historical Collection Configuration editor has a formula editor much like what you have in the Situation editor. Historical collection of a data sample occurs only if the values in the data row fulfill the filter criteria. For example, if the data sample has % Disk Write Time greater than 50%, it is saved to short-term history; otherwise the sample is not saved. See step 9 of "Creating a historical collection" in the *Tivoli Enterprise Portal User's Guide*.

The command-line interface **tacmd histcreatecollection** and **tacmd histeditcollection** commands have been updated to include the **filter** option for adding a filter formula; the **tacmd histviewcollection** and **tacmd histlistcollections** commands display any defined filter for a historical collection; and **tacmd histviewattributegroup** has been updated to add the

-v verbose option to include the table name and attribute names in the display. See *IBM Tivoli Monitoring Command Reference*.

Dynamic items for custom Navigator views

The Navigator Physical view is a discovered view and the Navigator Logical view is a predefined, customizable view. You can also create custom Navigator views for different logical hierarchies. Because custom Navigator views are user-defined, these views are not updated in the Tivoli Enterprise Portal when new managed systems come online. You must manually add them using Navigator editor.

Now you can assign managed system groups to Navigator items as *dynamic members* in the new **Dynamic items** tab of the Navigator item properties editor. As managed systems are added or removed from the group, the constituent members change dynamically within this branch of the Navigator. New managed systems that fit the filter criteria of the managed system group are added automatically.

See “Navigator item properties”.

Define schedules for running situation overrides

In previous releases you could apply a predefined schedule, such as Weekend, to specify when the situation expression override is run. You can now create your own calendar-based schedule or hourly schedule for each expression override that you create for a situation.

Also in previous releases, the only way to define a schedule for a situation expression override was through the command-line interface **tacmd addCalendarEntry**. As well as shared calendar-based schedules in the Situation editor, you can define repeating hourly schedules that apply to specific hourly time ranges.

See “Creating a situation override”.

The CLI **tacmd setOverride**, **tacmd deleteOverride**, **tacmd suggestBaseline**, and **tacmd acceptBaseline** commands have been updated to include the **inlinecal** option for specifying the inline (hourly) calendar schedule. See *IBM Tivoli Monitoring Command Reference*.

Model Expression dialog for testing situation override scenarios

Tivoli Monitoring V6.2.2 introduced the expression modeling feature to visually assist the situation author in establishing proper threshold values based on historical data and statistical analysis (see “Model Situation”). That same capability has now been added for situation expression overrides.

This new visual modeling capability also integrates the ability to dynamically build and associate daily schedules to expression overrides using drag-and-drop gesturing. See “Model Expression”.

Monitored baselining for charts indicate situation override expressions

The **Add Monitored Baseline** tool was added to the chart views in Tivoli Monitoring V6.2.2 for you to visualize the attribute threshold values from associated situations alongside the historical or real-time information displayed. If expression overrides are associated with the situation, you can now visualize them in the chart as well. You can optionally filter out situation and expression overrides that are inactive based on the data sample and time span being rendered in the view.

See “Adding monitored baselines to a chart”.

New and enhanced CLI tacmds

New **tacmd getfile** and **tacmd putfile** commands are available for transferring files between remote managed systems and a local computer.

New tacmd commands are available for replicating Navigator view or Navigator item managed system assignments:

listSysAssignments to list all managed systems and managed system groups that are assigned to a Navigator item.

exportSysAssignments to export all managed system assignments for the specified Navigator item or the entire Navigator view..

importSysAssignments to import all managed system assignments for a Navigator item or Navigator view.

deleteSysAssignments to delete the specified situation association for the given navigator item

The **tacmd listworkspaces** has been enhanced to include the Tivoli Enterprise Portal *objectid* in the workspace listing. By using the new **tacmd deleteWorkspace** command, you can indicate the workspace to delete by its objectid name.

New tacmd commands are available for changing situation associations with Navigator items:

listSitAssociations to list all situation associations defined for a navigator item

createSitAssociation to create a situation association for the given navigator item

exportSitAssociations to export all situation associations on the server

importSitAssociations to import all situation associations specified in a file

deleteSitAssociation to delete the specified situation association for the given navigator item

These commands have a new **inlinecal** option for specifying an inline (hourly schedule) calendar entry for situation overridses: **tacmd setOverride**, **tacmd suggestBaseline**, **tacmd acceptBaseline**, and **tacmd deleteOverride**.

These commands have a new **filter** option for adding a formula for pre-filtering data collection: **tacmd histcreatecollection** and **tacmd histeditcollection**. The **tacmd histviewattributegroup** command has a new **verbose** option for displaying the table name and attributes names defined for an attribute group.

See *IBM Tivoli Monitoring Command Reference*

New in Version 6.2.2 Fix Pack 1

This topic describes enhancements to the Tivoli Management Services administration since the release of Version 6.2.2.

TEDGEN can be run from the hub monitoring server or portal server

The TEDGEN tool provides an alternate method for generating a new XML file for EIF Slot Customization. There are now three places where you can run the TEDGEN tool if the necessary baroc files are on the same computer:

1. On the computer where the Tivoli Enterprise Console event server is installed.

2. On the computer where the hub Tivoli Enterprise Monitoring Server is installed.
3. On the computer where the Tivoli Enterprise Portal Server is installed.

See "Updating the XML used by the MCS Attribute Service" on page 115.

Send EIF events for private situations

You can configure a monitoring agent and an EIF event configuration file to emit life cycle events or private situation events, or both, directly to an EIF receiver such as the Tivoli Enterprise Console event server or Netcool/OMNIbus Probe for Tivoli EIF, with no connection to the Tivoli Enterprise Monitoring Server. See "EIF event configuration" on page 213.

Agent Management Services expanded support

Support has been added for these products:

- IBM Tivoli Monitoring for Databases: Oracle Agent
- IBM Tivoli Monitoring for Databases: Sybase Agent
- IBM Tivoli Monitoring for Messaging and Collaboration: Lotus Domino Agent (including 64-bit)
- IBM Tivoli OMEGAMON XE for Messaging

Also, Windows OS agents and Lotus Domino agents on 64-bit systems have Common Agent Package XML files installed in the `TMAITM6_x64\CAP` directory rather than `TMAITM\CAP`. See Chapter 11, "Agent Management Services," on page 149.

New in Version 6.2.2

This topic describes enhancements to the Tivoli Management Services that are relevant to this guide since the release of Version 6.2.1. Many of the changes are obvious as soon as you log on, such as the new toolbar icons. Others are changes in behavior or changes that are not apparent until you open workspaces or one of the editors.

The Tivoli Enterprise Portal client features are described in the online help and *IBM Tivoli Monitoring Tivoli Enterprise Portal User's Guide*.

User interface updates

The toolbar buttons and graphic icons have been updated and consolidated to further align the Tivoli user interfaces. Move the mouse pointer over a tool in the Tivoli Enterprise Portal to see its identity.

Tivoli Enterprise Portal Version 6.2.1

Desktop mode:

Browser mode:

Tivoli Enterprise Portal toolbar for Version 6.2.2

Some of the icons have been regrouped to align them with their function: **Switch to Home Workspace** has moved to the first position in the toolbar, next to **Back** in the desktop mode toolbar, and **Save** in the browser mode toolbar; and the **Situation Event Console**, **Common Event Console**, and **Tivoli Enterprise Console** views are grouped near the end of the toolbar.

Desktop mode:

Browser mode:

Workspace gallery for identification and selection

As well as the default workspace that opens when you click a Navigator item, additional workspaces are often available for the item and selectable through the Navigator pop-up menu or the View menu. Now you have the Workspace Gallery tool for showing you the workspaces that you can open for the current Navigator item. See Opening a workspace.

Manage Tivoli Enterprise Monitoring Server workspaces and situations

New self-monitoring workspaces and situations have been added and are accessible through the Enterprise Navigator item to help you monitor for and diagnose typical monitoring server configuration issues. See Tivoli Enterprise Monitoring Server status.

Configure historical data collection with distribution lists

The Historical Collection Configuration window has been redesigned. It looks similar to the Situation editor, with a tree on the left and user assistance on the right until you select a tree item: The summarization and pruning settings are displayed when you select a managed application; and the historical collection configuration is displayed when you select a collection name within the managed application branch.

You can now have multiple collection configurations for an attribute group. There is also a new distribution method called **Managed System (TEMA)** that enables you to specify the managed systems that data collection will occur on. (With this method, the managed systems must connect to a V6.2.2 Tivoli Enterprise Monitoring Server.) See Historical collection configuration.

Several tacmd commands have been added for creating, deleting, editing, listing, viewing the configuration of historical data collections. Although not new for this release, the *bulkExportSit -h* and *bulkImportSit -h* commands can be used to export and import historical collection configurations. See the IBM Tivoli Monitoring Command Reference.

Granular data collection with historical configuration object groups

Object grouping was introduced in the previous release for situations and managed systems. Now you can create historical configuration object groups, which can include managed system distribution, and assign historical collections to them. See Object group editor.

Managed system lists renamed to managed system groups

The term *managed system list* has been renamed to *managed system group* to follow the naming used in the Object Group editor.

Modeling conditions for situations

Now you can capture data from a query-based view and use it to model possible threshold scenarios, then apply the results to a situation. You can also select a situation from the Manage Situations at Managed System window, compare the criteria with current and historical data samples, for modeling, and use the results to edit the situation thresholds. See Modeling conditions for situations.

Baselining added to charts for trend analysis

The bar chart, plot chart, and area chart have a new **Add Monitored**

Baseline tool for selecting a situation to compare with the current sampling and anticipated values. The plot chart and area chart also have a new **Add Statistical Baseline** tool with statistical functions. In addition, the plot chart has a new **Add Historical Baseline** tool for comparing current samplings with a historical period. See Chart baselines.

Situation overrides can be assigned to subnodes
Situation overrides for dynamic thresholding has been extended to include subnodes. For situations that are distributed to managed systems that are subnodes of other managed systems, you can now apply expression overrides to the managed system subnodes. See Situation overrides for dynamic thresholding.

Tivoli System Monitor Agent
The Tivoli System Monitor Agent is a new category of monitoring agent. It is an OS agent that is installed and configured to have no dependency on nor any connection to a monitoring server. It must be installed on a computer where no other Tivoli Monitoring products or Tivoli Management Services components are installed, with the exception of agents created with Agent Builder.

The system monitor agent can run private situations that are defined in a situation configuration XML file; collect historical data, also defined in the situation configuration XML file, and send SNMP alerts that are defined in a trap destination XML file to an event receiver such as the Netcool/OMNIbus SNMP Probe. See Chapter 12, “Agent autonomy,” on page 157

Agent autonomy

Agent autonomy is the ability of a monitoring agent to perform independently of the Tivoli Enterprise Monitoring Server. With environment variables introduced in IBM Tivoli Monitoring V6.2.1, the Tivoli Enterprise Monitoring Agent can retain event information when communications with its monitoring server are interrupted. Now it is the default behavior that agent startup is independent of the monitoring server and information is collected at the agent and maintained even if the agent is stopped and started again, ready for transfer to the monitoring server when a connection is made.

Features introduced in this release that can be used by agents without a monitoring server connection are private situations, private history, SNMP alerts, and the Agent Service Interface. See Chapter 12, “Agent autonomy,” on page 157.

Create private situations that run locally, independent of the monitoring server
Private situations are created in a local private situation configuration XML file for a monitoring agent. 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 remain local to your workstation or can be sent as SNMP alerts to a receiver such as the Netcool/OMNIbus SNMP Probe.

Private situations have the same capability as enterprise situations, with some limitations. For example, only the **Value of expression** and **Check for Missing Items** formula functions can be used. See “Situation limitations” on page 167.

Create private history collections that run locally, independent of the monitoring server Private history is the collection and short-term storage of data from a local

monitoring agent. Define historical collection for each attribute group in a private situation configuration file for an agent, then use the Agent Service Interface to view the short-term history.

Private history is local only; there is no interaction with a warehouse proxy for long-term storage on the Tivoli Data Warehouse. However, when defining the private history, you can change the default 24-hour limit for historical data storage to as many hours as you like, limited only by the storage space on the computer. See "Private history" on page 190

Send SNMP traps
Send SNMP alerts from the local Tivoli Enterprise Monitoring Agent or Tivoli System Monitor Agent directly to a receiver, without ever connecting to a monitoring server. You create a trapcnfg.xml file and reference the situations and attribute information to send when the situation becomes true. When the agent is started, it sends any alerts for the situations defined in the file as SNMPv1/v2 traps or SNMPv3 informs to a receiver such as the Netcool/OMNIbus SNMP Probe. See "SNMP alerts" on page 197.

Create situation overrides in a local XML configuration file
As an alternative to creating expression overrides for enterprise situations in the Tivoli Enterprise Portal, you can define them in an XML configuration file. See Situation override XML specification.

Agent Service Interface
The IBM Tivoli Monitoring Service Index has links to service consoles for the components installed on the computers. Now, when you go to the Service Index, you will also see links to the Agent Service Interface. Use the Agent Service Interface to get reports for 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 choose reports of agent information, private situations, private history, and attribute descriptions and current values. You can also make a service interface request using provided XML elements. See "Agent Service Interface" on page 227.

Proxy Agent Services renamed to Agent Management Services
The Proxy Agent Services have been renamed to Agent Management Services to better describe their function. More base agents are supported and there are new capabilities to report and manage agent instances. See Chapter 11, "Agent Management Services," on page 149.

TIP Web Service for Tivoli Integrated Portal charts
In the previous release, users who wanted to see query-based views in the Tivoli Integrated Portal administrative console needed to have workspace administrator authority. This is no longer a requirement. Users can now view through the administrative console any Tivoli Enterprise Portal workspaces that their user ID has permission to view. For example, if the allowed applications for your user ID include Linux OS application but not DB2, then any Linux OS workspaces are available from the administrative console but not the DB2 workspaces. See "Enabling TIP Web Service for Tivoli Integrated Portal charts" on page 32.

Optional agent restart
After reconfiguring an agent, you can now choose whether to restart the agent immediately for the changes to take effect or to leave the agent running and restart it at a later time. The Tivoli Service Manager (Manage Tivoli Monitoring Services window) has a new column named

Configuration to show this status: **out-of-sync** for configuration changes that have not been implemented; or **up-to-date** for configuration changes that have been implemented by restarting the agent.

Client environment variables
`cnp.browser.installdir` to specify a different path for the browser view files, required if users will be running multiple instances of the portal client and possibly logging on to different versions of the portal server. See "Portal client parameter list" on page 36.

New in Version 6.2.1

This topic describes enhancements to the Tivoli Enterprise Portal and to the other components of Tivoli Management Services that are relevant to this guide since the release of Version 6.2.0.

The Tivoli Enterprise Portal client features are described in the online help and *IBM Tivoli Monitoring Tivoli Enterprise Portal User's Guide*.

Dynamic thresholding with situation override
Dynamic thresholding for situations means that you can override the expression values of a situation formula for a specific managed system (or group of managed systems) or for a specific time period or both. This capability enables you to adjust situations for conditions that are specific to a particular managed system or managed system group or to a particular time period.

Long situation names
Situation names can now be as long as 256 bytes and are no longer restricted to only letters, numbers, and underscores.

Organize situations and managed systems into named groups
The Object Group editor is a general, consistent mechanism for creating named groups of objects. The object types that can be grouped are situations and managed systems. The managed system group is being retired in this release and all the functions of the managed system group editor are now provided by the new object group editor.

Customize the EIF slots in the Situation editor for events that get forwarded to an EIF receiver
Through the EIF (Enterprise Integration Facility) tab of the Situation editor, you can now map situation events to the EIF receiver and customize the forward events.

Area chart
The new area chart view is similar to the plot chart. The difference is that the area from the X-axis and Y-axis to the plot point for each data series is filled with a pattern or color or both.

Chart area thresholds and markers, collapsible legend
Visual indicators for value ranges (chart thresholds) and for specific values (chart markers) can be added to the bar, plot, and area charts. Chart legends can be kept in a collapsible panel and expanded and hidden as needed to give maximum viewing space to the plot area. As well as colored labels for the attributes, you can specify a fill pattern.

Zooming in on chart areas
On bar charts, plot charts, and area charts, you can click and drag over an area of the chart that you would like to zoom into for closer scrutiny, then press Esc to return to the previous size.

Historical navigation mode to synchronize workspaces
When you open a workspace, the query-based views retrieve the latest values. Then, as you open or link to other workspaces, their assigned queries also retrieve the latest values. However, you might be performing analysis over multiple workspaces that requires review of a fixed time or time range. You can turn on *Historical navigation mode* with a time span that you specify. Then, all workspaces you open will align to that time period until you turn it off.

Options for more frequent warehouse intervals
When configuring attribute groups for historical data collection, you have more choices for the data rolloff from the history files to the Tivoli Data Warehouse. As well as 1 hour and 1 day, you now can select 15 minutes, 30 minutes, or 12 hours.

Find Navigator items
The Find feature for Navigator items enables you to search for and locate items by criteria such as product code or associated situation, and using formula functions. .

Terminal view links
You can now build contextual links from the table and chart views on an OMEGAMON XE workspace to a terminal view in another workspace.

Tabbed workspaces
When using the Tivoli Enterprise Portal browser client, you can open workspaces in new tabs if your browser enables them.

Agent deployment status workspaces
New workspaces have been added to the Enterprise Navigator item for showing deploy depot information and the status of the past, current, and scheduled agent deployments.

Single sign-on support with Java Web Start client
As well as the browser client, you can now use the Java Web Start client to launch into the Tivoli Enterprise Portal and out to other Tivoli web-based and web-enabled solutions without needing to re-enter your authentication credentials.

Tivoli Proxy Agent Services
New services can monitor the availability of agents and respond automatically (such as with a restart) if the agent operates abnormally or exits unexpectedly.

Configuring an HTTP proxy server for the browser view
The procedure for setting up an HTTP proxy server for portal browser view has been simplified and is the same for all Tivoli Enterprise Portal clients.

Enabling FIPS
The Tivoli Enterprise Portal Server has a new environment variable that can be enabled for conformance to the Federal Information Processing Standard (FIPS) 140–2 specification.

64-bit integers are now supported
Support for 64–bit integers has been added. Many of the Version 6.2.1 products have new attribute groups, attributes, situations, and workspaces that use the new 64–bit integer values. For example, there are workspaces with a superseded version that displays queries with a signed 32-bit maximum value (2,147,483,647) and a version with the same name (minus 'superseded') with queries that support values up to signed 64-bit max

(9,223,372,036,854,775,807). You will also see similar ‘superseded' notations for attribute groups, attributes, and situations that have a 64–bit counterpart. See your product user's guide for details.

New schema publication tool simplifies generation of SQL statements for creating the Tivoli Data Warehouse

With the new schema publication tool, you can now generate the SQL statements needed to create the database objects (data warehouse tables, indexes, functions, views, and ID table inserts) required for initial setup of the Tivoli Data Warehouse. For details, see “Generating SQL statements for the Tivoli Data Warehouse: the schema publication tool” in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Tivoli Data Warehouse now supports DB2 on z/OS

You can now create your Tivoli Data Warehouse repository using DB2 running on z/OS. Although the Warehouse Proxy agent still runs only on Windows, Linux, or UNIX, the data warehouse itself is stored in DB2 on z/OS databases. Data communication is supported using either an ODBC or a JDBC connection. See “Tivoli Data Warehouse solution using DB2 on z/OS” in the *IBM Tivoli Monitoring Installation and Setup Guide* for instructions on setting up your Tivoli Data Warehouse environment to run with a DB2 on z/OS repository.

Command Line Interface tacmds for new features

The CLI has dozens of new tacmds for many of the new Tivoli Enterprise Portal features and for features that are exclusive to the CLI, such as for exporting queries and custom Navigator views and their associated situations. See the *IBM Tivoli Monitoring Command Reference* for details.

Setting traces

On Linux and UNIX, the file that contains the KBB_RAS1 parameter for setting a trace has moved from the .config file in the *Install_dir*/config directory to ms.ini. On distributed operating platforms the value is no longer enclosed in single quotes. See the *IBM Tivoli Monitoring Troubleshooting Guide* for information on setting traces.

New in Version 6.2.0

This topic describes enhancements to the Tivoli Enterprise Portal and to the other components of Tivoli Management Services that are relevant to this guide in this release.

Event type tag for SOAP methods

The CT_Alert, CT_Acknowledge, CT_Reset, and CT_Resurface methods were modified to support the <type> tag, which specifies the event type.

User groups

The Administer Users window has been significantly enhanced to enable the creation of user groups, including assigned permissions.

Lockout and lockout override

Logon Permitted is a new permission that enables the administrator to lock out a user ID, preventing the user from logging on to the portal server or to override an automatic lock out, which occurs after a set number of invalid logon attempts.

User authentication

Support has been added to enable external authentication of Tivoli Enterprise Portal users with standards-based Lightweight Directory Access Protocol (LDAP) to shared registries. The hub monitoring server can be

configured to validate user IDs and passwords using either the local system registry or a central LDAP authentication and authorization system.

Flexible scheduling of summarization and pruning
The Summarization and Pruning agent configuration window has been enhanced to allow for flexible scheduling and to have the data warehouse and warehouse aggregation logs trimmed automatically after a specified number of days, months, or years.

The Defaults tab has been removed. Now, when you use the History Collection Configuration window to configure historical data collection for an attribute group, no summarization and pruning check boxes are selected for you by default.

Null values in summarized historical data
You now see `null` in a table cell or chart point when values that were identified as invalid were reported from a monitoring agent for a given summarization period.

More frequent intervals for historical data collection
One-minute and five-minute intervals have been added to the **Collection Interval** options, enabling you to save more frequently to the short-term history files at the monitoring agent or monitoring server.

There are no longer pre-selected check boxes for summarization and pruning when you configure historical data collection for an attribute group.

Common and Tivoli Enterprise Monitoring Server attributes
The common attribute groups, Local Time and Universal Time, have a new *Time* attribute for the time of the data sampling corrected for local time zone and daylight saving time.

Two of the Tivoli Enterprise Monitoring Server attribute groups also have new attributes: The Managed System Status attribute group adds a *Reason* attribute for the two-character reason code, if one exists, for the managed system status. The Situation Definition attribute group adds a new *Last Release* attribute to identify the release of the product to associate with the situation.

If your product was updated for this release of IBM Tivoli Monitoring, check the **New in this release** section of the product user's guide for a list of the new and updated attribute groups.

Some distributed products might require that you create new queries before you can see the new attributes in the query or queries for that attribute group or see no query for a new attribute group.

Seven event severities
The state of an event that opens for a true situation can be set to informational, warning, or critical. Now you have four additional states to choose from for associated situations, table view thresholds, and for filtering an event console view: Unknown, Harmless, Minor, and Fatal.

Table view threshold icons
The table view has a feature that highlights the background of any cell whose value exceeds a given threshold. Before this release, thresholding was limited to three background colors to indicate an informational,

warning, or critical severity. Now, as well as having four more severities available, you can choose to display either an icon in the cell (

) or a background color (

).

Table view style properties
A new option on the Style tab and in the workspace presentation cascading style sheet enables you to control the default font styling of table view header and footer text.

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

Situation editor EIF tab
The new EIF (event integration facility) tab has options for forwarding events that open for the situation to one or more EIF receivers and to specify the severity. The tecserver.txt mapping file that was used in version 6.1 is no longer needed.

Refresh Tivoli Enterprise Console information
A new CLI (Command Line Interface) `tacmd refreshTECInfo` command enables you to have the Tivoli Enterprise Console Event Forwarder reprocess updated event definitions, EIF configuration, and custom event mapping files without requiring a hub monitoring server recycle.

Situation event acknowledgement
Event acknowledgement has several new enhancements to facilitate quick acknowledgements, writing and reviewing notes, and attaching files to the acknowledgement.

Enterprise Status workspace
The first indication of the acknowledgement enhancements is in the Enterprise Status workspace, which adds a new view called My Acknowledged Events, as well as a new **Owner** column in the situation event console view that shows the ID of the user who acknowledged the situation event.

Home workspace
Initially, Enterprise Status is the first workspace to be displayed when you log on to the portal server. This is your home workspace. With the **Assign as Home Workspace** option, you can now establish another workspace, whether at this Navigator level or another and whether on this Navigator view or another, as your home workspace. The **Home** tool opens the home workspace.

Topology view
Topology view is a new type of query-based view that enables you to create views from relational data sources. Attributes from the query are rendered as objects and connected to related objects.

Another topology view that is available at the Enterprise level of the Navigator is the TMS (Tivoli Management Services infrastructure view, which visually expresses the relationships and linking of monitoring agents and other components to the hub monitoring server.

Self-Monitoring Topology workspace
The Enterprise Navigator item has a new Self-Monitoring Topology

predefined workspace. The purpose of this workspace is to introduce the self-monitoring capabilities that are available through the Tivoli Enterprise Portal.

Dynamic linking
A new link type has been added to the workspace link feature that enables the link author to identify the target workspace by the host identifier. The *dynamic* link type adds more opportunities for workspace linking, such as to provide links to workspaces of other types of monitoring agents.

Navigator view icon in the status bar
When the Navigator view has been collapsed, you can now restore it or open to another Navigator view by a click or right-click of  *<Navigator name>* in the Tivoli Enterprise Portal tatus bar.

Bar chart overlay
A new overlay feature has been introduced that allows one or more related attributes to be plotted against the bar chart. This can highlight the relationship of related values, and is useful for visualizing trends from historical data.

Plot chart overlay
In earlier releases the plot chart view was able to show data only from the first row of a data sampling. The plot chart properties have been enhanced for plotting multiple-row attribute groups (or historical data from a single-row attribute group) and multiple managed systems, and for controlling the refresh rate independent from the workspace as a whole.

The plot chart also has a new overlay feature that can be used to establish a secondary value axis.

Workflow editor
You can now launch the Situation editor from an activity in the Workflow editor to edit the situation that it references.

Application window
The banner artwork in the browser client has changed, the Navigator tabs have been modernized, as have the view title bars, which also have two new buttons for hiding or showing the view toolbar and for opening the Properties editor.

Creating a new view
After you click a view tool to create a new view, the mouse pointer adopts the view icon (instead of the  hand icon) on Windows® systems. And you can now press Escape or click somewhere in the toolbar if you then decide not to add the view.

Moving a view
You can now drag a view by its title bar to a new location in the workspace.

Searching in a view
The view toolbar for the table, notepad, and browser views has a new

 **Find** tool for quickly searching through text in the view.

Cell function
The  **See if Null (no value) has been detected** function can be used in the Filters and Thresholds formula editors to locate attributes for which no value has been retrieved.

operator
The **Value of expression** function has a new comparison operator that can be used in situations and in table view Filters and Thresholds to highlight specific text values.

Browser view
The browser view now supports most types of web content, such as JavaScript™, Applets, and PDF files.

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, IBM Tivoli Composite Application Manager, and IBM Tivoli Performance Analytics for Tivoli Enterprise Portal.

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 your Tivoli Enterprise Monitoring Agents.

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.

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.

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 client gathers the current values of the attributes and produces reports formatted into tables, charts, and relational table-based topology views. It can also test the values against a threshold and display an alert icon when that threshold is exceeded or a value is matched. These tests 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 to an Netcool/OMNIbus SNMP Probe.

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

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).

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

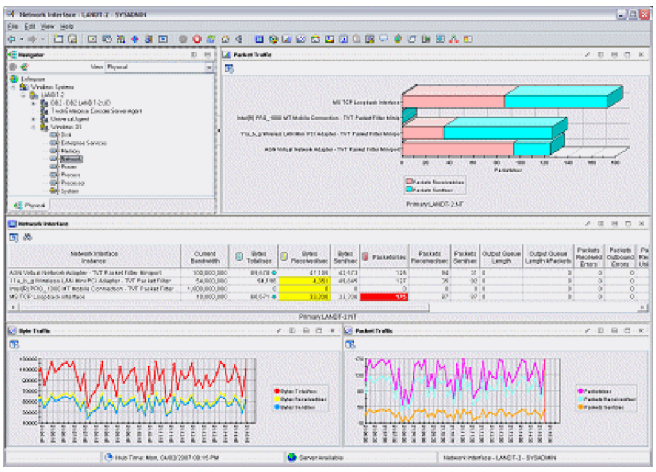
Tivoli Enterprise Portal client

Tivoli Enterprise Portal is the interface for your IBM Tivoli Monitoring 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 *IBM Tivoli Monitoring 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.

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

- Establishes 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

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, a check is done for IBM Java 1.5 on your system and, if not found, is installed automatically. This check also takes place when you install the desktop client, Java WebStart client, or log on from a browser with this difference: Sun Java 1.5.0_xx through 1.6.0_xx is also recognized as a valid JRE for the client (but not 1.6.0_xx for Firefox on Linux).

If you have different versions of the Java Runtime Environment installed locally, they can coexist. Tivoli Enterprise Portal V6.2.1 and V6.2.2 require IBM Java V1.5 or Sun Java V1.5. The desktop client must be at the same version as the portal server it connects to. This is also true for the browser client, but Java versioning is controlled at the portal server and upgraded automatically when you connect to a newer portal server.

Running IBM Tivoli Monitoring V6.1 agents and IBM Tivoli Monitoring V6.2 agents on the same computer requires Java 1.4.2 and 1.5 on that computer.

Allocating additional memory for the applet

When the browser client connects to the Tivoli Enterprise Portal Server, it downloads a Java applet. Before starting the browser client, allocate enough memory for the applet to avoid out-of-memory problems.

Before you begin

Specify the runtime parameters for the Java applet using the control panel for the appropriate JRE. Use the same user ID from where the browser client will be

launched to open the control panel and specify these parameters. Otherwise, the user-level `deployment.properties` file for the correct user ID will not be updated.

About this task

Take these steps to increase the memory allocated for the Tivoli Enterprise Portal Java applet.

Procedure

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. Example: `/opt/IBM/ibm-java2-i386-50/jre/bin/ControlPanel`.
2. Click the **Java** tab.
3. In the Java Applet Runtime 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 Runtime and that the JRE is in the correct path. For example, `C:\Program Files\IBM\Java50\jre\bin` for IBM Java on Windows.
5. Click in the **Java Runtime Parameters** field and set the parameters:

   ```
   -Xms128m -Xmx256m -Xverify:none
   ```

   The -Xms128m specifies the starting size of the Java heap (128 MB) and -Xmx256m specifies the maximum size. The -Xverify:none parameter disables Java class verification, which can improve the startup time.
   If you are using the IBM JRE on Linux, add the –Djava.protocol.handler.pkgs option:

   ```
   -Xms128m -Xmx256m -Xverify:none
   -Djava.protocol.handler.pkgs=sun.plugin.net.protocol
   ```

   This option is required for the IBM JRE on Linux due to a problem with the plug-in not caching jar files. If the parameter is left off, the Tivoli Enterprise Portal applet jar files will not be cached, making subsequent start ups of the applet slow.
6. Confirm that the Temporary Files settings are set to Unlimited:
   a. Click the **General** tab.
   b. Click **Settings**.
   c. Select the maximum amount of disk space for storing temporary files: 1000 MB.
7. Clear the browser cache:
   a. In the **General** tab, click **Delete Files**.
   b. In the window that opens, select **Downloaded Applets** .

What to do next

You can now start the browser client and connect to the portal server.

Note: The Sun JRE does not always support the same maximum heap values as the IBM JRE. The true maximum is calculated based on the resources available on the particular computer being configured, and the algorithm that is involved is different between the two JREs. The symptom of a memory problem is that the applet fails to load in the browser and you receive an error message. To resolve

this problem, reduce the value of the maximum heap setting in the Java Control panel in 32m or 64m increments until the error goes away. For example, if you start with the recommended value of -Xmx256m try reducing it to -Xmx224m or -Xmx192m. Eventually you will reach a value that is appropriate for your computer.

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 portal server (on Windows, the files reside in the <*itm_install_dir*>\cnb branch; on operating systems such as UNIX, they are in the <*itm_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 Tivoli Enterprise Portal. Otherwise, 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 Web site with Internet Explorer without changing your security settings. If you wish to keep your current security settings, you can add the Tivoli Enterprise Portal Web site 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 Web Start client or the browser client. Otherwise, Java exception errors are encountered during attempts to start the product.

Before users can download the Web Start 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.

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 *1920* is the port number for the browser client: `http://systemname:1920///cnp/client`
  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*.)
- IBM 32-bit Runtime Environment for Windows, Java 2, version 5.0 must be installed on the computer to which you want to download the desktop client.

  You can download the IBM JRE installer from the Tivoli Enterprise Portal Server. The IBM JRE must be installed as the system JVM.

If you want to run the desktop client on a system that already has a Tivoli Management Services base component installed (such as a monitoring server or the portal server), there is no need to install the IBM JRE. The correct version of the IBM JRE is installed with the Tivoli Management Services component.

If you run the desktop client using Web Start instead of installing it from the installation media, you must configure the JRE to enable tracing for the desktop client.

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 1.5. 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:

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 <*TEPS_host_name*> is the fully qualified host name of the computer where the portal server is installed (for example, `myteps.itmlab.company.com`):
   `http://<TEPS_host_name>:1920///cnp/kdh/lib/java/ibm-java2.exe`
3. When prompted, save the **java/ibm-java2.exe** file to a directory on your hard drive.
4. Change to the directory where you saved the **java/ibm-java2.exe** file and double-click the file to launch the JRE installer to start the installation program.
5. On the pop-up window, select the language from the drop-down list and click **OK**.
6. Click **Next** on the Welcome page.
7. Click **Yes** to accept the license agreement.
8. Accept the default location for installing the JRE or browse to a different directory. Click **Next**.
9. Click **NO** on the message asking if you want to install this JRE as the system JVM. Make Java 1.5 the system JVM only if there are no other JREs installed on the computer.
10. If another JRE is currently installed as the system JVM and you are prompted to overwrite the current system JVM, click **NO**. Overwriting the current system JVM 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://teps_hostname:1920///cnp/kdh/lib/java
     /ibm-java2-i386-jre-5.0-7.0.i386.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://teps_hostname:1920///cnp/kdh/lib/java
        /ibm-java2-i386-jre-5.0-7.0.i386.rpm
   ```

   where *teps_hostname* is the fully qualified host name of the computer where the portal server is installed (for example, `myteps.itmlab.company.com`).
3. When prompted, save the installer to disk.
4. Change to the directory where you saved the **ibm-java2-i386-jre-5.0-7.0.i386.rpm** file and launch the installer to start the installation program using the following command:

   ```
   rpm -ivh ibm-java2-i386-jre-5.0-7.0.i386.rpm
   ```

Enabling tracing for the JRE

Log files are not created for the desktop client launched through Web Start unless you enable tracing for the JRE.

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` directory. On Linux and UNIX computers, the logs are located in the `.java/deployment` directory of the home directory of the user ID under which the Java JRE was installed. Java Web Start will create a uniquely named trace file for every independent launch of the application. The files are named `javaws.`*`nnnnn`*`.trace`, where *nnnnn* is an arbitrary five-digit identifier.

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\Java50\jre\bin\javacpl.exe".
   - On Linux, change to `<install_dir>/jre/<platform>/bin` and run Control Panel: `./Control Panel`
2. Select the **Advanced** tab.
3. Expand the Debugging node in the **Settings** tree and check **Enable Tracing**.
4. Click **OK** to save the setting and close the Java Control Panel.

Downloading and running the desktop client

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 *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 <*TEPS_host_name*> is the fully qualified host name of the computer where the Tivoli Enterprise Portal Server is installed.

     `http://TEPS_host_name:1920///cnp/kdh/lib/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 1.5 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 *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 `./Control Panel`.
2. On the **General** tab, in the Temporary Internet Files section, click **Settings**. The Temporary Files Settings window is displayed.
3. Click **View Applications**.
4. On the **User** tab, select Tivoli Enterprise Portal, then click **Launch Online**.

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 Web Start from the command line:
  1. Open a command line window and change to the directory where Web Start is installed.
     Windows
     ```
     C:\Program Files\IBM\Java50\jre\bin
     ```
     Linux
     `<install_dir>/jre/<platform>/bin`
  2. Enter the following command, where *<TEPS_host_name>* is the fully qualified host name of the computer where the Tivoli Enterprise Portal Server is installed.
     Windows
     ```
     javaws http://<TEPS_host_name>:1920///cnp/kdh/lib/tep.jnlp
     ```
     Linux
     ```
     ./javaws http://<TEPS_host_name>:1920///cnp/kdh/lib/tep.jnlp)
     ```

  Web Start downloads and launches the desktop client.

Manually creating a shortcut for the Web Start client

On Windows, the Web Start executable file for the default Java JVM is copied to the Windows\System32 directory. When you let Web Start create a short cut for launching the desktop client, it uses the file in the System32 directory as the target. If the default JVM is not IBM Java 1.5, the shortcut will not launch the desktop client. You must create a shortcut manually.

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\Java50\jre\bin\javaws.exe
   ```
3. Click **Next** and type a name for the shortcut in the Select a Title for the Program window. For example:
   ```
   ITM Web Start client
   ```
4. Click **Finish**. The shortcut appears on your desktop.

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

1. On the computer where the desktop client is installed, start Manage Tivoli Monitoring Services:
   - Windows Select **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.
   - Linux UNIX Change directory (cd) to *Install_dir*/bin and enter `./itmcmd manage`.
2. Right-click **Tivoli Enterprise Portal – Desktop** 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**.

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 onto the 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 Tivoli Enterprise 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. If your organization has multiple managed networks, you can start a separate instance of the browser and log on to a different portal server from the same computer. There is no limit, other than the practical limit imposed by resources, to how many portal server environments you can manage from one workstation as long as you start a new instance of your browser for each portal server. You cannot log on to a portal server in a browser window and then, from the same window, log on to another portal server.

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 Monitoring Services window, right-click the **Tivoli Enterprise Portal Browser** entry and click **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 Tivoli Enterprise 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 36

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 Monitoring Services (**Start > (All) Programs > IBM Tivoli Monitoring > Manage Tivoli Monitoring Services**).
  2. In the Manage Tivoli 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).

Secure Socket Layer transmissions

Information transmitted over a network from one component of Tivoli Management Services to another can be encrypted. Changes to the Secure Socket Layer (SSL) configuration for the Tivoli Enterprise Portal Server can be made at anytime.

Security is enhanced by the use of the Global Security Toolkit (GSKit) for SSL communications and the iKeyMan utility for security certificates. A default certificate and key is provided with your installation. If you prefer to have a self-signed certificate, use the iKeyMan utilities to create and load the certificate and key database. A stash file provides the database password for unattended operation.

See "Using SSL between the portal server and the client" in the *Installation and Setup Guide*for more information.

Enabling TIP Web Service for Tivoli Integrated Portal charts

If your product is based on the Tivoli Integrated Portal infrastructure and you want to build Tivoli charts with values from your Tivoli Monitoring environment, enable the ITMWebService.

Before you begin

Single sign-on must be enabled for users of the Tivoli Integrated Portal administrative console and Tivoli Enterprise Portal.

About this task

Complete these steps on the computer where the Tivoli Enterprise Portal Server is installed.

Procedure

1. Copy the **kfwtipewas.properties** file to the portal server directory:

   Windows From `<itm_installdir>\CNPS\SQLLIB\` to `<itm_installdir>\CNPS`.

   Linux or UNIX From `<itm_installdir>/<platform>/cq/sqllib/` to `<itm_installdir>/<platform>/cq/`.
2. Reconfigure the Tivoli Enterprise Portal Server.

What to do next

In the previous release you needed to have workspace administrator authority for the user who wanted to see query-based views. As a Tivoli Enterprise Portal user you are entitled to see certain workspaces in the portal that belong to a particular monitored application based on your permissions. If you are entitled to see, say, Linux workspaces in the portal, then those workspaces will be available in the Tivoli Integrated Portal.

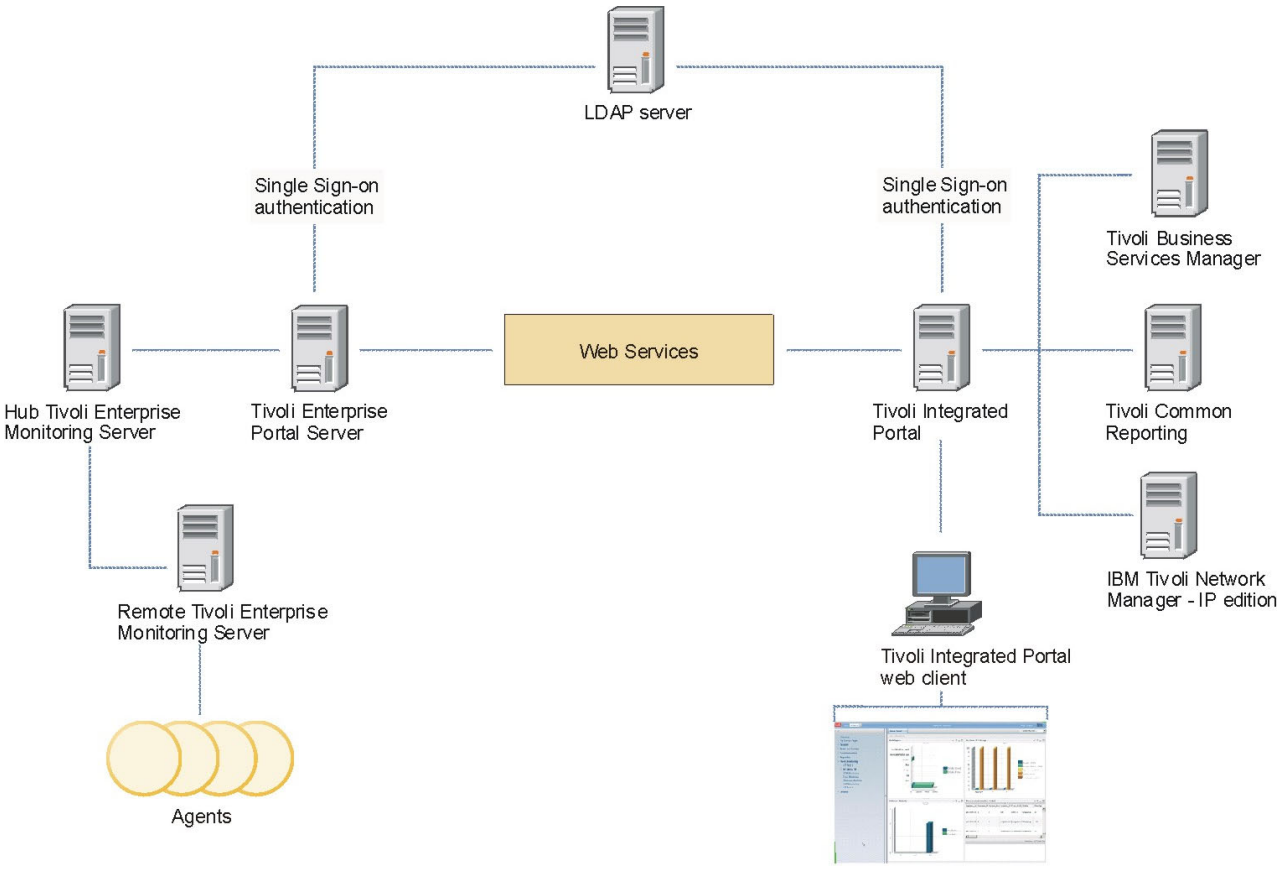


Figure 1. Tivoli Integrated Portal Web Services and the cross-product connections

Chapter 3. Editing the portal 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 Tivoli Enterprise Monitoring Server.

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 Web Start, 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 Monitoring Services. For the browser client and Web Start, this is the computer where the portal server is installed; otherwise, it is where the desktop client is installed.
   a. Windows Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.
   b. Linux Change to the <*itm_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 Web Start); 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"

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 47.

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 47.

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 Tivoli Enterprise Portal. 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 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 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 mode 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 41.

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**.

legacy_life cycle

With Sun Java versions 1.6.0_10 or higher, a new plug-in architecture was introduced and established as the default plug-in. IBM Tivoli Monitoring browser clients do not run with this new plug-in architecture. To use the Sun 1.6.0_10 (or higher) JRE, set this parameter to **true**.You will also need disable the *next-generation Java plug-in* on the computer where the browser client is being run: Launch the Java Control Panel for the Sun JRE. In the **Advanced** tab, expand the **Java Plug-in** branch. Clear the ☐ **Enable the next-generation Java Plug-in (requires browser restart)** check box.

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
Specifies the language code of the user's locale preference (de, en, es, fr, it, ja, ko, pt, zh). As well as the language, the time, date, currency, and number separator formats are converted for the locale. 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 mode users can enter the text below directly into their Java plug-in runtime parameters if they do not want to change these environment variables or their operating system locale.

-Duser.language=xx

-Duser.region=XX

where xx is the language and XX is the locale: de_DE, en_US, en_GB, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, zh_CN, and zh_TW (such as pt_BR for Brazilian Portuguese and zh_TW for Traditional Chinese).

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.

user.region
Specifies country code of user's locale preference (DE, US, UK, ES, FR, IT, JA, KR, BR, CN, TW). See also the description for **user.language**.

Related tasks

"Editing the client parameters" on page 35

"Controlling the size of event attachments" on page 47

"Starting the browser client on another portal server" on page 28

Related reference

"Portal server environment variables" on page 45

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 1. 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 *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 <*itm_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 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*.

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 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 Tivoli Enterprise 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 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 repository information is handled by TEPS/e and is not affected by Tivoli Management Services directly. See step 5 on page 69.

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"

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 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 monitoring server as described in the topic, "Controlling the number of logon attempts" on page 48.

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 44

Related reference

"Portal client parameter list" on page 36

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 Monitoring Services:
     Windows Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.
     Linux Change to the <*itm_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 Web Start); 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 onto 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 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 36

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"
  ```

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 process 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 process 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 supposed to remain connected to the monitoring server in your environment, consider defining a situation that reports that the agent is offline. Then you can take the actions required to reconnect it so that the situations will continue to be 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, a display item, reflex action, an Until clause, or dynamic thresholding with expression override

Duper situation _Z_ identifier

You can verify that a duper process 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 will be skipped.You can open the KBBENV file by right-clicking the **Tivoli Enterprise Monitoring Server** entry in the Manage Tivoli Monitoring Services window and clicking **Advanced > Edit ENV File**. Alternatively, you can open KBBENV in Linux UNIX `<itm_installdir>/tables/<tems_name>` or in Windows `<itm_installdir>\CMS`.

Enabling FIPS for the Tivoli Enterprise Portal Server

You must configure the Tivoli Enterprise Portal Server to enable the Federal Information Processing Standard (FIPS).

About this task

Follow these steps to enable FIPS 140–2 for the portal server.

Procedure

1. Do one of the following to open the Tivoli Enterprise Portal Server environment file on the computer where the portal server is installed:
   - In Manage Tivoli Monitoring Services, right-click the Tivoli Enterprise Portal Server and click **Advanced > Edit ENV file**.
   - From the Windows command line, change to the `<install_dir>\cnps` directory and open **kfwenv** in a text editor. (The default `<install_dir>` is `C:\IBM\ITM\`.)
   - From the Linux or UNIX command line, change to the `<install_dir>/config` directory and open **cq.ini** in a text editor. (The default `<install_dir>` is `/opt/IBM/ITM`.)
2. Save and close the environment file.
3. Recycle the portal server.

Results

FIPS is now enabled on the portal server.

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 Web site at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm.

All IP.SPIPE connections and SSL-enabled LDAP connections utilize TLS 1.0 protocol only. SSL must be enabled between the Tivoli Enterprise Portal client and the Tivoli Enterprise Portal Server, and is described in the *IBM Tivoli Monitoring Installation and Setup Guide* topic, "Using SSL between the portal server and the client". Failure to enable SSL might expose credentials.

Enable IP.SPIPE between all 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. Chapter 4 describes in detail how to replace cryptographic certificates. If your environment uses the the provided GSKit utilities, the '-fips' flag must be included in all operations. Refer to your local security administrator or to the NIST Web site for more details on FIPS 140-2 compliance.

Chapter 4. Setting up asymmetric encryption

Setting up asymmetric encryption through the use of public-private key files involves creating a new key database, requesting a new public-private key pair, adding the CA-signed digital certificate to your key database, and enabling components to access the certificate.

For additional information on these procedures, see the IKeyMan user guide on IBM developerWorks®

During installation, the key file names are specified with these parameters in the Tivoli Enterprise Portal Server environment file:

- 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

Setting the JRE for GSKit and starting Key Manager

You must set the path to the Java Runtime Environment before starting GSKit. Otherwise, you will get an error like "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. Change the directory to GSKit path\bin and run this command:
     gsk7ikm.exe
  5.
- Linux UNIX
  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 GSKit Key Manager by running the command that corresponds to your system:
     - **HP 32–bit**: GskitInstallDir/bin/gsk7ikm_32
     - **Linux, AIX®, or Solaris 32–bit**: GskitInstallDir/bin/gsk7ikm
     - **64–bit**: GskitInstallDir_64/bin/gsk7ikm_64

Creating a new key database

Create a new key database using iKeyman.

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 SSL 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 5. Enabling user authentication

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

Tivoli Enterprise Portal user profile
Each user IDs that is defined in the Tivoli Enterprise Portal is assigned a set of permissions that determine the portal 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.

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 and add more user accounts. No password is required to log on to the portal unless the hub monitoring server was configured to enable **Security: Validate User**.

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

User IDs that need to make SOAP Server requests (including user IDs that issue CLI commands that invoke SOAP server methods) 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.

Authentication through the portal server
User IDs authenticated through the portal server can be authenticated only by an external LDAP-enabled registry.

User IDs that require single sign-on (SSO) capability must be authenticated through the portal server to an LDAP registry.

Authentication through the hub monitoring server and the portal server
The hub monitoring server and portal server can connect to the same LDAP server. You can use the same user ID to log on to the portal server that you use for the CLI tacmd command login. To do this, you must go to **Administer Users** in the portal client to map the user ID to the ITMSSOEntry repository; *not* the DEFAULTWIMITMBASEDREALM repository.

Note: The ITMSSOEntry is the name of the portal server LDAP repository if AD2000, AD2003, or IDS6 is selected during portal server configuration. (See "Using Manage Tivoli Monitoring Services to configure the portal server for LDAP authentication" on page 68 or "Using the Linux or UNIX command line to configure the portal server for LDAP authentication" on page 71.)

Migrating LDAP authentication from the hub to the portal server

If your hub monitoring server has already been configured to authenticate users to an LDAP registry, and you now want to migrate to authentication through the portal server using the same LDAP registry, you must change the Distinguished Name that is set for the user ID in the Administer Users window of the Tivoli Enterprise Portal.

Related concepts

"User authentication through the portal server" on page 64

"User authentication through the hub monitoring server"

User authentication through the hub monitoring server

User authentication through the hub Tivoli Enterprise Monitoring Server hub can be done by either the local system 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 2. Tasks to complete before configuring authentication.

Task	Where to find information
Set up Tivoli Enterprise Portal user accounts.	"Adding a user ID" on page 86
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 *IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS*.
To use SSL (Secure Socket Layers) for communication between the hub and an LDAP server, set up a CMS key store and key store stash using GSKit and to import any required certificates.	Chapter 4, "Setting up asymmetric encryption," on page 53

If you intend to authenticate using the hub 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 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 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 this table from the LDAP administrator before configuring user authentication.

Table 3. 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 repository 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.
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 intend to use SSL communication between the hub monitoring server and the LDAP server, obtain the information described in this table.

Table 4. 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 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 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 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 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 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 monitoring server:

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

Configuring authentication by using Manage Tivoli Monitoring Services:

Configure authentication by using Manage Tivoli Monitoring Services.

About this task

To configure authentication by using Manage Tivoli 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 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 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 Tivoli Enterprise 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 SSL operations that are used in Tivoli Monitoring and has additional command-line options to support LDAP SSL searches. Tivoli Monitoring does not include ldapsearch with production installations. For information on Tivoli Directory Server ldapsearch, see *Tivoli Directory Server Command Reference*, Client utilities.

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 Web site 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 TEMS configuration file.

Table 5. ldapsearch command line options and corresponding monitoring server configuration parameters

Option	Description	Corresponding parameter in TEMS 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 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 SSL)

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

Use the following values to configure the ldapsearch command in an environment where 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 SSL)

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

Use the following values to configure the ldapsearch command in an environment where 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
...
```

User authentication through the portal server

Configure authentication by an external LDAP registry through the Tivoli Enterprise Portal Server. This is required if you intend to provide single sign-on (SSO) capability, whereby users can log on to the Tivoli Enterprise Portal and then launch to other Tivoli Web-based or Web-enabled applications without having to re-enter their credentials.

Prerequisites for configuring authentication on the portal server

Before configuring 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 repository 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 repository, two entries with the same name cause a conflict.

 A best practice is to not add **sysadmin** to the LDAP repository. 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.

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 repository.

Windows **The sysadmin password**

The Windows installer creates a **sysadmin** user account in the Windows registry 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.

Table 6. 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 • Tivoli Directory Server 6.x • Other **Other** is specified if you are configuring a different LDAP server or need to customize the LDAP configuration (or both). For example, if you are using a federated user registry with a base distinguished name that is different from the default **o=ITMSSOEntry**, select **Other** and specify the distinguished name in the TEPS/e administration console. See "TEPS/e administration console" on page 72.

Table 6. LDAP configuration parameters (continued)

Parameter	Description

LDAP base	This is the LDAP base node in the LDAP repository 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	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	can be omitted if the LDAP server is on the same computer as the portal server.

Information for SSO configuration

If you intend to configure SSO, obtain the information described in this table from the LDAP administrator.

Table 7. SSO parameters

Parameter	Description
Domain name	This is the Internet or Intranet domain for which SSO is configured. Only applications available in this domain or its subdomains are enabled for SSO. Example: `ibm.com`
Realm name	This is a parameter shared across applications that are using the SSO implementation and are configured for SSO within the specified domain. Applications configured for the same domain name, but for a different realm name will not work as a part of the same SSO infrastructure. Example: `ibm_tivoli_sso`

Keys to encrypt LTPA tokens

If you intend to export the keys used to encrypt LTPA tokens generated by the TEPS/e, as part of the configuration process, you must provide a name for the key file and password to use to encrypt the key. If you intend to import the keys used by other participating applications, you must provide the name of the key file and the password required to decrypt the file.

You can export and import keys at any time in Manage Tivoli Monitoring Services using the **Advanced → TEPS/e Administration** options for the Tivoli Enterprise Portal Server.

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 those applications, without having to re-enter their credentials. 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
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.

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.

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 registry that is shared by all participating Tivoli applications. All the participating applications must be configured for SSO and must belong to the same security domain and realm.

The domain is the Internet or Intranet domain for which SSO must be configured. Only applications available in this domain or its subdomains are enabled for the SSO.

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 (Lightweight Third Party Authentication) tokens. An LTPA token is encrypted data containing previously authenticated user credentials. Participating SSO applications pass LTPA tokens using browser cookies.

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. The Tivoli Enterprise Portal administrator must export the keys used by the portal server and make them available to participating applications. The administrator must also import keys from other SSO servers into the portal server.

Tasks to enable SSO

After the user IDs available for SSO have been established in the LDAP repository, enable SSO by completing the following tasks:.

- 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.
- Export portal server LTPA keys and import keys from participating Tivoli Web applications.
- Map Tivoli Enterprise Portal user IDs to LDAP distinguished names.

Related tasks

"Reconfiguring the browser client for SSO" on page 77

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

Configure the Tivoli Enterprise Portal Server to set up user authentication for an external LDAP repository and, optionally, single sign-on capability. This involves adding LDAP information such as the bind ID and port number to the portal server configuration through Manage Tivoli Monitoring Services.

Before you begin

If you want to export and 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.

Have the configuration information for the LDAP server at hand, as well as the realm and domain names for single sign-on if you plan to enabled it for users.:

About this task

Take these steps to reconfigure the portal server for user validation with an LDAP registry and, optionally, enable Single Sign-On.

Procedure

1. Start Manage Tivoli Monitoring Services:
   - Windows Click **Start → Programs →IBM Tivoli Monitoring → Manage Tivoli 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]`.
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 want to add the export and import LTPA keys for 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
   - **IDS6** for IBM Tivoli Directory Server Version 6.x.
   - **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. For example, you are using a federated user registry with a base distinguished name that is different from the default o=ITMSSOEntry 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 "TEPS/e administration console" on page 72.
     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 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 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:
   a. **LDAP base** is the LDAP base node in the LDAP repository that is used in searches for users.
   b. **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.
   c. **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.
   d. **LDAP port number** that the LDAP server is listening on. This value can be omitted if the port is 389.
   e. **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, click **OK** to close any other portal server configuration dialogs.
8. For SSO, specify the realm and domain in the Single Sign On dialog:
   a. **Realm name** is a parameter shared across applications that are using the SSO implementation and are configured for SSO within the specified domain. 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. Only applications available in this domain or its subdomains are enabled for SSO. .

9. For applications participating in SSO, if you want to export the key used to encrypt and decrypt the LTPA tokens generated by the portal server, 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 keys 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.
10. For applications participating in SSO, if you want to import keys used by other applications to encrypt their LTPA tokens, 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.

With the LDAP registry configured, you can map the Tivoli Enterprise Portal user IDs to the LDAP distinguished names to complete the 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.

Related concepts

"TEPS/e administration console" on page 72

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 configure the portal server for LDAP authentication, and optionally single sign-on, from the command line.

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:

   ```
   Edit 'ITM Connector' 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 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):
   ```

7. Enter 1 to begin configuration of LDAP authentication and provide the values for the LDAP parameters.

   ```
   LDAP type: [AD2000, AD2003, 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. For example, you are using a federated user registry with a base distinguished name that is different from the default o=ITMSSOEntry 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 "TEPS/e administration console" on page 72.

8. Continue to enter the values prompted for (see Table 6 on page 65):

   ```
   LDAP base: 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
   ```

9. 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):
   ```

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

10. Recycle the portal server.

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

What to do next

If you enable single sign-on, ensure that the Tivoli Enterprise Portal administrator exports the LTPA keys for exchange with other participating applications and import the keys from those applications.

TEPS/e administration console

The Tivoli Enterprise Portal Server extended services (TEPS/e) has an administrative console that you can access for configuring an LDAP registry that is not supported by the Tivoli Management Services installation and configuration utilities.

As well as to configure an LDAP registry that you specified as **Other** in the portal server LDAP configuration dialog, use the TEPS/e administration console for these tasks:

- Verify the LDAP configuration parameters.
- Configure SSL communication between the portal server and the LDAP server. (Use the administration console help for instructions to enable SSL communication.)
- To specify a different distinguished name suffix than the default o=ITMSSOEntry that is supplied when LDAP authentication is through the portal server.

Important: Any customizations 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. When **Other** is chosen, the repository information is handled by TEPS/e and is not affected by Tivoli Management Services directly. See step 5 on page 69.

Starting the TEPS/e administration console

Use the TEPS/e administration console to configure an "Other" 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 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*`/ls3263/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 theManage Tivoli 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
   ```
4. Log onto 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.

What to do next

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

SSL communication between the portal server and the LDAP server is configured using the TEPS/e administration console. Consult the administrative console help system for instructions.

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 an external LDAP data server

You must use the Tivoli Enterprise Portal Server extended services (TEPS/e) administration console to configure LDAP servers that are not supported by the installation and configuration utilities,

Procedure

1. In the TEPS/e administration consolenavigation tree, click **Security → Secure administration, applications, and infrastructure**.
2. On the page that is displayed, ensure that `Federated repositories` is selected as the realm definition, and click **Configure**.
3. Configure the federated repository:
   a. Verify or enter the **Realm Name** value.

b. On the same page, under **Related Items**, select `Manage repositories`.

4. Click **Add**. The page now displays the properties for the portal server to LDAP connection.
5. Provide the appropriate values for the parameters. (See "Prerequisites for configuring authentication on the portal server" on page 64.) For **Repository identifier**, enter a name for the repository that you find meaningful.
6. Click **OK** to accept the settings.
7. Return to the Secure administration, applications, and infrastructure page and **Configure** the federated repositories.
8. Select `Add Base entry to Realm` to attach the LDAP repository you just configured:
   - For Repository, use the repository identifier you just specified.
   - For the distinguished name of the base entry enter a value that uniquely identifies the set of entries in the realm, using the class `o`. For example, `o=itmssoentry`
   - For the distinguished name of the base entry, enter the node under which the SSO entries are located. For example, `ou=teps,dc=ibm,dc=us,dc=com`.
9. Click **OK** to accept the settings.
10. If you want to enable SSO:
    a. Return to the Secure administration, applications, and infrastructure page and **Configure** the federated repositories.
    b. Select the **Single sign-on (SSO)** link to complete the SSO configuration.
    c. Verify that SSO is enabled and that the Domain Name parameter is correct.
    d. 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. Restart the Tivoli Enterprise Portal Server.

Results

Users logging into the portal server are now authenticated through the external LDAP repository.

What to do next

You might also need to map the Tivoli Enterprise Portal user IDs to the LDAP UIDs. See "Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 75.

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

Important: Any customizations 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. When **Other** is chosen, the repository information is handled by TEPS/e and is not affected by Tivoli Management Services directly. See step 5 on page 69.

If your organization needs to use Tivoli Enterprise Portal IDs longer than 10 characters, download *TEPS/e User Administration using Active Directory* from the Open Process Automation Library (OPAL) and follow the method to enable Active Directory or other LDAP sources for the User ID mappings.

Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names

One of the items passed in an LTPA token is a valid user name. Because the user name is being authenticated by a central LDAP repository that is shared by participating Tivoli applications, this name is the user's unique identifier (UID) as known by the LDAP registry. 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 UIDs.

Before you begin

User IDs are mapped to LDAP UIDs in the portal Administer Users window by a user with administrator authority.

If 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 user IDs:

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.
   Example:`UID=TEPUSER,O=SS`.

   Note: The default suffix for LDAP servers that are configured through the Tivoli Enterprise Portal Server is o=ITMSSOEntry, which is created by the Tivoli Enterprise Portal Server extended services. The ITMSSOEntry repository is not created if the portal server was configured for LDAP authentication with an LDAP type of “Other”.
   If you want to use a federated user registry with a different DN suffix, you must configure the portal server for an LDAP type of “Other”, and then configure the LDAP server and the DN suffix through the TEPS/e administration console.
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

After you have configured a user ID for single sign-on, if the user is not successful at launching 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.

Related reference

"Tivoli Enterprise Portal distinguished names" on page 77

Importing and exporting LTPA keys

Export the keys used to encrypt the LTPA tokens that are generated by the TEPS/e to other applications participating in SSO. Import keys from other participating SSO applications if it was not already done as part of configuring SSO.

Before you begin

When you request an export or import operation, you must provide the name of the key file to export or import and the password to use to encrypt or decrypt the file. The Tivoli Enterprise Portal Server must be running for import and export operations to be performed.

About this task

Follow the steps for your environment to import or export LTPA keys:

Procedure

- From Manage Tivoli 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 keys 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 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 keys, 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 keys, run **./importKeys.sh *<filename>* *<password>***. The script is installed to *ITM_dir*/*platform*/iw/scripts.

Related tasks

"Migrating authentication from the monitoring server to the portal server" on page 78

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 Monitoring Services utility.
2. Right-click the Tivoli Enterprise Portal Browser entry and click **Reconfigure** to open the Configure Enterprise Portal Browser window.
3. In the **Host** field beneath TEP Server, type the fully-qualified name of the computer. Example: `myhost.mycompany.com`

Related concepts

"About single sign-on" on page 66

Tivoli Enterprise Portal distinguished names

A distinguished name (DN) is a unique name that unambiguously identifies a single entry in a tree-like structure called the Directory Information Tree (DIT). Each DN is constructed of a relative distinguished name (RDN), constructed from some attribute or attributes in the entry, following by the DN of the parent entry. The Tivoli Enterprise Portal Server requires distinguished names in addition to user IDs for each user account.

Portal server configured for LDAP authentication

If you are creating new users, and you have already configured the portal server with LDAP and all of your users have entries in the LDAP registry, then you will have to map new users to their corresponding LDAP distinguished names in the Tivoli Enterprise Portal Administer Users window.

Default DN

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

`UID=tep_userid,O=DEFAULTWIMITMBASEDREALM`

If you are upgrading from ITM V6.2, distinguished names are automatically created for existing users. This is also the default suffix for user IDs that authenticate through the hub monitoring server.

The default DN suffix for the TEPS/e user registry is o=defaultWIMFileBasedRealm. The wasadmin is the initial ID for TEPS/e:

`UID=wasadmin,o=defaultWIMFileBasedRealm`

Changing the default DN

When the portal server has been configured for LDAP authentication, the base DN is defaulted to **o=ITMSSOEntry**. If you want to use a federated user registry with a different base DN, you must specify an **LDAP type** of **Other** when configuring the portal server for LDAP. Then use the TEPS/e administration console to configure the LDAP server and to specify the base distinguished name that you plan to use. If you do not specify an LDAP type of "Other" when you change the base distinguished name, any subsequent reconfiguration of the portal server might result unexpected LDAP configuration changes.

Related tasks

"Mapping Tivoli Enterprise Portal user IDs to LDAP distinguished names" on page 75

Migrating authentication from the monitoring server to the portal server

If your environment has already been configured for LDAP configuration using the hub monitoring server and you now want to use authentication through the Tivoli Enterprise Portal Server, take the following steps.

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 disable security validation on the hub Tivoli Enterprise Monitoring Server, then change the Distinguished Name to the one associated with the portal server.

Procedure

1. Disable Tivoli Enterprise Monitoring Server security validation:
   - Windows Use the Manage Tivoli 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 to be through the portal server. Use the Manage Tivoli Monitoring Services utility or the command line to configure the portal server.
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.

Results

At this point, the migration is complete.

What to do next

You can always reinstate authentication through the Tivoli Enterprise Monitoring Server for users who want to use the Tivoli Web Services for sending SOAP requests.

Related concepts

Chapter 5, “Enabling user authentication,” on page 57

“User authentication through the portal server” on page 64

“User authentication through the hub monitoring server” on page 58

Related tasks

“Importing and exporting LTPA keys” on page 76

Chapter 6. User administration

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 repository, 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.

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.

Related tasks

"Adding a user ID" on page 86

"Viewing and editing a user ID" on page 88

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 lists. 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.

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 repository 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. 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.

About this task

Take these steps to add a new user:

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) repository 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 Logon window has a field for entering a password. If you want the user ID to include a password, you must define the same user ID, including a password, to your network domain user accounts or to the operating system where the hub monitoring server is installed. Also, the monitoring server must be configured to validate users, which is the default on the Windows-based hub monitoring server. (In **Manage Tivoli Monitoring Services**, right-click **Tivoli Enterprise Monitoring Server** and click **Reconfigure**.)

Related reference

"Administer Users" on page 81

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 repository. 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 81

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 Server. 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.

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 hub 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 Monitoring Services program:

   Windows Start → Programs →IBM Tivoli Monitoring → Manage Tivoli 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 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. 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*:1920///cnp/client to the IP address of the Tivoli Enterprise Portal Server. For example, http://10.21.2.166:1920///cnp/client.

   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. 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* 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 Tivoli Enterprise Console events

This section provides information about attribute mapping of situation events to Tivoli Enterprise Console events. You can use this mapping information when you forward a situation event to the Tivoli Enterprise Console and you want to write correlation rules in the Tivoli Enterprise Console.

The situation event forwarder generates a 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 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 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 8. 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 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 (EIF) 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 8. 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 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 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 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 Tivoli Enterprise Console event. In case the attribute name conflicts with the slot names in 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 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 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 9. Special characters for attribute groups and names in 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 10. Situation name suffix mapping to Tivoli Enterprise Console event severity

Situation name suffix	Assigned 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 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 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 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 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 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 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 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 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 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 Tivoli Enterprise Console command:

```
wlsesvrcfg -c
```

To set this rules cache size, run the 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 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 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 Tivoli Enterprise Console rules engine. Also, the situation event update is not forwarded to the originating Tivoli Enterprise Monitoring Server. This behavior results from theTivoli 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 Tivoli Enterprise Console database.

Both situations can be remedied by performing a 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 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 two following options:two options for running this command:

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 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 acknowledgement of the situation expires and the situation is still true, then a new situation event is opened in the Tivoli Enterprise Console. If the situation becomes false, then it resets itself in IBM Tivoli Monitoring and the event remains closed in the Tivoli Enterprise Console.

The default acknowledgement 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 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 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 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 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 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 IBM Tivoli Enterprise Console information center.

Tuning considerations

Integration parameters supporting actions at the 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.
- 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 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 Tivoli Enterprise Console, change the allowable range using the 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:

  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 Tivoli Enterprise Console class. The other classes usually derive from the Tivoli Enterprise Console EVENT class.

  In 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 Tivoli Enterprise Console Server. Although the tool is closely integrated with 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 Tivoli Enterprise Console product rule language also supports the inheritance nature of the Tivoli Enterprise Console class definitions. When a predicate in the Tivoli Enterprise Console rule is looking for a particular class, all classes that inherit from that particular class also satisfy the rule predicate.

In 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 Tivoli Enterprise Console Server. Although the tool is closely integrated with 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 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.
- 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 Tivoli Enterprise 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 Monitoring Services or with the CLI itmcmd config -S and then recycle the monitoring server and Tivoli Enterprise Portal Server.

See *IBM Tivoli Monitoring Installation and Setup Guide* 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:
   a. Windows In the Manage Tivoli Monitoring Services window, right-click Tivoli Enterprise Monitoring Server and click **Advanced → Edit EIF Configuration**.
   b. 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**
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 *IBM Tivoli Monitoring Command Reference* for a description.

Related reference

Tivoli Event Integration Facility Reference

Tivoli Monitoring Installation and Setup Guide

Tivoli Monitoring Command Reference

Specifying EIF forwarding for a situation event

When the Tivoli Enterprise Monitoring Server has been configured for the **Tivoli Event Integration Facility**, all situation events are forwarded to the event receiver. Use the Tivoli Enterprise Portal Situation editor to override this default 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*). 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 no event destinations are specified for a Tivoli Enterprise Console event, the event is 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*.

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 Monitoring Installation and Setup Guide

Tivoli Monitoring Command Reference

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 *IBM Tivoli Monitoring: CandleNet Portal User's Guide* for more information.

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 Tivoli Enterprise 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, 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, use the Tivoli Enterprise Console Event Definition Generator (TEDGEN) tool supplied in the TECLIB directory on the computer where the monitoring server or portal server is installed or on the Tools DVD from the event server installation media. The computer on which you generate the MCS XML file must have the necessary BAROC files, which are located in the TECLIB directory. The exception is on a Linux- or UNIX-based portal server if the **Install TEMS support for remote seeding** option was not selected during installation (see the install.sh instruction).

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 Tivoli Enterprise Console event server or are all in the same directory (depending on the options used when running the TEDGEN tool, see next) before running the TEDGEN utility to generate the MCS XML file.

About this task

These steps have you install the TEDGEN utility on the computer where you want to run the tool: at the Tivoli Enterprise Console event server, the hub Tivoli Enterprise Monitoring Server, or the Tivoli Enterprise Portal Server. After installing the utility, run the TEDGEN command to create a new XML file for EIF Slot Customization.

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, install the TEDGEN utility from the tools DVD that comes with the event server installation media (see the README.txt file). Then create a new XML file:

     Windows

     ```
     tedgen -itmDir Install_dir\{CMS|CNPS}
     \TECLIB -id server_id -xmlPath output_xml_file_path
     ```

     Linux UNIX

     If you are generating the MCS XML file on the portal server, the BAROC files are not present by default and you must install them by running the **install.sh** script on the portal server computer and selecting the **Install TEMS support for remote seeding** option. This action places the BAROC files on the portal server under the *Install_dir*/tables/cicatrsq/TECLIB directory.

     ```
     tedgen -itmDir
     Install_dir/tables/{tems_name|cicatrsq}/
     TECLIB -id server_id -xmlPath output_xml_file_path
     ```

     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 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 XML file.
   c. Save and close the environment file.

Displaying events from the Universal Agent on the Tivoli Enterprise Console

Because all Universal Agent applications dynamically generate their catalog, attribute, and ODI files, certain manual steps are required to have Universal Agent situation events properly displayed on the Tivoli Enterprise Console.

Before you begin

For the Universal Agent situation to be properly translated to a Tivoli Enterprise Console event by the Tivoli Enterprise Console event forwarder, the Universal Agent attribute files must reside on the hub Tivoli Enterprise Monitoring Server during hub initialization. If the Universal Agent is connected to a remote monitoring server, the Universal Agent catalog and attribute files are not propagated to the hub and the translation of Universal Agent situation events fails.

About this task

Complete these steps to ensure that the Universal Agent attribute files are on the hub monitoring server and to generate a BAROC file with the Universal Agent situation event definitions, which is required to properly parse and display the Universal Agent events on the Tivoli Enterprise Console.

Procedure

1. Ensure that the Universal Agent attribute files are on the hub:
   - Temporarily connect the Universal Agent to the hub monitoring server to enable uploading of the attribute files. After successful connection, the Universal Agent can be reconfigured to connect to the remote monitoring server.
   - Manually move the Universal Agent attribute files from the remote monitoring server to the hub. Attribute file location:
     Windows *Install_dir*\CMS\ATTRLIB
     Linux UNIX *Install_dir*/ tables/tems_name/ATTRLIB.
   - Recycle the hub monitoring server.
2. Obtain the required BAROC files for the Universal Agent application
   a. Search for and download the *BAROC File Generator* from the Tivoli Open Process Automation Library at http://www-01.ibm.com/software/brandcatalog/portal/opal.
   b. Run the BAROC generator, providing the ODI file for Universal Agent application as input. ODI file location (requires that the Universal Agent has successfully connected to the monitoring server) on the computer where the Tivoli Enterprise Portal Server is installed:
      Windows *Install_dir*\cnps
      Linux UNIX *Install_dir/platform*/cq/bin
      The format of the ODI filename is *xxx*odi*nn*, where *xxx* is the application name specified agent and *nn* is the version number.
   c. After generating the BAROC file, move it to the event server, then compile and load it.

Using the NetView console through the Tivoli Enterprise Console event viewer

You can launch the IBM Tivoli NetView® Java console from the 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

Tivoli Enterprise Console rules automatically synchronize the events forwarded by Tivoli NetView to the 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 IBM Tivoli Enterprise Console information center.

If you want to use the NetView console through the 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 <*itm_install_dir*>\cnp\cnp.bat
- Linux or UNIX <*itm_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 Tivoli Enterprise Console view.

See the Tivoli Enterprise Console product documentation for more detailed information about using the NetView console.

Chapter 8. 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.

The *IBM Tivoli Monitoring Installation and Setup Guide* provides the 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.

Default mapping of situation events to OMNIbus alerts

This topic provides information about attribute mapping of situation events to OMNIbus alerts. You can use this mapping information when you forward a situation event to the Tivoli Netcool/OMNIbus ObjectServer and want to write probe rules or SQL procedures and triggers in the ObjectServer.

The situation event forwarder generates an event integration facility (EIF) event with an event class based on the attribute group used in the situation. When the situation event is forwarded to the ObjectServer, the Tivoli Netcool/OMNIbus EIF probe translates the EIF event into an OMNIbus alert format. The EIF event contains all of the attributes described by the parent *Omegamon_Base* class.

The Omegamon_Base class is described as follows:

```
Omegamon_Base ISA EVENT
DEFINES {
   cms_hostname: STRING;
   cms_port: STRING;
   integration_type: STRING;
   master_reset_flag: STRING;
   appl_label:STRING;
   situation_name: STRING;
   situation_origin: STRING;
   situation_displayitem: STRING;
   situation_time: STRING;
   situation_status: STRING;
   situation_eventdata: STRING;
  situation_type: STRING;
  situation_thrunode: STRING;
   situation_group: STRING;
   situation_fullname: STRING;  }; END;
```

As part of the generic mapping for these situations, the IBM Tivoli Monitoring event forwarder assigns associated values for each of the event class attributes when forwarding an event to OMNIbus. In addition to these event class attributes, values are assigned to the host name, origin, severity, and message attributes that are inherited from the base EVENT class.

Table 11. Tivoli Netcool/OMNIbus ObjectServer attributes

Attributes	Values and meaning
appl_label	Application-specific data related to the event, if any.
cms_hostname	TCP/IP host name of the Tivoli Enterprise Monitoring Server that forwards the event.
cms_port	Tivoli Enterprise 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 host name of the managed system where the event originates, if available.
integration_type	Indicator to help OMNIbus performance: • N for a new event, the first time the event is raised • U for update event, subsequent event status changes
master_reset_flag	Master reset indicator set for master reset events. Value is NULL for all other events: • R for 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	Event data attributes in key-value pair format. The event data can be truncated because the event integration facility (EIF)) imposes a 2 KB size limit.
situation_group	One or more situation group names (up to five) 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. This name has the same value as sub_source.
situation_status	Current status of the situation event.
situation_time	Timestamp of the situation event.
situation_type	Indicator of whether the Tivoli Monitoring situation that caused the event is a sampled or pure situation.
situation_thrunode	The hub or remote Tivoli Enterprise Monitoring Server through which the event was forwarded.
source	Base EVENT class attribute that contains "ITM."

Table 11. Tivoli Netcool/OMNIbus ObjectServer attributes (continued)

Attributes	Values and meaning
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.

The Tivoli Netcool/OMNIbus EIF probe maps the attributes of the situation event into ObjectServer attributes, which are defined in the `alerts.status` table of the ObjectServer.

Table 12. Mapping of situation attributes to OMNIbus attributes

Situation attribute	OMNIbus attribute
situation_name + situation_origin +situation_displayitem + event_class	Identifier (for ITMProblem)
situation_name + situation_origin + situation_displayitem + event_class + ITMResolution	Identifier (for ITMResolution)
situation_name	AlertKey
situation_origin	Node
situation_origin	NodeAlias
source	Agent
default	Type (20) (for ITMProblem)
situation_status = “P” and integration_type = “U”	Type (21) (for ITMResolution)
situation_status = “D” and integration_type = “U”	Type (21) (for ITMResolution)
situation_status = “N” and integration_type = “U”	Type (21) (for ITMResolution)
situation_displayitem	ITMDisplayItem
situation_status	ITMStatus
situation_thrunode	ITMThruNode
situation_time	ITMTime
situation_type	ITMSitType
situation_eventdata	ITMEventData
cms_hostname	ITMHostname
master_reset_flag	ITMResetFlag
integration_type	ITMIntType
event_class	AlertGroup
msg	Summary
“tivoli_eif probe on ”+hostname()	Manager
6601	Class

Table 12. Mapping of situation attributes to OMNIbus attributes (continued)

Situation attribute	OMNIbus attribute

severity
`FATAL / 60 = Critical`
`CRITICAL / 50 = Critical`
`MINOR / 40 = Minor`
`WARNING / 30 = Warning`
`UNKNOWN / 10 = Indeterminate`
	Severity

getdate	LastOccurrence/FirstOccurrence
date	TECDate
repeat_count	TECRepeatCount
fqhostname	TECFQHostname
hostname	TECHostname
cms_port	ITMPort
situation_fullname	ITMSitFullName
situation_group	ITMSitGroup
appl_label	ITMApplLabel

Expanding the description of a generic event message situation

The OMNIbus EIF probe maps the message slot in the EIF event sent from the Tivoli Enterprise Monitoring Server into the summary attribute of the ObjectServer. The summary attribute gives you a descriptive way of looking at an alert in OMNIbus.

The situation name alone does not provide detailed event identification where there are large numbers of like-events from various sources. The situation name in the summary attribute that is sent from the hub monitoring server to the ObjectServer 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 that determines 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 value is optional and is used only if a display item is specified in the situation definition.

threshold Name-Value pairs
The raw data that the situation uses to evaluate whether it is triggered.

```
Examples:
NT_Criticial_Process [(Process_CPU > 4 AND Thread_Count > 50)
  ON IBM-AGX02:NT
```

```
  (Process_CPU = 8 AND Thread_Count = 56)]

NT_Disk_Full [(Free_Megabytes < 1000000)
  ON "IBM-AGX02:NT"
  ON D: (Free_Megabytes = 100)]
```

Generic mapping for agent specific slots

Generic mapping identifies the target event class based on information from a situation that is triggered and forwarded to the OMNIbus EIF probe.

For situation events that do not have a mapping specified for the forwarded event, an event is generated with a unique class based on the attribute group used in the situation. The class name of the EIF event is a combination of **ITM_** plus the attribute group name associated with the situation.

For example, a situation using the NT_Process attribute group generates a Tivoli Enterprise Console event with class *ITM_NT_Process*.

Additional event slot values are populated with situation attribute values from the situation event data. The slot names are the attribute names. These additional slot values can be used to write additional OMNIbus EIF probe rules.

For example, a situation using the Process_CPU attribute causes a process_cpu slot in the EIF event to be generated and forwarded to the OMNIbus EIF probe. If the attribute name conflicts with the slot names in Tivoli Enterprise Console EVENT class or Omegamon_Base class, the *applname* associated with the attribute group, for example: knt_ , is prepended 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 EIF event class used is derived from the first attribute group encountered in the situation event data of the triggering situation. For example, if a situation is written for the NT_Process and NT_System attribute groups, where NT_Process is the first attribute group, the EIF ITM_NT_Process event class is used. Additional event slots are generated based on the attributes of the attribute group only.

Table 13. Special characters for attribute groups and names in EIF events generated from forwarded situation events

Character:	**Converts to:**
<uppercase> (applies only to attribute name)	<lowercase> (applies only to attribute name)
% percent sign	pct_
I/O	io
/ forward slash	_per_
\ backward slash	_ (underscore)
<space>	_ (underscore)
(open parenthesis) close parenthesis	_ (underscore)
< open pointed bracket > close pointed bracket	_ (underscore)

All strings and timestamp types are mapped to STRING types, and all integer types are mapped to INTEGER in the event class definition. No default values are assigned to the attribute slots. Attributes that have a specified non-zero scale/precision value are mapped to the string type of REAL.

Note: If you are mapping from an attribute to a slot and the resulting slot name has a trailing underscore, the trailing underscore is removed in the final slot name, which never has a trailing underscore.

Localizing alert summaries

Edit the KMS_OMTEC_GLOBALIZATION_LOC variable to enable globalization of the EIF event message slots that get mapped to alert summaries by the OMNIbus EIF probe.

About this task

By default, this variable is set to American English and the message slots are filled with the American English messages. Take these steps to edit the variable to enable any 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.

Synchronizing situation events

During installation, the hub Tivoli Enterprise Monitoring Server was configured to forward situation events to the Netcool/OMNIbus Probe for Tivoli EIF. Changes in the status of events made on the Netcool/OMNIbus ObjectServer are reported back to the forwarding monitoring server so that events are synchronized on the two event management systems. You can change the event synchronization configuration and define additional monitoring servers for event forwarding.

Changing the configuration of the event synchronization on the event server

The following procedure can be used to change the configuration of the event synchronization on the event server.

About this task

If you want to change any of the settings for the event synchronization on the event server, use the **sitconfig.sh** command. 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 *<event_sync_installdir>*/etc and then run: **sitconfig.sh update *<config_filename>***
- Run the **sitconfig.sh** command directly, specifying only those settings that you want to change. See *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 *<event_sync_installdir>*/bin directory with stopSUF.sh and startSUF.sh.

Defining additional monitoring servers for the event synchronization on the ObjectServer

For each monitoring server that is forwarding situation events to the ObjectServer, 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.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process from the *<event_sync_installdir>*/bin directory: On Windows systems, run the stopSUF.cmd and startSUF.cmd commands; on operating systems such as UNIX, run the stopSUF.sh and stopSUF.sh commands.

Deleted or cleared sampled situation events

When a situation event from a sampled situation is forwarded to the Tivoli Netcool/OMNIbus ObjectServer and that event is subsequently deleted or cleared

in the ObjectServer, the behavior of event synchronization is to send a request to the Tivoli Enterprise Monitoring Server to acknowledge the situation with a specified timeout. Closing sampled situations interferes with the situation firing after the situation closes in IBM Tivoli Monitoring.

If the acknowledgement of the situation expires and the situation is still true, then a new situation event is opened in the ObjectServer. If the situation becomes false, then it resets itself in IBM Tivoli Monitoring and the event remains closed in the ObjectServer.

The default acknowledgement expiration time for sampled situations is 59 minutes. This timeout value can be changed in the situation timeouts configuration file on the ObjectServer: **sit_timeouts.conf**. Also, expiration times for individual situations can be configured in this file.

After editing this file, you can have the expiration times dynamically loaded into the ObjectServer using the `sitconf` refresh command in the *`Event_Sync_Installdir`*`/bin` directory. You must specify the `pathc` and `type` parameters with this command. See the Netcool/OMNIbus ObjectServer **sitconf** command in the *IBM Tivoli Monitoring Command Reference* for details.

Customizing the OMNIbus configuration

The get_config_parms procedure in the <event_sync_install_dir>/omnibus/itm_proc.sql file defines three configuration parameters:

```
set sit_ack_expired_def_action = 'REJECT'
set sit_resurface_def_action = 'ACCEPT'
set situpdate_conf_file = 'situpdate.conf'
```

The sit_ack_expired_def_action variable defines the action to be taken for an event by the OMNIbus server when acknowledgement expiration information is received for an event from a monitoring server. The default action is to Reject the request. OMNIbus sends information to change the state of the event to Acknowledge back to the monitoring server. If you want to change the action taken by the OMNIbus server to Accept the acknowledgement expiration, modify the statement to set sit_ack_expired_def_action = 'ACCEPT'.

The sit_resurface_def_action variable defines the action to be taken by the OMNIbus server when a situation event has resurfaced. The default action of the OMNIbus server is to Accept this request and Deacknowledge the event. If you want to change the action taken by OMNIbus server to Reject the resurface of the event, modify the statement to set sit_resurface_def_action = 'REJECT'. OMNIbus then sends information back to the monitoring server to change the state of the event back to Acknowledge.

The situpdate_conf_file variable specifies the name of the configuration file to be used by the SitUpdate Forwarder. If you want to change the name of the configuration file, modify the statement to set situpdate_conf_file = '*newname.conf*'.

After modifying itm_proc.sql, issue the following command:

Windows

```
%OMNIHOME%\..\bin\redist\isql -U <username>
      -P <password>
      -S <server_name>
      < <path_to_file>\itm_proc.sql
```

Linux or UNIX

```
$OMNIHOME/bin/nco_sql -user <username>
     -password <password>
     -server <server_name>
     < <path_to_file>/itm_proc.sql
```

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 size of the event cache.

Before you begin

When the **Tivoli Event Integration Facility** (EIF) has been enabled on the hub monitoring server and the default EIF server (Tivoli Enterprise Console Event Server or Netcool/OMNIbus EIF probe) and port number have been specified, the EIF configuration file is updated with the information. This is the default EIF receiver of forwarded situation events.

See the *IBM Tivoli Monitoring Installation and Setup Guide* 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:
   a. Windows In the Manage Tivoli Monitoring Services window, right-click Tivoli Enterprise Monitoring Server and click **Advanced** → **Edit EIF Configuration**.
   b. 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 OMNIbus EIF probe. 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=	OMNIbus EIF probe listening port, which is 5529 by default. Specify `0` if the OMNIbus EIF probe uses the port mapper. If you specified multiple server locations, add the corresponding port numbers here, separating each with a comma. Value: **[port:0]**

Option	Description
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**
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**
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 in to the command-line interface with **tacmd login** , then run **tacmd refreshTECinfo -t eif** to complete the EIF configuration. See the *IBM Tivoli Monitoring Command Reference*.

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 command, **tacmd createEventDest**.

Specifying situation events that send an OMNIbus event

When the Tivoli Enterprise Monitoring Server has been configured for the **Tivoli Event Integration Facility**, all situation events are forwarded to the Tivoli Netcool/OMNIbus Probe for Tivoli EIF. Use the Tivoli Enterprise Portal Situation editor to override this default for individual situations.

About this task

Complete these steps to specify the destination EIF receiver and severity for a forwarded event:

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*.
7. Save the situation definition with your changes by clicking **Apply**, to keep the Situation editor open, or **OK**, to close the Situation editor.

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 in the Tivoli Enterprise Portal situation editor, 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 Netcool/OMNIbus ObjectServer.

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, " quote, ' single quote, and & ampersand. This column is available only if no value is selected in the Mapped attribute column.

See the topic on "Forwarding the event to an EIF receiver" in the *IBM Tivoli Monitoring Tivoli Enterprise Portal User's Guide*.

Chapter 9. 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 Monitoring Services.
  2. In the Manage Tivoli Monitoring Services window, right-click **Tivoli Enterprise Portal Server**.
  3. In the menu, click **Reconfigure**.

4. In the first TEP Server Configuration window, click **OK**.
5. In the second TEP Server 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 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 10. Maintaining monitoring agents

The Navigator Physical view in the Tivoli Enterprise Portal displays all 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 to which the OS agent is connected.

The *IBM Tivoli Monitoring: Installation and Setup Guide* tells how establish an agent depot on the monitoring server and an OS agent on each computer where agents will be deployed. After that has been done, use the topics here to start, stop, configure, and remove a monitored agent from the managed network.

Also described is how to change the monitoring server designation for an agent.

To manage agents through the Tivoli Enterprise Portal, your user ID requires **Manage** permission for the **Agent Management** authority.

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. See the *IBM Tivoli Monitoring: Installation and Setup Guide* for more information.

Before you can remotely install and configure distributed monitoring agents on a computer, the computer must have an OS monitoring agent installed for the operating system the monitoring product will run under. When the OS monitoring 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 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 name of the monitoring agent to install and click **OK**. The New Managed System Configuration window opens with an **Agent** tab. Any other tabbed pages are specific to the agent. Move the mouse pointer over a field to see hover help.
4. Complete the fields to configure the agent, clicking **Next** and **Back** to move among the tabbed 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 you want to configure or upgrade.
2. Click **Configure** to open the Configure Managed System window. Any tabbed pages besides **Agent** are specific to the agent. Move the mouse pointer over a field to see hover help.
3. Edit the fields to configure the agent, clicking **Next** and **Back** to move among the tabbed pages.
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 agents

Whenever a new release of a distributed agent is available, you can use remote deployment to apply the updates. You can perform the updates through the Tivoli Enterprise Portal or at the command line.

- "Updating an agent through the Tivoli Enterprise Portal"
- "Updating an agent through the command-line interface" on page 145

Updating an agent through the Tivoli Enterprise Portal

Use the Configure Managed System window in the Tivoli Enterprise Portal client to apply a patch for a monitoring agent.

Before you begin

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. 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 *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.

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 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.

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
```

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 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.

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

Complete these steps to reassign a monitoring agent to a different monitoring server:

Procedure

1. In the Manage Tivoli Monitoring Services window, right-click the monitoring agent, and click **Reconfigure**.
2. Click **OK** in the first Agent Advanced Configuration window.
3. In the second Agent Advanced Configuration window, enter the **Hostname or IP Address** of the monitoring server to connect to; if the port you are using is different from the default 1918, enter the **Port** number.

What to do next

When reconfiguring a Universal Agent to connect to a different monitoring server, restart all the situations that are distributed to that managed system. Otherwise, the situations that are set to autostart will fail to start and an error will occur.

Chapter 11. 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.

Agent Management Services is a strategic approach to Tivoli monitoring agent management that provides these capabilities:

- 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 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 for these base agents: Linux OS, UNIX Logs, UNIX OS, Windows OS, Warehouse Proxy, Warehouse Summarization and Pruning, the Universal Agent, and the Agentless Monitoring Agents. Also supported are agents for the ITM WebSphere Message Broker Agent QI. Agent Management Services is not supported on i5/OS® and z/OS.
- Agent management workspaces with views of the information being collected by the Agent Management Services for many distributed Tivoli Enterprise Monitoring Agents. Refer to the documentation for your product to find out if Agent Management Services is supported.

The Tivoli Enterprise Portal is the user interface for the services, with predefined take action commands for manually starting or stopping management of an agent by 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 Monitoring Services and through the Tivoli Enterprise Portal Navigator pop-up menu.

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.

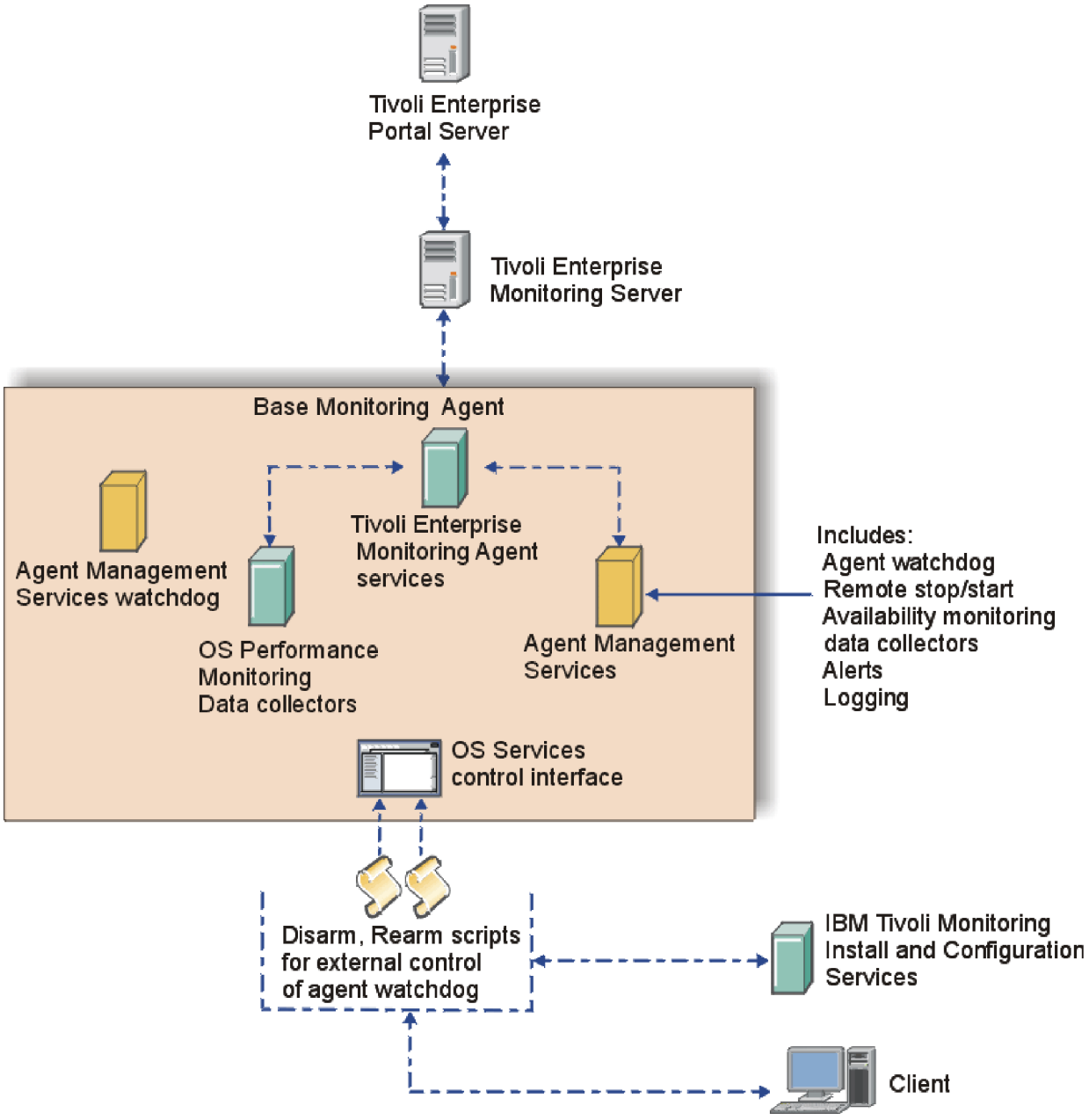


Figure 2. Interactions of Agent Management Services components with IBM Tivoli Monitoring components

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.

CAP files are being shipped with most agent products today. All agents in the Tivoli portfolio, with the exception of those on i5/OS and z/OS, are supported. For your convenience, the IBM Open Process Automation Library has downloadable IBM Tivoli Monitoring Agent Management Services Common Agent Package (CAP) files for agents that do not have them included with the product.

Agent Management Services watchdog
Who watches the watchdog? That is the job of the *Agent Management Services watchdog*. It is the same programmatically as the agent watchdog within the OS agent and behaves in the same way, but it is used *only* to watch 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 Monitoring Services and through the Tivoli Enterprise Portal Navigator pop-up menu.

Tivoli Agent Management Services installation and configuration

The Agent Management Servicesis 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.

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 is disabled at installation or upon upgrade to V6.2.2 Fix Pack 2 (or later) regardless of whether it was active or inactive prior to the upgrade. 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.

You can have one CAP file govern multi-instance monitoring agents or create a separate CAP file for each instance.

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: **30**.

<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 Universal Agent (um) on Linux (lz):

```
<startScript>
  <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/agentInstanceCommand.sh
  um START ##INSTANCE##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>

<startScript>
  <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/agentInstanceCommand.sh
  um 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
  um STOP ##INSTANCE## ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</stopScript>
```

For single instance agents on Linux, use this syntax:

```
<startScript>
  <command>su -c "$CANDLEHOME/bin/itmcmd agent start ul" -
  ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>

<stopScript>
  <command>su -c "$CANDLEHOME/bin/itmcmd agent stop ul" -
  ##USER##</command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</stopScript>
```

For single instance agents such as the UNIX Log 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 ul"
  </command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</startScript>

<stopScript>
  <command>su - ##USER## -c "$CANDLEHOME/bin/itmcmd agent stop ul"
 </command>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
  </returnCodeList>
</stopScript>
```

Upgrade the Universal Agent CAP file from V6.2.1 to V6.2.2 or higher

IBM Tivoli Monitoring V6.2.2 does not support multiple agent instances when <agentType> is set to **WinService** rather than the **ITM_Windows** or **ITM_UNIX** setting, which is new for V6.2.2.

- With a setting of **ITM_Windows** or **ITM_UNIX** for <agentType>, the standard Tivoli Monitoring `kincinfo/cinfo` installation utilities are used to discover monitoring agent instances.
- With a setting of **WinService** for <agentType> in V6.2.2 for multi-instance monitoring agents, Tivoli Monitoring instance names are no longer displayed. The agent type must now be **ITM_UNIX** or **ITM_Windows**.

The default CAP file for the Universal Agent name in V6.2.1 is kum.xml; upon upgrading to V6.2.2, the V6.2.1 kum.xml CAP is used by default. The new updated V6.2.2 default CAP file name for the kum agent, with multiple instance support, is kum_default.xml. Using the V6.2.1 kum.xml CAP file results in only the primary

instance being displayed in the Tivoli Enterprise Portal; other Universal Agent instances cannot be displayed. Either rename kum.xml to another file type such as kumxml.old or delete it to have the new, upgraded version of the CAP file created that supports instances used by Agent Management Services.

Related reference

Linux or UNIX installation considerations: Autostart scripts

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 10 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.

Related tasks

Configuring Agent Management Services on autonomous agents

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 Universal Agent for Windows (shows 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 "Universal Agent for Windows"`. 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 *IBM Tivoli Monitoring Tivoli Enterprise Portal User's Guide* and the user's guide for the specific agent.

Related reference

Take action commands

Chapter 12. Agent autonomy

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 agent should have, resource constraints, and how much dependency the agent should have on the monitoring server. A *Tivoli System Monitor Agent* is an OS agent that is installed and configured to have no dependency on nor any connection to a monitoring server.

Tivoli Enterprise 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.

Furthermore, 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.

OS agents and Agent Builder agents can also be installed and configured as Tivoli System Monitor Agents, which never connect 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.

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 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 160.

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 higher.

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 172.

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 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 (and 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 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 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 197 and “EIF events” on page 213.

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.

The HISTORY element includes an attribute for setting the number of hours to keep historical data on the computer before it is trimmed . 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. You can use the provided file conversion programs, such as krarloff, to move data out of the historical files to text files.

See "Private history" on page 190.

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 "Situation override XML specification" on page 192.

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 227.

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.

IBM Tivoli Monitoring Installation and Setup Guide provides instructions for installing and configuring the Tivoli System Monitor Agent. It also has a reference of the common agent environment variables in an appendix.

Tivoli Enterprise Monitoring Agent environment file

The environment variables are edited in or added to the Tivoli Enterprise Monitoring Agent environment file, where *pc* is the two-character product code:

Windows *Install_dir*\TMAITM6\k*pc*env. On system monitor agents, this file is *pc*.environment

Linux UNIX *Install_dir*/config/*pc*.ini. On system monitor agents, this file is *pc*.environment

i5/OS /qautotmp/kmsparm.kbbenv

z/OS member name K*PC*ENV in *&hilev.&rte.*RKANPARU

Best practices for z/OS

Use the "Specify Nonstandard Parameters" panel in the Configuration Tool (also called the Installation and Configuration Assistance Tool, or ICAT) to make changes to the members. Any changes you make using this editor are automatically preserved when the runtime environment is updated, which means your settings are not overwritten the next time the runtime environment is updated. See the "Adding, changing, or deleting a parameter in a runtime member" topic in *OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration*.

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.

Control autonomy in Tivoli Enterprise Monitoring Agents

. The following configuration parameters start and control autonomous behavior of Tivoli Enterprise Monitoring Agents.

IRA_AUTONOMOUS_LIMIT=50

This parameter determines the number of events that can be stored at the agent when it is in autonomous mode or allocates the amount of disk space that the events can occupy. When the event limit or disk space maximum has been reached, no further events are collected. The default is **50** events or **2MB**. Specify either the total number of events or the disk space limit, where *n* is the numeric value:

n = maximum number of events (sampled and pure) that can be saved. To estimate the space for each event, add 1200 to the average application row size.

nKB = n times 1024 bytes.

nMB = n times 1,024,000 bytes.

nGB = n times 1,024,000,000 bytes

IRA_AUTONOMOUS_MODE=Y

This parameter controls autonomous operation in enterprise monitoring

agents. By default, autonomy is enabled. To disable it, which sets the agent to the same dependency it had on the monitoring server prior to V6.2.1, set this parameter to **N**.

IRA_EIF_DEST_CONFIG=*filename*
Use this parameter in the agent environment file to specify the location of the EIF destination configuration XML file. You can specify the complete path or the path relative to the local configuration directory.

IRA_EIF_MSG_LOCALE=en_US
This 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.

IRA_EVENT_EXPORT_CHECKUSAGE_INTERVAL=180
Specifies the preferred interval in seconds to check if the IRA_AUTONOMOUS_LIMIT has been reached. The default interval is **180** seconds (3 minutes); the minimum interval that can be specified is **60** seconds.

IRA_EVENT_EXPORT_EIF=Y
This 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_EVENT_EXPORT_SIT_STATS=Y
You can get a report of the situation operation statistics through the Agent Service Interface. This parameter enables (Y) or disables (N) the basic situation operation statistics data collection:

Situation Name

Situation Type - Enterprise or Private

Application Name

Table Name

Sample interval

Row data size

Time stamp First Time situation started

Time stamp First Time situation raised event (passed filter)

Time stamp Last Time situation started

Time stamp Last Time situation stopped

Time stamp Last Time situation evaluated to TRUE

Time stamp Last Time situation evaluated to FALSE

Number of times situation recycled

Number of times situation in autonomous operation

Default: **Y**.

IRA_EVENT_EXPORT_SIT_STATS_DETAIL=N
When set to *Y*, this parameters enables collection of the following event metrics from the agent:

True sample count

False sample count

True Sample ratio

False Sample ratio

Number of data rows counted in 24 hours

Number of true samples counted in 24 hours

Number of false samples counted in 24 hours

The agent keeps these metrics for eight days on disk, with roll-off daily at midnight. Default: **N**.

IRA_EVENT_EXPORT_SNMP_TRAP=Y/N
IRA_EVENT_EXPORT_SNMP_TRAP=N disables agent SNMP alerts even if the *pc*_trapcnfg.xml file is present. Default: **Y**.

IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG
By default, the agent looks to see if a *Install_dir*/localconfig/*pc*/*pc*_trapcnfg.xml file exists. If the configuration file is located somewhere else or named something else, use this parameter to specify the path and name of the file. You can specify the complete path or a path relative to the local configuration directory.

z/OS The z/OS agent looks for the default *PC*TRAP member name in the *&hilev.&rte.*RKANDATV data set in the environment. If the SNMP trap member has a different name, specify the member name using this variable. If the member is in a different data set, also specify both the data set name and the member name, in the format: *member_name.dataset_name*.

For example, if the name of the configuration file is MYSNMP and it is in RKANDATV, specify IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG=MYSNMP.
If the configuration file is in a different data set, for example, TIVOLI.ITM622.TVT1006.MYFILES(MYSNMP), specify IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG=MYSNMP.MYFILES.

IRA_LOCALCONFIG_DIR
The default local configuration directory path that contains locally customized configuration files such as private situations, EIF event configuration, and SNMP trap configuration files is the localconfig subdirectory of the directory specified by the *CANDLE_HOME* environment variable; RKANDATV *DD* name on z/OS systems. Use this parameter to change the path.

KHD_REGWITHGLB
Normally, the Warehouse Proxy agent is registered with the hub Tivoli Enterprise Monitoring Server. If you want the warehouse proxy to have no dependency on the monitoring server, add KHD_REGWITHGLB=N to the warehouse proxy environment file (Windows khdenv; Linux UNIX hd.ini) to not register with the monitoring server.

KHD_WAREHOUSE_LOCATION
If the Warehouse Proxy agent does not register with the hub monitoring server, you must add this parameter to the environment file of every enterprise monitoring agent that has full autonomy. Enter the fully qualified name of each warehouse proxy that can transfer historical data from the agent to the Tivoli Data Warehouse, each separated by a semicolon (;). The syntax is KHD_WAREHOUSE_LOCATION=*family protocol*:*network address*[*port number*], for example, KHD_WAREHOUSE_LOCATION=ip.pipe:DEPT-XP[63358];ip:MY-XP[63358];ip.pipe:#9.44.255.253[65538].

KSY_AUTONOMOUS
Normally, the summarization and pruning settings for attribute groups are configured through the Tivoli Enterprise Portal or the command-line interface **tacmd histconfiguregroups** and saved in a WAREHOUSESUMPRUNE table on the Tivoli Data Warehouse.

If you want the summarization and pruning agent to have no dependency on the Tivoli Enterprise Portal Server, add KSY_AUTONOMOUS=Y to the summarization and pruning agent environment file and add the location of the agent description files using the KSY_AUTONOMOUS_ODI_DIR variable.

The summarization and pruning agent requires the agent application support files that are installed with the portal server. If you have set KSY_AUTONOMOUS=Y and the Summarization and Pruning agent is not installed on the same computer as the portal server, you must copy the required application support files to the same computer. With the exception of **dockcj**, which is not used, the support files are the **dock*pc*** (where *pc* is the two-character product code) files in the portal server directory: Windows *Install_dir*\cnps; Linux UNIX *Install_dir*/*arch*/cq/data. See the "Configuring a Summarization and Pruning agent to run autonomously" topic of *IBM Tivoli Monitoring Installation and Setup Guide*.

KSY_AUTONOMOUS_ODI_DIR

Although the Summarization and Pruning agent can be configured for autonomy from the portal server, you still need to have the portal server installed and application support for all agents that are configured to collect historical data because the application support files are needed by the summarization and pruning process when running autonomously. Use this parameter to enter the path to the application support files.

When the agent is configured to run autonomously, the summarization and pruning settings must be entered directly into the WAREHOUSESUMPRUNE table on the warehouse database using the SQL insert command.

Private situations

IRA_PRIVATE_SITUATION_CONFIG

Specifies the fully qualified private situation configuration file name. During agent initialization, a check is made for the private situation configuration file: `Install_dir/localconfig/pc/pc_situations.xml` where *pc* is the two-character product code.

z/OS A fully qualified path to the situation configuration file on z/OS, such as 'TIVOLI.ITM622.TVT1006.RKANDATV(MYPSSIT)' where DDNAME RKANDATV is TIVOLI.ITM622.TVT1006.RKANDATV: `IRA_PRIVATE_SITUATION_CONFIG=MYPSSIT`.
For a situation configuration file that is not a PDS member in DDNAME RKANDATV, specify 'TIVOLI.ITM622.TVT1006.MYFILES(MYPSSIT)' where DDNAME MYFILES is TIVOLI.ITM622.TVT1006.MYFILES: `IRA_PRIVATE_SITUATION_CONFIG=MYPSSIT.MYFILES`.

See "Private situations" on page 172 to learn about private situations.

Private history

CTIRA_HIST_DIR

Specifies the directory where agent-based short-term history data files will be stored. Does not apply to the Tivoli Enterprise Monitoring Server's short-term history data files. This is the default location for enterprise history or private history binary files.

Windows *Install_dir*\TMAITM6\logs

Linux UNIX *Install_dir*/*arch*/*pc*/hist

See “Private history” on page 190 to learn about private history data collection.

Situation expression overrides

CTIRA_THRESHOLDS
Specifies the fully qualified name of the XML-based adaptive (dynamic) threshold override file. By default, the agent looks to see if (where *pc* is the agent product code) a *Install_dir*/localconfig/*pc*/*pc*_thresholds.xml file exists. You can specify the complete path or the path relative to the local configuration directory.

z/OS The default file name is ***PC*THRES**. To specify the complete path, the PDS should be listed at the end (or omitted and allowed to default to RKANDATV).

IRA_ADAPTIVE_THRESHOLD_MODE
Specifies the adaptive (dynamic) threshold operation mode, either CENTRAL or LOCAL. The default mode is CENTRAL.

In CENTRAL mode, situation threshold overrides are created through the Tivoli Enterprise Portal or CLI **tacmd setOverride** command and distributed to the target agent through the Tivoli Management Services distribution framework.

You can set an agent to LOCAL mode to have the agent use a locally defined threshold configuration XML instead of the CENTRAL override distribution. In LOCAL mode, central distribution to the agent is inhibited (its affinity is not registered) and threshold overrides are locally created and managed. Use LOCAL mode with caution because it causes the Tivoli Enterprise Monitoring Server's thresholds and the agent's thresholds to be out of sync.

If you switch the agent from LOCAL mode back to CENTRAL mode, the CENTRAL override specification supersedes the local definitions and synchronizes with the CENTRAL overrides repository located at the monitoring server.

See “Situation override XML specification” on page 192 to learn about local situation overrides.

Agent Service Interface

These agent configuration parameters effect Service Interface operation:

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 uagent02 to identify a second installed Universal Agent instance on a system.

Default:
Windows system.*hostname_pc*
Linux UNIX *hostname_pc*

IRA_SERVICE_INTERFACE_DEFAULT_PAGE
Instructs the agent to open the named product-specific HTML page instead of the installed **navigator.htm** page upon log on to the agent service

interface. The HTML file must exist in the agent installation HTML subdirectory: *Install_dir*\localconfig\html\ or as specified by IRA_SERVICE_INTERFACE_DIR.

IRA_SERVICE_INTERFACE_DIR
Defines the path specification of the agent service interface HTML directory. In conjunction with the IRA_SERVICE_INTERFACE_DEFAULT_PAGE parameter, the agent constructs the file path to a specific, requested HTTP GET object. The default is *Install_dir*/localconfig on distributed systems.

Example: If IRA_SERVICE_INTERFACE_DIR="\mypath\private" and you enter http://localhost:1920///kuxagent/kuxagent/html/myPage.htm in your browser, myPage.htm is retrieved from \mypath\private\html\ instead of *ITM_dir*\localconfig\html\.

z/OS There is no directory path specification but instead a data set represented by the JCL DD (Data Definition) name. Therefore, IRA_SERVICE_INTERFACE_DIR is not used but the IRA_SERVICE_INTERFACE_HTML specification is in effect. The default is RKANDATV DD name.

See also the environment variables for Centralized Configuration that are prefixed with IRA_SERVICE_INTERFACE.

Diagnostics and troubleshooting

These parameters can be set in the agent environment file for troubleshooting. All diagnostic information goes to agent RAS (reliability, availability, and serviceability) trace log.

IRA_DEBUG_AUTONOMOUS=N
When set to **Y**, this parameter enables trace logging of all autonomous agent operation. The default setting is **N**.

IRA_DEBUG_EIF=N
When set to **Y**, this parameter enables trace logging of EIF emitter operations. The default setting is **N**.

IRA_DEBUG_EVENTEXPORT=N
When set to **Y**, this parameter enables trace logging of event export activity such a SNMP traps. The default setting is **N**.

IRA_DUMP_DATA=N
When set to **Y**, this parameter enables trace logging of all remote procedure call (RPC) data. The default setting is **N**.

IRA_DEBUG_PRIVATE_SITUATION=N
When set to **Y**, all the trace information regarding private situation problems are entered in the RAS trace log. The default setting is **N**.

IRA_DEBUG_SERVICEAPI=N
When set to **Y**, this parameter enables trace logging of all agent service interface processing. The default setting is **N**.

KEF_DEBUG=N
When set to **Y**, this parameter enables trace logging of EIF library operations. The default setting is **N**.

Related tasks

Editing the agent environment file

Related reference

Environment variables for central configuration

Access Authorization Group Profile

Situation limitations

The types of formula functions that can be used in a private situation are limited. As well, the types of formula functions in an enterprise situation that can be processed by an agent when it is disconnected from its Tivoli Enterprise Monitoring Server are limited.

Table 14. Availability of situation formula functions when an enterprise agent is connected or disconnected, or when the situation is private.

	Event emitted from the monitoring server	**Event emitted from the enterprise monitoring agent**		
Formula function	**Supported in enterprise situations**	**Enterprise situation Agent connected to the monitoring server Evaluates at the monitoring server**	**Enterprise situation Agent disconnected from the monitoring server Evaluates at the agent**	**Supported in private situations Evaluates at the agent[1]**
Cell functions				
CHANGE	available	available	available	not available
DATE	available	available	available	not available
MISSING	available	available	available	available
PCTCHANGE	available	available	available	not available
SCAN	available	available	available	not available
STR	available	available	available	not available
TIME	available	not available	not available	not available
VALUE	available	available	available	available
IN	available	available	available	not 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	not available
COUNT	available	not available	not available	not available
MAX	available	not available	not available	not available
MIN	available	not available	not available	not available
SUM	available	not available	not available	not available

Table 14. Availability of situation formula functions when an enterprise agent is connected or disconnected, or when the situation is private. (continued)

	Event emitted from the monitoring server	**Event emitted from the enterprise monitoring agent**		
Formula function	**Supported in enterprise situations**	**Enterprise situation Agent connected to the monitoring server Evaluates at the monitoring server**	**Enterprise situation Agent disconnected from the monitoring server Evaluates at the agent**	**Supported in private situations Evaluates at the agent[1]**
Situation characteristics				
Embedded, including correlated situations	available	not available	not available	not available
Multiple attribute groups	available	not available	not available	not available
Persistence enabled	available	available[2]	available[2]	not available
Display item selected	available	available	not available[3]	not available
Uses duper process	available	available	not available[4]	not available
Distribution to managed system group	available	available	available	not available

[1] This column also applies to system monitor agents.

[2] Situation persistence is not evaluated at the agent. 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] Situations that include a display item (available for multiple row attribute groups) 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] Traps are emitted but situations are not evaluated when the agent is disconnected from the monitoring server.

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 />
    <CRITERIA><![CDATA[ *IF *VALUE KAB_LOGFILE.Message *NE '' ]]>
   </CRITERIA>
   <INTERVAL>000000</INTERVAL>
   <DISTRIBUTION>AB:uxlog:LOG</DISTRIBUTION>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>KAB_Log1only_Message />
    <CRITERIA><![CDATA[ *IF *VALUE KAB_LOGFILE.Message *NE '' ]]>
   </CRITERIA>
   <INTERVAL>000000</INTERVAL>
   <DISTRIBUTION>AB:uxlog1:LOG</DISTRIBUTION>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>KAB_Log2only_Message />
    <CRITERIA><![CDATA[ *IF *VALUE KAB_LOGFILE.Message *NE '' ]]>
   </CRITERIA>
   <INTERVAL>000000</INTERVAL>
   <DISTRIBUTION>AB:uxlog2:LOG</DISTRIBUTION>
 </PRIVATESIT>
</PRIVATECONFIGURATION>
```

Related concepts

Customizing event integration with Tivoli Netcool/OMNIbus

Customizing event integration with Tivoli Enterprise Console

Related reference

"Private situation operation" on page 172

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 the "Configuring hub and remote monitoring servers on z/OS" topics in *Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

i5/OS

(EIF event destination configuration is not supported, nor are SNMPv3 informs.) Because UTF-8 is not easily displayed or edited on i5/OS, 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.

Related reference

"Private situation XML specification" on page 175

"Trap configuration XML specification" on page 199

"EIF event mapping XML specification" on page 216

"EIF event destination configuration XML specification" on page 221

Configuring the Tivoli Enterprise Monitoring Server on z/OS

i5/OS National language support (NLS) considerations

Configuring Agent Management Services on Tivoli System Monitor Agents

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 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 />
    <CRITERIA>
     <![CDATA[ *MISSING NT_Process.Process_Name *EQ ('schedule')]]>
    </CRITERIA>
    <INTERVAL>001000</INTERVAL>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>NT_Paging_File_Critical_pr />
    <CRITERIA>
     <![CDATA[ *VALUE NT_Paging_File.%_Usage *GE 80 ]]>
    </CRITERIA>
    <INTERVAL>001500</INTERVAL>
  </PRIVATESIT>
  <PRIVATESIT>
   <SITUATION>Disk_Queue />
    <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*.)

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 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 227 and "Agent Service Interface request - Private situation control" on page 246.

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 Chapter 13, "Centralized Configuration," on page 255.

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.
- Have a separate namespace to avoid naming conflicts with enterprise situations.
- 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.

- Include no support for wildcards.
- One attribute group in a situation. Two different attribute groups are not supported unless one of the groups is a global attribute group such as Local Time.
- 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 Management Services is 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.
- Enterprise and private situations must have unique situation names.

Related reference

"Situation limitations" on page 167

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 dataset

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. See "Private situations" on page 164 and "Control autonomy in Tivoli Enterprise Monitoring Agents" on page 161.

Elements

The elements and their attributes are case-insensitive. For example, you can enter <PRIVATESIT>, <PrivateSit>, or <privatesit>.

<PRIVATECONFIGURATION>

PRIVATECONFIGURATION is the root element identifying this as an agent private situation configuration document.

```
<PRIVATECONFIGURATION>
<PRIVATESIT>
<SITUATION NAME="Check_Process_CPU_Usage"  INTERVAL="000500" /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>
<CMD><![CDATA[netstat >.\logs\netstat.dat]]></CMD>
<AUTOSOPT When="N" Frequency="N" />
</PRIVATESIT>
</PRIVATECONFIGURATION>
```

<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" />
```

<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 INTERVAL 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.

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.

<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 comma (,). The default is the agent managed system name or all known subnodes.

<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.

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>
```

<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. The XML parser processes duplicated <HISTORY> as update scenarios. The final updated attribute value will be the value in effect and always output to agent's Operation Log.

TABLE= This parameter specifies the application attribute group name.

DISTRIBUTION If the agent has subnodes, specify the subnode name in DISTRIBUTION begin and end tags. Separate multiple subnode names with , commas.

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 will be collected for the specified attribute group. Default:**“15”**.

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"**.

Examples:
The agent collects WTSYSTEM 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 agent collects System table data every 5 minutes and maintains 3 days of short-term history.

```
<HISTORY TABLE="System" Interval="5" RETAIN="72"  />
```

The agent collects WTLOGCLDSK table data every minute.

```
<HISTORY TABLE="NT_Logical_Disk" INTERVAL="1"  />
```

The 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>
```

Related reference

"Private history" on page 190

"Agent Service Interface - History" on page 234

"Agent Service Interface - Situations" on page 233

"UTF-8 encoded XML files" on page 170

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* 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 167 for other restrictions.

Elements

The elements and their attributes are case-insensitive. 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.

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.

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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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_pr />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 />
  <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 situations. There can be only one active history data collection per application table (attribute group).

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 APPL=KN3 to APPL=PVT.
   c. Change TABLE=KN3BPG to TABLE=PN3BPG.
2. Add application tables OVERRIDE statement in data set RKANPARU member KN3PG.
   a. Copy the table KN3BPG OVERRIDE statement, if any.
   b. Change APPL=KN3 to APPL=PVT.

After completion of the two configuration steps, Private History can be configured and retrieved as shown in this example:

```
<HISTORY TABLE= VTAM_Buffer_Usage_By_Category Interval=15 Retain=24>
<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.

Related reference

"Private situation XML specification" on page 175

"Agent Service Interface - History" on page 234

Converting short-term history files to delimited flat files

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.

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

i5/OS ctira_sit_path/hostname_pc_thresholds.xml

See "Situation expression overrides" on page 165 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).

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.

i5/OS The i5/OS 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 to a Tivoli Universal Agent monitoring 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="UAStatMon" 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.

The elements and their attributes are case-insensitive. For example, you can enter ADDRESS, Address, or address.

Related reference

"UTF-8 encoded XML files" on page 170

"SNMP PassKey encryption: itmpwdsnmp" on page 207

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 15. 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
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

Table 15. TrapDest element XML specification (continued)

Attribute	Description	Required	Default	SNMPv1/v2 or SNMPv3
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
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

Table 15. TrapDest element XML specification (continued)

Attribute	Description	Required	Default	SNMPv1/v2 or SNMPv3
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

Related reference

"SNMP PassKey encryption: itmpwdsnmp" on page 207

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 16. 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 17. 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	

Table 17. Situation element XML specification (continued)

Attribute	Description	Required	Default
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
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.	

Table 17. Situation element XML specification (continued)

Attribute	Description	Required	Default
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 18. 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 ITM 6.2.2, the default value of CTIRA_MAX_RECONNECT_TRIES has been changed to 0, so the agent will never shutdown.	1 – Indeterminate	3 – Status

Table 18. Agent life cycle status traps (continued)

Attribute	Description	Severity	Category
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 19. 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.exe
Linux UNIX *Install_dir*/bin/itmpwdsnmp.sh

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
```

Related reference

"Trap configuration XML specification" on page 199
"TrapDest element" on page 199

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.

Related reference

MIB SNMP agent event descriptions

OMNIbus configuration for SNMP

You must configure your 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 Support 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 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 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 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/kpc`. 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
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 Support 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();
```

Related reference

"EIF life cycle event" on page 224

"EIF heartbeat event" on page 225

EIF events

Send private situation events directly from a Tivoli Monitoring Agent to an EIF receiver without going through the Tivoli Enterprise Monitoring Server.

Related concepts

Situation event integration with Tivoli Enterprise Console

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 Tivoli Enterprise Console event server or Netcool/OMNIbus Probe for Tivoli EIF.

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 *ITM_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 *ITM_dir*\localconfig*pc**pc*_eventmap.map

Linux UNIX *ITM_dir*/localconfig/*pc*/*pc*_eventmap.map

z/OS RKANDATV, with file name *P*CEVMAP

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 OS agent of Version 6.2.2 Fix Pack 1 (or later) 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

The elements and their attributes are case-insensitive. For example, you can enter DEFAULT=, Default=, or default=.

<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 runtime. See "Dynamic EIF classname" on page 219.

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 runtime, 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.

Related reference

"UTF-8 encoded XML files" on page 170

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= Destination index, from 0 to 999. Default: **0**

type= Optional. Destination type: T=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 event listener. A setting of **port=0** applies only to Tivoli Enterprise Console on Linux or operating systems such as UNIX and indicates that the event listener is using portmapper. Default: **0**

<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.

Example

The following example is an event destination configuration file containing two event destinations:

```
<EventDest>
 <Destination  id="0" type="M" stat="Y" master_reset="Y">
  <Server  location="omniserver.ibm.com" port="9999"  />
  <Server  location="192.0.2.1" port="9999" />
    <StatEvent name="EE_HEARTBEAT" interval="5"/>
 </Destination>
 <Destination  id="1"  type="T" default="Y" master_reset="Y" stat="Y">
  <Server location="tecserver.ibm.com"  port="5529"/>
    <StatEvent name="EE_HEARTBEAT" interval="5"/>
 </Destination>
</EventDest>
```

Sample EIF files are provided on the Tivoli Monitoring Agent installation media in the `PrivateConfigSamples/EIF` directory.

Related reference

"UTF-8 encoded XML files" on page 170

"EIF heartbeat event" on page 225

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 20. 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".
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 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

Table 20. Set of common slots for emitted EIF events. (continued)

Slots	Values and meaning
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**
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 21. EIF life cycle events.

Event	Meaning
EE_HEARTBEAT	Agent heartbeat.

Table 21. EIF life cycle events. (continued)

Event	Meaning
EE_AUTO_ENTER	The situation has entered autonomous mode operation.
EE_AUTO_EXIT	The situation has exited autonomous mode operation.
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 22. 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
severity	"HARMLESS"
msg	Message describing the life cycle event

Related tasks

"Enabling OMNIbus heartbeat automation" on page 212

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 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 Tivoli Enterprise Console event server. It is on the

tools DVD that is included with the event server installation media. 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
```

Related tasks

"Enabling OMNIbus heartbeat automation" on page 212

Related reference

"EIF event destination configuration XML specification" on page 221

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 shipped Netcool/OMNIbus or Tivoli Enterprise Console rule closes all the opened events from this particular agent and its subnodes.

Table 23. 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
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

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 i5/OS The Agent Service Interface is not available for installation on i5/OS and z/OS operating systems. You can, however, download the ITMSUPER Tools (NavCode 1TW10TM6L) from the Tivoli Open Process Automation Library. As part of the tool set, **itmsa.htm** provides direct access to agent reports on i5/OS, z/OS, Windows, Linux, and UNIX platforms; and **itmsuper.htm** provides Web Services tools that are accessible through the hub Tivoli Enterprise Monitoring Server. Additionally

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 ***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*/config/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 24. 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
Overrides		x	x
AAGP			x
FileObj			x

The **Restricted** group definition is required. If it is not included in the AAGP, the agent default specification shown in Table 24 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 259.

If the AAGP contains no `<AAGROUP>` specification, the agent default specification shown in Table 24 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:

<AAGP REFRESH="Y"> Deletes the current active AAGP and replaces it with the AAGP definition from this AAGP specification.

<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>SitStat</SIAPI>
  <SIAPI>SitSummary</SIAPI>
  <SIAPI>HistRead</SIAPI>
  <SIAPI>Report</SIAPI>
 </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>
 </AAGROUP>
 <AAGROUP>
  <GROUPNAME>Administrative</GROUPNAME>
  <INCLUDE>Operation</INCLUDE>
  <SIAPI>FileObj</SIAPI>
  <SIAPI>AAGP</SIAPI>
 </AAGROUP>
 <AAUSER>
  <ID>dyang</ID>
  <ASSIGN>OP</ASSIGN>
 </AAUSER>
 <AAUSER>
  <ID>ksmith</ID>
  <ASSIGN>OP</ASSIGN>
 </AAUSER>
 <AAUSER>
  <ID>jmlake</ID>
  <ASSIGN>AD</ASSIGN>
 </AAUSER>
 <AAUSER>
  <ID>tcharris</ID>
  <ASSIGN>OP</ASSIGN>
 </AAUSER>
 <AAUSER>
  <ID>acwills</ID>
  <ASSIGN>Operation</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.

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\GSK7
* CSV1_PATH=d:\IBM\ITM\GSK7\lib
```

```
* CSV2_PATH=d:\IBM\ITM\GSK7\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\java50\jre
* KBB_IGNOREHOSTENVIRONMENT=N
* PATH=d:\IBM\ITM\GSK7\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)

Related reference

"Private situation XML specification" on page 175

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.

Related reference

"Private history" on page 190

"Private situation XML specification" on page 175

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 25. Agent Service Interface - Queries sample attribute listing

Name	Display	Type	Length	Minimum	Maximum	ENUMS
ORIGINNODE	NT_IP_Address.System_Name	2	64			
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 26. 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 27. 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 28. 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
<STATUS>	Return status code bracketed by begin and end tag

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>0008000000000000000000000000000004000000u0a4</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\GSK7
   CSV1_PATH=C:\IBM\ITM\GSK7\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\Java50\jre
   PATH=C:\IBM\ITM\GSK7\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 29. 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 30. 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 31. 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 32. 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 33. 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 Universal Agent TS2ATR00 attribute file:

```
<READATTR>
  <ATTRFILE>TS2ATR00</ATTRFILE>
</READATTR>
```

Report output

Table 34. Agent Service Interface <READATTR> request output.

Output tag	Description
<ATTRFILE>	Attribute file name.
<ATTRDATA>	Attribute file records.

This example shows the TS2ATR00 attribute file contents:

```
<ATTRFILE>TS2ATR00</ATTRFILE>
  <ATTRDATA>
 <![CDATA[//1090428005244020 TS200/06.00.00
 //Generated by Universal Agent
  //
  entr ATTR
  name TS2TCPIOQ00.Node_Name
  affi 00000000000000080000000000000000000000000000
  acod TS200
  usag I
  appl TS200
  stmp 1090428005244020
  cvrm 06.00.00
  lvrm 06.00.00
  tabl TS24601600
  mult 1
  samp 3
  colu ORIGINNODE
  type 2
  slng 32
  msid KUM0000
  opgr 0
  atid 065535
  //entr ATTR
  name TS2TCPIOQ00.LocalApplAddress
  atom y
  affi 00000000000000080000000000000000000000000000
  acod TS200
  colu UA1
  type 2
  slng 24
  msid KUM0000
  opgr 2
  atid 065535
  //
  entr ATTR
  name TS2TCPIOQ00.TargetApplAddress
  affi 00000000000000080000000000000000000000000000
  acod TS200
  colu UA2
  type 2
  slng 24
  msid KUM0000
  opgr 2
  atid 065535
  //
  entr ATTR
  name TS2TCPIOQ00.SendQueueSize
  affi 00000000000000080000000000000000000000000000
  acod TS200
  colu UA3
  type 1
  msid KUM0000
  opgr 2
  atid 065535
  mini -2147483648
  maxi 2147483647
  //
  entr ATTR
  name TS2TCPIOQ00.RecvQueueSize
  affi 00000000000000080000000000000000000000000000
  acod TS200
  colu UA4
  type 1
  msid KUM0000
  opgr 2
  atid 065535
```

```
  mini -2147483648
  maxi 2147483647
  //
  entr ATTR
  name TS2TCPIOQ00._LocalTimeStamp
  affi 00000000000000080000000000000000000000000000
  acod TS200
  colu UA5
  type 2
  slng 16
  msid KUM0000
  opgr 2
  atid 065535
  //
  entr HIDDEN
  name TS2TCPIOQ00.KUMHELP
  affi 00000000000000080000000000000000000000000000
  colu KUMHELP
  type 3
  opgr 0
  cost 9
  vali ^APPLICATION
  vale "No Application Help Defined"
  vali ^ATTRGROUP[TCPIOQ]
  vale "No attribute group Help Defined"
  vali LocalApplAddress
  vale "No attribute Help Defined"
  vali TargetApplAddress
  vale "No attribute Help Defined"
  vali SendQueueSize
  vale "No attribute Help Defined"
  vali RecvQueueSize
  vale "No attribute Help Defined"
  vali _LocalTimeStamp
  vale "Universal Agent inserted attribute per metafile keyword
  AddTimeStamp specification. It is the 16-byte timestamp value
  when the data arrived."
 ]]></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 35. Agent Service Interface <REPORT> request

Tag	Description
<REPORT>	Enter begin and end REPORT tags to retrieve application table data for the table specified.
<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.

Table 35. Agent Service Interface <REPORT> request (continued)

Tag	Description
<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 175.

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 36. Agent Service Interface <REPORT> request output.

Output tag	Description
<REPORTDATA>	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.

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 37. 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 38. 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 39. 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 40. 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>300</STATUS>
<FILOBJ>
```

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 41. Agent Service Interface <SITSUMMARY> request.

Tag	Description
<SITSUMMARY>	Define dynamic threshold override specification.

```
<SITSUMMARY>
</SITSUMMARY>
```

The output from the request looks like the private situation configuration files shown in the "Private situation examples" on page 185.

Sample request 1: Download the configuration file to the agent

```
<SITSUMMARY>
 <CONTROL>PUSH</CONTROL>
 <INSTRUCTS>
  <FILE><![CDATA[name="setenv150.bat"
   path="%candle_home%\tmaitm6" recDLM="$@$" ]]>
  </FILE>
  <DATA>
   <![CDATA[ set path=.;%WINDIR%\system32\npp;%PATH%$@$
  set KFC_DEBUG_API=Y$@$
  set KFC_DEBUG_FILTER=N$@$
  set KFC_DEBUG_STORAGE=N$@$
  set KFC_DEBUG_STORAGE_STAT=N$@$
  set KFC_DEBUG_TIMESYNC=N$@$
  set KFC_TIME_SYNC_REQUIRED=N$@$
  set KFC_API_MEDIASERVER_LISTEN_PORT=12125$@$
  set kbb_ras1=ERROR$@$
  set kbb_ras1_log=.\logs\kfc1.log$@$ ]]>
  </DATA>
 </INSTRUCTS>
</SITSUMMARY>
```

Sample request 2: Retrieve trace log file from the agent

```
<SITSUMMARY>
 <CONTROL>PULL</CONTROL>
 <INSTRUCTS>
 <FILE>
  <![CDATA[name="kntcma-%seq%.log" range="3500-3750"
   path="%candle_home%\tmaitm6\logs"]]>
 </FILE>
 </INSTRUCTS>
</SITSUMMARY>
```

Report output

Table 42. Agent Service Interface <SITSUMMARY> request output.

Output tag	Description
<ROW>	Defines an output data row.

Table 42. Agent Service Interface <SITSUMMARY> request output. (continued)

Output tag	Description
<DATA>	Data tags enclose the download file contents.
<STATUS>	Return status code bracketed by begin and end tag.

Sample output 1: The agent returns the process status

```
<SITSUMMARY>
 <STATUS>0</STATUS>
<FILOBJ>
```

Sample output 2: The agent returns the trace log file contents

```
<SITSUMMARY>
  <STATUS>0</STATUS>
  <ROWCOUNT>5</ROWCOUNT>
 <ROW>
 <DATA><![CDATA[+49BDCB34.001C 00000000 3018060A 2B060106 03010104
 0100060A 0...+..... ]]></DATA>
 </ROW>
 <ROW>
 <DATA><![CDATA[ +49BDCB34.001C 00000010 2B060104 018D0301 0315
 +.........]]></DATA>
 </ROW>
 <ROW>
 <DATA><![CDATA[ (49BDCB34.001D-B90:kraaest1.cpp,92,
 "IRA_ConstructTrapVarBindV1") *TRAP-INFO: IRA_ConstructTrapVarBindV1
 - Entry pPDU<39B4C6A> pTrapWork<3CDA0A8> pTrapSit<2B8F098>
 dataBuffer<39BC948> offset<1363> resetTrap<0>]]> </DATA>
 </ROW>
 <ROW>
 <DATA><![CDATA[ (49BDCB34.001E-B90:kraaesti.cpp,289,
 "addVarBindStringData") <0x39B4C6A,0x16>
 *TRAP-INFO: VarBind 1.3.6.1.4.1.1667.1.2.1.10.1.1 KNT ]]> </DATA>
 </ROW>
 <ROW>
 <DATA><![CDATA[ +49BDCB34.001E 00000000 3014060D 2B060104 018D0301
 02010A01 0...+..... ]]></DATA>
 </ROW>
</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 43. 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 know 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 44. 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.
<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 45. Agent Service Interface <HISTREAD> request.

Tag	Description
<HISTREAD>	Retrieve history table data
<SQLTABLE>	Identify SQL table definition set.
<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 46. 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>
```

Related reference

"Access Authorization Group Profile" on page 228

Chapter 13. 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 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 higher, or it can be any Web server, such as WebSphere, IBM HTTP, Microsoft IIS, or Apache.

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.

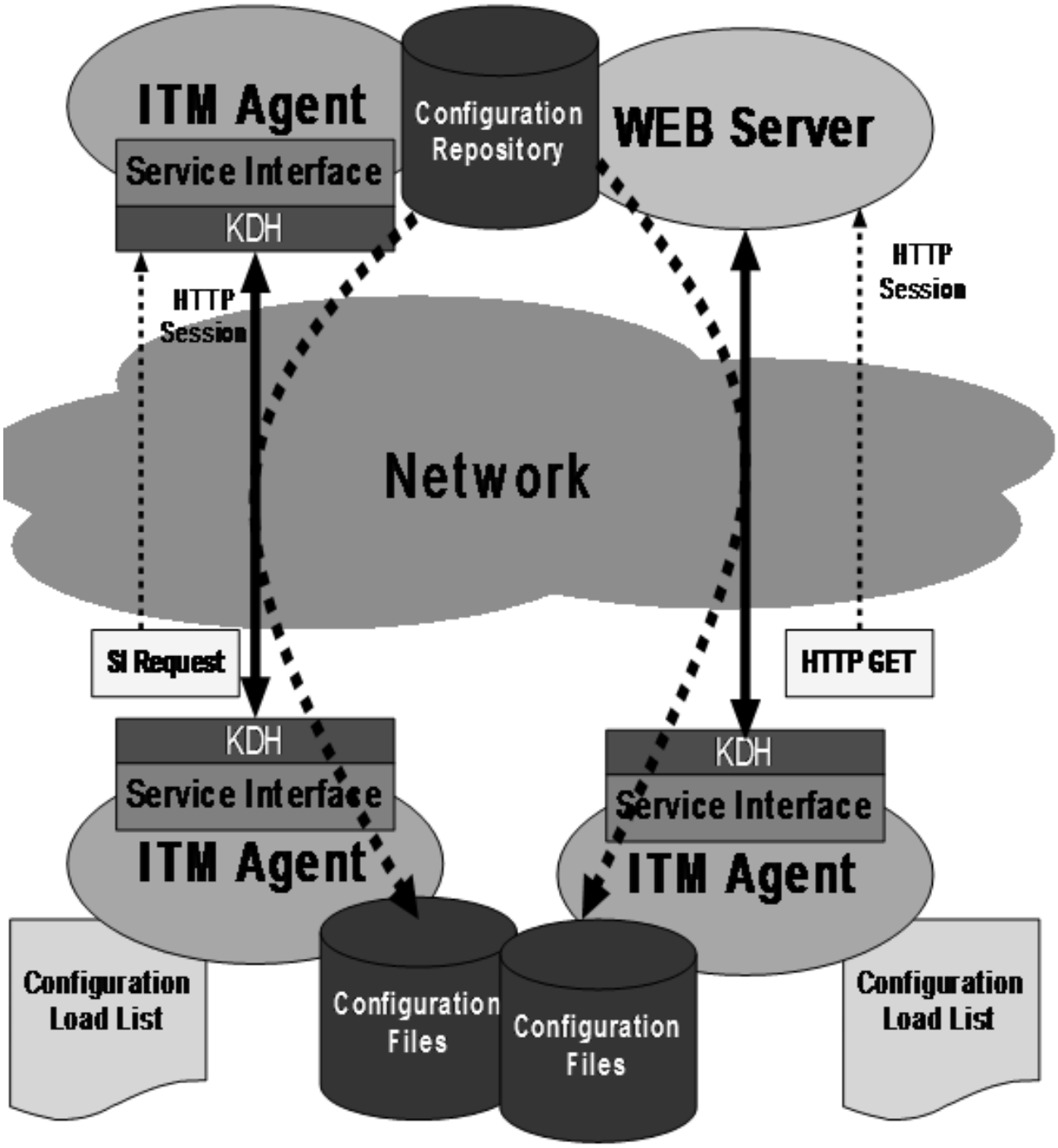


Figure 3. Central configuration components

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_cnfglist.xml
lz_trapcnfg.xml

nt
nt_cnfglist.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

i5/OS 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.

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 252.

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

i5/OS /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 270.

Elements

The elements and their attributes are case-insensitive. For example, you can enter <CONFIGSERVER>, <ConfigServer>, or <configserver>.

<ConfigurationArtifact> </ConfigurationArtifact>

ConfigurationArtifact is the root element identifying this 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="THRESHOLDS.XML"
    Path="ITM/Config/@PRODUCT@"
    Disp="threshold"
    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 207.

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 268."

<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

THRESHOLD
Situation threshold override 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 228.)

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 following Disp types require the agent to restart in order to read the updated configuration files:

TRAPCNFG

EIFCNFG

EIFMAP

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 also available for Disp="CUSTOM".

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 eventing from the monitoring agent is 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 11, "Agent Management Services," on page 149 for details on using Agent Management Services to monitor the availability of agents.

YES Instructs the agent to merge the downloaded file into the current file. New changes take affect dynamically without agent restart. This option is valid only for the following DISP values: CNFGLIST, PVTSIT, threshold, and THRESH.

Table 47. 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	N/A	Available	Default
EIFMAP	N/A	Available	Default
THRESHOLD	Available	Available	Default
UAMETA	N/A	Available	Default
PASCAP	CAP files are activated on download, therefore the Activate attribute does not apply.		
AAGP	Default	N/A	N/A

Table 47. Configuration load list <ConfigFile> element and the Activate options available for the Disp type. (continued)

Disp=	Activate="Yes"	Activate="Restart"	Activate="NO"
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.

Table 48. Keywords for the configuration load list.

@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. Examples: linux; win2003.
@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 257.

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 *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 267 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

i5/OS /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

i5/OS /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 uagent02 to identify a second installed Universal 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 of *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 is an example of 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 Support 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 207.

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 267.

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 higher.

Windows C:\ibm\ITM\TMAITM6\itmpwdsnmp.bat

Linux UNIX /opt/IBM/ITM/bin/itmpwdsnmp.sh

Here is an example of the display at the Linux command line:

```
itmpwdsnmp.sh
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.sh

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 will just use this file to locate the unique configlist that the agent should use. This CNFGLIST will only be in effect for a few seconds, we will define AAGP and other files in the In our sample, the agent looks in the directory identified by the @PRODUCT@ keyword for a file called cnfglist.xml. It is best to come to the config server for the bootstrap CNFGLIST, because that will allow you to modify the way you have agents identify their CNFGLIST by changing this file 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 278 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 269.)

Initiating with enterprise monitoring agent environment variables

Use the 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 Monitoring Services or from the command line:
   - Start Manage Tivoli Monitoring Services:
     Windows Click **Start → Programs →IBM Tivoli Monitoring → Manage Tivoli 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 267 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 *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 267 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 261.

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

   i5/OS 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* 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 started 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 143.
- 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 141) or through the CLI using **tacmd addSystem** from the monitoring server (see *IBM Tivoli Monitoring Command Reference*.
4. If the agent was not restarted started 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* describes the tacmds and their syntax and includes examples.

Procedure

1. Log onto the to the hub Tivoli Enterprise 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 in the kshsoap 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 *OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration* for instructions.

Chapter 14. 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 "Setting up data warehousing" topics in *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.

IBM Tivoli Monitoring Command Reference 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 higher. 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 retries 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 **hist** tacmd 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*.

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.

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 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 the Comprehensive List of IBM Tivoli Monitoring: ITM 6.X Product Codes at http://www-01.ibm.com/support/docview.wss?uid=swg21265222&myns=swgtiv&mynp=OCSSZ8F3&mync=E.
  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 Enterprise Monitoring 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 (described in “Creating a historical collection” and “histcreatecollection command”). 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 (described in “Creating another query”).

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 Enterprise Monitoring 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.

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.

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:

- i5/OS 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 the "IBM Tivoli product codes" appendix of 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 = $NODE$
```

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.

  – i5/OS 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.

Logging successful exports of historical data

Every successful export of historical data is logged in the data warehouse in a table called WAREHOUSELOG. The WAREHOUSELOG contains information such as origin node, table to which the export occurred, number of rows exported, time the export took place, and so forth. You can query this table to learn about the status of your exported history data.

Furthermore, the WAREHOUSELOG table contains error messages that explain the reasons that export failed.

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 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 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 49. 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 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 Tivoli Open Process Automation Library (OPAL) 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 server,

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.

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.

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.

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 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 Web site. 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 desired options 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 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 History Collection Configuration window:

Procedure

1. In the Tivoli Enterprise Portal, click the History Collection Configuration window 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 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.

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.

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:

- <*itm_install_dir*>\cms directory on the Tivoli Enterprise Monitoring Server.
- <*itm_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:

- <*itm_install_dir*>\cms directory on the monitoring server
- <*itm_installdir*>\tmaitm6\logs directory on the managed systems

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.

Note: The krarloff program will not output values that contain UTF8 data. All attributes that might contain UTF8 data will output the value BLANK.

Krarloff command on Windows

Run the krarloff rolloff program from the directory in which the Tivoli Enterprise Monitoring Server or the monitoring agent is run by entering the following at the command prompt, where the [ ] square brackets denote the optional parameters and the { } curly braces denote a required parameter:

```
krarloff [-h] [-d delimiter] [-g] [-m metafile] [-r rename-to-file]
[-o output-file] {-s source | source-file name}
```

Krarloff rolloff program parameters

The following table lists the krarloff rolloff program parameters, their purpose, and default values.

Table 50. 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.
-d	tab	Delimiter used to separate fields in the output text file. Valid values are any single character (for example, a comma).
-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.
-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 an i5/OS 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 i5/OS 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 i5/OS 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 i5/OS 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 i5/OS 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'] ['-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 i5/OS 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 i5/OS 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 Enterprise 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.

See *IBM Tivoli Monitoring: Command Reference* for additional information about **itmcmd history**.

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 ] 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 *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 <*itm_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, as described in the "Enable historical data store maintenance" topic of the *OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration* guide, 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.

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 the "Creating a historical collection" topic of the *Tivoli Enterprise Portal User's Guide*.

Related reference

"Manual archiving procedure" on page 324

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 KBDXTRA 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 datasets are kept in read/write state even if they are not active. This makes the datasets unavailable if the Tivoli Enterprise Monitoring Server is running. Thus, jobs cannot be run against the active datasets and the inactive datasets 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 51. 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 52. KPDXTRA parameters

Parameter	Default Value	Description
PREF=	none	Required parameter. Identifies the high level qualifier where the output files are written.
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 *&hilev.&midlev.*RKANMOD or *&hilev.&midlev.*TKANMOD library, where:

- *&hilev* is the library in which the Tivoli Enterprise Monitoring Server was installed
- *&midlev* is the name you have provided at installation time.

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.

Chapter 15. 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.

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 Open Process Automation

Library for a number of Tivoli Monitoring products. You can search "Tivoli Common Reporting" to find report packages on OPAL.

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 space.

- 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 space.

For more information about Tivoli Common Reporting, including information about installing and administering Tivoli Common Reporting and creating reports, refer to the IBM Tivoli Common Reporting for Asset and Performance Management information center.

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 Version 6.2 Fix Pack 1 or later
- Tivoli Common Reporting Version 1.1 or later

  Tivoli Common Reporting for Asset and Performance Management Version 1.3 is included with IBM Tivoli Monitoring Version 6.2 Fix Pack 2 (or later) and it is the recommended version to use with the monitoring agent reports. This version of TCR includes Cognos Business Intelligence and Reporting Version 8.4.

  For other monitoring agents, install Tivoli Common Reporting Version 1.1 or later.

  If you have not done so already, install and configure Tivoli Common Reporting, using the information found in the IBM Tivoli Common Reporting for Asset and Performance Management information center. To ensure that Tivoli Common Reporting is running, go to `https://computer_name:16316/ibm/console/`. The default port number for http is 16315. If you already have an earlier version of Tivoli Common Reporting and you installed Version 1.3 in a different directory, 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.
- Historical data stored in a database manager product supported by IBM Tivoli Monitoring Version 6.2 Fix Pack 1 or later

  BIRT reports in this guide are historical reports, reporting against data collected in Tivoli Data Warehouse 6.2 Fix Pack 1 or later. For information about supported databases, refer to "Supported databases for the Tivoli Enterprise Portal Server and 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, which describes each report. In particular, they contain 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'
```

Notes:

1. In Tivoli Common Reporting for Asset and Performance Management, both BIRT and Cognos report engines can co-exist.
2. Although it is not required, you can install the Eclipse BIRT Report Designer, Version 2.2.1. Eclipse BIRT Report Designer, along with the *Tivoli Common Reporting: Development and Style Guide*, can be used to edit report templates or create new reports.

For software requirements for running the BIRT Report Designer and to download it, refer to Business Intelligence and Reporting Tools on the Eclipse web site or the IBM developerWorks Tivoli Common Reporting space. Download the development and style guide from the IBM Tivoli Common Reporting for Asset and Performance Management information center.

Upgrading from a previous version

BIRT OS monitoring agent reports continue to be delivered on OPAL. For other monitoring agents, which were previously delivered on OPAL and ran under Tivoli Common Reporting V1.1.1, you can upgrade your Tivoli Common Reporting product level to the version (V1.3 or later) included with Tivoli Monitoring V6.2.2 Fix Pack 2 (or later) monitoring agent without reinstalling the report packages that were downloaded from OPAL or from the product media.

BIRT report packages are available on the IBM Open Process Automation Library for the base OS monitoring agents. A DVD is available for the Cognos version of the IBM Tivoli Monitoring Agent Reports. Download the package that corresponds to your Tivoli Management Services infrastructure version.

Tivoli Common Reporting Version 1.1.1 ran under the Integrated Solutions Console and installed into a different location from Tivoli Common Reporting Version 1.2 and higher, which runs under the Tivoli Integrated Portal and now relies on that product for infrastructure support. Versions 1.1.1 and 1.3 can coexist on the same computer or you can migrate the reports you downloaded from OPAL to Version 1.3. You do not need to reinstall the report packages. There are two options for migrating reports from Version 1.1.1 to Version 1.3:

- During installation of Tivoli Common Reporting Version 1.3, the installer program detects if Version 1.1.1 is installed and asks if you want to migrate these reports. Say **Yes**.

  Note: If you migrated the report package you downloaded from OPAL to Tivoli Common Reporting Version 1.3, be sure that the previously installed reports are overwritten. When you import the report package, click on **Advanced Options** in the **Import Report Package** text box and select the **Overwrite** check box.
- Migrate report packages manually.

Both of these options are explained in the Version 1.3 *Tivoli Common Reporting User's Guide*.

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.

Limitations

The limitations of the reports produced by Tivoli Common Reporting are described in this section.

- 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 will 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 and Changing global configuration settings.
- Reports that cover a long time period or a processing-intensive attribute might cause SQL arithmetic overflow.
- These 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 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).
- The IBM Tivoli Monitoring OS Agents Cognos reports are coded to connect to a data connection in Tivoli Common Reporting with the name "TDW".

Importing and running Cognos reports

You must create and populate the Cognos dimensions tables and then run the reports installer to enable Tivoli Common Reporting for the Tivoli Monitoring OS agents.

Creating shared dimensions tables and populating the time dimensions table

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:

- Schema IBM_TRAM. TRAM stands for Tivoli Reporting and Analytics Model which is the common data model used by Tivoli products.
- Table TIME_DIMENSION with 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.
- Table MONTH_LOOKUP globalizes the month names for Time Dimension.
- Table WEEKDAY_LOOKUP globalizes the weekday names for Time Dimension.
- 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.

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:

     ```
     start 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 generates from 12/31/2009 to 12/31/2010 with 60-minute granularity.

     ```
     db2 call IBM_TRAM.CREATE_TIME_DIMENSION('2009-12-31-00.00.00.000000',
     '2010-12-31-00.00.00.000000', 60);
     ```
- **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 to.
  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. **createSchema.sql**
     b. **createProcedure.sql**
     c. **clean.sql**
     d. **populateTimeDimension.sql** Also, modify the boundary parameters for the time dimension and granularity. For example,

```
@startDate    = '2010-01-01 00:00:00',
@endDate      = '2010-07-08 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:

   ```
   @reports/myreports/populateTimeDimension.sql
   '2008-12-31 00:00' '2011-12-31 00:00' '60'
   ```

Results

The shared dimensions and time dimensions tables are complete.

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

Before starting the procedure to build the resource dimension table, you must first 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

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 YQMQDH
  -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 -t "knt" -o "NT Computer Information" -u sysadmin
```

See "Historical collection configuration" or **tacmd histconfiguregroups**, **tacmd histcreatecollection**, and **tacmd histstartcollection** in the *IBM Tivoli Monitoring Command Reference* for details.

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. Create the tables:

     ```
     db2 -tf gen_resources.db2
     ```

     After the command completes successfully, a new table is shown under the ITMUSER schema: ManagedSystem.
  4. Create the stored procedure to populate the ManagedSystem table:

     ```
     db2 -td@ -vf populate_resources.db2
     ```
  5. To populate ManagedSystem table, call the stored procedure:

     ```
     db2 call ITMUSER.POPULATE_OSAGENTS();
     ```
- **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
     ```
  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.

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.

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.

Installing and running IBM Cognos reports

Use the reports installer to enable Tivoli Common Reporting for Tivoli Monitoring OS agents.

Although these instructions are for the OS agent reports, you can follow them for your product reports and replace the OS agent entries with those for your product.

Before you begin

The monitoring agent report models are based on IBM Cognos. You must have IBM Tivoli Common Reporting 1.3 (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 shared dimensions tables and populating the time dimensions table" on page 330 and "Creating and populating the resource dimension table" on page 333, and connected to the data warehouse as described in "Connecting to the Tivoli Data Warehouse using the database client over ODBC" on page 335.

Procedure

1. From the Cognos reports disk or the directory where the report package was extracted, launch the setup script:
   - Windows `setup.bat` *`tcr_install_dir`*. (If Tivoli Common Reporting is installed in the default directory, you do not need to specify an installation directory.)
   - Linux UNIX `./setup.sh` *`tcr_install_dir`*.
2. Click **Next** on the installer welcome page.
3. Select **OS Agents Reporter** from the **Cognos: available report sets** field and move it to the **To be installed** field.
4. Read all instructions that appear in the next window to check if your system meets all prerequisites for installing the reports. Proceed according to the instructions. Click **Next**.
5. Specify a set of parameters and click **Next**:
   - Enter the Cognos name space. (OS Agents reports do not have their own namespace; they use what is assigned by Tivoli Common Reporting.)
   - Enter the Cognos user name and password. Example: `tipadmin/tippass`
   - Specify the Cognos dispatcher URL. Example of the URL for the VMware VI monitoring agent: `http://VMMProvider:tipadmin:tippass@localhost:16315/tarf/servlet/dispatch`. The port number is 16316 for https or might be different if Tivoli Common Reporting V1.3 was installed in an independent install directory (for example, the user chose not to upgrade) on a computer that Tivoli Common Reporting V1.2 is currently installed on. For that scenario, the correct http port is 16345 and the https port is 16346.
6. This step lets you skip defining the required data sources. Select the check box if you want to do so. Do not skip defining the data sources unless you already have them created. They are required for the agent reports to work properly.
7. Unless you have chosen to skip this step in the previous window, set the data source parameters in the window that appears:

- Data source type: Oracle, Microsoft SQL Server (ODBC) or DB2. After you have chosen the data source type, the **Copy** button automatically enters default values in the JDBC driver class name, JDBC URL value, and JDBC user ID fields.
- DB2 Database Name, Oracle Database Name, or Microsoft SQL Server Database Name depending on the data source type you have specified in the previous field
- Data source user name
- Data source password

8. Click **Next**. This installation step takes some time to complete.
9. After the summary information is displayed, read it carefully to check if the information is correct. If it is, click **Install**. Use the **Back** button if you want to change any of the previously specified parameters. A window appears that shows the progress of your installation.
10. After the post-installation report appears, check that the installation was successfully completed, and click **Finish**.

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" topics.

Importing and running BIRT reports

If you have the Eclipse BIRT (Business Intelligence and Reporting Tools) Report Designer and have downloaded the OS Agent report packages from OPAL, you can import and run the reports.

Ensure that historical reporting is enabled

The first step in preparing to run BIRT reports is to ensure that historical reporting is enabled.

About this task

The reports in this report package run against long-term historical data that is stored in the Tivoli Data Warehouse. Before you can run these 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 *IBM Tivoli Monitoring Installation and Setup Guide*.)
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 *Configuring the Tivoli Enterprise Monitoring Server on z/OS*) and *OMEGAMON XE and Tivoli Management Services on z/OS Common Planning and Configuration*.)

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.
4. Install the Summarization and Pruning Agent. (See *IBM Tivoli Monitoring Installation and Setup Guide*.) Use the "Historical collection configuration" feature in the Tivoli Enterprise Portal to enable summarization and pruning for historical data collection. The attribute groups used to provide summary reports are identified in the product help or user's guide.

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.

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

Take these steps to import a report package:

Procedure

1. Launch the Tivoli Integrated Portal administrative console and log in.
2. In the navigation tree, expand the **Reporting** item.
3. Click **Common Reporting with BIRT**.
4. In the report navigation window, click the **Navigation** tab (default).
5. Right-click the root node of the navigation tree (Report Sets) and click **Import Report Package**.
6. Specify the location and file name of the report package .zip file to import. You must specify the name of a security set. You can type directly in the entry field or click **Browse** to open a file selection window from which you can select the report package file. If you want to overwrite existing reports, select ☑ **Overwrite** to indicate that any existing file with the same name as an imported file is to be overwritten.

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.

Related reference

Tivoli Common Reporting Information Center - Importing a report package

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*`/products/tcr/lib/birt-runtime-2_2_1/ReportEngine/plugins/ org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919/drivers`

- For a DB2 data source, copy the DB2 JDBC drivers as well as the license jar file to the same location. You can copy db2jcc.jar and db2jcc_licence_cu.jar file on the db2 server system from *db2_installdir*/javalocation (for example, C:\Program Files\IBM\SQLLIB\java).

3. Right-click the name of a report and click **Data Sources**.
4. Click **Edit** in the Report Data Sources dialog. Open the Enabling a JDBC Driver list to see the location for the JDBC drivers.
5. In the **JDBC Driver** field, enter the path to the JDBC driver:

   IBM DB2
   ```
   com.ibm.db2.jcc.DB2Driver
   ```

   Microsoft SQL Server
   ```
   com.microsoft.sqlserver.jdbc.SQLServerDriver
   ```

   Oracle
   ```
   oracle.jdbc.driver.OracleDriver
   ```
6. In the **JDBC URL** field, enter the URL:

   IBM DB2
   ```
   jdbc:db2://<HOSTNAME>:<DBPORT>/
   <DBNAME>:currentSchema=<SCHEMANAME>;
   ```

   Example: `jdbc:db2://tivrepo rting.raleigh.ibm.com:50000/ WAREHOUS:c urrentSchema=ITMUSER;`

   Microsoft SQL Server
   ```
   jdbc:sqlserver://<HOSTNAME>:<DBPORT>;databasename=<DBNAME>;
   ```

   Example: `jdbc:sqlserver:// tivreporting.raleigh.ibm.com:1433;databasename=WAREHOUS;`

   Oracle
   ```
   jdbc:oracle:thin:@<HOSTNAME>:<DBPORT>:<DBNAME>
   ```

   Example: `jdbc:oracle:thin:@tivreporting.raleigh.ibm.com:1521:WAREHOUS`
7. Change your username and password in the window to be the same as your database manager login ID (for example, your DB2 username).
8. Click **Save**.

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 space.

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*.

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 the Tivoli Common Reporting Browser. From the Start menu, select **All Programs > Tivoli Common Reporting > Launch Reporting Browser** and log into the Tivoli Integrated Portal.
2. In the navigation tree on the left, expand the **Reporting** item.
3. Click **Common Reporting with BIRT**. 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 information about data sources, refer to 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, default heap size might be your problem:

   ```
   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 on how to increase the default heap size.
7. Click the icon beside a report name to launch the parameters window and produce a report in the desired format. You can select from these report format by right-clicking the report name and selecting one of these report formats: 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 press Enter to accept all displayed defaults.
8. Click **Run** to generate a report matching your parameter definitions.

Results

You will see an hour glass while Tivoli Common Reporting gathers report data and creates formatted output. After processing finishes, the report viewer opens in a new browser tab or instance, displaying the formatted report using the appropriate browser plug-in. You can view the report in your browser or save the formatted output using the browser or plug-in capabilities.

What to do next

If no report is generated or you see a message indicating that the requested data is unavailable, refer to 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 16. 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.

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 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 higher 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 Tivoli Enterprise 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 Tivoli Enterprise Monitoring Server and importing into another.

tacmd bulkExportPcy
tacmd bulkImportPcy
: Export all Tivoli Monitoring policies from one hub Tivoli Enterprise 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 CandleNet 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/lnxenvnocms 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/lnxenvnocms 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. Tivoli Enterprise Monitoring Web Services

This appendix describes the Tivoli Enterprise Monitoring Web Services feature. The Tivoli Enterprise 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.

Tivoli Enterprise Monitoring Web Services implements a client/server architecture. The client sends Simple Object Access Protocol (SOAP) requests to the SOAP server. The server receives and processes the SOAP requests from the client.

Predefined SOAP methods let you perform many functions within the monitored environment. 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.

See the *IBM Tivoli Monitoring: CandleNet Portal User's Guide* for instructions on installing and configuring this product on the Tivoli Enterprise Monitoring Server.

Note:

1. 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.
2. If you will be issuing SOAP request on Internet Explorer version 5.0x, you must first install service pack MSXML 3.0. Otherwise, the SOAP requests will not succeed. Internet Explorer version 6.0x includes this service pack.

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 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 Monitoring Services:
   a. Windows Click **Start → Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**.
   b. Linux or UNIX Change directory to `<itm_install_dir>/bin` and enter: `./itmcmd manage`
2. Right-click Tivoli Enterprise Monitoring Server and click **Reconfigure**.
3. Select or clear the ☐ **Security: Validate User** field.
4. Open Manage Tivoli 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 53. 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 54. SNA Fields in Hub Specification Dialog

Field	Description
Network Name	Your site SNA network identifier.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU6.2 LOGMODE	The name of the LU6.2 logmode. Default: CANCTDCS.
TP Name	The Transaction Program name for the monitoring server.

    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.

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**.

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 Monitoring Services:

Procedure

1. Change to the *<itm_install_dir>*/bin directory and start Manage Tivoli Monitoring Services by entering the following command:

   ```
   ./itmcmd manage
   ```

   The Manage Tivoli 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 acknowledgements. 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 acknowledgements (*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 acknowledgement 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 acknowledgement 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.

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).

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.

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.

Note: Before you can access newly created Universal Agent objects, the hub monitoring server where the SOAP server is running must be recycled. See the *IBM Tivoli Monitoring Installation and Setup Guide* for instructions on configuring the hub monitoring server.

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 higher 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 fill in 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.
- 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.

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>
```

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.

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>
```

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 any single character. ' * ' 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.

The lower-level methods are:

- <CT_Get>
- <CT_Redirect>
- <attach>
- <insert>

The **<CT_Get>** and **<CT_Redirect>** tags are used as described in "SOAP methods" on page 356. The **<attach>** tag is used to load a file. The file must be located in the `\ibm\itm\cms\html` directory. 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>
  <insert>
    <insertelement>
      <insertdata>
        This data has been inserted complements of CT SOAP server.
      </insertdata>
    </insertelement>
  </insert>
  <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.

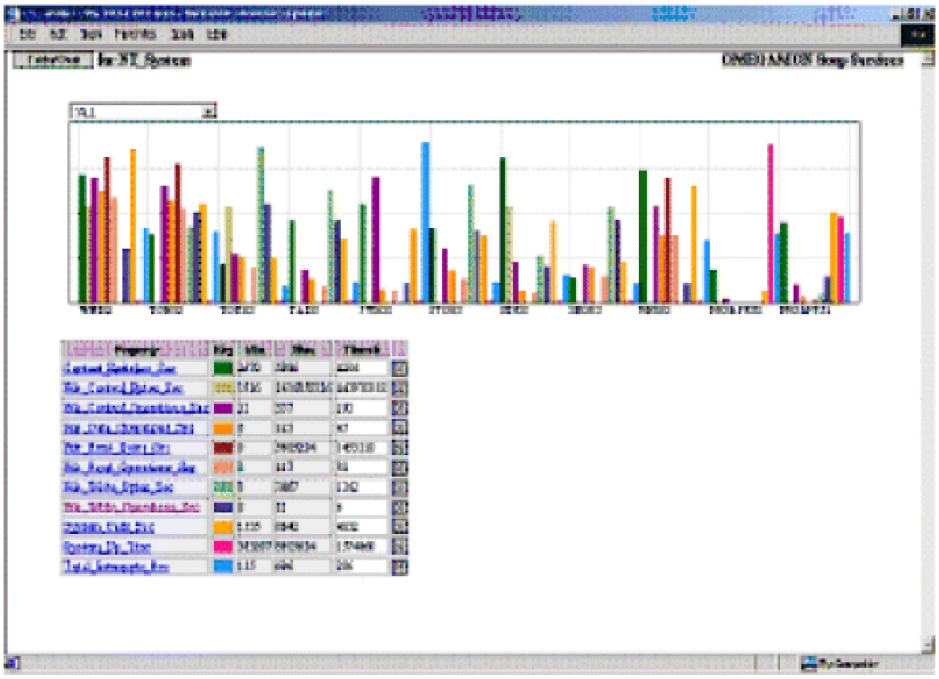


Figure 4. Data snapshot chart and table

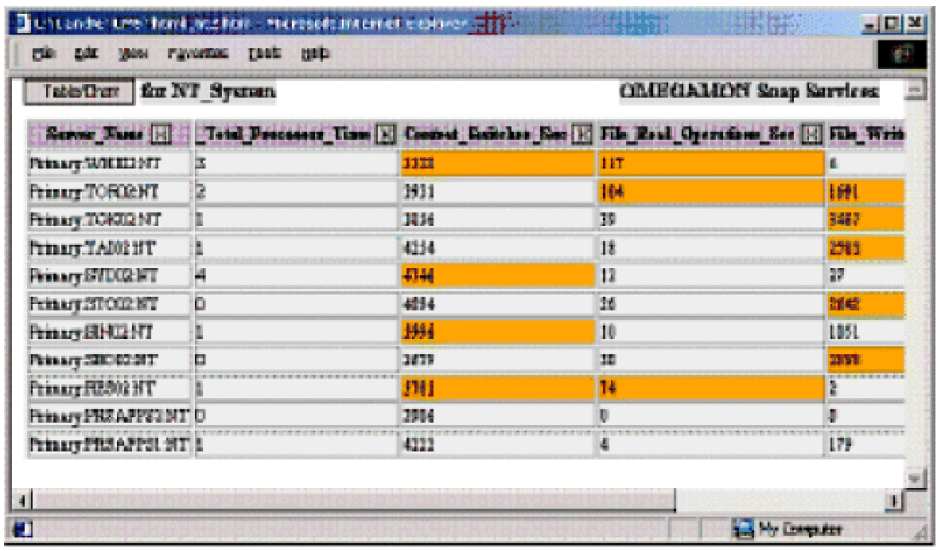


Figure 5. Data snapshot table

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 IOM

You can create a System Automation for Integrated Operations Management REXX application that calls JSCRIPT SOAP functions to forward any SA IOM trapped message and display it on a Universal Message console. You can use SA IOM 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 IOM 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.

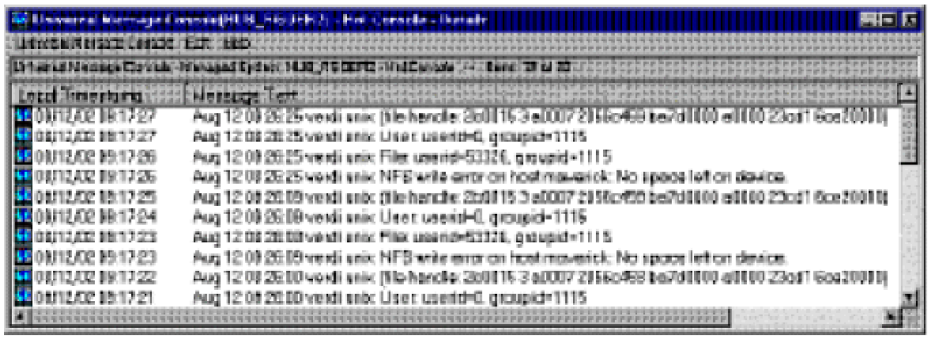


Figure 6. Universal Message Console Showing Messages Received

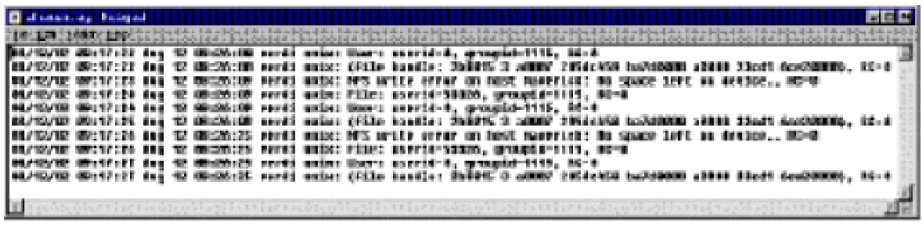


Figure 7. Message Log Details

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 higher):

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 acknowledgement
3. The alert acknowledgement 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. Using the Tivoli Management Services Discovery Library Adapter

Use the Discovery Library Adapter (DLA) program for scanning your monitored environment to identify the managed systems. You can then feed this information (an XML output file) into the Tivoli Application Dependency Discovery Manager's (TADDM) Change and Configuration Management Database (CCMDB).

The 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 DLA gathers information by querying the hub Tivoli Enterprise 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.

For example, "IMN1:SYS1:IMS" is the managed system name for an OMEGAMON XE for IMS™ agent. The DLA discovers the following information:

- A z/OS computer named "IMN1"
- An IMS subsystem named "SYS1"
- A relationship between the SYS1 z/OS computer and IMN1 IMS

For agents that use IP, IP.PIPE, or IP.SPIPE, the DLA can discover the IP address where the agent is running. As well, the DLA discovers the operating system for the computer where the agent is running, regardless of whether an OS monitoring agent is running on that computer.

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 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 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 the CCMDB, you must transfer the XML file to the Discovery Library File Store and then use the Discovery Library Bulk Loader.

Related reference

Tivoli Change and Configuration Management Database

Problems with the Tivoli Monitoring DLA

Tivoli Monitoring Command Reference

Appendix C. 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 55. 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 55. 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 197 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 56. 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 56. 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 56. 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 57. 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 57. 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 57. 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 D. 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

This appendix contains information about the publication related to IBM Tivoli Monitoring and to the commonly shared components of Tivoli Management Services.

These publications are listed in the following categories:

- IBM Tivoli Monitoring library
- Related publications

See the *IBM Tivoli Monitoring Documentation Guide* for information about accessing and using the publications. You can find the *Documentation Guide* in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp. To open the *Documentation Guide* in the information center, select **Using the publication** in the **Contents** pane.

To find a list of new and changed publications, click **What's new** on the Welcome page of the IBM Tivoli Monitoring and OMEGAMON XE Information Center. To find publications from the previous version of a product, click **Previous versions** under the name of the product in the **Contents** pane.

IBM Tivoli Monitoring library

The following publications provide information about IBM Tivoli Monitoring and about the commonly shared components of Tivoli Management Services:

- *Quick Start Guide*, GI11-8058

  Introduces the components of IBM Tivoli Monitoring.
- *Installation and Setup Guide*, GC32-9407

  Provides instructions for installing and configuring IBM Tivoli Monitoring components on Windows, Linux, and UNIX systems.
- *Program Directory for IBM Tivoli Management Services on z/OS*, GI11-4105

  Gives instructions for the SMP/E installation of the Tivoli Management Services components on z/OS.
- *Configuring the Tivoli Enterprise Monitoring Server on z/OS*, SC32-9463Gives detailed instructions for using the Configuration Tool to configure Tivoli Enterprise Monitoring Server on z/OS systems. Includes scenarios for using batch mode to replicate monitoring environments across the z/OS enterprise. Also provides instructions for setting up security and for adding application support to a Tivoli Enterprise Monitoring Server on z/OS.
- *Administrator's Guide*, SC32-9408

  Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.
- *High-Availability Guide for Distributed Systems*, SC23-9768

  Gives instructions for several methods of ensuring the availability of the IBM Tivoli Monitoring components.
- Tivoli Enterprise Portal online help

Provides context-sensitive reference information about all features and customization options of the Tivoli Enterprise Portal. Also gives instructions for using and administering the Tivoli Enterprise Portal.

- *Tivoli Enterprise Portal User's Guide*, SC32-9409

  Complements the Tivoli Enterprise Portal online help. The guide provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.

- *Command Reference*, SC32-6045

  Provides detailed syntax and parameter information, as well as examples, for the commands you can use in IBM Tivoli Monitoring.

- *Troubleshooting Guide*, GC32-9458

  Provides information to help you troubleshoot problems with the software.

- *Messages*, SC23-7969

  Lists and explains messages generated by all IBM Tivoli Monitoring components and by z/OS-based Tivoli Management Services components (such as Tivoli Enterprise Monitoring Server on z/OS and TMS:Engine).

- *IBM Tivoli Universal Agent User's Guide*, SC32-9459

  Introduces you to the IBM Tivoli Universal Agent, an agent of IBM Tivoli Monitoring. The IBM Tivoli Universal Agent enables you to use the monitoring and automation capabilities of IBM Tivoli Monitoring to monitor any type of data you collect.

- *IBM Tivoli Universal Agent API and Command Programming Reference Guide*. SC32-9461

  Explains the procedures for implementing the IBM Tivoli Universal Agent APIs and provides descriptions, syntax, and return status codes for the API calls and command-line interface commands.

- *Agent Builder User's Guide*, SC32-1921

  Explains how to use the Agent Builder for creating monitoring agents and their installation packages, and for adding functions to existing agents.

Documentation for the base agents

If you purchased IBM Tivoli Monitoring as a product, you received a set of base monitoring agents as part of the product. If you purchased a monitoring agent product (for example, an OMEGAMON XE product) that includes the commonly shared components of Tivoli Management Services, you did not receive the base agents.

The following publications provide information about using the base agents.

- Operating system agents:
  - *Windows OS Agent User's Guide*, SC32-9445
  - *UNIX OS Agent User's Guide*, SC32-9446
  - *Linux OS Agent User's Guide*, SC32-9447
  - *i5/OS Agent User's Guide*, SC32-9448
  - *UNIX Log Agent User's Guide*, SC32-9471
- Agentless operating system monitors:
  - *Agentless Monitoring for Windows Operating Systems User's Guide*, SC23-9765
  - *Agentless Monitoring for AIX Operating Systems User's Guide*, SC23-9761
  - *Agentless Monitoring for HP-UX Operating Systems User's Guide*, SC23-9763
  - *Agentless Monitoring for Solaris Operating Systems User's Guide*, SC23-9764

  - *Agentless Monitoring for Linux Operating Systems User's Guide*, SC23-9762
- Warehouse agents:
  - *Warehouse Summarization and Pruning Agent User's Guide*, SC23-9767
  - *Warehouse Proxy Agent User's Guide*, SC23-9766
- System P agents:
  - *AIX Premium Agent User's Guide*, SA23-2237
  - *CEC Base Agent User's Guide*, SC23-5239
  - *HMC Base Agent User's Guide*, SA23-2239
  - *VIOS Premium Agent User's Guide*, SA23-2238
- Other base agents:
  - *Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint User's Guide*, SC32-9490

Related publications

You can find useful information about related products in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp.

Other sources of documentation

You can also obtain technical documentation about IBM Tivoli Monitoring and related products from the following sources:

- IBM Tivoli Open Process Automation Library (OPAL)

  http://www.ibm.com/software/tivoli/opal

  OPAL 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/probsub.html, or more directly through your product Web site, which contains a link to Technotes (under **Solve a problem**).
- Tivoli wikis on the IBM developerWorks Web site

  Tivoli Wiki Central at http://www.ibm.com/developerworks/wikis/display/tivoli/Home is the home for interactive wikis that offer best practices and scenarios for using Tivoli products. The wikis contain white papers contributed by IBM employees, and content created by customers and business partners.

  Two of these wikis are of particular relevance to IBM Tivoli Monitoring:
  - Tivoli Distributed Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivolimonitoring/Home provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Management products.
  - Tivoli System z Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivoliomegamon/

Home provides information about the OMEGAMON XE products, NetView for z/OS, Tivoli Monitoring Agent for z/TPF, and other System z monitoring and application management products.

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 Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.
- Go to the IBM Tivoli Distributed Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivolimonitoring/Home. Feel free to contribute to this 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 http://www.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 Web site 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 Web site at http://www.ibm.com/software/support.
2. Click **My support** in the far upper-right corner of the page under **Personalized support**.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. The **Edit profile** tab is displayed.
5. In the first list under **Products**, select **Software**. In the second list, select a product category (for example, **Systems and Asset Management**). In the third list, select a product sub-category (for example, **Application Performance & Availability** or **Systems Performance**). A list of applicable products is displayed.
6. Select the products for which you want to receive updates.
7. Click **Add products**.
8. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
9. In the **Documents** list, select **Software**.
10. Select **Please send these documents by weekly email**.
11. Update your e-mail address as needed.
12. Select the types of documents you want to receive.
13. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

Online
Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

By phone
Call 1-800-IBM-4You (1-800-426-4968).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects. The easiest way to obtain that assistance is to open a PMR or 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 Web site at http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm .

  By phone
  For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with Linux, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html and click the name of your geographic region for phone 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.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Glossary

activity
One phase within a sequence of predefined steps called a **policy** that automate system responses to a **situation** that has fired (that is, become true).

administration mode
See workspace administration mode.

affinity
A label that classifies objects by **managed system**.

agent Software installed on systems you want to monitor that collects data about an operating system, subsystem, or application running on each such system. Because an executable file gathers information about a **managed system**, there is always a one-to-one correspondence between them.

agentless monitor
An agentless monitor uses a standard API (such as SNMP or CIM) to identify and notify you of common problems with the operating system running on the same computer. Thus, as their name implies, the agentless monitors can retrieve monitoring and performance data without requiring OS agents on the computers being monitored. The agentless monitors provide monitoring, data gathering, and event management for Windows, Linux, AIX, HP-UX, and Solaris systems.

agentless monitoring server
A computer with an OS agent installed that has one or more agentless monitors running on it. Each agentless monitoring server can support up to 10 active instances of the various types of agentless monitors, in any combination. Each instance can communicate with up to 100 remote nodes, which means a single agentless monitoring server can support as many as 1000 monitored systems.

alert A warning message that appears at a console to indicate that something has occurred that may require intervention.

associate
The process of linking a situation with a Navigator item that enables a light to go on and a sound to play for an open event. Predefined situations are associated automatically, as are situations created or edited through the Navigator item pop-up menu. When you open the Situation editor from the toolbar, any situations you create cannot be associated with a Navigator item during this editing session. You must close the Situation editor, then open it again from the pop-up menu of the Navigator item with which the situation should be associated.

attribute
(1) A system or application element being monitored by the , such as Disk Name and Disk Read/Writes Per Second. (2) A characteristic of a managed object; that is, a field in the data structure of a managed object or in the workspace associated with that managed object. (3) A field in an ODBC-compliant database.

attribute group
A set of related attributes that can be combined in a data view or a situation. When you open the view or start the situation, data samples of the selected attributes are retrieved. Each type of has its own set of attribute groups.

browser client
The software installed with the that is downloaded to your computer when you start in browser mode. The browser client runs under control of a web browser.

chart A graphical view of data returned from a . A data point is plotted for each attribute chosen and, for bar and pie charts, a data series for each row. Types of charts include pie, bar, plot, and gauge.

class file
A file containing Java object code for a single Java object class.

class loader
A Java component that loads Java **class files**.

client An application that receives requested data from a **server**.

client/server architecture
An architecture in which the client (usually a personal computer or workstation) is the machine requesting data or services and the server is the machine supplying them. Servers can be microcomputers, minicomputers, or mainframes. The client provides the user interface and may perform application processing. In the is the client to the , whereas the is the client to the .

A **database server** maintains the databases and processes requests from the client to extract data from or to update the database. An **application server** provides additional business-support processing for the clients.

Common Object Request Broker Architecture
An industry specification for the design and standardization of different types of object request brokers (ORBs). ORBs allow different computers to exchange object data; CORBA enables ORBs from different software vendors (often running under dissimilar computer systems and operating systems) to exchange object data. CORBA is an architecture and specification that facilitates communication among program components in a network using objects. The is a CORBA implementation.

Configure History permission
Your user ID must have Configure History permission to open the History Collection Configuration window for setting up history files and data rolloff. If you do not have this permission, you will not see the menu item or tool for historical configuration.

CORBA
See Common Object Request Broker Architecture.

critical state
The indication that a situation associated with a Navigator item is in an unacceptable state and that you must take corrective action. The critical state is represented by the color red.

Custom Navigator Views permission
Your user ID has a Modify checkbox for the Custom Navigator Views feature. This permission must be enabled for you to open the Navigator view editor to maintain and update Navigator views.

datasource name
The name that is stored in the database server and that enables you to retrieve information from the database through ODBC. The DSN includes such information as the database name, database driver, user ID, and password.

datasources
Data pertaining to J2EE data sources, which are logical connections to database subsystems.

DB2 on the workstation
IBM's DB2 Database for Linux, UNIX, and Windows systems is a relational database management system that runs on desktop computers. You install a DB2 database on the same system as the Tivoli Enterprise Portal Server; it stores the portal server's queries, customized workspaces, user IDs, and custom Navigator views. DB2 on the workstation can also serve as the data repository for the Tivoli Data Warehouse, which stores historical monitoring information.

Demilitarized Zone
The area of a Worldwide Web application that enables a company to host Internet services without allowing unauthorized access.

Derby An open-source, public-domain, relational database management system implemented in Java and designed to conform to accepted database standards (such as **SQL** and **JDBC**). Derby came about when IBM contributed its Cloudscape database manager to the Apache project and features a small machine footprint. implements Derby as an embedded database within its ; in other words, the database is installed with the , and it runs within the 's Java virtual machine.

desktop client
Software supplied with that you install on a workstation that you plan to use for interacting with the and the . The desktop client provides the graphical user interface into the network.

detailed attribute name
The name used in formulas, expert advice, Take Action commands, and headers and footers when referencing a attribute. In the Properties and Situation editors, you click Show Formula, then check Show detailed formula to see the detailed attribute name.

display item
An attribute designated to further qualify a situation. With a display item set for a multiple-row attribute group, the situation continues to look at the other rows in the sampling and opens more events if other rows qualify. The value displays in the event workspace and in the message log and situation event console views. You can select a display item when building a situation with a multiple-row attribute group.

distribution
The managed systems on which the situation is running.

DLL See Dynamic Link Library.

DMZ See Demilitarized Zone.

DSN See datasource name.

Dynamic Link Library
A composite of one or more executable objects that is bound together by a linking procedure and loaded at run time (rather than when the application is linked). The code and data in a dynamic link library can be shared by several applications simultaneously. DLLs apply only to Windows operating environments

EIB See Enterprise Information Base.

endcode
You assign endcodes in a **policy** as you connect one **activity** to another. The endcode indicates the result of this activity that will trigger the next activity.

Enterprise Information Base
A database used by the that serves as a repository of shared objects for all systems across your enterprise. The EIB stores all persistent data, including situations, policies, user definitions, and managed-object definitions.

enterprise situation
A situation that is created for a Tivoli Enterprise Monitoring Agent that reports events to the Tivoli Enterprise Monitoring Server it connects to. Enterprise situations are centrally defined at the monitoring server and distributed at agent startup. See also situation.

event An action or some occurrence, such as running out of memory or completing a transaction, that can be detected by a situation. Events cause a change in the state of a managed object associated with a situation, thereby make the situation true and causing an alert to be issued.

event indicator
The colored icon that displays over a Navigator item when an event opens for a situation.

event item
A Navigator item that shows when you open the event workspace for a true situation (by selecting it from the event flyover listing or from the situation event console pop-up menu).

event sound
The sound file that plays when an event opens. This sound file is set in the Situation editor when the situation is associated with a Navigator item and can differ for different Navigator items.

expert advice
A description within the Situation editor of each situation provided with a monitoring agent to help you quickly understand and interpret events arising from it.

Extensible Markup Language
A data-description language derived from Standard Generalized Markup Language (SGML). A tool for encoding messages so they describe their own fields, XML allows you to format a document as a data structure. As program objects, such documents can have their contents and data hidden within the object, which allows you to control who can manipulate the document and how. In addition, documents can carry with them the object-oriented procedures called **methods**. The XML standard aids in exchanging data between applications and users.

filter criteria
These limit the amount of information

returned to the data view in response to a query. You can apply a prefilter to the query to collect only certain data, or apply a postfilter to the view properties to show only certain data from the information collected.

georeferenced map
A special type of graphic that has built-in knowledge of latitude and longitude and can be zoomed into and out of quickly. The uses proprietary .IVL files generated with the map-rendering component. These files cannot be opened or saved in a graphic editor.

historical data management
The procedures applied to short-term binary history files that roll off historical data to either the or to delimited text files (the krarloff utility on UNIX or Windows; ddname KBDXTRA for the z/OS **Persistent Datastore**), and then delete entries in the short-term history files over 24 hours old, thereby making room for new entries.

hot standby
A redundant that, if the primary or hub should fail, assumes the responsibilities of the failed .

hub (1) A central host system that collects the status of situations running on your systems. (2) The that your site has selected to act as the focal point to which all s and remote s in this monitored network connect. A **remote** passes its collected data to the hub to be made available to clients, creating an enterprise-wide view.

IBM Tivoli Monitoring
A **client/server** implementation for monitoring enterprise-wide computer networks that comprises a , an application server known as the , one or more clients, and multiple s that collect and distribute data to the .

integral web server
A proprietary web server developed for that is installed and configured automatically with the . You enter the URL of the integral web server to start the client in **browser mode**.

Interoperable Object Reference
Connects clients to the . The IOR identifies a remote object, including such information as name, capabilities, and how to contact it. The URL may include an IOR because it goes through the web server; the uses it to tell the client which IOR to fetch. Once it does that, the extracts the host and port information and tells the client where to route the request.

interval
The number of seconds that have elapsed between one sample and the next. A sample is the data that the collects for the server.

IOR See Interoperable Object Reference.

Java Database Connectivity
A standard API that enables application developers to access and update relational databases (RDBMSes) from within Java programs. The JDBC standard is based on the X/Open SQL Call Level Interface (CLI) and complies with the SQL-92 Entry Level standard; it provides a DBMS-independent interface that enables SQL-compliant database access for Java programmers.

JDBC See Java Database Connectivity.

location broker
The component of the that manages connections for the hub , enabling it to find all other Tivoli Monitoring components, including remote s, the , and s.

managed object
An icon created in the from a managed object template that represents resources you monitor using situations. Managed objects are converted to items in the Navigator Logical view.

managed system
A particular operating system, subsystem, or application in your enterprise where a is installed and running. This is any system that is monitoring.

managed system list
A named list of **managed systems** of the same type that is maintained by the . Example: a list of Linux managed systems for a geographic region named LINUX_LONDON. You can see and select a managed system list when you distribute a situation or policy, edit a

query specification, or assign managed systems to Navigator items in custom Navigator views.

If a managed system list is updated (usually when a constituent managed system is added or deleted), then all the situations and policies that use that list are redistributed to all managed systems in the list.

middleware
Software that enables the exchange of information between components in a distributed computing environment. The middleware is the data-exchange and communications channel that allows programs to cooperate with each other without having to know details about how they are implemented or where they are deployed. Middleware typically provides a range of related facilities such as persistence, auditing, and the ability to build a transactional unit of work. IBM's CICS and WebSphere MQ are examples of middleware.

monitor interval
A specified time, scalable to seconds, minutes, hours, or days, for how often the checks to see if a situation has become true. The minimum monitor interval is 30 seconds; the default is 15 minutes.

NAT See Network Address Translation.

Navigator
The upper-left pane of the window. The Navigator Physical view shows your network enterprise as a physical hierarchy of systems grouped by platform. You can also create other views to create logical hierarchies grouped as you specify, such as by department or function.

Network Address Translation
A scheme used by local-area networks (LANs) to establish an internal and external set of IP addresses. Internal IP addresses are kept private and must be translated to and from the external addresses for outbound and inbound communications. NAT is often used in firewall configurations.

non-agent bundles
These custom bundles let you remotely deploy components that need not connect to a Tivoli Enterprise Monitoring Server, such as those that support other Tivoli products like Netcool/OMNIbus.

object An instance of a **class**, which comprises an implementation and an interface. An object reflects its original, holding data and methods and responds to requests for services. **CORBA** defines an object as a combination of state and a set of methods characterized by the behavior of relevant requests.

ODBC
See Open Database Connectivity.

Open Database Connectivity
A standard for accessing different database systems using procedural, non-object-based languages such as C. The Query editor enables you to write custom SQL queries for creating views that retrieve data from ODBC-compliant databases.

platform
The operating system upon which the managed system is running, such as z/OS or Linux. The Navigator's Physical mapping places the platform level under the Enterprise level.

policy A set of automated system processes that can perform actions, schedule work for users, or automate manual tasks, frequently in response to **events**. Policies are the automation tool; they comprise a series of automated steps, called **activities**, whose order of execution you control.

In most cases, a policy links a Take Action command to a situation that has turned true. Once begun, the policy's workflow progresses until all activities have been completed or until the user manually stops the policy. You can create both policies that fully automate workflow strategies and those that require user intervention. As with **situations**, policies are distributed to the managed systems you want to monitor and to which you are sending commands.

private situation
A situation that is defined in an XML "privateconfiguration" file for the local Tivoli Enterprise Monitoring Agent or Tivoli System Monitor Agent and does not interact with a Tivoli Enterprise

Monitoring Server. Events are viewed through the Agent Service Interface or can be sent as EIF events or SNMP alerts to a receiver. See also situation.

product code
The three-letter code used by to identify the product component. For example, the product code for for WebSphere Application Server is KWE.

Properties editor
A multi-tabbed window for specifying the properties of the individual views that make up a workspace, as well as the general workspace properties.

pure event
A pure event is one that occurs automatically, such as when a paper-out condition occurs on the printer or when a new log entry is written. Situations written to notify you of pure events remain true until they are manually closed or automatically closed by an UNTIL clause.

query A particular view of specified attributes of selected instances of a set of managed-object classes, arranged to satisfy an end-user request. Queries are written using **SQL**.

Remote Procedure Call
A protocol based on the Open Software Foundation's Distributed Computing Environment (DCE) that allows one program to request services from a program running on another computer in a network. RPC uses the **client/server** model: the requesting program is the client, and the responding program is the server. As with a local procedure call (also known as a **function call** or a **subroutine call**, an RPC is a synchronous operation: the requesting program is suspended until the remote procedure returns its results.

remote
A remote collects monitoring data from a subset of your site's s and passes its collected data to the hub to be made available to one or more clients via the , thereby creating an enterprise-wide view.

RPC See Remote Procedure Call.

sample
The data that the collects for the instance. The **interval** is the time between data samplings.

sampled event
Sampled events happen when a situation becomes true. Situations sample data at regular intervals. When the situation becomes true, it opens an event, which gets closed automatically when the situation goes back to false (or when you close it manually).

server An application that satisfies data and service requests from **clients**.

Simple Network Management Protocol
A TCP/IP transport protocol for exchanging network management data and controlling the monitoring of network nodes in a TCP/IP environment. The SNMP software protocol facilitates communications between different types of networks. uses SNMP messaging to discover the devices on your network and their availability.

Simple Object Access Protocol
The Simple Object Access Protocol is an **XML**-based interface that vendors use to bridge **remote procedure calls** between competing systems. SOAP makes it unnecessary for sites to choose between **CORBA**/Java/EJB and Microsoft's COM+.

Because XML and SOAP are platform- and language-neutral, users can mix operating systems, programming languages, and object architectures yet maintain business-component interoperability across platforms: using SOAP, applications can converse with each other and exchange data over the Internet, regardless of the **platforms** on which they run.

situation
One or more monitored conditions running on a **managed system**. The set of conditions that, when met, creates an **event**. A situation comprises an attribute, function (such as  Value of expression or

 Percent change in value), a relational operator such as greater than or equal to, and a value to be compared. It can be read as “If – system condition – compared to – value – is true”. An example of a

situation is: If – value of CPU usage > 90% – TRUE. The expression “CPU usage > 90 is the **situation condition**. See also

SNMP
See Simple Network Management Protocol.

SOAP See Simple Object Access Protocol.

sockets
Refers to the sockets method of passing data back and forth between a networked **client** and **server** or between program layers on the same computer.

sound The **WAV file** that plays whenever a situation becomes true for the current Navigator item. Sound is assigned to the Navigator item for a situation in the same way a **state** is assigned.

SQL See Structured Query Language.

state The severity of the situation event: critical, warning, or informational. Indicated by a colored event indicator, state is set in the Situation editor and can be different for different Navigator items.

status The true or false condition of a situation.

Structured Query Language
A programming language for extracting information from and updating information within a relational database. The **Query editor** enables you to write SQL queries to **ODBC** datasources for retrieval and display in table and chart views.

subnetwork
A configuration wherein a single IP network address is split up so it can be used on several interconnected local networks. Subnetworking is a local configuration; outside it appears as a single IP network.

summarization and pruning agent
One of the Tivoli Monitoring base agents, the summarization and pruning agent is used to keep the data warehouse from growing too large by summarizing and pruning your stored historical data at intervals that you set. For every attribute group that has data collection configured, you specify how often to aggregate (summarize) data in the Tivoli Data Warehouse and the length of time to delete (prune) data from the warehouse.

symbol
Represents a variable that can be added to header or footer text for data views, expert-advice text, or query specification. The detailed attribute name is enclosed in dollar signs, such as **$ORIGINNODE$**, and resolves to the attribute's value. For queries, `== $NODE$` specifies the managed systems from which to retrieve data. For queries to be used in link target workspaces, you can create symbols for attributes using the *$symbolname$* format.

System Monitor Agent
These agents were introduced with V6.2.2 for nodes that run the desktop operating systems (Windows, Linux, UNIX). System Monitor Agents operate only autonomously (that is, they run without a connection to a) and pass SNMP trap data about operating system performance to an SNMP Event Collector such as IBM Tivoli Netcool/OMNIbus.

No other agents or other components should be installed on the same node as a system monitor agent. The only exception to this rule is agents created with the Agent Builder tool (V6.2.2 or later).

Take Action
A dialog from which you can enter a command or choose from a list of predefined commands. It also lists systems on which to effect the command, which is usually a response to an **event**.

Take Action command
A Take Action command allows you to send commands to your **managed systems**, either automatically, in response to a **situation** that has fired (that is, turned true), or manually, as the operator requires. With Take Action commands, you can enter a command or select one of the commands predefined by your product and run it on any system in your managed network. Thus you can issue Take Action commands either against the managed system where the situation fired or a different managed system in your network.

TCP/IP
See Transmission Control Protocol/Internet Protocol.

TDW See Tivoli Data Warehouse.

telnet A terminal emulation program used on TCP/IP networks. You can start a telnet session with another system and enter commands that execute on that system. A valid user ID and password for that remote system are required.

threshold
A level set in the system at which a message is sent or an error-handling program is called. For example, in a user auxiliary storage pool, the user can set the threshold level in the system values, and the system notifies the system operator when that level is reached.

Tivoli Data Warehouse
This member of the product family stores Tivoli Monitoring agents' monitoring data in separate relational database tables so you can analyze historical trends using that enterprise-wide data. Reports generated from data provide information about the availability and performance of your monitoring environment over different periods of time.

Tivoli Enterprise Monitoring Server
The host data-management component for . It receives and stores data from either the agents or other s.

Tivoli Enterprise Portal Server
The server you log onto and connect to via the client. The connects to the hub ; it enables retrieval, manipulation, and analysis of data from your enterprise's s.

Tivoli Enterprise Web Services
An open standards-based interface to the that uses SOAP requests. Using SOAP, any can be dynamically queried, which means that its performance and availability data can be processed by external applications not a part of .

Tivoli Management Services
An integrated, layered architecture consisting of data-access, communication, and presentation components that enable cross-platform operation and integration of enterprise-wide data for systems-management applications. The software foundation that supports the development and operations of the , the and , and their s.

Transmission Control Protocol/Internet Protocol
An open, portable communications protocol that is the software basis for the Internet.

value of expression
A function in a situation condition, query specification, or data view filter or threshold that uses the raw value of an attribute. A value can be a number, text string, attribute, or modified attribute. Use this function with any operator.

view A window pane, or frame, in a workspace. It may contain data from an agent in a chart or table, or it may contain a terminal session or browser, for example. A view can be split into two separate, autonomous views.

warehouse proxy agent
One of the Tivoli Monitoring base agents, the warehouse proxy agent passes data from either a monitoring agent or the Tivoli Enterprise Monitoring Server to the Tivoli Data Warehouse. This multi-threaded server process can handle concurrent requests from multiple monitoring agents to roll off data from their short-term history files to the data warehouse.

WAV file
Waveform audio format for storing sound in files, developed jointly by Microsoft and IBM.

wildcard
An * (asterisk) used to represent any characters that may follow or precede those entered, such as Sys* to find System and SysTray. Used in formulas with the VALUE function or MISSING function (in the Missing Task List). Used also with the SCAN function, but at the beginning of the text as in *Z to find markZ and typeZ.

workspace
The viewing area of the window, excluding the Navigator. Each workspace comprises one or more views. Every Navigator item has its own default workspace and may have multiple workspaces.

workspace administration mode
A global parameter set in the Administer Users editor but which is available only

for user IDs with administrator authority. When enabled for a user ID, customization of workspaces, links, and terminal-session scripts automatically becomes available to all users connected to the same .

XML See Extensible Markup Language.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Readers' Comments — We'd Like to Hear from You

IBM Tivoli Monitoring
Administrator's Guide
Version 6.2.2 Fix Pack 2 (Revised June 2010)

Publication No. SC32-9408-03

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Send your comments to the address on the reverse side of this form.

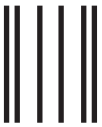
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
SC32-9408-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Fold and Tape | **Please do not staple** | Fold and Tape

SC32-9408-03

Cut or Fold
Along Line

Printed in USA

SC32-9408-03