IBM Tivoli Monitoring
Version 6.3

Command Reference

SC22-5448-00

IBM Tivoli Monitoring
Version 6.3

Command Reference

SC22-5448-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 349.

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

Contents

Tables

Chapter 1. Using the commands

IBM Tivoli Monitoring supports a number of command line interface (CLI) commands.

These commands can be used to set up the environment, to deploy and configure framework components and monitoring agents, and to run and manage the environment. Many of these commands duplicate functions that can be performed using a graphical interface such as Manage Tivoli Enterprise Monitoring Services or the Tivoli Enterprise Portal. In general, the CLI commands are used for automation of these functions.

Tivoli command syntax

Use special characters to define the Tivoli® command syntax.

[ ] Identifies elements that are optional. Required elements do not have brackets around them.

... Indicates that you can specify multiple values for the previous element. Separate multiple values by a space, unless otherwise directed by command information.

If the ellipsis for an element follows a closing bracket, use the syntax within the brackets to specify multiple values. For example, to specify two administrators for the option [**–a** *admin*]..., use **–a admin1 –a admin2**.

If the ellipsis for an element is within the brackets, use the syntax of the last element to specify multiple values. For example, to specify two hosts for the option [**–h** *host*...], use **–h host1 host2**.

| Indicates mutually exclusive information. You can use the element on either the left or right of the vertical bar.

{ } Delimits a set of mutually exclusive elements when a command requires one of them. Brackets ([ ]) are around elements that are optional.

The following example illustrates the typeface conventions used in Tivoli command syntax:

itmcmd agent [-l] [-h *install_dir*] [-o *instance*] [-p *option*] [-c] [-s] start|stop {*pc*|all}

The **start|stop** and {***pc*|all**} options are the only required options for the **itmcmd agent** command. The brackets around the **-l**, **-h**, **-o**, **-p**, **-c**, and **-s** options indicate that they are optional. The braces around {***pc*|all**} indicate that you must either specify a product code (pc) or choose to start or stop all components.

Note: When using the **"** character while executing commands, you must use the escape character (\), which is a general command-line restriction, for example:

```
[root@vger ksh]# tacmd createsit -s abc\"123 -b Linux_Process_High_Cpu
```

Running commands

You must run commands one at a time. You cannot run multiple commands at the same time on one system. This limitation includes opening two windows on the same system and running commands in parallel.

Note: Only IBM Tivoli Monitoring v6.3.0 commands are supported to connect to a IBM Tivoli Monitoring v6.3.0 hub monitoring server.

Typeface conventions

This publication employs a variety of typeface conventions.

Bold

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

Italic

- Citations (examples: titles of publications, diskettes, and CDs
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide.

`Monospace`

- Examples and code examples
- File names, directory names, and path names
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This publication employs the UNIX convention for specifying environment variables and for directory notation.

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

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

New in this release

For IBM® Tivoli Monitoring v6.3, a variety of commands have been added.

- New tivcmd commands associated with granular security
  - tivcmd addtorole
  - tivcmd copyrole
  - tivcmd createrole
  - tivcmd exclude
  - tivcmd grant
  - tivcmd help
  - tivcmd listdomains
  - tivcmd listobjecttypes
  - tivcmd listresourcetypes
  - tivcmd listroles
  - tivcmd login
  - tivcmd logout
  - tivcmd removefromrole
  - tivcmd revoke
- New commands:
  - tacmd addSdaInstallOptions
  - tacmd deleteSdaInstallOptions
  - tacmd deleteSdaSuspend
  - tacmd editSdaInstallOptions
  - tacmd listSdaInstallOptions
  - tacmd listSdaStatus
  - tacmd listtrace
  - tacmd resumeSda
  - tacmd settrace
  - tacmd suspendSda
- New command options:

  tacmd clearDeployStatus -i | --inprogress
  Specifies the option to clear an in-progress transaction.

  tacmd clearDeployStatus -y | --yes
  Specifies the performing of an action without requiring confirmation.
- The **tacmd listappinstallrecs** command no longer displays SDA configuration settings or monitoring server SDA status (that is, 5530 records).
- The **tacmd tepsLogin** command supports HTTPS by default with port 15200.
- For the following tacmd commands, the list of valid characters has been expanded to include strings of letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ):
  - tacmd createsystemlist
  - tacmd editsystemlist
  - tacmd cleanMS
  - tacmd executeAction
  - tacmd executecommand
  - tacmd putfile
  - tacmd getfile
  - tacmd listSit
  - tacmd deleteOverride
  - tacmd setOverride
  - tacmd acceptbaseline
  - tacmd suggestBaseline

– tacmd listSystems

Chapter 2. tacmd CLI commands

You can run the tacmd CLI commands to manage your monitoring environment.

Table 1. tacmd CLI commands

Command	Description
"tacmd acceptBaseline" on page 13	Sets a situation override based on the baseline (situation override) values calculated by using one of several statistical functions for a situation attribute based on historical data from the Tivoli Data Warehouse.
"tacmd addBundles" on page 17	Add one or more deployment bundles to the local agent deployment depot.
"tacmd addCalendarEntry" on page 19	Create a calendar entry on the Tivoli Enterprise Monitoring Server.
"tacmd addgroupmember" on page 20	Add a group member to the specified group.
"tacmd addSdaInstallOptions" on page 23	Add a version to the Self-Describing Agent (SDA) install option record for a product type.
"tacmd addSystem" on page 24	Deploy a monitoring agent to a computer in your IBM Tivoli Monitoring environment.
"tacmd bulkExportPcy" on page 28	Export all the available policies from the Tivoli Enterprise Monitoring Server.
"tacmd bulkExportSit" on page 30	Export all the available situations from the Tivoli Enterprise Monitoring Server.
"tacmd bulkImportPcy" on page 31	Import all the available policy objects to the Tivoli Enterprise Monitoring Server from BULK_OBJECT_PATH.
"tacmd bulkImportSit" on page 32	Import all the available objects to the Tivoli Enterprise Monitoring Server from BULK_OBJECT_PATH.
"tacmd checkprereq" on page 34	Check for prerequisites required for deploying an agent to a managed system.
"tacmd cleanMS" on page 36	Clear all the offline entries present in the Tivoli Enterprise Monitoring Server.
"tacmd clearAppSeedState" on page 37	Clear the value of the SEEDSTATE column of an installation record that has status IC and SEEDSTATE value I (Incomplete) or E (Error).
"tacmd clearDeployStatus" on page 38	Remove entries from the table that stores the status of the asynchronous agent deployment operations.
"tacmd configurePortalServer" on page 40	Configure a user-defined portal server data source.
"tacmd configureSystem" on page 42	Edit the configuration options of an existing agent.
"tacmd createAction" on page 46	Create a new Take Action.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd createEventDest" on page 47	Create a new event destination definition on the server.
"tacmd creategroup" on page 49	Create a new group on the server.
"tacmd createNode" on page 50	Deploy an OS agent to a remote computer.
"tacmd createSit" on page 54	Create a new situation.
"tacmd createSitAssociation" on page 58	Creates one or more situation associations for a Tivoli Enterprise Portal navigator item.
"tacmd createSysAssignment" on page 60	Assigns one or more managed systems or managed system lists to a Tivoli Enterprise Portal navigator item.
"tacmd createsystemlist" on page 62	Create a new managed system group.
"tacmd createUser" on page 63	Create a new user in the Tivoli Enterprise Portal.
"tacmd createUserGroup" on page 65	Create a new group in the Tivoli Enterprise Portal.
"tacmd deleteAction" on page 66	Delete a Take Action.
"tacmd deleteappinstallrecs" on page 67	Delete application support install records on the server.
"tacmd deleteCalendarEntry" on page 68	Delete a calendar entry on the Tivoli Enterprise Monitoring Server.
"tacmd deleteEventDest" on page 69	Delete an event destination server definition from the server.
"tacmd deletegroup" on page 69	Delete a specified group member from the Tivoli Enterprise Monitoring Server.
"tacmd deletegroupmember" on page 70	Delete a specified group member from the Tivoli Enterprise Monitoring Server.
"tacmd deleteOverride" on page 71	Delete the situation overrides defined for a specified situation on a managed system or list of managed systems.
"tacmd deleteSdaInstallOptions" on page 73	Remove a version from a Self-Describing Agent (SDA) install option record for a product type.
"tacmd deleteSdaOptions" on page 74	Delete Self-Describing Agent (SDA) option configuration entries.
"tacmd deleteSdaSuspend" on page 75	Delete the Self-Describing Agent (SDA) Suspend record from the database. **CAUTION:** **Do not use the deleteSdaSuspend command unless directed by IBM Software Support.**
"tacmd deleteSit" on page 76	Delete a situation from your environment.
"tacmd deleteSitAssociation" on page 77	Dissociates one or more situations from a Tivoli Enterprise Portal navigator item.
"tacmd deleteSysAssignment" on page 78	Deletes one or more managed system assignments from a Tivoli Enterprise Portal navigator item.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd deletesystemlist" on page 80	Delete a managed system group.
"tacmd deleteUser" on page 81	Delete the existing user from Tivoli Enterprise Portal Server.
"tacmd deleteUserGroup" on page 82	Delete the existing group from Tivoli Enterprise Portal Server.
"tacmd deleteWorkspace" on page 83	Delete a global or user-customized Tivoli Enterprise Portal workspace from the Tivoli Enterprise Portal Server.
"tacmd describeSystemType" on page 84	Display the configuration options that are required for an agent type.
"tacmd editAction" on page 86	Edit a Take Action.
"tacmd editCalendarEntry" on page 87	Edit a calendar entry on the Tivoli Enterprise Monitoring Server.
"tacmd editEventDest" on page 88	Modify an existing event destination server definition on the server.
"tacmd editGroup" on page 90	Edit a group definition.
"tacmd editgroupmember" on page 91	Edit a groupmember definition.
"tacmd editSdaInstallOptions" on page 93	Edit the version of a Self-Describing Agent (SDA) install option record for a product type.
"tacmd editSdaOptions" on page 95	Edit a situation definition that exists on a server or that was exported to a local system.
"tacmd editSit" on page 98	Edit a situation.
"tacmd editsystemlist" on page 100	Add or delete managed systems to or from an existing managed system group on the server.
"tacmd editUser" on page 101	Edit a user definition in the Tivoli Enterprise Portal.
"tacmd editUserGroup" on page 103	Edit a group definition on the Tivoli Enterprise Portal Server.
"tacmd executeAction" on page 105	Execute the system command provided in the given Take Action command.
"tacmd executecommand" on page 109	Execute the system command provided in the given Take Action command.
"tacmd exportBundles" on page 114	Export remote deployment bundles from installation media or from a remote deployment depot.
"tacmd exportCalendarEntries" on page 116	Export all the calendar entries available in the Tivoli Enterprise Portal Server to the specified XML file.
"tacmd exportNavigator" on page 117	Export a Tivoli Enterprise Portal logical navigator and all workspaces, queries, and situation associations referenced within the logical navigator from the Tivoli Enterprise Portal Server to an XML file.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd exportQueries" on page 118	Export one or more Tivoli Enterprise Portal queries from the Tivoli Enterprise Portal Server to an XML file.
"tacmd exportSitAssociations" on page 120	Exports all situation associations for a Tivoli Enterprise Portal navigator, or optionally, a particular navigator item within the navigator, to an XML file.
"tacmd exportSysAssignments" on page 121	Exports all managed system assignments for a Tivoli Enterprise Portal navigator, or optionally, a particular navigator item within the navigator, to an XML file.
"tacmd exportWorkspaces" on page 123	Export one or more portal server workspaces to a file.
"tacmd getDeployStatus" on page 126	Display the status of the asynchronous agent deployment operations.
"tacmd getfile" on page 128	Transfer a file from a remote managed system to a local destination.
"tacmd help" on page 131	Display the name of all the available CLI commands, along with a short description of each command.
"tacmd histconfiguregroups" on page 140	Display the historical configuration information of the specified attribute group.
"tacmd histcreatecollection" on page 142	Create the given collection using specified inputs.
"tacmd histdeletecollection" on page 145	Create the given collection using specified inputs.
"tacmd histeditcollection" on page 146	Edit the given collection using specified inputs.
"tacmd histlistattributegroups" on page 147	List all of the attribute groups for the specified product name for historical data collection and configuration feature.
"tacmd histlistcollections" on page 148	List all the collections that are started for a managed system or managed system group, or that are defined for an attribute group
"tacmd histlistproduct" on page 150	List all of the products available for the historical data collection and configuration feature.
"tacmd histstartcollection" on page 151	Unconfigure the given attribute groups using provided inputs for historical data collection.
"tacmd histstopcollection" on page 153	Unconfigure the given attribute groups using provided inputs for historical data collection.
"tacmd histunconfiguregroups" on page 154	Unconfigure the given attribute groups using provided inputs for historical data collection.
"tacmd histviewattributegroup" on page 156	Display the historical configuration information of the specified attribute group.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd histviewcollection" on page 157	Display the configuration information of a specified collection
"tacmd importCalendarEntries" on page 158	Import all the calendar entries available in specified XML file to the Tivoli Enterprise Portal Server.
"tacmd importNavigator" on page 159	Import a Tivoli Enterprise Portal logical navigator view, workspaces, queries, and situation associations from an XML file to the Tivoli Enterprise Portal Server.
"tacmd importQueries" on page 161	Import Tivoli Enterprise Portal queries from an XML file to the Tivoli Enterprise Portal Server.
"tacmd importSitAssociations" on page 162	Imports all situation associations from an XML file to the Tivoli Enterprise Portal Server.
"tacmd importSysAssignments" on page 164	Imports all managed system assignments from an XML file to the Tivoli Enterprise Portal Server.
"tacmd importWorkspaces" on page 166	Import the workspaces contained in a file into the portal server.
"tacmd listAction" on page 168	Display the list of the Take Action commands in the server.
"tacmd listappinstallrecs" on page 168	List application install records.
"tacmd listBundles" on page 171	Display the details of one or more deployment bundles that are available for deployment to the local deployment depot.
"tacmd listCalendarEntries" on page 173	List calendar entries on the Tivoli Enterprise Monitoring Server.
"tacmd listEventDest" on page 173	Display the server ID, name, and type for every event destination definition on the server.
"tacmd listGroups" on page 174	Display a list of known groups.
"tacmd listNavigators" on page 174	Display the server ID, name, and type for every event destination definition on the server.
"tacmd listOverrideableSits" on page 175	Display a list of override-eligible situations for a given application.
"tacmd listOverrides" on page 176	Display the situation overrides defined for a specified situation on a managed system or list of managed systems.
"tacmd listQueries" on page 177	Import Tivoli Enterprise Portal queries from an XML file to the Tivoli Enterprise Portal Server.
"tacmd listSdaInstallOptions" on page 178	List the Self-Describing Agent (SDA) install options records.
"tacmd listSdaOptions" on page 180	Lists the Self-Describing Agent (SDA) options.
"tacmd listSdaStatus" on page 180	Lists the Self-Describing Agent (SDA) Enablement status for a monitoring server.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd listSit" on page 184	List the situations on the hub monitoring server.
"tacmd listSitAssociations" on page 186	Displays a list of all situations associated with or eligible for association with a Tivoli Enterprise Portal navigator item. Optionally, the command can display a list of situations that are eligible for association with the specified navigator item.
"tacmd listSitAttributes" on page 187	List attribute names that are eligible for use with dynamic thresholding (override) commands for a given situation.
"tacmd listsystemlist" on page 190	List the available managed system groups.
"tacmd listSysAssignments" on page 188	Displays a list of managed systems or managed system lists that are assigned to a Tivoli Enterprise Portal navigator item.
"tacmd listSystems" on page 191	Display a list of agents, optionally filtering for only those on a given managed system or one or more product codes, or both.
"tacmd listtrace" on page 192	Display the RAS1 logging level on an ITM endpoint.
"tacmd listUsers" on page 193	List all the available users or users belonging to a particular group.
"tacmd listUserGroups" on page 194	List all the available groups.
"tacmd listworkspaces" on page 196	List all of the portal workspaces on the server.
"tacmd login" on page 198	Log on to a monitoring server and create a security token used by subsequent commands.
"tacmd logout" on page 199	Log out of the monitoring server and disable the security token created by the tacmd login command.
"tacmd managesit" on page 199	Start or stop situations in the Tivoli Enterprise Monitoring Server.
"tacmd pdcollect" on page 201	Execute the pdcollect script in the specified host computer and fetch the resultant jar file to the local computer.
"tacmd putfile" on page 203	Transfer a file from a local source to a remote managed system.
"tacmd refreshCatalog" on page 206	Update the catalog file.
"tacmd refreshTECinfo" on page 207	Trigger the Event Forwarder to reprocess any updated event destinations, EIF configurations, and custom event mapping files without recycling the HUB Tivoli Enterprise Monitoring Server.
"tacmd removeBundles" on page 208	Remove one or more deployment bundles from the local deployment depot.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd removeSystem" on page 209	Remove one or more instances of an agent or uninstall an agent from a managed system. Using the bulk deployment option, the command removes all agents in a deployment and bundle group combination.
"tacmd restartAgent" on page 212	Start or restart the given agents or the agents for the given managed systems.
"tacmd restartFailedDeploy" on page 215	Restart all the failed entries in the status table or filter the table entries to restart from the status table entries to a specific deployment operation.
"tacmd resumeSda" on page 216	Resume the Self-Describing Agent (SDA) installation.
"tacmd setAgentConnection" on page 217	Edit connection properties and environment variables of agents running on the target node.
"tacmd setOverride" on page 221	Define a situation override for a specified situation on a managed system or list of managed systems.
"tacmd settrace" on page 224	Modify the RAS1 logging level on a remote managed system.
"tacmd startAgent" on page 226	Start the given agent or agents for the given managed systems.
"tacmd stopAgent" on page 229	Stop the given agent or agents for the given managed systems.
"tacmd suggestBaseline" on page 232	Calculate a baseline (situation override) value using one of several statistical functions for a situation attribute based on historical data from the Tivoli Data Warehouse.
"tacmd suspendSda" on page 236	Suspend the Self-Describing Agent (SDA) installation.
"tacmd tepsLogin" on page 237	Log in to Tivoli Enterprise Portal Server.
"tacmd tepsLogout" on page 238	Log off Tivoli Enterprise Portal Server.
"tacmd updateAgent" on page 239	Install an agent update on a specified managed system.
"tacmd viewAction" on page 241	Display the details of a Take Action.
"tacmd viewAgent" on page 242	Display the details of the given agent or the agent for a given managed system.
"tacmd viewCalendarEntry" on page 243	View information about a calendar entry on the Tivoli Enterprise Monitoring Server.
"tacmd viewDepot" on page 243	Display the types of agents you can install from the deployment depot on the server which you are logged on to or the specified remote server.
"tacmd viewEventDest" on page 244	Display all properties for the specified event destination definition on the server.
"tacmd viewgroup" on page 245	Display details of the specified group.

Table 1. tacmd CLI commands (continued)

Command	Description
"tacmd viewgroupmember" on page 246	Display the details of the specified group member.
"tacmd viewNode" on page 246	Display the details of a node, including the installed components.
"tacmd viewSit" on page 247	Display the definition of a situation in your monitored environment.
"tacmd viewsystemlist" on page 248	List the configuration of a managed system group to be displayed or saved in an export file.
"tacmd viewUser" on page 249	Display the details of a specified user.
"tacmd viewUserGroup" on page 251	Display details of the specified group.
"kincinfo" on page 265	On Windows systems, view information for your monitoring server, including inventory of installed IBM Tivoli products, configuration settings, installed CD versions, and a list of running IBM Tivoli processes.
"KinCli.exe command" on page 271	Generate response files.

Input files for tacmd commands

Description

You have the option to provide all of the command-line options by using an input file. The following syntax is available for all tacmds commands:

```
tacmd subcommand inputfile
```

where:

subcommand
Specifies the command name, such as addSystem or configureSystem.

input file
Specifies a relative or fully qualified path to the text file containing the desired command-line switches for the command.

See the following example command-line input for the addSystem command:

```
tacmd addSystem -n tivm163:LZ -t
 r2 -p INSTANCE=snmp1
KR2_DP_SELECT.DATA_PROVIDER=SNMP
KQZ_SNMP.SNMP_PORT=161
KQZ_SNMP.SNMP_VERSION=snmpV1
KQZ_SNMPV1.SNMP_COMMUNITY=public
WIN:tivg19.SNMP_HOST=tivg19
```

can be entered through an input file as follows:

```
tacmd addSystem tivm163_lz_r2.txt
```

where `tivm163_lz_r2.txt` contains the following lines of code:

```
--node
tivm163:LZ
--type
r2
--property
```

```
INSTANCE=snmp1
KR2_DP_SELECT.DATA_PROVIDER=SNMP
KQZ_SNMP.SNMP_PORT=161
KQZ_SNMP.SNMP_VERSION=snmpV1
KQZ_SNMPV1.SNMP_COMMUNITY=public
WIN:tivg19.SNMP_HOST=tivg19
```

Remote Deployment commands

Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes the following commands:

- tacmd addBundles
- tacmd addSystem
- tacmd checkprereq
- tacmd clearDeployStatus
- tacmd configureSystem
- tacmd createNode
- tacmd exportBundles
- tacmd getDeployStatus
- tacmd listBundles
- tacmd removeBundles
- tacmd removeSystem
- tacmd restartAgent
- tacmd restartFailedDeploy
- tacmd setAgentConnection
- tacmd startAgent
- tacmd stopAgent
- tacmd updateAgent
- tacmd viewAgent
- tacmd viewDepot

tacmd acceptBaseline

Description

Use the **tacmd acceptBaseline** command to set the overrides for a situation based on the baseline (situation override) values calculated by using one of several statistical functions for a situation attribute based on historical data from the Tivoli Data Warehouse. This command yields identical calculations to the **suggestBaseline** command; however, you can use the **acceptBaseline** command to calculate and set baseline values with a single command invocation.

Note:

- The managed system specified with the -m|--system option must be online to run the command.
- For a managed system group, the overrides are only applied to members of the list that are override-eligible. Overrides are not distributed to ineligible managed systems.
- If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **acceptBaseline**

command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

- The total number of characters used in all the expression overrides defined for a situation should not exceed 4000 bytes. The actual size requirement for a single override varies depending on the names and values of the key columns and the override expression. In one case the limit might be 25 or, in a simpler case, it might be higher. The symptom of exceeding the 4000-byte limit is that the overrides do not work and the monitoring server trace log shows an "exceeds limit 4000" override error.

CLI syntax

tacmd acceptBaseline

{-s | --situation} *SITNAME*
{-m | --system} *SYSTEM | SYSTEM_LIST*
{-p | --predicate} *PREDICATE*
{-f | --function} *STATISTICAL_FUNCTION*
{-d | --startdata} *START_TIMESTAMP*
{-e | --enddata} *END_TIMESTAMP*
[{-u | --userid} *TEPS_USERID***]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-t | --inlinecal} *INLINE_CAL_ENTRY...***]**
[{-c | --calendarentry} *CALENDAR_ENTRY...***]**
[{-k | --key} *KEY_CONDITION* **...]**
[{-h | --tepshostname} *TEPS_HOSTNAME***]**

where:

-s | --situation
Specifies the situation to calculate the baseline value and set the overrides for. If you include either the & character or the < character in the situation name, you must include quotation marks around the name, for example, "abc&def" or "abc<def".

-m | --system
The name of the managed system or managed system group to calculate the baseline value and set the overrides for. Historical data results from the warehouse used for statistical calculations is restricted to values recorded for the managed system or managed systems specified. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-c | --calendarentry
Specifies the name of the calendar entry that defines the time period when the override is active. If one or more calendar entries are entered, historical data results from the warehouse will be filtered such that only the results that fall within each calendar entry are used to calculate the baseline value. A separate baseline value is calculated for each calendar entry.

-t | --inlinecal
Specifies the name of the Hourly Schedule entry that defines the time period when the override is active. The situation override is always active if you do not enter a Hourly Schedule interval. For the INLINE_CAL_ENTRY variable, use the [HHmm,HHmm] format, where HH is for hours in 00-23 notation and mm stands for minutes.

If one or more Hourly Schedule intervals are entered, historical data results from the warehouse are filtered such that only the results that fall within each Hourly Schedule are used to calculate the baseline value. A separate baseline value is calculated for each Hourly Schedule interval.

-d | --startdata

Specifies the starting time from which historical data from the warehouse will be used. Historical results queried from the warehouse are bounded by the start and end times. The start time value is specified as a timestamp in the format CYYMMDDHHmmSS or CYYMMDDHHmmSSsss, where:

- C=the century identifier (use 1 for year 2000 and later, 0 for earlier)
- YY=the year (for example, '08' for 2008)
- MM=the month (for example, '01' for January, or '12' for December)
- DD=the day of the month (for example, '06' for the 6th, or '31' for the 31st)
- HH=the hour of the day (for example, '08' for 8 A.M. or '17' for 5 P.M.)
- mm=the minute of the hour (for example, '00' for 'on the hour', '30', and so on)
- SS=the second (for example, '01' for one second past the minute)
- sss=milliseconds (for example, '500' for half a second). This value is optional.

-e | --enddata

Specifies the ending time from which historical data from the warehouse will be used. Historical results queried from the warehouse are bounded by the start and end times. The end time value is specified as a timestamp in the format CYYMMDDHHmmSS or CYYMMDDHHmmSSsss, where:

- C=the century identifier (use 1 for year 2000 and later, 0 for earlier)
- YY=the year (for example, '08' for 2008)
- MM=the month (for example, '01' for January, or '12' for December)
- DD=the day of the month (for example, '06' for the 6th, or '31' for the 31st)
- HH=the hour of the day (for example, '08' for 8 A.M. or '17' for 5 P.M.)
- mm=the minute of the hour (for example, '00' for 'on the hour', '30', and so on)
- SS=the second (for example, '01' for one second past the minute)
- sss=milliseconds (for example, '500' for half a second). This value is optional.

-f | --function

Specifies the statistical function that is used to calculate baseline values for the historical data queried from the warehouse. The statistical function is specified in the format:

```
{ mode | percent NUM | avg[{+|-}NUM] | min[{+|-}NUM] | max[{+|-}NUM] }
```

where:

```
min[{+|-}NUM] : minimum value +/- NUM percent of the value
max[{+|-}NUM] : maximum value +/- NUM percent of the value
avg[{+|-}NUM] : average value +/- NUM standard deviations
percent NUM   : value for the NUM percentile
mode          : most frequently observed value
```

When the mode calculation yields multiple results, the first result is used by the **acceptBaseline** command for the purposes of setting the override value.

-p | --predicate
Specifies the situation formula predicate for which the baseline value is calculated. The predicate must be enclosed in double quotation marks and entered in the format "ATTRIBUTE OPERATOR VALUE" with spaces between ATTRIBUTE, OPERATOR, and VALUE. The predicate OPERATOR must be one of the following: "EQ", "NE", "GT", "LT", "GE", or "LE". Historical data results from the warehouse used for statistical calculations are restricted to values recorded for the attribute specified by this predicate.

The attribute can be entered by using either the formula name or the display name for the attribute. Run the **tacmd listSitAttributes -s** ***SITNAME*** command to view the eligible attribute names for the situation.

-k | --key
Specifies the key condition or key conditions restricting the predicate attribute for which the baseline value will be calculated. Each key condition must be enclosed in double quotation marks and entered in the format "ATTRIBUTE OPERATOR VALUE" with spaces between ATTRIBUTE, OPERATOR, and VALUE. The key condition OPERATOR is restricted to the value "EQ". Historical data results from the warehouse used for statistical calculations are restricted to values recorded for the predicate attribute where all of the key conditions (where ATTRIBUTE equals VALUE) are satisfied.

The key condition attribute name can be entered by using either the formula name or the display name for the attribute. Run the **tacmd listSitAttributes -s** ***SITNAME*** command to view the eligible key condition attribute names for the situation.

-u | --userid
Specifies the existing User ID to log on to the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

CLI example

This example calculates and sets baseline values by using the average value plus 1 standard deviation for managed system Primary:LEVER:NT for the NT_NotesServerProcess situation, where the "Binary Path" attribute value is equal to "`C:\Notes\NotesServer\nserver.exe`" . Baseline values for the calendar entries WeekdayMorning and WeekdayAfternoon are calculated by using metrics stored in the Tivoli Data Warehouse between 5:59 a.m. July 28th, 2008, and 1 a.m. August 29th, 2008:

```
tacmd acceptbaseline --userid sysadmin --password ******** --system Primary:LEVER:NT
--situation NT_NotesServerProcess --predicate "% Processor Time GE 50"
--function AVG+1 --startdata 1080728055900 --enddata 1080829010000
--key "Binary Path EQ C:\Notes\NotesServer\nserver.exe"
--calendarentry WeekdayMorning WeekdayAfternoon
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd addBundles

Description

Use the **tacmd addBundles** command to add one or more deployment bundles to the local agent deployment depot. By default, this command also adds all deployment bundles that are prerequisites of the deployment bundle being added, if the prerequisite bundles do not already exist in the depot. This command can only be run from a Tivoli Enterprise Monitoring Server installation with a depot.

If you do not already have an agent depot, the bundles are added to the location defined by the DEPOTHOME environment variable in the KBBENV environment file.

This command must be run locally on a monitoring server containing a depot. If the current operating system user has the proper permissions, it is not necessary to run the login command before the **addbundles** command.

Note: The prerequisites for the bundle must be in the same image directory as the bundle being added. The addBundles command does not look for the prerequisites in the depot. The command attempts to copy the bundle as well as all of the bundle's prerequisites from the image directory to the depot. In most cases, the patches and their prerequisites reside in different image directories unless you manually move them all to the same directory. Each patch bundle must be added to the depot by using the "-n | --noPrereq" option.

CLI syntax

tacmd addBundles
{-i | --imagePath} *IMAGEPATH*
[{-t | --product | --products} *PRODUCT* **...]**
[{-p | --platform | --platforms} *PLATFORM* **...]**
[{-v | --version | --versions} *VERSION* **...]**
[{-n | --noPrereq | --noPrerequisites }]
[{-x | --version | --excludeOptional}]
[{-f | --force }]

where:

-i | --imagePath
Specifies a directory that contains the deployment bundles to be added to the depot.

-t | --product | --products
Specifies the product code or codes of the agents to add. This value corresponds to the value that will be displayed in the *Product Code* field that is displayed by the **viewDepot** or **listBundles** command.

-p | --platform | --platforms
Specifies the platform code or codes of the agents to add. This value corresponds to the value that will be displayed in the *Host Type* field that is displayed by the **viewDepot** or **listBundles** command.

-v | --version | --versions
Specifies the version or versions of the bundles to add. This value corresponds to the value that is displayed in the *Version* field that is displayed by the **viewDepot** or **listBundles** command.

-n | --noPrereq | --noPrerequisites
Indicates that prerequisite bundles are not automatically added.

-x | --version | --excludeOptional
Prevents adding optional prerequisite bundles specified in the descriptor file to the depot.

-f | --force
Installs any matching deployment bundles to the depot without prompting for confirmation first.

CLI example

The following example copies every agent bundle, including its prerequisites, into the agent depot on a UNIX from the installation media (cd image) located at `/mnt/cdrom/`:

```
tacmd addBundles -i /mnt/cdrom/unix
```

The following example copies all agent bundles for the Oracle agent into the agent depot on a UNIX computer from the installation media (cd image) located at `/mnt/cdrom/`:

```
tacmd addBundles -i /mnt/cdrom/unix -t or
```

The following example copies all agent bundles for the Oracle agent into the agent depot on a Windows computer from the installation media (cd image) located at `D:\WINDOWS\Deploy`:

```
tacmd addBundles -i D:\WINDOWS\Deploy -t or
```

The following example copies the agent bundle for the Oracle agent that runs on the AIX® version 5.1.3 operating system into the agent depot on a UNIX computer from the installation media (cd image) located at `/mnt/cdrom/`:

```
tacmd addBundles -i /mnt/cdrom/unix -t or -p aix513
```

Return values

See Table 8 on page 272.

Related commands

"tacmd listBundles" on page 171

"tacmd removeBundles" on page 208

"tacmd viewDepot" on page 243

Return to Table 1 on page 5.

tacmd addCalendarEntry

Description

Use the **tacmd addCalendarEntry** command to create the calendar entry on the Tivoli Enterprise Monitoring Server. The data for the calendar entries is given in CRON format. The format of the data must be given as a quintuple (5 places) value separated by a space within double quotation marks if specified by using the -c|--cron option. The format of the data must also follow the sequential order as [minute hour day_of_month month day_of_week]. The order must not be changed and you must specify an asterisk (*) in place of the values that you do not want to provide. Do not skip any values.

The valid values for the cron attributes are as follows:

- Minute - integer values between 0-59
- Hour - integer values between 0-23
- Day Of Month - integer values between 0-31
- Month - integer values between 1-12 or the first three letters of the month names. For example, JAN.
- Day Of Week - integer values between 0-7, or the first three letters of the day. For example, SUN. Both 0 and 7 indicate Sunday.
  - 0 = Sunday
  - 1 = Monday
  - 2 = Tuesday
  - 3 = Wednesday
  - 4 = Thursday
  - 5 = Friday
  - 6 = Saturday
  - 7 = Sunday

This example of the –c|--cron option specifies 4:30 AM on the 1st and 15th of the month and every Friday:

```
30 4 1,15 * 5
```

If the -c|--cron option is not used, the cron data can also be given by using one or more of the (-i|--min;-h|--hour;a|--daym|--dayofmonth;-m|--month;-w|--dayw|--dayofweek) options. The values that are not specified are considered asterisks (*), meaning every minute, hour, and so on.

You must log in by using the **login** command before running the **tacmd addCalendarEntry** command.

CLI syntax

tacmd addCalendarEntry
{-n|--name} *CALENDAR_ENTRY_NAME*
{-c|--cron} *CRON_SPEC*
[{-d|--description} *DESCRIPTION* **]**

tacmd addCalendarEntry
{-n|--name} *CALENDAR_ENTRY_NAME*
[{-i|--min} *MIN* **]**
[{-h|--hour} *HOUR* **]**

[{a | --daym | --dayOfMonth} *DAY_OF_MONTH* **]**
[{-m | --month} *MONTH* **]**
[{-w | --dayw | --dayOfWeek} *DAY_OF_WEEK* **]**
[{-d | --description} *DESCRIPTION* **]**

where:

-n | --name
Specifies the name of the calendar entry.

-c | --cron
Specifies the CRON specification of the calendar entry.

-d | --description
Specifies the description of the calendar entry.

-i | --min
Specifies the minute value of the CRON specification.

-h | --hour
Specifies the hour value of the CRON specification.

-a | --daym | --dayOfMonth
Specifies the day of the month value of the CRON specification.

-m | --month
Specifies the month value of the CRON specification.

-w | --dayw | --dayOfWeek
Specifies the day of the week value of the CRON specification.

CLI example

The following example adds the calendar entry Clean_Temp:

```
tacmd addCalendarEntry -n Clean_Temp -d "Clean Temporary directory on weekend"
-c "30 21 * * SUN"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd addgroupmember

Description

Use the **tacmd addgroupmember** command to add a group member to the specified group. You must log in by using the **login** command before running the **addgroupmember** command.

CLI syntax

Adding a child group:

tacmd addgroupmember
{-g | --group} *GROUPNAME*

{-m|--member} *MEMBERNAME*
{-t|--groupType} *DEPLOY|BUNDLE|SITUATION|COLLECTION*

Adding a bundle group member:

tacmd addgroupmember
{-g|--group} *GROUPNAME*
{-m|--member} *MEMBERNAME*
{-t|--groupType} *BUNDLE*
{-y|--productType} *PRODUCT_TYPE*
[-i|--platform] *PLATFORM*
[-v|--version] *VERSION*
[-p|--property|--properties *PROPERTY...***]**

Adding a deployment group member:

tacmd addgroupmember
{-g|--group} *GROUPNAME*
{-m|--member} *MEMBERNAME*
{-t|--groupType} *DEPLOY*
[-p|--property|--properties *PROPERTY...***]**

Adding a situation or collection group member:

tacmd addgroupmember
{-g|--group} *GROUPNAME*
{-m|--member} *MEMBERNAME*
{-t|--groupType} *SITUATION|COLLECTION*

Adding multiple members by using member file:

tacmd addgroupmember
{-g|--group} *GROUPNAME*
{-x|--file} *FILE*

where:

-g|--group
Specifies the name of the group that the new group member is added to.

-m|--member
Specifies the name of the group member.

-t|--groupType
Specifies the group type name. Acceptable type names are DEPLOY, BUNDLE, SITUATION, or COLLECTION. When adding a child group with -t COLLECTION, the collection setting member is distributed and started on the target system.

-p|--property|--properties
Specifies one or more *name=value* pairs that identify the configuration properties to be persisted for the group member. See "Configuration options and properties" on page 252 for information on these properties.

-y|--productType
Specifies the product type code. The product value corresponds to the value that is displayed in the Product Code field as a result of running the **viewDepot** or **listBundles** command.

-i | --platform
Specifies the platform code of the product. The platform value corresponds to the value that is displayed in the Host Type field as a result of running the **viewDepot** or **listBundles** command.

-v | --version
Specifies the version number of the deployment bundle being added as a bundle group member.

-x | --file
Specifies the Comma Separated Value (CSV) file containing one or more group members to add. When using the -x option for this command the CSV file should have the following syntax:

```
[MEMBERNAME],[DEPLOY|BUNDLE|SITUATION|COLLECTION],{[-y PRODUCTCODE] |
[-v PRODUCTVERSION] | [-i PRODUCTARCHITECTURE]},{[KEYWORD=VALUE]}
```

CLI example

This example adds the deployment member w099o002.tivlab.raleigh.ibm.com to the group NewWindowsDeployGroup:

```
tacmd addGroupMember -g NewWindowsDeployGroup -t DEPLOY
-m w099o002.tivlab.raleigh.ibm.com
-p KDYRXA.installDir=c:\\IBM\\ITM KDYRXA.RXAUSERNAME=Administrator
KDYRXA.RXAPASSWORD=****
```

Note: To add a member to a group, you need to create a group first. In the example above, create a deployment group NewWindowsDeployGroup by using the **createGroup** command, and then execute the **addGroupMember** command.

This example adds the situation group member NT_Disk_Space_Low to the group NEW_NT_SITUATION_GROUP:

```
tacmd addgroupmember -g NEW_NT_SITUATION_GROUP -m NT_Disk_Space_Low -t SITUATION
```

Note: To add a member to a group, you need to create a group first. In the example above, create a situation group NEW_NT_SITUATION_GROUP by using the **createGroup** command, and then execute the **addGroupMember** command.

This example adds the bundle member specified in the first column of the CSV file to the bundle group NewBundleGroup:

```
tacmd addGroupMember -g NewBundleGroup -x c:\bulk_bundle_list.csv
```

The CSV file's format includes the following variables:
member,*type*,*cmdLine_options*,*properties*

In the preceding example, the `bulk_bundle_list.csv` CSV file contents are in the following format:

member	type	cmdLine_options	properties
unixBundle	BUNDLE	-y UX	KDYRXA.RXA protocol= ssh KDYRXA.RXAport =22
db2Bundle	BUNDLE	-y UD	-v 062000000
f50pa2d.tivlab. raleigh.ibm.com	BUNDLE	-y UD	INSTANCE=db2inst1

member	type	cmdLine_options	properties
amssol19.tivlab.raleigh.ibm.com	BUNDLE	-y UM	UA.CONFIG='file.mdl'

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd addSdaInstallOptions

Description

Use the **tacmd addSdaInstallOptions** command to add a version to product versions configured to be allowed for Self-Describing Agent (SDA) install. The hub monitoring server does not allow an SDA-enabled product to install its application support on the enterprise-wide monitoring server or portal server without the product version specification.You must log in by using the **tacmd login** command before running the **tacmd addSdaInstallOptions** command.

CLI syntax

tacmd addSdaInstallOptions
{-t|--type} DEFAULT
{-i|--install} ON|OFF
[{-f|--force}]

OR

tacmd addSdaInstallOptions
{-t|--type} ***PRODUCT_TYPE...***
{-v|--version} ***VERSION***
[{-f|--force}]

where:

{-t|--type} DEFAULT
Specifies the DEFAULT SDA installation option for all products that do not have a specific version defined for their product type.

Note: Any product with a specific version defined will only allow SDA installation for the allowed versions listed, not the DEFAULT SDA install option. When no DEFAULT SDA install option is defined, use OFF.

{-t|--type}*PRODUCT_TYPE...*
Specifies one or more managed system types (product codes) to update.

{-i|--install} ON|OFF
Required with the {-t|--type} DEFAULT option.
- ON enables SDA installation
- OFF disables SDA installation

-v | --version
Specifies the product versions configured to be allowed for SDA install for the product code identified. The version is an eight-digit identifier in the format *VVRRMMFF*, where *VV* specifies Version, *RR* specifies Release, *MM* specifies Modification, and *FF* specifies PTF Level. For example, the *VVRRMMFF* designation for ITM 623 FP2 is 06230200.

-f | --force
Specifies to add the SDA install option without prompting for confirmation.

CLI example

Run the following command to disable SDA installation for all products that do not have a specific product versions configured to be allowed for SDA install defined for their product type:

```
tacmd addSdaInstallOptions -t DEFAULT -i OFF
```

Run the following command to enables SDA installation for the ITM623 FP1 Windows product type, without prompting for user response:

```
tacmd addSdaInstallOptions -t NT -v 06230100  -f
```

Run the following command to enable SDA installation for the ITM623 FP2 Windows product type:

```
tacmd addSdaInstallOptions -t NT -v 06230200
```

Run the following command to enable SDA installation for ITM623 FP3 for the Linux and Unix product types.:

```
tacmd addSdaInstallOptions -t LZ UX -v 06230300
```

Return values

See Table 8 on page 272.

Related commands

"tacmd deleteSdaInstallOptions" on page 73

"tacmd editSdaInstallOptions" on page 93

"tacmd listSdaInstallOptions" on page 178

"tacmd listSdaOptions" on page 180

Return to Table 1 on page 5.

tacmd addSystem

Description

Use the **tacmd addSystem** command to deploy a monitoring agent to a computer in your IBM Tivoli Monitoring environment. The **tacmd addSystem** command deploys an agent and other required components if they are not already installed on the node. This command is also available for non-agent bundles. When using this command to set or modify an environment variable, ensure that the value you

assign to the variable is correct. An incorrect value assignment might impact the agent behavior and possibly prevent the agent from starting.

Note: When this command is issued against a managed node that already has the specified agent deployed to it, a message is issued informing you to specify that the "-p" options are required, such as _UNIX_STARTUP_.Username.

By using the bulk deployment option, the agents specified in the bundle group are deployed on the managed systems specified in the deployment group.

Any computer to which you want to deploy an agent must already have an OS agent installed. You can either install the OS agent by using the installation wizard or with the “tacmd createNode” on page 50 command.

Note: You cannot use this command to add a non-default Universal Agent instance that you created manually. You must use the **itmcmd config** command with the **-o** option to create additional Universal Agent instances.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Single IBM Tivoli Monitoring agent deployment:

tacmd addSystem
{-t|--type} *TYPE*
{-n|--node} *MANAGED-OS*
{-p|--property} *SECTION.NAME=VALUE...*
[{-e|--environment} *NAME=VALUE* **...]**
[{-o|--option|--options} *NAME=VALUE* **...]**

Bulk IBM Tivoli Monitoring or System Service Monitors agent deployment:

tacmd addSystem
{-g|--deploygroup *DEPLOY_GROUP_NAME***}**
{-b|--bundlegroup *BUNDLE_GROUP_NAME***}**
[{-e|--environment} *NAME=VALUE* **...]**
[{-o|--option|--options} *NAME=VALUE* **...]**
[{-x|--noexecute}]

Standard input option:

tacmd addSystem
{-stdin|--stdin}

where:

-t|--type
Specifies the type (product code) of agent to add to the monitoring system. See the *IBM Tivoli Monitoring Installation and Setup Guide* for a listing of agent product codes.

-n|--node
Identifies the node, or the directory on the monitoring system where the OS agent is installed, to which you want to add the agent. A node is identified by the managed operating system that it contains. The name of a

node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed.

-p | --property
Specifies *section.name=value* pairs that identify agent configuration properties and their values, where *section* specifies the name of the section containing the key value pair, *name* specifies the name of the configuration property, and *value* specifies the property value. You can specify the instance name of the system to be configured via the *instance* property for a system that can have multiple instances. If more than one option is specified, each*section.name=value* pair should be separated by a space.

See the agent user's guide for the agent that you are configuring for a list of available configuration properties.

In addition to the agent-specific configuration properties, you can also configure the Run-as settings, specifying the user ID under which an agent runs. Use the following options:

_UNIX_STARTUP_.Username=*user*
On UNIX, the username under which to run the agent. Note that you can only use this option if the OS agent running on the UNIX computer is started as the root user (or another user with privileges to super user). You cannot change the Run-as setting if your OS agent runs as a non-root user.

If you have already configured the Run-as user (for example, through the Manage Tivoli Enterprise Monitoring Services utility), this value defaults to what you have already set. If you have not configured the Run-as user previously, the default user is the user that is running the OS agent on the computer.

_WIN32_STARTUP_.Username=*user*
On Windows, the username under which to run the agent.

If you have already configured the Run-as user (for example, through the Manage Tivoli Enterprise Monitoring Services utility), this value defaults to what you have already set. If you have not configured the Run-as user previously, the default user is LocalSystem (InteractWithDesktop=0).

_WIN32_STARTUP_.Password=*pwd*
The password for the Run-as user that you specified with **_WIN32_STARTUP_Username**.

_WIN32_STARTUP_.LocalSystem={0 | 1}
Indicates whether you want to use the LocalSystem user to start the agent. Specify 1 if you want to use the LocalSystem user. Specify 0 if you do not want to use the LocalSystem user.

You must also specify the **_WIN32_STARTUP_.InteractWithDesktop={0 | 1}** option.

_WIN32_STARTUP_.InteractWithDesktop={0 | 1}
Indicates whether the LocalSystem can interact with the computer desktop. Use 1 to specify that it can and 0 to specify that it cannot.

-e | --environment
Specifies one or more NAME=VALUE pairs that identify environment variables to update, where NAME specifies the name of the environment

variable, and VALUE specifies the value to be assigned. If more than one environment variable is specified, each NAME=VALUE pair should be separated by a space. For example:

```
-e CTIRA_HOSTNAME=aixnode CTIRA_HEARTBEAT=8
```

Refer to the "Agent configuration and environment variables" appendix of the *IBM Tivoli Monitoring: Installation and Setup Guide* for a list of supported variables.

-o | --option | --options
One or more configuration parameters that can be used to customize the operation of this program. The valid options are: COLLECTALL, EXECPREREQCHECK, IGNOREPREREQCHECK. The values are to be specified in KEY=VALUE format.

-g | --deploygroup
Specifies the name of the deployment group to which the agents in the bundle group will be deployed.

-b | --bundlegroup
Specifies the name of the bundle group containing the agents that will be deployed to the managed systems in the deployment group.

-x | --noexecute
Causes the command to display which bundles will be deployed to which managed systems.

-stdin | --stdin
Indicates that all command-line parameters are processed from standard input (in the same command-line format) instead of being parsed from the command-line arguments.

CLI example

This command deploys universal agent (type UA) to the monitoring system named *HDCHASDSTC0213* with the *file.mdl MDL* file.

```
tacmd addSystem -t UM -n Primary:HDCHASDSTC0213:NT -p UA.CONFIG="file.mdl"
```

Each agent bundle has its own unique configuration properties that you need to provide in the tacmd addSystem command (by using the -p option). You can view the configuration parameters by running the tacmd describeSystemType command. The following example shows the configuration options that are available to use with the tacmd addSystem command for the Universal Agent (product code um) to be installed on a remote Windows system (platform WINNT):

```
tacmd describeSystemType -t um -p WINNT
```

The MDL file is made available in the %CANDLEHOME%/cms/depot/UACONFIG directory and $CANDLEHOME\tables*tems*\depot\UACONFIG in the case of a UNIX system.

This command deploys universal agent (type UA) to the monitoring system named *HDCHASDSTC0213* with the *script.mdl* file.

```
tacmd addSystem -t UM -n Primary:HDCHASDSTC0213:NT -p UA.SCRIPT="script.mdl"
```

The mdl file is made available in the %CANDLEHOME%/cms/depot/UASCRIPT directory and $CANDLEHOME\tables*tems*\depot\UASCRIPT directory in the case of a UNIX system.

This command includes the -e option to specify an environment variable setting for CTIRA_MAX_RECONNECT_TRIES.

```
tacmd addsystem -t ul -n amssol11:KUX -e CTIRA_MAX_RECONNECT_TRIES=10
```

The following is an example for the bulk deployment option. The agents specified in the bundle group are deployed to the host systems specified in the deployment group:

```
tacmd addSystem -g UnixGroup -b ULBundle
```

Note: Before executing the preceding command, create groups UnixGroup and ULBundle by using the **createGroup** command and add members on it by using the **addGroupMember** command as follows:

```
tacmd createGroup -g UnixGroup -d "unix deploy group" -t DEPLOY
-p KDY.SERVER=IP.PIPE:\\r111o001.tivlab.raleigh.ibm.com:1918
KDYRXA.TIMEOUT=300 KDYRXA.CONNECT_TIMEOUT=100 KDYRXA. RXAPROTOCOL=smb
KDYRXA.RXAPORT=4230

tacmd addGroupMember -g UnixGroup -t DEPLOY -m f50pa2d.tivlab.raleigh.ibm.com
-p KDYRXA.installDir=/data/aut/r111o001 KDYRXA.RXAUSERNAME=root
KDYRXA.RXAPASSWORD=**** KDYRXA.RXAPROTOCOL =ssh KDYRXA. RXAPORT =22

tacmd addGroupMember -g UnixGroup -t DEPLOY -m amssol19.tivlab.raleigh.ibm.com
-p KDYRXA.installDir=/data/aut/r111o001 KDYRXA.RXAUSERNAME =root KDYRXA.
RXAPASSWORD =Agnt2tst KDYRXA. RXAPROTOCOL =ssh KDYRXA. RXAPORT =22

tacmd createGroup -g ULBundle -d "UL bundle" -t BUNDLE

tacmd addGroupMember -g ULBundle –m linuxULBundle -t BUNDLE -y UL
–v 062100000 –i li6263
```

These additional examples include the -e option to specify an environment variable setting for CTIRA_HEARTBEAT.

```
tacmd addSystem -g UnixGroup -b ULBundle -e CTIRA_HEARTBEAT=4

tacmd addSystem -t UM -n Primary:HDCHASDSTC0213:NT -p UA.CONFIG="file.mdl"
-e CTIRA_HOSTNAME=WIN2008 CTIRA_HEARTBEAT=9
```

Return values

See Table 8 on page 272.

Related commands

"tacmd createNode" on page 50

"tacmd describeSystemType" on page 84 (to view configuration properties for an installed agent)

"cinfo" on page 275 (to return the list of product codes installed on the computer)

Return to Table 1 on page 5.

tacmd bulkExportPcy

Description

Use the **tacmd bulkExportPcy** command to export all the available policies from the Tivoli Enterprise Monitoring Server. You can filter for a specified system type, a list of specified system types, a specified policy name or names, or a listfile containing policy names.

You must log in by using the **login** command before running the **bulkExportPcy** command.

CLI syntax

tacmd bulkExportPcy
[-p|--path *POLICYFILEPATH*]
[-t|--type| --types *TYPE*]
[-n|--names *POLICYNAMES*]
[-f|--force]
[-d|--distribution]

tacmd bulkExportPcy
[-p|--path *POLICYFILEPATH*]
[-l|--listfile *FILENAME*]
[-f|--force]
[-d|--distribution]

where:

-t|--type| --types
Specifies one or more system types. Specify the two-digit character code of the system type name to export policies to. Specify 'Others' to export policies that are not related to any of the two-digit product code.

-n|--names
Specifies the list of policy names to export.

-p|--path
Specifies the path name where the policy XML files are to be created. If not specified, picks from either the environmental variable BULK_OBJECT_PATH or the current path.

-l|--listfile
Specifies the file name that contains the list of policy names to export.

-f|--force
Exports the policy files without prompting for confirmation from the user.

-d|--distribution
Exports the distribution list of the policies along with other details.

CLI example

This command exports all policies for the "NT" application type to multiple files in the `C:\IBM\ITM\BIN\Bulk\Policy\NT` directory. The file name for each exported policy corresponds to the policy name with the "`.xml`" file extension. The path `C:\IBM\ITM\BIN\` would be picked up from the BULK_OBJECT_PATH environment variable; if BULK_OBJECT_PATH is not defined in the environment, the current working directory would be used.

```
tacmd bulkExportPcy -t NT  -f
```

This command exports the NT_Disk_Busy policy details along the distribution details to an xml file in the `C:\IBM\ITM\BIN\Bulk\Policy\NT` directory.

```
tacmd bulkExportPcy –n NT_Disk_Busy –d –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd bulkExportSit

Description

Use the **tacmd bulkExportSit** command to export all the available situations from the Tivoli Enterprise Monitoring Server. You can filter for a specified system type, a list of specified system types, a specified situation name, or a listfile containing situation names. You must log in by using the **login** command before running the **bulkexportsit** command.

CLI syntax

tacmd bulkExportSit
[-p|--path *SITUATION_FILEPATH* **]**
[-t|--type| --types *TYPE***]**
[-n|--names *SITUATIONNAMES***]**
[-d|--distribution]
[-h|--historical]
[-f|--force]

tacmd bulkExportSit
[-p|--path *SITUATION_FILEPATH* **]**
[-l|--listfile] *FILENAME*
[-d|--distribution]
[-h|--historical]
[-f|--force]

where:

-t|--type| --types
One or more system types. Specify a two-digit character code of the system type name to export situations to. Specify 'Others' to export situations that are not related to any of the two-digit product code.

-l|--listfile
Specifies the file name which contains the list of situation names to export.

-n|--names
Specifies the list of situation names to export.

-p|--path
Specifies the path name where the situation XML files are to be created. If not specified, picks from either the environmental variable BULK_OBJECT_PATH or the current path.

-f|--force
Exports the situation files without prompting for a confirmation from the user.

-d|--distribution
Exports the distribution list of the situations along with other details.

-h|--historical
Exports only the collections, which are historical situations.

CLI example

This command exports all situations for the "NT" application type to multiple files in the C:\IBM\ITM\BIN\Bulk\Situation\NT directory. The file name for each exported situation corresponds to the situation name with ".xml" extension . The path C:\IBM\ITM\BIN\ would be picked up from the BULK_OBJECT_PATH environment variable; if BULK_OBJECT_PATH is not defined in the environment, the current working directory would be used.

```
tacmd bulkExportSit -t NT -f
```

This command exports all the NT situation details along with their distribution details to multiple xml files in the C:\IBM\ITM\BIN\Bulk\Situation\NT directory.

```
tacmd bulkExportSit -t NT –d -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd bulkImportPcy

Description

Use the **tacmd bulkImportPcy** command to import all the available policy objects to the Tivoli Enterprise Monitoring Server. You can filter for a specified system type, a list of specified system types, a specified object name, or a listfile containing object names. You must log in by using the **login** command before running the **bulkimportpcy** command.

CLI syntax

tacmd bulkImportPcy [-p|--path *POLICYFILEPATH* **]**
[-t|--type| --types *TYPE* **]**
[-n|--names *POLICYNAMES* **]**
[-l|--listfile *POLICYFILENAME* **]**
[-d|--distribution]
[-f|--force]

where:

-p|--path
Specifies the path name from where the files are to be imported. The XML files for import should be made available in the POLICYFILEPATH\Bulk\POLICY\Productcode\ directory. The following example would import files from C:\temp\Bulk\POLICY\NT\ :

```
tacmd bulkimportpcy -p C:\temp -t NT
```

-t|--type| --types
One or more system types. Specify a two-digit character code of the system type name to import objects. Specify 'Others' to import objects that do not pertain to any of the two-digit product codes.

-n|--names
One or more object names. Specify a list of object names to import.

-l|--listfile
Specifies the file name that contains the list of object names to import.

-f|--force
Imports the objects without confirmation.

-d|--distribution
Imports the distribution list of the policies along with other details.

CLI example

For Windows systems, this command imports all policies in the subdirectories under the path `C:\temp\Bulk\Policy\` directory into the Tivoli Enterprise Monitoring Server that the user is logged on to. The command executes without prompting for confirmation because the "-f" option was provided.

```
tacmd bulkImportPcy –p C:\temp -f
```

For UNIX systems, this command imports all the policy objects that are available from the respective product code subdirectories under the path `/tmp/Bulk/POLICY/`.

```
tacmd bulkimportpcy -p /tmp
```

Note: POLICY in the directory path is case sensitive in case of UNIX.

This example imports the policy NT_Disk_Busy with its distribution. Note: the above example will import the distribution only if the policy was previously exported by specifying the –d option.

```
tacmd bulkImportPcy –n NT_Disk_Busy –d –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd bulkImportSit

Description

Use the **tacmd bulkImportSit** command to import all the available objects to the Tivoli Enterprise Monitoring Server from BULK_OBJECT_PATH. You can filter for a specified system type, a list of specified system types, a specified object name, or a listfile containing object names. You must log in by using the **login** command before running the **bulkimportsit** command.

CLI syntax

tacmd bulkImportSit
[-p|--path *SITUATIONFILEPATH* **]**
[-t|--type| --types *TYPE***]**
[-n|--names *SITUATIONNAMES***]**
[-l|--listfile *SITUATIONFILENAME***]**
[-d|--distribution]
[-f|--force]

where:

-p|--path
Specifies the path name from where the objects are imported. The XML files for import should be made available in `SITUATIONFILEPATH\Bulk\SITUATION\Productcode\` directory. The following example would import files from `C:\temp\Bulk\SITUATION\NT\`:

```
tacmd bulkimportsit -p C:\temp -t NT
```

-t|--type| --types
One or more system types. Specify a two-digit character code of the system type name to import objects. Specify 'Others' to import objects that do not pertain to any of the two-digit product codes.

-n|--names
One or more object names. Specify a list of object names to import.

-l|--listfile
Specifies the file name that contains the list of object names to import. Specify the file name without the file name extension. For example, *object_name*`.xml` would be *object_name*`.`

-f|--force
Imports the objects without confirmation.

-d|--distribution
Imports the distribution list of the situations along with the other details.

CLI example

For Windows systems, this command imports all situations from the application type subdirectories (NT, UX, and so on) under the path `C:\temp\Bulk\Situation\` directory into the Tivoli Enterprise Monitoring Server that the user is logged on to. The command executes without prompting for confirmation because the "-f" option was provided.

```
tacmd bulkImportSit –p C:\temp -f
```

For UNIX systems, this command imports all the situation objects that are available from the respective product code subdirectories under the path `/tmp/Bulk/SITUATION/`.

```
tacmd bulkimportsit -p /tmp
```

Note: SITUATION in the directory path is case sensitive in case of UNIX.

This example imports the situation NT_Service_Error with its distribution.

Note: This imports the distribution only if the situation was previously exported by specifying the –d option.

```
tacmd bulkimportsit -n NT_Service_Error –d
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd checkprereq

Description

Use the **tacmd checkprereq** command to check for prerequisites required for deploying an agent to a managed system. The **tacmd checkprereq** command deploys a prerequisite checking tool to determine if the target system meets the requirements for the agent. A global transaction ID is immediately returned. You can then use the Deployment Status workspace in the Tivoli Enterprise Portal, or run the **tacmd getDeployStatus** command, to view the status of the queued operation. The pass or fail status information is saved in *candlehome*`logs/checkprereq_results`.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

For a single monitoring agent with endpoint credentials:

```
tacmd checkprereq {-h|--host} [ {smb|ssh|rexec|rsh}:// ] HOST [ :PORT ]
                  {-t|--type} TYPE
                  {-u|--username} USERNAME
                  {-w|--password} PASSWORD
                  [ {-d|--dir|--directory} DIRECTORYPATH ]
                  [ {-v|--version} VERSION ]
                  [ {-p|--property|--properties} NAME=VALUE ... ]
                  [ {-o|--option|--options} NAME=VALUE ... ]
                  [ {-c|--collectall} ]
```

For a single monitoring agent with an OS agent at the endpoint:

```
tacmd checkprereq {-n|--node MANAGED-OS}
                  {-t|--type} TYPE
                  [ {-v|--version} VERSION ]
                  [ {-p|--property|--properties} NAME=VALUE ... ]
                  [ {-o|--option|--options} NAME=VALUE ... ]
                  [ {-c|--collectall} ]
```

For bulk execution of agents:

```
tacmd checkprereq {-g|--deploygroup DEPLOY_GROUP_NAME}
                  {-t|--type} TYPE
                  [{-u|--username} USERNAME ]
                  [{-w|--password} PASSWORD ]
                  [ {-d|--dir|--directory} DIRECTORYPATH ]
                  [ {-v|--version} VERSION ]
                  [ {-o|--option|--options} NAME=VALUE ... ]
                  [ {-c|--collectall} ]
```

where:

-h|--host
Identifies the host where the prerequisite check runs. Optionally, a specific

connection protocol and a port can be specified. If you specify an OS agent product code, the correct product code for the target is automatically chosen.

-n | --node
Identifies the node or monitoring system where you want to execute the prerequisite check.

-g | --deploygroup
Identifies the name of the deployment group to which the prerequisite checker will be deployed.

-t | --type
Specifies the type of agent to add to the monitoring system.

-u | --username
A valid user login ID on the specified host. The software prompts you for the user name if you do not specify one.

-w | --password
The password for the specified user name. The software prompts you for the password if you do not specify one.

-d | --dir | --directory
Specifies the location on the specified host where the agent is to be installed. This location must be specified as a directory, in absolute path format.

-v | --version
Specifies the version of the agent.

-o | --option | --options
One or more configuration parameters that can be used to customize the operation of this program. The valid options are the following: TIMEOUT, CONNECT_TIMEOUT, TEMP, VERSION, AUTOCLEAN, KEYFILE, PASSPHRASE, AGENT, JLOG_APPENDING, JLOG_SET_FILE_DIR, JLOG_SET_FILE_NAME, JLOG_SET_MAX_FILES, JLOG_SET_MAX_FILE_SIZE, ENV_env_variable_name. The values are to be specified in KEY=VALUE format.

-p | --property | --properties
Specifies one or more configuration properties that identify configuration properties of the new system and their values. Values can differ per system. The following properties are valid for an IBM Tivoli Monitoring OS agent: ENCRYPT, KEY, IP_PIPE, IP_SPIPE, PROTOCOL, PROTOCOL1, PROTOCOL2, PROTOCOL3, PORT, SERVER, SNA_NETNAME, SNA_LOGMODE, SNA_LUNAME, SNA_TPNAME, BACKUP, BSERVER, BPROTOCOL, BPROTOCOL1, BPROTOCOL2, BPROTOCOL3, BSNA_NETNAME, BSNA_LOGMODE, BSNA_LUNAME, BSNA_TPNAME, FOLDER, BPORT, BIND_TO_NIC. The values are to be specified in KEY=VALUE format. See the **tacmd describeSystemType** command for a list of valid IBM Tivoli Monitoring application agent properties.

-c | --collectall
Specifies the collection of all results for all members in the deployment group, meaning that both success and failure results are retrieved. The default without specifying this option is to retrieve only failed results.

If you want to collect results for a selected group member only, you can specify KDYRXA.COLLECTALL=TRUE in the properties list for the deployment group member.

CLI example

The following example runs the command on the amsntx00 host, where a Windows OS Agent is being added. Both the success and failure results are retrieved:

```
tacmd checkprereq --host amsntx00 --type NT --collectall -u administrator
```

The following example, for bulk execution of agents, runs the command on the UX_deploy_group, for UNIX agents. Both the success and failure results are retrieved:.

```
tacmd checkprereq -g UX_deploy_group -t UX -c
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd cleanMS

Description

Use the **tacmd cleanMS** command to delete the entries for offline managed systems from the Tivoli Enterprise Monitoring Server. You must log in by using the **tacmd login** command before you run the **tacmd cleanMS** command.

Note: Specifying **tacmd cleanMS** shows you the prompt information. You must specify either the **-m** option, the **-a** option, or the **-g** option to clear any offline entries.

Note: If an error is encountered while clearing one or more offline managed system entries, the command logs the name of the managed system and the nature of the failure to the log file and continues to process the other offline entries.

Note: You cannot use the **-m** option if you are also using the **-p** option.

Note: In large-scale environments with significant numbers of nodes registered (for example, 10-20 thousand), using the **tacmd cleanMS** command or removing a node through the TEPS console can take several seconds to complete. If the **-a** option is used with the **tacmd cleanMS** command, and there are many offline nodes, the **tacmd cleanMS** command might take a long time to complete (that is, hours).

CLI syntax

tacmd cleanMS {-m|--systems} *MANAGEDSYSTEMNAME...*

tacmd cleanMS {-g|--age} *NUMDAYSOFFLINE* **[-p|-- preview]**

tacmd cleanMS {-a|--all} [-p|--preview]

where:

-m | --systems
Specifies the name of the offline managed systems in the Tivoli Enterprise Monitoring Server. Valid values include letters (upper or lowercase), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-g | --age
Clears all the offline managed system entries present in the Tivoli Enterprise Monitoring Server that have been offline for NUMDAYSOFFLINE or more.

-a | --all
Clears all the offline managed system entries present in the Tivoli Enterprise Monitoring Server.

-p | --preview
Checks the number of offline managed system entries present in the Tivoli Enterprise Monitoring Server, according to chosen condition. Only a simulation of clearing occurs to obtain the number of offline affected entries. Also specifies the names of eligible managed systems to be deleted.

CLI example

This example clears all the offline managed systems present on the Tivoli Enterprise Monitoring Server.

```
tacmd cleanms –a
```

This example clears the offline entries for the two managed systems specified:

```
tacmd cleanms –m Primary:HDCHASDSTC0061:NT  HDCHASDSTC0061:UA
```

Note: When Universal Agent nodes that have sub nodes are specified with the **-m** option, the command will delete the offline entries for the sub nodes as well after prompting for your confirmation.

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd clearAppSeedState

Description

Use the **clearAppSeedState** command to clear the value of the SEEDSTATE column of an installation record that has status IC (Installation Complete) and SEEDSTATE value I (Incomplete) or E (Error). You must log in by using the **tacmd login** command before running the **tacmd clearAppSeedState** command.

CLI syntax

tacmd clearAppSeedState
{-n | --temsname} ***TEMS NAME***
{-t | --type} ***TYPE***

{-v | --version} *PRODUCT_VERSION*
{-i | --idver} *ID_VERSION*
[{-f | --force}]

where:

-n | --temsname
Specifies the Tivoli Enterprise Monitoring Server name where you want to clear the seed state value of the record.

-t | --type
Specifies the product code of the records to be cleared.

-v | --version
Specifies the product version of the records to be cleared. PRODUCT_VERSION must be in the format XXXXXXXX (8 integers). For example, 06230000.

-i | --idver
Specifies the ID product version of the record to be cleared. ID_VERSION must be in the format XXXXXXXX (8 integers). For example, 06230000.

-f | --force
Specifies to clear the record without prompting for confirmation.

CLI example

Run the following command to clear the value of the SEEDSTATE column for the LZ product code on nc11722_HUB machine:

```
tacmd clearappseedstate -t LZ -v 06230000 -i 06230000 -n nc117122_HUB
```

Return values

See Table 8 on page 272.

Related commands

"tacmd listSdaOptions" on page 180

Return to Table 1 on page 5.

tacmd clearDeployStatus

Description

Use the **tacmd clearDeployStatus** command to remove entries from the table that stores the status of the asynchronous agent deployment operations. The command allows you to clear all the entries in the status table. This command also provides the option of filtering the table entries which deletes specific entries from the status table entries down to a specific deployment operation. You must log in by using the **tacmd login** command before running the **tacmd clearDeployStatus** command.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Clear all the entries in the status table, except in-progress entries:

tacmd clearDeployStatus {-a|--all}

Clear all the entries in the status table, including in-progress entries:

tacmd clearDeployStatus {-a|--all} {-i|–inprogress}

Clear specific deployment operation in the status table:

tacmd clearDeployStatus
[{-g|--transactionID}*TRANSID* **...]**
[{-c|--command} *COMMAND***]**
[{-h|--hostname} *HOSTNAME***]**
[{-p|--platform} *PLATFORM***...]**
[{-v|--version} *VERSION***]**
[{-t|--product}]
[{-f|--failed}]
[{-s|--successful}]
[{-q|--queued}]
[{-r|--retryable}]
[{-i|--inprogress}]
[{-y|--yes}]
[{-a|--all}]

where:

--g|--transactionID
Specifies global transaction ID.

c|--command
Specifies the type of the deployment operation. Acceptable operations are: START, RESTART, STOP, INSTALL, REMOVE, CONFIGURE, UPDATE, CHECKPREREQ, or SETAGENTCONN.

-h|--hostname
Specifies the hostname of the deployment operation that is deleted from the status table.

-p|--platform
Specifies the platform of the deployment operation that is deleted from the status table.

-v|--version
Specifies the version of the deployment operation that is deleted from the status table.

-t|--product
Specifies the product type of the deployment operation that is deleted from the status table.

-f|--failed
The flag to filter the result by the failed transaction.

-s|--successful
The flag to filter the result by the successful transaction.

-q | --queued
The flag to filter the result by the queued transaction.

-r | --retryable
The flag to filter the result by the retryable transaction.

-i | --inprogress
The flag to allow the clearing of an i-progress transaction.

Note: This flag cannot be used on its own and the user must confirm the action of clearing an in-progress transaction (unless the -y | --yes option in specified).

-y | --yes
The flag specifying the performing of actions without requesting confirmation.

-a | --all
This option clears all the entries in the Remote Deploy status table, except for the in-progress entries (unless the -i | --inprogress option is specified).

CLI example

The following command clears the deployment status for the transaction ID "12173047037190000015724" on the Tivoli Enterprise Monitoring Server you are logged on to:

```
tacmd cleardeploystatus -g 12173047037190000015724
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd configurePortalServer

Description

Use the **tacmd configurePortalServer** command to configure a user-defined portal server data source. If the data source already exists, use this command to change the configuration. If the data source does not exist, it is created by this command. You can also use this command to remove a data source. This command can only be run from a Tivoli Enterprise Portal Server installation.

When the **tacmd configurePortalServer** command is run on a system that does not have a Tivoli Enterprise Portal Server installed, a cq.ini file is created. When a Tivoli Enterprise Portal Server is not installed, this command fails with an error message indicating that this command should only be run on a Tivoli Enterprise Portal Server or that a Tivoli Enterprise Portal Server configuration file was not found. Determine whether this system has a Tivoli Enterprise Portal Server installed and ensure that the path is specified correctly with the -d <CANDLEHOME> option or CANDLEHOME variable correctly exported in the CLI's environment before running this command.

Note:

1. System defined data sources cannot be edited with this command. Only user-defined data sources can be edited with this command. Names for these data sources are in the format "DSUSER1", "DSUSER2"...etc.
2. You must recycle the portal server to get a newly created connection to show up as a data source.

CLI syntax

tacmd configurePortalServer
{-s|--datasource} *DATASOURCE*
{-p|--property|--properties} *NAME=VALUE* **...**
[{-d|--directory} *CANDLEHOME*] [**{-f|--force** }]

tacmd configurePortalServer
{-s|--datasource} *DATASOURCE*
{-r|--remove} [**{-d|--directory}** *CANDLEHOME*] [**{-f|--force** }]

tacmd configurePortalServer {-s|--datasource} *DATASOURCE*
{-v|--view} [**{-d|--directory}** *CANDLEHOME*]

where:

-s|--datasource
Specifies the name of a new or existing data source. If the data source already exists and the remove option is not given, then an edit operation occurs. If the data source does not already exist, and the remove option is not specified, then an add operation occurs.

-r|--remove
Removes the named data source.

-v|--view
Display the properties of a datasource and their values. Password properties have their values displayed in an encrypted form.

-p|--property|--properties
A list of property names and values required to configure the data source. The list can be different for each data source type but usually includes at least user ID (key name *UID*) and password (key name *PWD*). Each property is stored as a key=value pair. Property values are encrypted before being stored to the configuration file or the Windows Registry. The combined properties create your datasource connection string. The following is a list of typical properties:

CONNECTION_TYPE=
Required if other than ODBC. Valid values are JDBC or DB2.

CONNECTION_LIMIT=
Optional. Controls the limit on how many simultaneous connections for ODBC or DB2 can be opened by the portal server. This property does not affect JDBC.

KFWDSURL=
If you are using JDBC, this is required.

KFWJDBCDRIVER=
If you are using JDBC, this is required.

Note: Your datasource connection string might require other properties not documented here.
The following is an example of datasource connection string for Oracle:

```
DSUSER2 =DSN=myJDBC;UID=scott;PWD=tiger;CONNECTION_TYPE=JDBC;
KFWJDBCDRIVER=/somewhere/ojdbc14.jar;
KFWDSURL=jdbc:oracle:thin:@myhost:1521:orcl
```

-d|--directory
The server's home directory.

-f|--force
Performs actions without asking confirmation.

CLI example

The following example modifies the *DSUSER1* data source with user ID *db2user* and password *db2password*. The DSUSER1 data source is created if it does not already exist.

```
tacmd configurePortalServer -s DSUSER1 -p UID=db2user PWD=db2password
```

The following example shows the configuration settings for the DSUSER1 data source:

```
tacmd configurePortalServer -s DSUSER1 -v
```

The following output is displayed:

```
DSN=DSUSER1
UID=db2user
PWD={AES256:keyfile:a}HW0LxUxCJ5tj9biXUWhCIQ==
CONNECTION_LIMIT=32
```

Return values

See Table 8 on page 272.

Note: To verify that the data source was correctly configured, log on to the portal server, click **Query** in the main toolbar to open the Query editor, and then click **New Query** to open the Create Query window. The name of the ODBC data source you configured is displayed in the Data Sources list.

Related commands

Return to Table 1 on page 5.

tacmd configureSystem

Description

Use the **tacmd configureSystem** command to edit configuration options of an existing managed system. By default, the managed system monitoring agent is restarted so the new configuration parameters can take effect. When using this command to set or modify an environment variable, ensure that the value you assign to the variable is correct. An incorrect value assignment might impact the agent behavior and possibly prevent the agent from starting. Note that you can use this command with the -e|--environment option to specify environment variables for an OS agent. However, in the case of an OS agent, you cannot use the -p (properties) option to update the OS agent connection properties. To configure OS agent connection properties, use the **tacmd setAgentConnection** command.

Specify the configuration data through the parameter pair *SECTION.NAME=VALUE*. If an optional parameter is specified with an equal sign but without a value, the specified parameter is removed from the agent configuration. By using the bulk deployment option, the agents specified in the bundle group are configured on the managed systems specified in the deployment group.

The environment option is supported for both individual configuration and group configuration where there are environment variables that need to be changed or defined at the target node for the agent to work properly. Only those environment variables should be set or updated that are incorrect or not set and are required for the agent. You can query deployment workspaces if you are using the Tivoli Enterprise Portal or run the **tacmd getDeployStatus** command to determine the result for the **tacmd configureSystem** command.

The -x|--noexecute option is intended to allow you to determine which configuration properties are used to configure which managed systems.

If you specify the -n|--noagentrestart option instead of the default, the managed system monitoring agent is reconfigured, but the agent is not restarted after the configuration.

Note: You cannot use this command to configure a non-default Universal Agent instance that you created manually. Use the **itmcmd config** command with the -o option instead to configure a non-default Universal Agent instance.

Note: Use the **tacmd describeSystemType** command to view available configuration properties.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Configuring a single IBM Tivoli Monitoring agent:

tacmd configureSystem
{-m|--system} ***SYSTEM***
{-p|--property|--properties} ***SECTION.NAME=VALUE*** **...**
[{-e|--environment} ***NAME=VALUE*** **...]**
[{-n|--noagentrestart}]
[{-f|--force}]

Configuring a single System Service Monitors agent:

tacmd configureSystem
{-h|--host} ***HOST[:PORT]***
[{-c|--configfile|--configfiles} ***CONFIG_LIST*****]**
[{-l|--filelist} ***FILE_LIST*****]**
[{-p|--property|--properties}] ***SECTION.NAME=VALUE*** **...**
[{-r|--reboot}]
[{-f|--force}]

Note: At least one of -p, -c, or -l must be specified.

Configuring bulk IBM Tivoli Monitoring or System Service Monitors agents:

tacmd configureSystem
{-g | --deploygroup *DEPLOY_GROUP_NAME***}**
{-b | --bundlegroup *BUNDLE_GROUP_NAME***}**
[{-e | --environment} *NAME=VALUE* **...]**
[{-x | --noexecute}]
[-n | --noagentrestart]
[{-f | --force}]

Standard input option:

tacmd configureSystem
{-stdin | --stdin}

where:

-m | --system
Identifies the agent (managed system) for which to update the configuration.

-h | --host
Identifies the location of the host of the System Service Monitors agent.

-c | --configfile | --configfiles
List of one or more configuration files (separated by spaces) that are executed on the System Service Monitors agent. Configuration files must be located in the depot under `depot/SSMCONFIG`.

-l | --filelist
List of one or more files (separated by spaces) that are transferred to the System Service Monitors agent, but *not* installed as configuration files. Configuration files must be located in the depot under `depot/SSMCONFIG`.

-e | --environment
Specifies one or more *NAME=VALUE* pairs that identify environment variables to update, where *NAME* specifies the name of the environment variable, and *VALUE* specifies the value to be assigned. If more than one environment variable is specified, each NAME=VALUE pair should be separated by a space. For example:

```
-e CTIRA_HOSTNAME=aixnode CTIRA_HEARTBEAT=8
```

Refer to the "Agent configuration and environment variables" appendix of the *IBM Tivoli Monitoring: Installation and Setup Guide* for a list of supported variables.

Note: If you update the environment variable CTIRA_HOSTNAME, you should use the **tacmd cleanMS** command to delete the entries for offline managed systems from the Tivoli Enterprise Monitoring Server before running other remote deploy commands to the target system. Use the **tacmd listSystems** command to display the list of managed systems.

-p | --property | --properties
Specifies one or more *SECTION. NAME=VALUE* pairs that identify configuration properties to update, where *SECTION* specifies the configuration section containing the configuration property, *NAME* specifies the name of the configuration property, and *VALUE* specifies the property value. Specify the instance name of the system to be configured by using the *INSTANCE* property for a system that can have multiple instances. If more than one property is specified, each NAME=VALUE pair should be separated by a space.

-r | --reboot
Forces a restart of the System Service Monitors agent.

-g | --deploygroup
Specifies the name of the deployment group to which the agents in the bundle group are configured.

-b | --bundlegroup
Specifies the name of the bundle group containing the agents which are configured to the managed systems in the deployment group.

-x | --noexecute
Causes the command to display the configuration properties used to configure specific managed systems.

-n | --noagentrestart
Specifies that when configuration completes, the agent is not rebooted.

-f | --force
Allows execution of the command without prompting for confirmation.

-stdin | --stdin
Indicates that all command-line parameters are processed from standard input (in the same command-line format) instead of being parsed from the command-line arguments.

CLI example

This command reconfigures the universal agent on *stone* with the *file_unix.mdl MDL* file.

```
tacmd configureSystem -m stone:UA -p UA.CONFIG="file_unix.mdl"
```

The following is an example for the bulk deployment option. The agents specified in the bundle group are configured to the host systems specified in the deployment group:

```
tacmd configureSystem -g UnixGroup -b ULBundle
```

Note: Before executing the above command, create groups UnixGroup and ULBundle by using the **createGroup** command and add members to it by using the **addGroupMember** command as follows:

```
tacmd createGroup -g UnixGroup -d "unix deploy group" -t DEPLOY
-p KDY.SERVER=IP.PIPE:\\s108o001.tivlab.raleigh.ibm.com:1918
KDYRXA.TIMEOUT=300 KDYRXA. CONNECT_TIMEOUT =100 KDYRXA. RXAPORT =4230

tacmd createGroup -g ULBundle -d "unix log Bundle group" -t BUNDLE

tacmd addGroupMember -g UnixGroup -t DEPLOY -m f50pa2d.tivlab.raleigh.ibm.com
-p KDYRXA.installDir=/data/aut/s108o001 KDYRXA.RXAUSERNAME=root
KDYRXA.RXAPASSWORD=**** KDYRXA.RXAPROTOCOL =ssh KDYRXA. RXAPORT=22

tacmd addGroupMember -g UnixGroup -t DEPLOY -m amssol19.tivlab.raleigh.ibm.com
-p KDYRXA.installDir=/data/aut/s108o001 KDYRXA.RXAUSERNAME=root
KDYRXA.RXAPASSWORD=**** KDYRXA.RXAPROTOCOL =ssh KDYRXA. RXAPORT=22

tacmd addGroupMember -g ULBundle –m linuxULBundle -t BUNDLE -y UL
–v 062100000 –i li6263
```

This additional example includes the -e option to specify an environment variable setting for CTIRA_HEARTBEAT.

```
tacmd configureSystem -m stone:UX -e CTIRA_HOSTNAME=stoneUNIX CTIRA_HEARTBEAT=9
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addSystem" on page 24

Return to Table 1 on page 5.

tacmd createAction

Description

Use the **tacmd createAction** command to create a new Take Action. You must log in by using the **tacmd login** command before running the **tacmd createAction** command.

CLI syntax

tacmd createAction
{-n|--name} *ACTIONNAME*
{-t|--type} *TYPE*
{-p|--property|--properties} *NAME=VALUE*
[{-d|--detailtextname} *TYPEDESC* **]**

where:

-n|--name
The name of the action to be created.

-t|--type
The application type name. Specifies the application two-digit code for which the action has to be created. Note that you cannot create a Take Action command for the Tivoli Enterprise Portal Server managed system (type CQ).

-d|--detailtextname
Application detail text name. Specify detail text of system type name to create the action.

-p|--property|--properties
Specifies one or more NAME=VALUE pairs that identify the properties of the new action and their values. The Cmd property in the -p option is mandatory for creating the new action. Valid property entries for *name* are:

Desc or Description
The description of the Take Action command to be created. Input given as text enclosed between double quotation marks, such as:

```
-p Desc="Stops the specified services"
```

Cmd or Command
The system command to be executed.

Key The unique key value to identify the Take Action command. If you have not specified the key value, a random key value would be generated. This value does not support non-ASCII characters.

CLI example

This example creates a new action command of name "Test Alerter Service "of type NT in the logged in server.

```
   tacmd createAction -n "Test Alerter Service" -t NT -p cmd="net start Alerter"
desc="To start the alerter service" key="Test Alerter"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd createEventDest

Description

Use the **tacmd createEventDest** command to create a new event destination definition on the server.

Note: The hub monitoring server needs to be recycled or refreshed for this action to take effect. If this is the first time you are configuring EIF forwarding after an upgrade, both the portal server and the Tivoli Enterprise Portal client must be recycled.

CLI syntax

tacmd createEventDest
{-i|--id| --serverID} *ID*
[{-p|--property|--properties} *NAME=VALUE...*
[{-f|--force }]

where:

-i|--id| --serverID
Identify the Server Destination ID of the event destination server definition to create on the server. The value must be a value between 1 and 999, inclusive, and an event destination server definition with the same ID cannot already be defined on the server.

-f|--force
Delete the event destination server definition on the server without prompting for confirmation.

-p|--property|--properties
Specifies one or more NAME=VALUE pairs that identify the properties of the new event server destination and their values. The NAME|SERVERNAME, and HOST1 properties are required.

Host properties should be specified in the format:

```
HOST{1|2|3|4|5|6|7|8}=HOSTNAME[:PORT]
```

Host entries must be defined such that they are sequential in existence; for example, you cannot specify the HOST3 property if HOST2 is not also specified. If a port value is not provided for a host entry, the port will default to 0.

A maximum of 5 default servers are allowed. To designate this event destination server as a default server, specify the DEFAULT | DEFAULTSERVER property with a value of Y.

The property TYPE indicates whether the event destination server is a TEC server, an OMNIbus server, or a WS-Notification. Permitted values for the TYPE property include "T" for TEC, "M" for Micromuse/OMNIbus (Netcool®), and "W" for WS-Notification.

The following property names are valid:

- DESC | DESCRIPTION
- NAME | SERVERNAME
- TYPE | SERVERTYPE
- DEFAULT | DEFAULTSERVER
- HOST1
- HOST2
- HOST3
- HOST4
- HOST5
- HOST6
- HOST7
- HOST8

CLI example

This command creates a new event destination definition on the server *bigTECserver:4567* with the Server Destination ID of *123* .

```
tacmd createEventDest -i 123 -p host1=bigTECserver:4567 name=myTEC
```

This command creates a new event destination definition on the server *<hostname>:9899* with the Server Destination ID of *330* .

```
tacmd createEventDest -i 330 -p HOST1=<hostname>:9899 NAME=test_NetCool TYPE=M
```

with the following values:

```
tacmd vieweventdest -i 330

Server Id  : 330
Server Name: test_NetCool
Server Type: Micromuse/Omnibus
Description:
Default    : N
Host1      : clisoap.romelab.it.ibm.com:9899
Host2      : Not set
Host3      : Not set
Host4      : Not set
Host5      : Not set
Host6      : Not set
Host7      : Not set
Host8      : Not set"
```

These commands set multiple default event servers.

```
tacmd createEventDest -i 123 -p host1=bigTECserver:4567 default=Y name=myTEC
tacmd createEventDest -i 124 -p host1=bigTECserver1:4577 default=Y name=myTEC1
```

or with the **tacmd editEventDest** command, you can set DEFAULT=Y for existing event servers.

To send a situation to all three default event servers (the two that are defined and the basic one), specify an empty destination for the situation, as depicted in the following example.

```
C:\ODI>tacmd viewsit -s test_tec1
Name                    : test_tec1
Full Name               :
Description             :
Type                    : Windows OS
Formula                 : *IF *VALUE NT_Cache.Copy_Read_Hits_% *EQ 1
Sampling Interval       : 0/0:15:0
Run At Start Up         : Yes
Distribution            :
Text                    :
Action Location         : Agent
Action Selection        : System Command
System Command          : *NONE
True For Multiple Items: Action on First Item only
TEC Severity            : Critical
TEC Forwarding          : Y
TEC Destination         :
```

Note: The TEC Destination field is empty, but TEC Forwarding is set to **Y**.
In this example, the situation is sent to the <*default receiver*>, myTEC and myTEC1 event servers. In the TEC interface, only <*Default EIF Receiver*> displays in the left column (Assigned EIF Receivers), while myTEC and myTEC1 display in the right column (Available EIF Receivers), despite being set as DEFAULT servers. This is a known Tivoli Enterprise Portal limitation.

To change a situation from a specified TEC destination to an unspecified (empty) TEC destination, perform the following steps:

1. Export the situation to an XML file:

   ```
   C:\ODI>tacmd viewsit -s test_tec1 -e c:\test_tec1.xml
   ```

2. Edit the XML file and change

   ```
   <SITINFO>
   <![CDATA[SEV=Critical;TFWD=Y;~;"]] >
   </SITINFO>
   ```

   accordingly, to specify a null destination server.

3. Delete the original situation and import the situation back to the server:

   ```
   tacmd createsit -i c:\test_tec1.xml
   ```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd creategroup

Description

Use the **tacmd creategroup** command to create a new group on the server. You must log in by using the **login** command before running the **creategroup** command.

CLI syntax

tacmd creategroup
{-g|--group} *GROUPNAME*
{-t|--grouptype} *DEPLOY|BUNDLE|SITUATION|COLLECTION*
[-d|--description] *DESCRIPTION*
[-p|--property|--properties] *PROPERTY...*
[-l|--list] *MANAGED_SYSTEM_NAME | MANAGED_SYSTEM_LIST*

where:

-g|--group
Specifies the name of the group to be created.

-t|--grouptype
Specifies the group type name. Acceptable type names are DEPLOY, BUNDLE, SITUATION or COLLECTION.

-d|--description
Specifies the description for the group to be created.

-l|--list
Specifies one or more managed systems or managed system groups to be assigned to the group. This option is valid only for situation and collection groups.

-p|--property|--properties
Specifies one or more NAME=VALUE or SECTION.NAME=VALUE pairs that identify the configuration properties of the new group to be created. See "Configuration options and properties" on page 252 for information on these properties.

CLI example

The following example creates a new deployment group "NewWindowsDeployGroup" on the server:

```
tacmd createGroup -g NewWindowsDeployGroup -t DEPLOY -p KDYRXA.RXAUSERNAME=testuser
KDYRXA.RXAPASSWORD=1235 -d "Deploy Group"
```

The following creates a situation group that assigns the specified managed system to its distribution list:

```
tacmd creategroup -g newGroup -t situation -l Primary:test1:NT
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd createNode

Description

Use the **tacmd createNode** command to deploy an IBM Tivoli Monitoring node or a System Service Monitors agent to a host. This command also creates a *node*, the directory into which not only the OS agent is installed, but where any non-OS

agents are deployed. A create node request is sent to the deployment controller on the hub Tivoli Enterprise Monitoring Server, and a global transaction ID is immediately returned to the user. You can then use the Deployment Status workspace on the Tivoli Enterprise Portal, or execute the **getDeployStatus** CLI, to view the status of the queued operation. When using this command to set or modify an environment variable, ensure that the value you assign to the variable is correct. An incorrect value assignment might impact the agent behavior and possibly prevent the agent from starting. You must log in by using the **tacmd login** command before running the **tacmd createNode** command.

The node is created on the local computer if no host is specified.

Note:

1. On UNIX computers, you must run the **tacmd createNode** command as a root user.
2. Use **tacmd createNode** to deploy an OS agent when there are no OS agents already on the target computer. Attempting to deploy multiple OS agents on the same computer can cause unpredictable results.
3. Only Secure Shellversion 2 communication protocol is supported, Secure Shell version 1 is not supported.
4. Properties provided when using the **tacmd creatnode** command for a bulk deployment override the properties given inside the group and group members. For more information about the order of precedence, see the Properties precedence table in the Bulk agent deployment subsection in the *IBM Tivoli Monitoring Installation and Setup Guide*.
5. Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Execution for a single IBM Tivoli Monitoring or System Service Monitors agent:

tacmd createNode
{-h|--host} [{smb|ssh|rexec|rsh}://] *HOST [:PORT]*
{-u|--username} *USERNAME*
{-w|--password} *PASSWORD*
[{-o|--option|--options} *NAME=VALUE* ...]
[{-d|--dir|--directory} *DIRECTORYPATH*]
[{-p|--property|--properties} *NAME=VALUE* ...]
[{-t|--type}*ITM|SSM*]
[{-e|--environment}NAME=VALUE ...]
[{-k|--securegroup} *ITMGROUP*]
[{-f|--force}]

Bulk execution for IBM Tivoli Monitoring or System Service Monitors agents:

tacmd createNode
{-g|--deploygroup *DEPLOY_GROUP_NAME*}
{-b|--bundlegroup *BUNDLE_GROUP_NAME*}
{-u|--username} *USERNAME*
{-w|--password} *PASSWORD*
[{-s|--serverlist} *SERVER_LIST*]
[{-o|--option|--options} *NAME=VALUE* ...]
[{-d|--dir|--directory} *DIRECTORYPATH*]
[{-t|--type}*ITM|SSM*]

[{-e | --environment}NAME=VALUE ...]
[{-k | --securegroup} ***ITMGROUP*****]**
[{-f | --force}]

where:

-h | --host
Identifies the host on which to create a node. Optionally, a specific connection protocol and a port can be specified.

-g | --deploygroup
The name of the deployment group to which the agents in the bundle group will be deployed.

-b | --bundlegroup
The name of the bundle group containing the agents which is deployed to the managed systems in the deployment group.

-s | --serverlist
One or more server names, separated by space, from which the bulk createNode operations should be issued from.

u | --username
A valid user log in ID on the specified host. The software prompts you for the username if you do not specify one.

Note: On UNIX computers, you must run the **tacmd createNode** command as a root user. When using this command to deploy a Monitoring Agent for Windows, you must specify a user that has Administrator privileges on the remote computer. Local user accounts must be specified.

-w | --password
The password for the specified user ID. The software prompts you for the password if you do not specify one.

-o | --option | --options
One or more configuration parameters that can be used to customize the operation of this command. The valid options are: TIMEOUT, CONNECT_TIMEOUT, TEMP, VERSION, AUTOCLEAN, KEYFILE, PASSPHRASE, AGENT, JLOG_APPENDING, JLOG_SET_FILE_DIR, JLOG_SET_FILE_NAME, JLOG_SET_MAX_FILES, JLOG_SET_MAX_FILE_SIZE, COLLECTALL, EXECPREREQCHECK, IGNOREPREREQCHECK, or ENV_env_variable_name. The values are to be specified in KEY=VALUE format. If more than one option is specified, each KEY=VALUE pair should be separated by a space.

See “Configuration options and properties” on page 252 for information on these options.

-d | --dir | --directory
Specifies the location on the specified host where the agent is installed. This location must be specified as a directory in absolute path format.

-p | --property | --properties
Specifies one or more configuration properties that identify configuration properties of the new system and their values. Values can differ per system.

- The valid properties for an IBM Tivoli Monitoring agent are: ENCRYPT, KEY, IP_PIPE,IP_SPIPE, PROTOCOL,PROTOCOL1, PROTOCOL2, PROTOCOL3, PORT,SERVER, SNA_NETNAME, SNA_LOGMODE, SNA_LUNAME, SNA_TPNAME, BACKUP, BSERVER, BPROTOCOL,

BPROTOCOL1, BPROTOCOL2, BPROTOCOL3, BSNA_NETNAME, BSNA_LOGMODE, BSNA_LUNAME, BSNA_TPNAME, FOLDER,BPORT, or BIND_TO_NIC. See "Configuration options and properties" on page 252 for information on these properties.

- The valid properties for an SSM agent are : SVCUSERNAME, SVCPASSWORD, SNMPPORT, SNMPCOMMUNITY, COEXIST, OVERWRITE, SERVER_GUI,MS_SNMP_OVERRIDE, DISABLE_SNMPV1, DISABLE_SNMPV2, V3AUTHPROTOCOL, V3AUTHPASSWORD, V3PRIVPROTOCOL, CORE_ONLY, V3PRIVPASSWORD, MANUAL_SERVICE, SERVER, BSERVER, CLUSTER_INST, CLUSTER_GROUP, CORE_CONFIG_DISK, AGENTLOG ,BYPASS_RECONFIG, AGENTLOGSIZE, SNMPTRAPVER, CONFIGDIR, or INST_CONSOLE. The values are to be specified in KEY=VALUE format. If more than one property is specified, each NAME=VALUE pair should be separated by a space.

-t|--type
The type of agent to deploy to the unmanaged host. The valid values are ITM and SSM. Default is ITM.

-e|--environment
Specifies one or more NAME=VALUE pairs that identify environment variables to update, where NAME specifies the name of the environment variable, and VALUE specifies the value to be assigned. If more than one environment variable is specified, each NAME=VALUE pair should be separated by a space. For example:

```
-e CTIRA_HOSTNAME=aixnode CTIRA_HEARTBEAT=8
```

Refer to the "Agent configuration and environment variables" appendix of the *IBM Tivoli Monitoring: Installation and Setup Guide* for a list of supported variables.

-k|--securegroup
Specifies the ITMGROUP.

-f|--force
Executes the create node query without user confirmation.

CLI example

This example installs the OS agent in the `/opt/IBM/ITM` directory on *stone.ibm.com*. The installation is performed as *root*.

```
tacmd createNode -h stone.ibm.com -d /opt/IBM/ITM -u root
```

This example installs the OS agent in the `/opt/IBM/ITM` directory on *stone.ibm.com* and includes the -e option to specify an environment variable setting for CTIRA_HEARTBEAT. The installation is performed as *root*.

```
tacmd createnode -h stone.ibm.com -u root -d /opt/IBM/ITM -e CTIRA_HEARTBEAT=3
```

This example installs the OS agent on the Windows system *stone.ibm.com*. The installation is performed as the user Administrator.

```
tacmd createNode -h stone2.ibm.com -u Administrator
```

The following is an example for the bulk deployment option, where the Monitoring Agent for UNIX OS is deployed to the host systems specified in the deployment group:

```
tacmd createnode -g UnixGroup
```

Note: Before executing the above command, create a deployment group UnixGroup by using the **createGroup** command and add members to it by using the **addGroupMember** command as follows:

```
tacmd createGroup -g UnixGroup -d "unix deploy group" -t DEPLOY
-p KDY.SERVER=IP.PIPE://topaz.raleigh.ibm.com:1918
KDYRXA.TIMEOUT=300 KDYRXA.CONNECT_TIMEOUT=100 KDY.port=4230
```

```
tacmd addGroupMember -g UnixGroup -t DEPLOY -m jade.raleigh.ibm.com
-p KDYRXA.installDir=/home/root/ITMOSAgent KDYRXA.RXAUSERNAME=root
KDYRXA.RXAPASSWORD=***** KDYRXA.RXAPROTOCOL=ssh KDYRXA.RXAPORT=22
```

```
tacmd addGroupMember -g UnixGroup -t DEPLOY -m sapphire.raleigh.ibm.com
-p KDYRXA.installDir=/home/root/ITMOSAgent1 KDYRXA.RXAUSERNAME=achan
KDYRXA.RXAPASSWORD=***** KDYRXA.RXAPROTOCOL=ssh KDYRXA.RXAPORT =22
```

This example includes the -e option to specify an environment variable setting for CTIRA_MAX_RECONNECT_TRIES, CTIRA_HEARTBEAT, and KHD_HISTRETENTION.

```
tacmd createnode --deploygroup deploy_ux --environment
CTIRA_MAX_RECONNECT_TRIES=10 CTIRA_HEARTBEAT=4 KHD_HISTRETENTION=48
```

This example includes the -e option to specify an environment variable setting for CTIRA_HEARTBEAT.

```
tacmd createNode -h stone.ibm.com -d /opt/IBM/ITM -u root -e CTIRA_HOSTNAME=stone
UNIX CTIRA_HEARTBEAT=9
```

See the following example for deploying an SSM agent:

```
tacmd createNode -h smb://ruby.raleigh.ibm.com -t SSM -u root –w ****
-d c:\SSMAgent\ssm
```

Return values

See Table 8 on page 272.

Related commands

“tacmd addSystem” on page 24

Return to Table 1 on page 5.

tacmd createSit

Description

Use the **tacmd createSit** command to create a new situation.

The **tacmd createSit** command enables you to create situations without using the Tivoli Enterprise Portal. However, if you make a mistake in typing the name of an attribute when using this command, you do not receive an error. The situation is created, skipping the attribute that you meant to type in the command. If the created situation had, for example, 6 attributes to monitor, the new created situation has only 5 if you made a mistake in typing 1 of the attribute names in the command.

If you are using the IBM Tivoli Monitoring command line **tacmd createSit** function for situation creation, you can use the Situation editor in the Tivoli Enterprise Portal to validate your specified attributes.

You must log in by using the **login** command before running the **createsit** command.

Note: You cannot use this command to create UADVISOR situations.

CLI syntax

tacmd createsit
{-s|--situation} *SITNAME*
{-b|--basedOn} *BASENAME*
[{-p|--property|--properties} *NAME=VALUE* **]**

tacmd createSit {-i|--import} *FILENAME*

where:

-s|--situation
This is the name of the new situation, up to 32 letters, numbers, underscores (_). If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-b|--basedOn
Specifies the name of the situation on which you will be basing the new situation. The new situation is identical to the base situation except for the name and any situation properties that are specified on the command line.

-p|--property|--properties
Specifies one or more *name=value* pairs that identify the properties of the new situation and their values. Valid property entries for *name* are:

Desc or Description
The descriptive of the situation, consisting of fewer than 64 characters. Input given as text enclosed between double quotation marks, such as:

```
-p Desc="Alerts user to save the work.."
```

Interval
Sampling interval. Input Given in format ddd/hhmmss within double quotation marks, such as:

```
-p Interval="000/001500"
```

To change the interval. The format is *ddd/hh:mm:ss* where *ddd* is the number of days, from 0 to 999; *hh* is the hour, from 0 to 23; mm is the minute, from 0 to 59; and *ss* is the second, from 0 to 59, except if *ddd, hh, mm* are 0, in which case it is 30 to 59.

Formula
The situation formula for the conditions to test. Input given within double quotation marks. Keywords are prefixed with *, such as:

```
-p Formula="*IF *VALUE Local_Time.Minutes *GT 31"
```

Use the **tacmd viewSit** command to see the format of the base situation formula:

```
"*IF CONDITION [*UNTIL (*SIT SITUATION | *TTL INTERVAL |
                        *SIT SITUATION *OR *TTL INTERVAL)]"
```

CONDITION can be one condition or a list of conditions, each separated by an *AND or *OR logical operator and, if needed, grouped in parentheses. The condition is: a *function*, an *attribute* or *situation*, a *comparison operator*, and a *value*. More specifically:

```
{*VALUE|*CHANGE|*PCTCHANGE|*MISSING|*SCAN|*STR|*DATE|*TIME|*AVG|
  *COUNT|*MAX|*MIN|*SUM}
ATTRIBUTE_GROUP.ATTRIBUTE
{*EQ|*NE|*GT|*GE|*LT|*LE}
VALUE

 or

*SIT SITUATION *EQ *TRUE
```

When using the *MISSING function with multiple values, separate each entry with a comma and enclose the list in parentheses, such as "*MISSING NT_Process.Process_Name *EQ (Notepad, System)".

The *STR function requires that you specify at what position in the string to begin looking for a match. For example, *IF *STR NT_Process.Process_Name *EQ 4,hos will find any process names that have "hos" as the fourth, fifth and sixth characters.

The comparison operators are *EQ for equal, *NE for not equal, *GT for greater than, *GE for greater than or equal, *LT for less than, and *LE for less than or equal.

Depending on the function specified, the VALUE can be a word, text enclosed in single quotation marks, a number, or time.

If you specify an interval for the UNTIL clause, use the syntax d:hh:mm:ss, such as *UNTIL (*TTL 5:01:00:00) for five days and one hour.

See the Formula Functions appendix of the *IBM Tivoli Monitoring User's Guide* for a complete description of each function and any restrictions or special syntax requirements. Sample formulas are:

```
"*IF *VALUE NT_Event_Log.Event_ID *EQ 529"

"*IF *VALUE NT_Process.%_Processor_Time *GE 65 *AND
     *VALUE NT_Process.Priority_Base *NE 0"

"*IF (*SIT NT_Memory_Pages_Sec *EQ *TRUE) *AND
     (*SIT NT_Percent_Processor_Time *EQ *TRUE)"
```

Note: You cannot nest double-quotation marks inside the double-quotation marks that surround the formula, although you can have one pair of single quotation marks within the double-quotation marks.

Distribution

The situation description. Input should be a valid managed system name or names, such as:

```
-p Distribution="Primary:HDCHASDSTC0219:NT,
Primary:HDCHASDSTC0420:NT"
```

For multiple managed systems, separate each with a comma (,). To distribute to all monitoring agents of a given type, specify that type (such as '*NT_SYSTEM', '*ALL_UNIX', and '*LINUX_SYSTEM').

Advice

Expert Advice for situation. Input given as text enclosed between double quotation marks, such as:

```
-p Advice="Save the work.."
```

Action

Action to be performed when the situation becomes true. Program name or command to be executed, given as text enclosed between quotation marks.

```
-p Action="net send HDCHASDSTC0219 Save ur Work.."
```

You might need to use the situation editor to test the proper values. Limitations: It is not possible to specify in the CLI that the action is to be taken on each item should the condition be true for more than one item, nor that the action should be executed at the managed system, nor that the action is to take place at each interval should the condition stay true over multiple intervals.

RunOnStart

Determines whether the situation will start running on the managed systems upon creation, after editing, and whenever the monitoring agent or the monitoring server is started, a 'Yes' setting; or whether it required a manual start, a 'No' setting.

```
-p RunOnStart=Yes
```

SitInfo

Holds the Tivoli Enterprise Console® EIF data. It is all or any one of the following separated by ';' The SitInfo parameters must be enclosed in double quotation marks.

- Sev=*severity*

  can take values Critical or Warning or Minor or Harmless or Unknown.
- TFWD=[Y|N]
- TDST=n1[,n2...,n5]

  TDST can take up to 5 valid Tivoli Enterprise Console destination server IDs. For finding valid server IDs use the tacmd listeventdest command.

```
-p SitInfo="SEV=Critical;TFWD=Y;TDST=100"
```

-i|--import

Specifies the situation definition to import.

CLI example

The command in this example creates a new situation called *Sit1* based on an existing situation *NT_Service_Error* with the run on startup option set to *no*.

```
tacmd createSit -s Sit1 -b NT_Service_Error -p runonstart=no
```

The command in this example creates a new situation called *LogSpaceLow* with a formula that tests the usage percentage attribute from the Windows OS Monitored Logs group. A 90% or higher capacity causes an event to open and the advice given is to clear the log.

```
tacmd createSit -s LogSpaceLow  -b NT_Log_Space_Low
                -p formula="*IF *VALUE NT_Monitored_Logs_Rep *GE 90"
                Advice="Clear log." runonstart=yes
```

The command in this example creates a new situation called *DiskSpaceLow* with a Critical severity, TEC event forwarding, and the Tivoli Enterprise Console destination.

```
tacmd createSit -s new_test_Sit  -b NT_Disk_Space_Low
                -p SitInfo="SEV=Critical;TFWD=Y;TDST=1,2"
```

The command in this example creates a new situation called *ServiceError* with a Critical severity, TEC event forwarding, and the Tivoli Enterprise Console destination. An elapsed time of more than 31 minutes causes an event to open and the advice given is to save your work.

```
tacmd createsit -s SaveWork -b NT_Service_Error -p Desc="Alerts User to save.."
Formula="*IF *VALUE Local_Time.Minutes *GT 31" Advice="Please save your work..."
Interval="000/001500" Distribution="Primary:HDCHASDSTC0219:NT"
Action="net send HDCHASDSTC0219 Please Save your Work.." RunOnStart=Yes
SitInfo="SEV=Critical;TFWD=Y;TDST=100"
```

Return values

See "Return codes" on page 272

Related commands

"tacmd deleteSit" on page 76

"tacmd editSit" on page 98

"tacmd listSit" on page 184

"tacmd viewDepot" on page 243

"tacmd viewSit" on page 247

Return to Table 1 on page 5.

tacmd createSitAssociation

Description

Use the **tacmd createSitAssociation** command to create one or more situation associations for a Tivoli Enterprise Portal navigator item.

Optionally, you can also create one or more managed system or managed system list assignments for the navigator item. If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd createSitAssociation** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **createSitAssociation** command.

CLI syntax

tacmd createSitAssociation
[{-i|--situation|--situations} ***SITUATION...*****]**
{-a|--navItem} ***NAVIGATOR_ITEM***

[{-n|--navigator} *NAVIGATOR_NAME*]
[{-t|--state} *SITUATION_SEVERITY*]
[{-m|--system|--systems} *MSN_OR_MSL...*]
[{-s|--server} *TEPS_HOSTNAME[:PORT]*]
[{-u|--username} *TEPS_USER*]
[{-p|--password} *TEPS_PASSWORD*]
[{-f|--force}]

where:

-i|--situation|--situations
The name of the situation or situations to associate to the navigator item.

-a|--navItem
The fully qualified name of the navigator item to associate the situation or situations to, and optionally to assign the managed systems or managed system lists to. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n|--navigator
The name of the navigator view that the navigator item belongs to. By default, the Physical navigator view is used.

-t|--state
The state is indicated when the conditions have been met and the situation becomes true. An event indicator for the state overlays the Navigator item icon. By default, the Critical state is used.

-m|--system|--systems
The name of one or more managed systems or managed system lists to assign to the navigator item.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Situation' object enabled on the server to execute the **createSitAssociation** command. If the -m|--system option is provided, the user must also have the 'Modify' permission enabled for the Custom Navigator Views object.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f|--force
Performs the action without prompting for confirmation.

CLI example

The following example associates the UNIX_System_Busy_Critical situation to the Enterprise/child_logical navigator item that belongs to the Logical navigator:

```
tacmd createsitassociation -a Enterprise/child_logical -n Logical
-i UNIX_System_Busy_Critical
```

Return values

See “Return codes” on page 272

Related commands

“tacmd deleteSitAssociation” on page 77

“tacmd exportSitAssociations” on page 120

“tacmd importSitAssociations” on page 162

“tacmd listSitAssociations” on page 186

Return to Table 1 on page 5.

tacmd createSysAssignment

Description

Use the **tacmd createSysAssignment** command to assign one or more managed systems or managed system lists to a Tivoli Enterprise Portal navigator item. The command verifies that the system exists in the target Tivoli Monitoring environment. If the system is not in the the target Tivoli Monitoring environment, the command fails. Contrast this logic with that of the **tacmd createSitAssociation** command that you use to create one or more situation associations.

For the **tacmd createSysAssignment** command, a "system assignment" is a logical relation between a system and a navigator item that is used as the event indicator for situations. If you have no managed systems assigned to this navigator item, no events are displayed for it unless they are part of a roll-up display of events. In addition, the Situation editor will not be available from the menu. As a result, do not create a "system assignment" if the specified managed system does not exist on the target.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd createSysAssignment** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **createSysAssignment** command.

CLI syntax

tacmd createSysAssignment
{-a|--navItem} *NAVIGATOR_ITEM*
{-m|--system|--systems} *MSN_OR_MSL...*
{-n|--navigator} *NAVIGATOR_NAME*

[{-s|--server} *TEPS_HOSTNAME[:PORT]*]
[{-u|--username} *TEPS_USER*]
[{-p|--password} *TEPS_PASSWORD*]
[{-f|--force}]

where:

-a|--navItem
The fully qualified name of the navigator item to assign the managed systems or managed system lists to. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-m|--system|--systems
The name of one or more managed systems or managed system lists to assign to the navigator item.

-n|--navigator
The name of the navigator view that the navigator item belongs to.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f|--force
Performs the action without prompting for confirmation.

CLI example

The following example assigns the nc117242:KUX managed system to the Enterprise/child_logical navigator item that belongs to the Logical navigator:

```
tacmd createsysassignment -a Enterprise/child_logical -n Logical -m nc117242:KUX
```

Return values

See “Return codes” on page 272

Related commands

“tacmd deleteSysAssignment” on page 78

“tacmd exportSysAssignments” on page 121

“tacmd importSysAssignments” on page 164

"tacmd listSysAssignments" on page 188

Return to Table 1 on page 5.

tacmd createsystemlist

Description

This command creates a new managed system group. You must log in by using the **login** command before running the **createsystemlist** command.

Note: The correct name to use in commands for the Unix Logs agent is "Unix Logs". "Monitoring agent for Unix Logs" has been superseded.

CLI syntax

tacmd createsystemlist
{-l|--list} *LISTNAME*
{-b|--basedOn} *BASELISTNAME*
[{-m|--system} *SYSTEM***]**

tacmd createsystemlist
{-l|--list} *LISTNAME*
[{-t|--type} *TYPE***]**
{-m|--system} *SYSTEM*

tacmd createsystemlist
{-i|--import} *FILENAME*

where:

-l|--list
Name of the new managed system group to be created. Specify a string of letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ) up to a maximum length of 32 characters.

-m|--system
Name or names of the managed systems. Specify a string of letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ). This option is required when specifying -t|--type and is optional when specifying -b|--basedOn.

-b|--basedOn
Name of the managed system group on which to base the new system list. The new system list is identical to the base system list except the name (LISTNAME) and any systems that are specifically changed. Specify a string of letters (upper or lower case), numbers, underscores (_), or asterisks (*). This option is mutually exclusive with -t|--type.

-i|--import
Import the system list definition. Specify the name of a readable file containing a valid system list definition.

-t|--type
The type of the new system list. Specify a string for the managed system type name or its associated 2-character code. The string might consist of

letters (upper or lower case), numbers, underscores (_), slashes (/), left parenthesis "(", right parenthesis ")", or spaces ( ). If not specified, a type of "All Managed Systems" is used. This option is mutually exclusive with -b|--basedOn.

CLI example

This example creates a system list testList1 on the server https://10.102.22.123:3661.

```
tacmd createsystemlist -l testList1 -t NT
                       -m Primary:HDCHASDSTC0420:NT HUB_HDCHASDSTC0420
```

Return values

See "Return codes" on page 272.

Related commands

"tacmd editsystemlist" on page 100

"tacmd deletesystemlist" on page 80

"tacmd viewsystemlist" on page 248

"tacmd listsystemlist" on page 190

Return to Table 1 on page 5.

tacmd createUser

Description

Use the **tacmd createUser** command to create a new user in the Tivoli Enterprise Portal Server. The user ID and password for Tivoli Enterprise Portal Server log in are required by this command. To create a new Tivoli Enterprise Portal user, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **createUser** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd createUser
{-i|--id} *NEW_USERID*
[{-u|--userid} *TEPS_USERID* **]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-b|--base} *BASEDON_USERID***]**
[{-s|--server} *TEPS_HOSTNAME***]**
[{-n|--name} *NAME***]**

[{-dn | --distname} *DISTINGUISHEDNAME*]
[{-d | --desc} *DESCRIPTION*]

where:

-i | --id
(required) Specifies the new User ID to be created. The User ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character.

-b | --base
Specifies the ID, based on which the new user has to be created. The new user is created with the same properties as that of the base user. If not specified, the new user is created based on the Default User. The base user ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character.

-u | --userid
Specifies an existing User ID to log on to Tivoli Enterprise Portal. The software prompts you for the User ID if you do not specify one.

-w | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname where the user has to be created. If not specified, the user is created in the local Tivoli Enterprise Portal Server.

-n | --name
Specifies the name of the user.

-dn | --dname
Specifies the distinguished name for the user.

-d | --desc
Specifies a description for the new user.

CLI example

This example will create user TESTUSER based on sysadmin on the server HDCHASDSTC0219.

```
   tacmd createUser –i TESTUSER –b sysadmin –u sysadmin –w "tivoli123"
–s HDCHASDSTC0219 –n sysadmin
–dn UID=TESTUSER,O=DEFAULTWIMITMBASEDREALM
–d administration
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd createUserGroup

Description

Use the **tacmd createUserGroup** command to create a new user group in the Tivoli Enterprise Portal. A group can have members and also be a member of another group as allowed by the Tivoli Enterprise Portal Server. The user ID and password for Tivoli Enterprise Portal Server log in are required by this command. To create a new group, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **createUserGroup** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd createUserGroup
{-g|--gid} *NEW_GROUPID*
[{-u|--userid} *TEPS_USERID* **]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-b|--base} *BASEDON_GROUP***]**
[{-s|--server} *TEPS_HOSTNAME***]**
[{-n|--name} *NAME***]**
[{-d|--desc} *DESCRIPTION***]**

where:

-g|--gid
Specifies the new Group ID to be created. The Group ID must not contain any blank spaces characters in it. Its maximum allowed length is 31.

-b|--base
Specifies the based-on group name. If specified, the new group will inherit the Permissions, Applications, Navigator Views and Member Of from the based-on group. If not specified the new group is created based on the Default User.

-u|--userid
Specifies an existing User ID to log on to Tivoli Enterprise Portal. The software prompts you for the User ID if you do not specify one.

-w|--password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname where the user has to be created. If not specified, the new group user is created in the local Tivoli Enterprise Portal Server.

-n|--name
Specifies the name for the group.

-d | --desc
Specifies a description for the new group.

CLI example

This example will create group *TESTGRP based on *ADMINISTRATOR on the server HDCHASDSTC0219.

```
   tacmd createUserGroup –g *TESTGRP –b *ADMINISTRATOR –u sysadmin
–w "tivoli123" –s HDCHASDSTC0219 –n ADMINISTRATOR –d
"test group with full permissions"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteAction

Description

Use the **tacmd deleteAction** command to delete a Take Action. You must log in by using the **tacmd login** command before running the **tacmd deleteAction** command.

CLI syntax

tacmd deleteAction
{-n | --name} *ACTIONNAME*
[{-t | --type} *TYPE*]
[{-d | --detailtextname} *TYPEDESC*]
[{-f | --force}]

where:

-n | --name
The name of the action to be deleted.

-t | --type
Application type name. Specify a two-digit character code of the system type name to delete the action.

-d | --detailtextname
Application detail text name. Specify detail text of system type name to delete the action.

-f | --force
Deletes the action without prompting user for confirmation.

CLI example

This example deletes the action named "Test Action" of type WE and which has type name "WebSphere Application Server" after user's confirmation.

```
   tacmd deleteAction -n "Test Action" –t we  –d  "WebSphere Application Server"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteappinstallrecs

Description

Use the **tacmd deleteappinstallrecs** command to restart a failed installation by deleting application support installation records on the server. By default, this command deletes installation records only if a monitoring server installation error has occurred. If you specify the **--allstates** option instead of the default, the installation records are deleted even if no error records exist. Note that the error records are re-inserted the next time the agent of that type registers with the monitoring server. The **tacmd deleteappinstallrecs** command does not remove the product support, just the indication of the product support. You must log in by using the **tacmd login** command before running the **tacmd deleteappinstallrecs** command.

CLI syntax

tacmd deleteappinstallrecs
{-t|--type} *TYPE*
{-v|--version} *VERSION*
[{-n|--temsname} *TEMS_NAME***]**
[{-a|--allstates}]
[{-e|--alltems}]
[{-f|--force}]

Note: Either the -n option or the -e option must be specified.

where:

-t|--type
Specifies the product code of the records to be deleted.

-v|--version
Specifies the product version of the records to be deleted.

-n|--temsname
Specifies the Tivoli Enterprise Monitoring Server name where the records are to be deleted.

-a|--allstates
Specifies deleting installation records even if there are no existing error records.

-e|--alltems
Specifies deleting installation records from all existing online monitoring servers where a monitoring server installation error has occurred.

-f|--force
Deletes the records without prompting for confirmation.

CLI example

The following example deletes the application support installation records from the Linux product code, 623 version, and HUB_PCRIDDU monitoring server:

```
tacmd deleteappinstallrecs -t LZ -v 06230000 -n HUB_PCRIDDU
```

Return values

See Table 8 on page 272.

Related commands

"tacmd listSdaOptions" on page 180

"tacmd editSdaOptions" on page 95

"tacmd deleteSdaOptions" on page 74

"tacmd listappinstallrecs" on page 168

Return to Table 1 on page 5.

tacmd deleteCalendarEntry

Description

Use the **tacmd deleteCalendarEntry** command to delete an existing calendar entry on the Tivoli Enterprise Monitoring Server. You must log in by using the **login** command before running the **tacmd deleteCalendarEntry** command.

CLI syntax

tacmd deleteCalendarEntry
{-n | --name} *CALENDAR_ENTRY_NAME*
[{-f | --force}]

where:

-n | --name
Specifies the name of the calendar entry.

-f | --force
Deletes the calendar entry on the server without prompting for confirmation.

CLI example

The following example deletes the calendar entry Run_Bat:

```
tacmd deleteCalendarEntry -n "Run_Bat"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteEventDest

Description

Use the **tacmd deleteEventDest** command to delete an event destination server definition from the server.

CLI syntax

tacmd deleteEventDest
{-i|--id|--serverId} *ID*
[{-f|--force}]

where:

-i|--id|--serverID
Identifies the Server Destination Id of the event destination server definition to delete from the server. The value must be a value between 1 and 999, inclusive.

-f|--force
Deletes the event destination server definition from the server without prompting for confirmation.

CLI example

This example deletes the event destination 150 from the server:

```
tacmd deleteEventDest  -i 150
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deletegroup

Description

Use the **tacmd deletegroup** command to delete a specified group member from the Tivoli Enterprise Monitoring Server. You must log in by using the **login** command before running the **deletegroup** command.

CLI syntax

tacmd deletegroup
{-g|--group} *GROUPNAME*
{-t|--grouptype} *DEPLOY|BUNDLE|SITUATION|COLLECTION*
[-f|--force]

where:

-g|--group
Specifies the name of the group to be deleted.

-t | --grouptype
Specifies the type of the group to be deleted. Acceptable type names are DEPLOY, BUNDLE, SITUATION, COLLECTION. Note that the defined object group is also stopped.

-f | --force
Deletes the specified group without asking for confirmation.

CLI example

The following example deletes the deployment group "NewWindowsDeployGroup" from the server:

```
tacmd deleteGroup -g NewWindowsDeployGroup -t DEPLOY –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deletegroupmember

Description

Use the **tacmd deletegroupmember** command to delete a specified group member from the Tivoli Enterprise Monitoring Server. This command stops and undistributes the collections on the target system. You must log in by using the **login** command before running the **deletegroupmember** command.

CLI syntax

tacmd deletegroupmember
{-g | --group} *GROUPNAME*
{-m | --member}
{-t | --grouptype} *DEPLOY | BUNDLE | SITUATION | COLLECTION*
[-f | --force]

where:

-g | --group
Specifies the name of the group whose member has to be deleted.

-m | --member
Specifies the name of the member to be deleted.

-t | --grouptype
Specifies the type of the group member to be deleted. Acceptable type names are DEPLOY, BUNDLE, SITUATION, or COLLECTION.

-f | --force
Deletes the specified member without asking for confirmation.

CLI example

This example deletes the deployment member w099o002.tivlab.raleigh.ibm.com that belongs to the group NewWindowsDeployGroup:

```
tacmd deleteGroupMember -g NewWindowsDeployGroup -t DEPLOY
-m w099o002.tivlab.raleigh.ibm.com -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteOverride

Description

Use the **tacmd deleteOverride** command to delete the situation overrides defined for a specified situation on a managed system or list of managed systems.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **deleteOverride** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd deleteOverride
{-s|--situation} *SITNAME*
{-m|--system} *SYSTEM|SYSTEM_LIST*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-h|--tepshostname} *TEPS_HOSTNAME***]**
[{-f|--force}]

tacmd deleteOverride
{-s|--situation} *SITNAME*
{-m|--system} *SYSTEM|SYSTEM_LIST*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
{-p|--predicate} *PREDICATE ...*
[{-k|--key} *KEY_CONDITION ...***]**
[{-c|--calendarentry} *CALENDAR_ENTRY***]**
[{-t|--inlinecal} *INLINE_CAL_ENTRY***]**
[{-h|--tepshostname} *TEPS_HOSTNAME***]**
[{-f|--force}]

where:

-s|--situation
Specifies the situation to delete override definitions for. If you include either the & character or the < character in the situation name, you must use quotation marks around the name, for example, "abc&def" or "abc<def".

-m|--system
Specifies the name of the managed system or managed system group to delete override definitions for. Valid values include letters (upper or lower

case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-u | --userid
Specifies the existing User ID to log on to the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-c | --calendarentry
Specifies the name of the calendar entry of the override to delete.

-t | --inlinecal
Specifies the Hourly Schedule entry to remove. For the INLINE_CAL_ENTRY variable, use the [HHmm,HHmm] format, where HH is for hours in 00-23 notation and mm stands for minutes.

-p | --predicate
Specifies the situation formula predicate or predicates for the override to delete. All predicates for the override to delete must be entered. Predicates must be enclosed in double quotation marks and entered in the format "ATTRIBUTE OPERATOR VALUE" with spaces between ATTRIBUTE, OPERATOR, and VALUE. The predicate OPERATOR must be one of the following: "EQ", "NE", "GT", "LT", "GE", or "LE".

The attribute can be entered by using either the formula name or the display name for the attribute. Run the **tacmd listOverrides** command to view the defined overrides for the situation and managed system.

-k | --key
Specifies the key condition or key conditions for the override to delete. All conditions for the override to delete must be entered. Each key condition must be enclosed in double quotation marks and entered in the format "ATTRIBUTE VALUE" with spaces between ATTRIBUTE and VALUE. The key condition OPERATOR is restricted to the value "EQ".

The attribute can be entered by using either the formula name or the display name for the attribute. Run the **tacmd listOverrides** command to view the defined overrides for the situation and managed system.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

-f | --force
Deletes the specified member without asking for confirmation.

CLI example

This example deletes an override with an associated key condition and calendar entry:

```
tacmd deleteoverride -u sysadmin -w ******** -m Primary:LEVER:NT
-s NT_NotesServerProcess -c Weekend -p "% Processor Time GE 10"
-k "Binary Path EQ C:\Notes\NotesServer\nserver.exe"
```

This example deletes an override with no associated calendar entries or key conditions, by using the force option to suppress the confirmation prompt:

```
tacmd deleteoverride -u sysadmin -w ******** -m Primary:LEVER:NT
-s NT_NotesServerProcess -p "% Processor Time GE 20"
```

This example deletes all overrides for a managed system group:

```
tacmd deleteoverride -u sysadmin -w ******** -m *NT_SYSTEM -s NT_Disk_Space_Low
```

This example deletes an inline calendar entry for a managed system group:

```
tacmd deleteoverride -u sysadmin -w ******** -m *NT_SYSTEM -t 2201
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteSdaInstallOptions

Description

Use the **tacmd deleteSdaInstallOptions** command to delete a version from a Self-Describing Agent (SDA) install option record for a product type. You must log in by using the **tacmd login** command before running the **tacmd deleteSdaInstallOptions** command.

CLI syntax

tacmd deleteSdaInstallOptions
{-t|--type} DEFAULT
[{-f|--force}]

OR

tacmd deleteSdaInstallOptions
{ {-t|--type} ***PRODUCT_TYPE...***
[{-v|--version} ***VERSION*****]**
| {-a|--all } }
[{-f|--force}]

where:

{-t|--type} DEFAULT
Removes the DEFAULT SDA installation option for all products that do not have a specific SDA install option record defined for their product type. When the DEFAULT SDA install option is not defined, SDA installation support is disabled.

{-t|--type} *PRODUCT_TYPE...*
Specifies one or more managed system types (product codes) to update.

-v|--version
Specifies the VERSION to delete from the SDA install option record for the product code identified. Including a VERSION is optional and if not provided, all versions are deleted from the specified product. After all versions have been removed for a product, SDA installation support will be governed by the setting of the DEFAULT record. The version is an eight-digit identifier in the format *VVRRMMFF*, where *VV* specifies Version, *RR* specifies Release, *MM* specifies Modification, and *FF* specifies PTF Level. For example, the *VVRRMMFF* designation for ITM 623 FP2 is 06230200.

-a | --all
Deletes all SDA install options for all products; the DEFAULT install record is unaffected.

-f | --force
Specifies to delete the SDA install option without prompting for confirmation.

CLI example

Run the following command to disable SDA Installation support for the ITM623 FP2 Windows product type:

```
tacmd deleteSdaInstallOptions -t NT -v 06230200
```

Run the following command to disable SDA Installation support for the ITM623 FP2 Linux and Unix product types without confirmation:

```
tacmd deleteSdaInstallOptions --type LZ UX --version 06230200 --force
```

Run the following command to remove the DEFAULT SDA Installation option record:

Note: When no DEFAULT record exists, SDA installation will be disabled for all product types that do not have an SDA install option record defined.

```
tacmd deleteSdaInstallOptions -t DEFAULT
```

Run the following command to remove all versions from all product SDA Installation records:

Note: From this point, SDA Installation support for all products will be governed by the setting of the DEFAULT record which is unaffected by the command. If no DEFAULT record exists, SDA Installation support will be disabled for all products.

```
tacmd deleteSdaInstallOptions -a
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addSdaInstallOptions" on page 23

"tacmd editSdaInstallOptions" on page 93

"tacmd listSdaInstallOptions" on page 178

"tacmd listSdaOptions" on page 180

Return to Table 1 on page 5.

tacmd deleteSdaOptions

Description

Use the **tacmd deleteSdaOptions** command to delete Self-Describing Agent (SDA) option configuration entries.

The **tacmd deleteSdaOptions** command removes the configuration that controls how product definitions for the specified agent types are applied at the hub monitoring server. When the application support for a product is applied to the hub monitoring server, commonly called *seeding*, the definitions are added to the hub and these are automatically propagated to any active remote monitoring server. Configuration for the SDA seeding for a product type specifies how the distribution targets for definitions are applied. The option to control the seeding is provided to prevent prior customization from being lost. Removing seeding configuration will not prevent or disable the seeding process for a SDA install.

Note: You must log in using the **tacmd login** command before running the **deleteSdaOptions** command.

CLI syntax

tacmd deleteSdaOptions {–t|--type} *TYPE...*| {-a|–all} [-f|--force]

where:

-t|--type} *TYPE...*
Specifies one or more product codes to delete.

-a|--all
Delete all SDA seeding configuration records.

-f|--force
Delete the configuration options without confirmation.

Return values

See “Return codes” on page 272

Related commands

“tacmd listSdaOptions” on page 180

“tacmd editSdaOptions” on page 95

“tacmd listappinstallrecs” on page 168

“tacmd deleteappinstallrecs” on page 67

Return to Table 1 on page 5.

tacmd deleteSdaSuspend

Description

Use the **tacmd deleteSdaSuspend** command to delete the Self-Describing Agent (SDA) Suspend record from the database.

CAUTION:
Do not use the deleteSdaSuspend command unless directed by IBM Software Support. Instead, use the suspendSda and resumeSda commands for administering SDA activity.

You must log in by using the **login** command before running the **deleteSdaSuspend** command.

CLI syntax

tacmd deleteSdaSuspend
[{-f | --force }]

where:

-f | --force
Delete the Self-Describing Agent (SDA) Suspend record from the database without prompting for confirmation first.

CLI example

This command deletes the Self-Describing Agent (SDA) Suspend record from the database without prompting for confirmation first:

```
tacmd deleteSdaSuspend -f
```

Return values

See Table 8 on page 272.

Related commands

"tacmd resumeSda" on page 216

"tacmd suspendSda" on page 236

Return to Table 1 on page 5.

tacmd deleteSit

Description

Use the **tacmd deleteSit** command to delete a situation from your environment.

Note: You cannot use this command to delete UADVISOR situations.

CLI syntax

tacmd deleteSit
{–s | --situation} ***SITNAME*** **[{-f | --force}]**

where:

-s | --situation
Specifies the name of the situation to delete. If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-f | --force
Disables the message that asks if you are sure you want to delete the situation.

CLI example

The command in this example deletes the situation named *My_Situation* without asking the user to confirm.

```
tacmd deleteSit -s My_Situation -f
```

Return values

See "Return codes" on page 272

Related commands

"tacmd createSit" on page 54

"tacmd editSit" on page 98

"tacmd listSit" on page 184

"tacmd viewSit" on page 247

Return to Table 1 on page 5.

tacmd deleteSitAssociation

Description

Use the **tacmd deleteSitAssociation** command to dissociate one or more situations from a Tivoli Enterprise Portal navigator item.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd deleteSitAssociation** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **deleteSitAssociation** command.

CLI syntax

tacmd deleteSitAssociation
{-a|--navItem} *NAVIGATOR_ITEM*
[{-n|--navigator} *NAVIGATOR_NAME* **]**
[{-i|--situation|--situations} *SITUATION...***]**
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**
[{-f|--force} **]**

where:

-a|--navItem
The fully qualified name of the navigator item from which to dissociate the situation or situations. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put

double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n | --navigator
The name of the navigator view that the navigator item belongs to. By default, the Physical navigator view is used.

-i | --situation | --situations
The name of the situation or situations to dissociate from the navigator item.

-s | --server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Situation' object enabled on the server to execute the **deleteSitAssociation** command.

-p | --password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f | --force
Performs the action without prompting for confirmation.

CLI example

The following example disassociates situations from the Enterprise/child_logical navigator item that belongs to the Logical navigator:

```
tacmd deletesitassociation -a Enterprise/child_logical -n Logical
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSitAssociation” on page 58

“tacmd exportSitAssociations” on page 120

“tacmd importSitAssociations” on page 162

“tacmd listSitAssociations” on page 186

Return to Table 1 on page 5.

tacmd deleteSysAssignment

Description

Use the **tacmd deleteSysAssignment** command to delete one or more managed system assignments from a Tivoli Enterprise Portal navigator item.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd deleteSysAssignment** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **deleteSysAssignment** command.

CLI syntax

tacmd deleteSysAssignment
{-a|--navItem} *NAVIGATOR_ITEM*
[{-m|--system|--systems} *MSN_OR_MSL* **]**
{-n|--navigator} *NAVIGATOR_NAME*
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**
[{-f|--force}]

where:

-a|--navItem
The fully qualified name of the navigator item whose managed system assignments will be deleted. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-m|--system|--systems
The name of one or more managed systems or managed system lists whose assignments will be deleted from the navigator item. If this option is not provided, all managed system assignments for the navigator item will be deleted.

-n|--navigator
The name of the navigator view that the navigator item belongs to.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f|--force
Performs the action without prompting for confirmation.

CLI example

The following example deletes a managed system assignment from the Enterprise/child_logical navigator item that belongs to the Logical navigator:

```
tacmd deletesysassignment -a Enterprise/child_logical  -n Logical
```

Return values

See "Return codes" on page 272

Related commands

"tacmd createSysAssignment" on page 60

"tacmd exportSysAssignments" on page 121

"tacmd importSysAssignments" on page 164

"tacmd listSysAssignments" on page 188

Return to Table 1 on page 5.

tacmd deletesystemlist

Description

This command deletes the specified managed system group.

CLI syntax

tacmd deletesystemlist
{-l|--list} LIST [{-f|--force}]

where:

-l|--list
Name of the managed system group to be deleted. Specify a string of letters (upper or lower case), numbers, or underscores (_) up to a maximum length of 32 characters.

-f|--force
Do not confirm with the user the managed system group to be deleted. If not specified, the user is prompted for confirmation.

CLI example

This example deletes the managed system group testList1 from server https://10.102.22.123:3661 after prompting the user.

```
tacmd deletesystemlist –l testList1
```

Return values

See "Return codes" on page 272.

Related commands

"tacmd createsystemlist" on page 62

"tacmd editsystemlist" on page 100

"tacmd viewsystemlist" on page 248

"tacmd listsystemlist" on page 190

Return to Table 1 on page 5.

tacmd deleteUser

Description

Use the **tacmd deleteUser** command to delete the existing user from Tivoli Enterprise Portal Server To delete a Tivoli Enterprise Portal user, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **deleteUser** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd deleteUser
{-i | --id} *USERID*
[{-u | --userid} *TEPS_USERID* **]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOSTNAME***]**
[{-f | --force}]

where:

-i | --id
Specifies the User ID to be deleted. This is a mandatory option. The User ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character.

-u | --userid
Specifies an existing User ID to log in to Tivoli Enterprise Portal. The software prompts you for the User ID if you do not specify one.

-w | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname, from where the user has to be deleted. If not specified, the given user is deleted from the local Tivoli Enterprise Portal Server.

{-f | --force}
Deletes the specified user ID from Tivoli Enterprise Portal Server without any confirmation from the user.

CLI example

This example deletes user TESTUSER from the server HDCHASDSTC0219.

```
tacmd deleteUser –i TESTUSER –u sysadmin –w "tivoli123" –s HDCHASDSTC0219 –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteUserGroup

Description

Use the **tacmd deleteUserGroup** command to delete the existing user group from the Tivoli Enterprise Portal Server. To delete a Tivoli Enterprise Portal group, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **deleteUserGroup** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd deleteUserGroup {-g | --gid} *GROUPID*
[{-u | --userid} *TEPS_USERID* **]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOSTNAME***]**
[{-f | --force}]

where:

-g | --gid
Specifies the new Group ID to be created. The Group ID must not contain any blank spaces characters in it. Its maximum allowed length is 32 characters, and it must begin with "_" or " * ".

-u | --userid
Specifies an existing User ID to log on to the Tivoli Enterprise Portal. The software prompts you for the User ID if you do not specify one.

-w | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname from where the group has to be deleted. If not specified, the given group ID is deleted from the local Tivoli Enterprise Portal Server.

{-f | --force}
Deletes the specified group ID from Tivoli Enterprise Portal Server without any confirmation from the user.

CLI example

This example deletes the group *TESTGRP from the server HDCHASDSTC0219.

```
   tacmd deleteUserGroup –g *TESTGRP –u sysadmin –w "tivoli123"
–s HDCHASDSTC0219 –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd deleteWorkspace

Description

Use the **tacmd deleteWorkspace** command to delete a global or user-customized Tivoli Enterprise Portal workspace from the Tivoli Enterprise Portal Server.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **deleteWorkspace** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd deleteWorkspace
{-w | --workspace} *WORKSPACE* | **{-i | --objectid}** *OBJECT_ID* **}**
[{-r | --workspaceUser} *USERID* **]**
[{-t | --type} *TYPE* **]**
[{-o | --deletereadonly}]
[{-f | --force}]
[{-u | --username} *TEPS_USER* **]**
[{-p | --password} *TEPS_PASSWORD* **]**
[{-s | --server} *TEPS_HOSTNAME[:PORT]* **]**

where:

-w | --workspace
Specifies the name of the workspace to delete.

-i | --objectid
Specifies the object identifier of the workspace to delete. This option cannot

be used with the -w|--workspace option. You can retrieve the workspace object identifier by running the **listworkspaces** command with the -i|--objectid option.

r|--workspaceUser
Specifies the Tivoli Enterprise Portal User ID for the workspace that is to be deleted. If this option is not provided, the global workspace will be deleted. If you find that you cannot delete a workspace due to issues with your user credentials, use the -r option.

-t|--type
An IBM Tivoli Monitoring application type. If a 2-character type is entered, the letter 'k' will be prepended automatically to form a 3-character product code.

-o|--deletereadonly
Deletes the workspace even if the workspace was created or saved with the 'Do not allow modifications' option.

-u|--username
Specifies the identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have both 'Workspace Administration Mode' and 'Workspace Author Mode' Workspace Administrator permissions enabled on the server to run the **deleteWorkspace** command. The 'Workspace Administration Mode' permission is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p|--password
Specifies the password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

{-f|--force}
Deletes the workspace without confirmation.

CLI example

The following example deletes the workspace with the object identifier of klz.System_Information_621:

```
tacmd deleteworkspace  -i klz.System_Information_621
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd describeSystemType

Description

Use the **tacmd describeSystemType** command to display the configuration options that are available to use with the **configureSystem** or **addSystem** commands. If no

version is supplied, configuration options for the latest version are displayed. This command is also available for non-agent bundles.

This command can only be run from a Tivoli Enterprise Monitoring Server containing a depot.

CLI syntax

tacmd describeSystemType
{-t|--type} *TYPE*
{-p|--platform} *PLATFORM*
[{-v|--version} *VERSION*]

where:

-t|--type
The product code for the agent that you want to describe.

-p|--platform
The platform code or codes of the agents to describe. This value corresponds to the value that is displayed in the Host Type field as a result of executing the viewDepot or listBundles command. For example, "-p sol826" shows valid configuration options for the 'sol286', 'sol296' and 'sol506' operating system types. Use only architectures listed as Host Types as seen in the listBundles or viewDepot output.

-v|--version
The version of the agent to describe.

CLI example

This command displays the configuration options that are available to use with the **configureSystem** or **addSystem** commands for the Universal Agent (type *UM*) for the Windows platform *WINNT*, version *060100000*.

```
tacmd describeSystemType –t UM -p WINNT -v 060100000
```

This command displays the configuration options that are available to use with the **configureSystem** or **addSystem** commands for the latest version of the Universal Agent (type *UM*) for the Windows platform *WINNT*.

```
tacmd describeSystemType –t UM -p WINNT
```

Return values

See Table 8 on page 272.

Related commands

“tacmd configureSystem” on page 42

“tacmd addSystem” on page 24

Return to Table 1 on page 5.

tacmd editAction

Description

Use the **tacmd editAction** command to edit a Take Action. You must log in by using the **tacmd login** command before running the **tacmd editAction** command.

CLI syntax

tacmd editAction
{-n|--name} *ACTIONNAME*
[{-p|--property|--properties} *NAME=VALUE* **]**
[{-t|--type} *TYPE***]**
[{-d|--detailtextname} *TYPEDESC* **]**
[{-f|--force} **]**

where:

-n|--name
The name of the action to be edited.

-p|--property|--properties
Specifies one or more NAME=VALUE pairs that identify the properties of the action and their values. Valid property entries for *name* are:

Desc or Description
The description of the Take Action command to be created. Input given as text enclosed between double quotation marks, such as:

```
-p Desc="Stops the specified services"
```

Cmd or Command
The system command to be executed.

-t|--type
Specifies the application two-digit code of the action to be edited.

-d|--detailtextname
Specifies the application type Name of the action to be edited.

-f|--force
Edits the action without prompting for confirmation.

CLI example

This example edits the command and description of the action named "Test Alerter Service" without prompting for confirmation.

```
tacmd editAction -n "Test Alerter Service" -p cmd="net stop Alerter"
desc="To stop the alerter service" –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd editCalendarEntry

Description

Use the **tacmd editCalendarEntry** command to edit an existing calendar entry on the Tivoli Enterprise Monitoring Server. The data for the calendar entries are to be given in CRON format. This has to be given as a quintuple value separated by space within double quotation marks if specified by using the -c | --cron option. The rules for –c | --cron are the same as those for the **tacmd addCalendarEntry** command. The previous value of cron is replaced by the new value. There is no merging operation. The rule for the second syntax is that at least one of the -i, -h, -a, -m, or -w options is required, and must be provided. The value of the CRON specification is taken from the server and is replaced by the new value provided for these options, -i, -h, -a, -m, and -w. The values that are not specified are considered as asterisk (*), meaning every min, hour, and so on.

You must log in by using the **login** command before running the **tacmd editCalendarEntry** command.

CLI syntax

tacmd editCalendarEntry
{-n | --name} *CALENDAR_ENTRY_NAME*
{-c | --cron} *CRON_SPEC*
[{-d | --description} *DESCRIPTION* **]**
[{-f | --force}]

tacmd editCalendarEntry
{-n | --name} *CALENDAR_ENTRY_NAME*
[{-i | --min} *MIN* **]**
[{-h | --hour} *HOUR* **]**
[{-a | --daym | --dayOfMonth} *DAY_OF_MONTH* **]**
[{-m | --month} *MONTH* **]**
[{-w | --dayw | --dayOfWeek} *DAY_OF_WEEK* **]**
[{-d | --description} *DESCRIPTION* **]**
[{-f | --force}]

where:

-n | --name
Specifies the name of the calendar entry.

-c | --cron
Specifies the CRON specification of the calendar entry.

-d | --description
Specifies the description of the calendar entry.

-i | --min
Specifies the minute value of the CRON specification.

-h | --hour
Specifies the hour value of the CRON specification.

-a | --daym | --dayOfMonth
Specifies the day of the month value of the CRON specification.

-m | --month
Specifies the month value of the CRON specification.

-w | --dayw | --dayOfWeek
Specifies the day of the week value of the CRON specification.

-f | --force
Modifies the specified calendar entry on the server without prompting for confirmation.

CLI example

The following example forces a time change to the Run_Bat calendar entry:

```
tacmd editCalendarEntry -n "Run_Bat" -c "30 17 * * 1-5" -d "Changed the time" -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd editEventDest

Description

Use the **tacmd editEventDest** command to modify an existing event destination server definition on the server.

CLI syntax

tacmd editEventDest
{-i | --id | --serverID} ID
{-p | --property | --properties} NAME=VALUE...
[{-f | --force}]

where:

-i | --id | –serverID
Identifies the Server Destination Id of the event destination server definition to modify on the server. The value must be a value between 1 and 999.

-f | --force
Modifies the event destination server definition on the server without prompting for confirmation.

-p | --property | --properties
Specifies one or more NAME=VALUE pairs that identify the properties and values to modify.

Host properties should be specified in the format:

```
HOST{1|2|3|4|5|6|7|8}=HOSTNAME[:PORT]
```

Host entries must be defined such that they are sequential in existence; for example, you cannot specify the HOST3 property if HOST2 is not also specified. If a port value is not provided for a host entry, the port will default to 0.

A maximum of 5 default servers are allowed. To designate this event destination server as a default server, specify the DEFAULT|DEFAULTSERVER property with a value of Y.

To delete the value for a property, specify the property in the format NAME=. The NAME|SERVERNAME, TYPE|SERVERTYPE, and HOST1 properties cannot be deleted. When you delete a HOST property, all subsequent HOST properties are shifted down, such that if you delete HOST3, HOST3 will assume the value for HOST4, HOST4 will assume the value for HOST5, and so on.

The property TYPE indicates whether the event destination server is a TEC server or OMNIbus server. Although this indication alters the internal behavior of the code, it is transparent to the user.

The following property names are valid:

- DESC|DESCRIPTION
- NAME|SERVERNAME
- TYPE|SERVERTYPE
- DEFAULT|DEFAULTSERVER
- HOST1
- HOST2
- HOST3
- HOST4
- HOST5
- HOST6
- HOST7
- HOST8

CLI example

This example modifies an existing event destination server definition 150 on the server, changes the TYPE from 'T' (TEC) to 'M' (Micromuse/Omnibus), adds a new HOST entry and also modifies the description:

```
tacmd editEventDest  -i 150  -p  TYPE=M HOST2=HDCHASDSTC0816:5529
DESCRIPTION="Local OMNI server"
```

These commands set multiple default event servers.

```
tacmd createEventDest -i 123 -p host1=bigTECserver:4567 default=Y name=myTEC
tacmd createEventDest -i 124 -p host1=bigTECserver1:4577 default=Y name=myTEC1
```

or with the **tacmd editEventDest** command, you can set DEFAULT=Y for existing event servers.

To send a situation to all three default event servers (the two that are defined and the basic one), specify an empty destination for the situation, as depicted in the following example.

```
C:\ODI>tacmd viewsit -s test_tec1
Name                  : test_tec1
Full Name             :
Description           :
Type                  : Windows OS
Formula               : *IF *VALUE NT_Cache.Copy_Read_Hits_% *EQ 1
Sampling Interval     : 0/0:15:0
Run At Start Up       : Yes
Distribution          :
```

```
Text                    :
Action Location         : Agent
Action Selection        : System Command
System Command          : *NONE
True For Multiple Items: Action on First Item only
TEC Severity            : Critical
TEC Forwarding          : Y
TEC Destination         :
```

Note: The TEC Destination field is empty, but TEC Forwarding is set to **Y**. In this example, the situation is sent to the *<default receiver>*, myTEC and myTEC1 event servers. In the TEC interface, only *<Default EIF Receiver>* displays in the left column (Assigned EIF Receivers), while myTEC and myTEC1 display in the right column (Available EIF Receivers), despite being set as DEFAULT servers. This is a known Tivoli Enterprise Portal limitation.

To change a situation from a specified TEC destination to an unspecified (empty) TEC destination, perform the following steps:

1. Export the situation to an XML file:
   ```
   C:\ODI>tacmd viewsit -s test_tec1 -e c:\test_tec1.xml
   ```
2. Edit the XML file and change
   ```
   <SITINFO>
   <![CDATA[SEV=Critical;TFWD=Y;~;"]] >
   </SITINFO>
   ```
   accordingly, to specify a null destination server.
3. Delete the original situation and import the situation back to the server:
   ```
   tacmd createsit -i c:\test_tec1.xml
   ```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd editGroup

Description

Use the **tacmd editGroup** command to edit a group definition.

CLI syntax

tacmd editGroup
{-g|--group} *GROUP_NAME*
{-t|--grouptype} *DEPLOY|BUNDLE|SITUATION|COLLECTION*
{-p|--property|--properties} *PROPERTY...*
[-d|--description] *NEWDESCRIPTION*
[-a|--add] *MANAGED_SYSTEM_NAME* **|***MANAGED_SYSTEM_LIST*
[-r|--remove] *MANAGED_SYSTEM_NAME* **|** *MANAGED_SYSTEM_LIST*
[-f|--force]

where:

-g | --group
Specifies the group name to be edited.

-t | --grouptype
Specifies the group type to be edited. Acceptable group types are DEPLOY, BUNDLE, SITUATION, COLLECTION.

-p | --property | --properties
Specifies one or more NAME=VALUE or SECTION.NAME=VALUE pairs that identify the configuration property to be changed for the group.

-d | --description
Specifies description of the group.

-a | --add
Assigns the specified managed systems or managed system groups to the distribution list of the situation or collection group. This option is valid only for situation or collection groups. Note that the distribution also starts the defined object group on the added managed systems and or managed system groups.

-r | --remove
Deletes the specified managed systems or managed system groups from the distribution for the situation or collection group. This option is valid only for situation or collection groups. Note that the defined object group is stopped on the removed managed systems and managed system groups.

-f | --force
Edits the specified group without prompting for confirmation from the user.

CLI example

The following example edits the property of the deployment group "NewWindowsDeployGroup" on the server:

```
tacmd editGroup -g NewWindowsDeployGroup –t DEPLOY –p KDY.password=1234
```

The following edits the distribution of a situation group. The managed systems Primary:test1:NT are added and the system list Primary:test2:NT is deleted from the situation group's distribution list:

```
tacmd editgroup -g new2 -t situation –a Primary:test1:NT -r Primary:test2:NT-f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd editgroupmember

Description

Use the **tacmd editgroupmember** command to edit a groupmember definition. Situation and collection groupmembers cannot be edited by using this command. You must log in by using the **login** command before running the **editgroupmember** command.

CLI syntax

Editing a bundle group member:

tacmd editgroupmember
{-g | --group} *BUNDLE_GROUP_NAME*
{-m | --member}*MEMBER_NAME*
{-t | --grouptype} *BUNDLE*
{-y | --productType} *PRODUCT_TYPE*
[{-v | --version} *VERSION***]**
[{-i | --platform} *VERSION***]**
[-p | --property | --properties *NAME=VALUE* **...]**
[{-f | --force}]

Editing a deployment group member:

tacmd editgroupmember
{-g | --group} *DEPLOY_GROUP_NAME*
{-m | --member} *MEMBER_NAME*
{-t | --grouptype} *DEPLOY*
{-p | --property | --properties} *NAME=VALUE* **...**
[{-f | --force}]

Editing multiple members by using member file:

tacmd editgroupmember
{-g | --group} *GROUP_NAME*
{-x | --file} *MEMBER_FILE*

where:

-g | --group
Specifies the group name whose member has to be edited.

-m | --member
Specifies the member name to be edited.

-t | --grouptype
Group type name. Acceptable type names are DEPLOY or BUNDLE.

-v | --version
Specifies the version number of the deployment bundle being added as a bundle group member.

-y | --productType
Specifies the product type code. The product value corresponds to the value that is displayed in the Product Code field as a result of running the **viewDepot** or **listBundles** command.

-i | --platform
Specifies the platform code of the product. The platform value corresponds to the value that is displayed in the Host Type field as a result of running the **viewDepot** or **listBundles** command.

-p | --property | --properties
Specifies one or more *NAME=VALUE* or *SECTION.NAME=VALUE* pairs that identify the configuration property to be changed for the group member. See "Configuration options and properties" on page 252 for information on these options.

-x|--file
Specifies the file containing one or more group members to edit.

-f|--force
Edits the specified group without prompting for confirmation from the user.

CLI example

This example edits the property of deployment member w099o002.tivlab.raleigh.ibm.com belonging to the group NewWindowsDeployGroup:

```
tacmd editGroupMember -g NewWindowsDeployGroup -m w099o002.tivlab.raleigh.ibm.com
-p KDYRXA.installDir=c:\\IBM\\ITM KDYRXA.RXAusername=SYSADMIN
KDYRXA.RXApassword=****
```

This example edits the bundle member specified in the first column of the CSV file which belongs to the bundle group NewBundleGroup:

```
tacmd editGroupMember -g NewBundleGroup -x c:\bulk_bundle_list.csv
```

The CSV file's format includes the following: *member,type,cmdLine_options,properties*

In the example above, the `bulk_bundle_list.csv` CSV file contents are the following:

member	type	cmdLine_options	properties
unixBundle	BUNDLE	-y UX	KDYRXA.RXA protocol=rxec KDYRXA. RXAport=67
db2Bundle	BUNDLE	-y UD	-v 062000001
f50pa2d.tivlab. raleigh.ibm.com	BUNDLE	-y UD	INSTANCE=db2inst3
amssol19.tivlab. raleigh.ibm.com	BUNDLE	-y UM	UA.CONFIG ='file2.mdl'

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd editSdaInstallOptions

Description

Use the **tacmd editSdaInstallOptions** command to edit the version configured to be allowed for Self-Describing Agent (SDA) install for a product type. You must log in by using the **tacmd login** command before running the **tacmd editSdaInstallOptions** command.

CLI syntax

tacmd editSdaInstallOptions
{-t|--type} DEFAULT
{-i|--install} ON|OFF
[{-f|--force}]

OR

tacmd editSdaInstallOptions
{-t|--type} ***PRODUCT_TYPE...***
{-v|--version} ***EXISTING_VERSION/NEW_VERSION***
[{-f|--force}]

where:

{-t|--type} DEFAULT
Defines the DEFAULT SDA installation option for all products that do not have a specific version configured to be allowed for SDA install defined for their product type.

Note: Any product with a specific product version defined will only allow SDA installation for the allowed versions listed in the record. They will not use the DEFAULT SDA install option. When no DEFAULT SDA install option is defined, OFF is used.

{-t|--type} *PRODUCT_TYPE...*
Specifies one or more managed system types (product codes) to be updated. If a non-DEFAULT product code is specified, the VERSION command option must be specified and the INSTALL command option omitted. A list of product codes can be specified but cannot include the DEFAULT product code.

{-i|--install} ON|OFF
Required with the {-t|--type} DEFAULT option.

- ON enables SDA installation.
- OFF disables SDA installation.

-v|--version
Identifies an existing SDA install version and an update SDA install version. The VERSION option cannot be specified when the product type is DEFAULT. The input format is the existing version separated by forward slash (/) followed by the update version. The version is an eight-digit identifier in the format *VVRRMMFF*, where *VV* specifies Version, *RR* specifies Release, *MM* specifies Modification, and *FF* specifies PTF Level. For example, the *VVRRMMFF* designation for ITM 623 FP2 is 06230200.

-f|--force
Specifies to edit the SDA install option without prompting for confirmation.

CLI example

Run the following command to modify configuration for SDA installation support of Windows product type from version ITM623 FP1 to ITM623 FP2:

```
tacmd editSdaInstallOptions -t NT -v 06230100/06230200
```

Run the following command to modify configuration for SDA installation support of Windows product type from version ITM623 FP1 to ITM623 FP2, without prompting for user response:

```
tacmd editSdaInstallOptions -t NT -v 06230100/06230200 -f
```

Run the following command to modify configuration for SDA installation support of Linux and Unix product types from version ITM623 FP1 to ITM623 FP2:

```
tacmd editSdaInstallOptions -t LZ UX -v 06230100/06230200
```

Run the following command to modify or addconfiguration for the DEFAULT SDA installation configuration:

Note: The default SDA behavior is configured as ON.

```
tacmd editSdaInstallOptions -t DEFAULT -i ON
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addSdaInstallOptions" on page 23

"tacmd deleteSdaInstallOptions" on page 73

"tacmd listSdaInstallOptions" on page 178

"tacmd listSdaOptions" on page 180

Return to Table 1 on page 5.

tacmd editSdaOptions

Description

Use the **tacmd editSdaOptions** command to edit Self-Describing Agent (SDA) option configuration entries.

The **tacmd editSdaOptions** command removes the configuration that controls how product definitions for the specified agent types are applied at the hub monitoring server. When the application support for a product is applied to the hub monitoring server, commonly called *seeding*, the definitions are added to the hub and these are automatically propagated to any active remote monitoring server. Configuration for the SDA seeding for a product type specifies how the distribution targets for definitions are applied. The option to control the seeding is provided to prevent prior customization from being lost. Editing seeding configuration will not modify how the existing SDA was seeded.

Note: You must log in using the **tacmd login** command before running the **editSdaOptions** command.

CLI syntax

tacmd editSdaOptions {**-o**|**--options**} *KEY=VALUE...KEY=VALUE* | *KEY_GROUP=VALUE* {**-t**|**--type**} *TYPE...* [{**-f**|**--force**}]

where:

-o | --options
Specifies the *KEY=VALUE...KEY=VALUE* pairs to set, separated by a space.

KEY *INSTALL_SEED*
Valid values include NEW, NONE, and ALL. Specify NEW, NONE, or ALL to choose how distribution definitions are applied during the pristine install of the product identified.

- NEW: A configuration choice of NEW allows all product distribution definitions to be added to the TEMS during a Pristine Install.
- NONE: A configuration choice of NONE prevents any product distribution definitions from being added to the TEMS during a Pristine Install.
- ALL: A configuration choice of ALL allows all product distribution definitions to be added to the TEMS during a Pristine Install.

key *UPGRADE_SEED* **(for the Pristine Install)**
Valid values include NEW, NONE, and ALL. Specify NEW, NONE, or ALL to choose how distribution definitions are applied during the upgrade install of the product identified.

- NEW: A configuration choice of NEW prevents any product distribution definitions to be added to the TEMS during a Upgrade Install.
- NONE: A configuration choice of NONE prevents any product distribution definitions from being added to the TEMS during a Upgrade Install.
- ALL: A configuration choice of ALL allows all product distribution definitions to be added to the TEMS during a Upgrade Install.

KEY_GROUP *SEEDING*
Valid values include NEW, NONE, ALL, DISABLE, and ENABLE. Specify NEW, NONE, or ALL to choose how distribution definitions are applied during the upgrade install and pristine install of the product identified.

- NEW: A configuration choice of NEW is equivalent to specifying UPGRADE_SEED=NEW and INSTALL_SEED=NEW.
- NONE: A configuration choice of NONE is equivalent to specifying UPGRADE_SEED=NONE and INSTALL_SEED=NONE.
- ALL: A configuration choice of ALL is equivalent to specifying UPGRADE_SEED=ALL and INSTALL_SEED=ALL.
- DISABLE: A configuration choice of DISABLE prevents all product monitoring definitions from being added to the TEMS during an Upgrade or Pristine Install.

  Note: After you have disabled SEEDING for a TYPE using the -o SEEDING=DISABLE option, you cannot modify the INSTALL_SEED and UPGRADE_SEED keys until SEEDING is re-enabled for the TYPE using the -o SEEDING=ENABLE option.

- ENABLE: A configuration choice of ENABLE clears the DISABLED values and sets UPGRADE_SEED and INSTALL_SEED to INSTALL_SEED=ALL and UPGRADE_SEED=NEW.

-t|--type
Specifies one or more product types. If a product type does not have a seeding option configuration, the DEFAULT configuration is used.

-f|--force
Edit the SDA configuration options without prompting for confirmation.

If no configuration is provided, the SDA seeding uses default installation values. The default installation value for a pristine install is ALL. The default installation value for an upgrade installation is NEW. If only a single installation value is provided to an editSitOptions command, the value for the unspecified options is set to the described default installation value (i.e., ALL for pristine install, NEW for upgrade install).

CLI example

The command in the example edits configuration options for product type LZ.

```
tacmd editSdaOptions -t LZ -o INSTALL_SEED=NEW UPGRADE_SEED=NONE
```

The command in the example specifies a configuration for all product types that only the UPGRADE_SEED option is updated.

```
tacmd editSdaOptions -t DEFAULT -o UPGRADE_SEED=ALL
```

The command in the example disables seeding for product type NT.

```
tacmd editSdaOptions -t NT -o SEEDING=DISABLE
```

The command in the example restores seeding to the factory default for product type NT.

```
tacmd editSdaOptions -t NT -o SEEDING=ENABLE
```

The command in the example specifies that if R1 R2 R3 R4 R5 do not have entries, then new configuration options are created with factory default values.

```
tacmd editSdaOptions -t R1 R2 R3 R4 R5 -o SEEDING=ENABLE
```

Return values

See “Return codes” on page 272

Related commands

“tacmd listSdaOptions” on page 180

“tacmd deleteSdaOptions” on page 74

“tacmd listappinstallrecs” on page 168

“tacmd deleteappinstallrecs” on page 67

Return to Table 1 on page 5.

tacmd editSit

Description

Use the **tacmd editSit** command to edit a situation.

Note: You cannot use this command to edit UADVISOR situations.

CLI syntax

tacmd editsit
{-s|--situation} *SITNAME*
{-p|--property|--properties} *NAME=VALUE*
[-f|--force]

tacmd editSit
{-l|--local} *FILENAME*
{-p|--property|--properties} *NAME=VALUE*

where:

-s|--situation
Specifies the name of the situation to edit. If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-p|--property|--properties *NAME=VALUE*
Specifies one or more *NAME=VALUE* pairs that identify the properties of the modified situation and their values. Valid property names are:

Desc or Description
Description of the situation. Input given as text enclosed between double quotation marks, such as

```
-p Desc="Alerts user to save the work.."
```

Interval
Sampling Interval. Input Given in format ddd/hhmmss within double quotation marks, such as

```
-p Interval="000/001500"
```

Formula
Situation Formula. Input given within double quotation marks. Keywords are prefixed with *, such as:

```
-p Formula="*IF *VALUE Local_Time.Minutes *GT 31"
```

Distribution
Situation Distribution. Input should be a valid managed system name or names, such as:

```
-p Distribution="Primary:HDCHASDSTC0219:NT,
Primary:HDCHASDSTC0420:NT"
```

Advice
Expert Advice for situation. Input given as text enclosed between double quotation marks, such as:

```
-p Advice="Save the work.."
```

Action

Action to be performed when the situation becomes true. Program name or command to be executed, Given as text enclosed between quotation marks, such as:

```
-p Action="net send HDCHASDSTC0219 Save ur Work.."
```

RunOnStart

Specifies whether the situation has to be executed on start. Input Yes or No, such as:

```
-p RunOnStart=Yes
```

SitInfo

Holds the Tivoli Enterprise Console EIF data; a combination of SEV, TFWD, TDST separated by ";" The SitInfo parameters must be enclosed in double quotation marks. SEV can take values Critical or Warning or Minor or Harmless or Unknown. TFWD=Y or N. TDST can take up to 5 valid Tivoli Enterprise Console destination server IDs separated by ",", such as:

```
-p SitInfo="SEV=Critical;TFWD=Y;TDST=100"
```

For finding valid server IDs, use the **tacmd listeventdest** command.

-l|--local

Indicates the file name of the local situation definition to edit, so no changes are made to the situation definition on the monitoring server.

-f|--force

Edits the situation without confirmation.

CLI example

The command in the example edits the *No_Transactions* definition to not run at startup, which requires the situation to be started manually.

```
tacmd editSit -s No_Transactions -p RunOnStart=NO
```

The command in the example edits the *SaveWork* definition to run at startup.

```
tacmd editsit -s SaveWork -p Desc="Alerts User to save.."
Formula="*IF *VALUE Local_Time.Minutes *GT 31" Advice="Please save your work..."
Interval="000/001500" Distribution="Primary:HDCHASDSTC0219:NT"
Action="net send HDCHASDSTC0219 Please Save your Work.." RunOnStart=Yes
SitInfo="SEV=Critical;TFWD=Y;TDST=100"
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSit” on page 54

“tacmd deleteSit” on page 76

“tacmd listSit” on page 184

“tacmd viewSit” on page 247

Return to Table 1 on page 5.

tacmd editsystemlist

Description

This command is used to add or delete managed systems to or from an existing managed system group on the server. It can also be used to edit (add or delete system list names to/from) an existing managed system group in a file.

CLI syntax

tacmd editsystemlist {-l|--list} LIST
{[{-a|--add} SYSTEM ...] [{-d|--delete} SYSTEM ...]}
[{-f|--force}]

tacmd editsystemlist {-e|--edit} FILENAME
{[{-a|--add} SYSTEM ...] [{-d|--delete} SYSTEM ...]}

where:

-l|--list
Name of the managed system group to be edited. Specify a string of letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ) up to a maximum length of 32 characters.

-a|--add
Name or names of the managed systems to be added to the managed system group. Specify a string of letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ). Note that at least one of -a|--add or -d|--delete must be specified and both can be used in the same command invocation.

-d|--delete
Name or names of the managed systems to be deleted from the managed system group. Specify a string of letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ). Note that at least one of -a|--add or -d|--delete must be specified and both can be used in the same command invocation.

-e|--edit
Name of the managed system group file to be edited. Specify a valid file name consisting of letters (upper or lower case), numbers, underscores (_), colons (:), periods (.), slashes (/), back slashes (\), or tildes (~).

-f|--force
Do not confirm with the user the managed systems to be added or deleted. If not specified, you are prompted for confirmation.

CLI example

This example updates the managed system group testList1 on server https://10.102.22.123:3661.

```
tacmd editsystemlist -l testList1 -a Primary:HDCHASDSTC0422:NT -f
```

This example updates the managed system group definition file `sys200.xml` by both adding an entry and deleting entries.

```
tacmd editsystemlist -e sys200.xml
                     -a Primary:HDCHASDSTC0420:NT hdchasdstc0420ASFSdp:UAGENT00
                     -d HDCHASDSTC0420:Warehouse
```

Return values

See "Return codes" on page 272.

Related commands

"tacmd createsystemlist" on page 62

"tacmd deletesystemlist" on page 80

"tacmd viewsystemlist" on page 248

"tacmd listsystemlist" on page 190

Return to Table 1 on page 5.

tacmd editUser

Description

Use the **tacmd editUser** command to edit a user definition in the Tivoli Enterprise Portal. To edit the properties of a user, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

This command explicitly assigns navigators, so you must ensure that you have appended the new navigator to the list of navigators already assigned, or the other assignments will be lost. The Tivoli Enterprise Portal client does not receive a refresh event notification when assignments are made through the CLI, so a client restart is required to pick up the changes. The default navigator view is the first navigator in the list assigned to the "NavigatorViews" property. In the following example, "MyNavigator" would be the default navigator view when the Tivoli Enterprise Portal client is restarted.

```
tacmd edituser -u sysadmin -w mypassword -i sysadmin
-p NavigatorViews=MyNavigator,Physical
```

Note:

1. If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **editUser** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.
2. If you try to modify user permissions that are inherited from a group, you will receive an error message. Inherited group permissions cannot be modified.
3. When using this command to set two different permissions, it is best to use two separate commands; one for each permission setting.

CLI syntax

tacmd editUser

{-i | --id} *USERID*
{-p | --property | --properties} *NAME=VALUE* **...**
[{-u | --userid} *TEPS_USERID* **]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOSTNAME***]**
[{-n | --name} *NEWNAME***]**
[{-dn | --distname} *NEW_DISTINGUISHED_NAME***]**
[{-d | --desc} *NEWDESCRIPTION***]**
[{-f | --force}]

where:

-i | --id
Specifies the user ID for which users are to be listed. The User ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character.

-p | --property | --properties
Specifies one or more NAME=VALUE pairs that identify what information is to be changed for the user. The valid options for NAME are:

- Permissions
- Applications
- NavigatorViews
- MemberOf

The values for -p option must be provided within double quotation marks. For example:

```
-p "Permissions=Action.View=T" "Applications=Windows OS"
```

Commas are used as delimiters to specify more than one value for a property. For example:

```
-p "Permissions=Action.View=T,Policy.View=T"
"Applications=Windows OS, Linux OS"
```

You can use the **viewUser** command to identify the possible input values for Permissions, Applications, NavigatorViews, and MemberOf.

If you want to change a permission under a different realm (namespace) from the default realm (KFW), you must prefix the realm in the permission property, for example:

```
 -p Permissions=KCF.Configure.View=f
```

If no realm is specified, the default one is used (KFW). The **viewuser** command helps to identify the possible realms for Permissions.

-u | --userid
Specifies the existing user ID to log in to Tivoli Enterprise Portal Server. The software prompts you for the User ID if you do not specify one.

-w | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname from where the user details to be edited. If not specified, the user details are edited from the local Tivoli Enterprise Portal Server.

n | --name
Specifies the name of the user.

-d | --desc
Specifies description of the user.

-dn | --distname
Specifies the distinguished name of the user.

f | --force
Edits the specified user without prompting for confirmation from the user.

CLI example

This example edits permission, name, and description information for the user TESTUSER.

```
tacmd editUser –i TESTUSER –u sysadmin –w "tivoli123" –s HDCHASDSTC0219
–p "Permissions=Action.view=T, Action.modify=F" –n testuser –d testuser –f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd editUserGroup

Description

Use the **tacmd editUserGroup** command to edit a user group definition on the Tivoli Enterprise Portal Server. To edit the properties of a group, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **editUserGroup** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd editUserGroup
{-g | --gid} *GROUPID*
{-p | --property | --properties} *NAME=VALUE* **...**
[{-u | --userid} *TEPS_USERID* **]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOSTNAME***]**
[{-n | --name} *NEWNAME*
[{-d | --desc} *NEWDESCRIPTION***]**
[{-f | --force}]

where:

-g | --gid
(required) Specifies the new Group ID to be created. The Group ID must not contain any blank spaces characters in it. Its maximum allowed length is 32 characters, and it must begin with "_" or " * ".

-p | --property
Specifies one or more NAME=VALUE pairs that identify what information to be changed for the user group. The valid options for NAME are:

- Permissions
- Applications
- NavigatorViews
- MemberOf
- Members

The values for the -p option must be provided within double quotation marks. For example:

```
-p "Permissions=Action.View=T" "Applications=Windows OS"
```

Commas can be used as a delimiter to specify more than one value for a property. For example:

```
-p "Permissions=Action.View=T,Policy.View=T"
"Applications=Windows OS, Linux OS"
```

Use the **viewUserGroup** command to identify the possible input values for Permissions, Applications, NavigatorViews, MemberOf, and Members. If you want to change a permission under a different realm (namespace) from the default realm (KFW), you must prefix the realm in the permission property, for example:

```
 -p Permissions=KCF.Configure.View=f
```

If no realm is specified, the default one is used (KFW). The **viewuser** command helps to identify the possible realms for Permissions.

-u | --userid
Specifies an existing User ID to log in to the Tivoli Enterprise Portal Server. The software prompts you for the User ID if you do not specify one.

-w | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname from where the group details are to be edited. If not specified, the group details are edited from the local Tivoli Enterprise Portal Server.

-n | --name
Specifies the name of the group.

-d | --desc
Specifies the description of the group.

-f | --force
Edits the specified group without prompting for confirmation from the user.

CLI example

This example edits permission, name and description information for the group *TESTGRP.

```
   tacmd editUserGroup –g *TESTGRP –u sysadmin –w "tivoli123"
–s HDCHASDSTC0219 –n testgrp –d testgrp
 –p "Permissions=Action.view=T, Action.modify=F" –f
```

This command explicitly assigns navigators, so you must ensure that you have appended the new navigator to the list of navigators already assigned, or the other assignments will be lost. The Tivoli Enterprise Portal client does not receive a refresh event notification when assignments are made through the CLI, so a client restart is required to pick up the changes. The default navigator appears to be the first navigator in the list assigned to the "NavigatorViews" property. In the following example, "MyNavigator" would be the default navigator view when the Tivoli Enterprise Portal client is restarted.

```
tacmd editusergroup -u sysadmin -w mypassword -i sysadmin -p
NavigatorViews=MyNavigator,Physical
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd executeAction

Description

Use the **tacmd executeAction** command to execute the system command provided in the given Take Action command. You must log in by using the **tacmd login** command before running the **tacmd executeAction** command. If the following options are specified, certain Take Actions fail:

- -e | --stderr
- -o | --stdout
- -r | --returncode
- -l | --layout
- -p | --path

For more information, see 'The tacmd executeaction command fails for certain Take Actions' entry in the *IBM Tivoli Monitoring Troubleshooting Guide*.

Note: This command is intended for executing short CLI based actions/commands on the remote target. It should not be used to execute long running, GUI, or interactive executables.

CLI syntax

tacmd executeAction
{-n|--name} ***ACTIONNAME***
{-m|--system} ***MANAGEDSYSTEMNAME***
[{-t|--type} ***TYPE*****]**
[{-d|--detailtextname} ***TYPEDESC*** **]**
[{-o|--stdout}]
[{-e|--stderr}]
[{-r|--returncode}]
[{-l|--layout}]
[{-p|--path}]

[{-u|--timeout} ***TIMEOUT*****]**
[{-c|--cmdvalue} ***COMMANDVALUE*****]** ***VALUE=.....***

where:

-n|--name
Specifies the name of the action to be executed.

-m|--system
Specifies one or more managed systems on which the action is to be executed. For this command, the specified system cannot be a Tivoli Enterprise Portal Server managed system. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-t|--type
Specifies the application type name. Specify a two-digit character code of the system type name to execute the action.

-d|--detailtextname
Specifies the application detail text name. Specify detail text of system type name to execute the action.

Note: The -d|--detailtextname option should be used along with -t|--type option to filter the action having the same name and the same type.

-o|--stdout
Requests the standard output to be captured. For section separators, use the -l|--layout option. If the action command redirects standard output, then nothing will be captured.

To use this option, the hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. Otherwise, the option is ignored. This option is ignored if the specified system is an i5/OS or z/OS monitoring agent.

-e|--stderr
Requests the standard error to be captured. For section separators, use the -l|--layout option. If the action command redirects standard error, then nothing will be captured.

To use this option, the hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. Otherwise, the option is ignored. This option is ignored if the specified system is an i5/OS or z/OS monitoring agent.

-r|--returncode
Requests the return code to be captured. For section separators, use the -l|--layout option.

To use this option, the hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. Otherwise, the option is ignored. This option is ignored if the specified system is an i5/OS or z/OS monitoring agent.

-l|--layout
Requests the command string to be captured and adds section separators to the results file.

To use this option, the hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. Otherwise, the option is ignored. This option is ignored if the specified system is an i5/OS or z/OS monitoring agent.

-p|--path

Specifies the name of the directory where the results files are saved. You must specify one of the following options before the directory and results files are created, either when the directory name is specified or when the default directory is created:

- -o|--stdout
- -e|--stderr
- -r|--returncode

For section separators, use the -l|--layout option. If the specified directory does not exist, then it is created. If the -p|--path option is not specified, but if any of the following options is defined, then a default directory is created where the command issued:

- -o|--stdout
- -e|--stderr
- -r|--returncode

The default directory name has the following format:

`TACMD_CXA_`*`ActionName_Timestamp_RND8Char`*

In this directory, one file for each managed system is created by using the following format:

`TACMD_CXA_`*`ExecutedAction_ManagedSystemName_Timestamp_SEQChar`*`.log`

Where:

ManagedSystemName
Specifies the managed system name, and is filtered removing special characters such as colons (:) and substituting them with underscores (_).

ActionName
Specifies the name of the action.

ExecutedAction
Specifies the command run on the managed system, and is filtered removing special characters such as colons (:), and it is truncated to 64 characters.

Timestamp
Specifies the timestamp, and is given in the format YYYYMMDDHHMMSS.

RND8Char
A random string composed of eight random numbers.

SEQChar
A random string composed of sequential numbers starting from 0.

To use this option, the hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. Otherwise, the option is ignored. This option is ignored if the specified system is an i5/OS or z/OS monitoring agent.

-u|--timeout

Specifies the timeout value of the command request. The TIMEOUT is the number of seconds allowed for the specified command to complete. The range is 1 to 32400 seconds. The command is not stopped if the timeout limit is reached, but the output is not captured.

To use this option, the hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. Otherwise, the option is ignored. This option is ignored if the specified system is an i5/OS or z/OS monitoring agent.

-c|--cmdvalue

Specifies the input value for the command to execute. If a Take Action command value requires some input from the user to execute the command, use the -c|--cmdvalue option to specify those values. The command input value should be given within double quotation marks under the property name VALUE as value=INPUTCOMMANDVALUE. A comma is used as a delimiter to specify more than one input value. For example

```
value=inputvalue1,inputvalue2,inputvalue3
```

Certain input values are optional. You can provide a value for those inputs. You can also specify a comma for those that you do not want to provide a value. For example:

```
value=inputvalue1,,inputvalue3
```

The **viewaction** command helps to identify the input value. Checking for the number of ampersand characters under the column command tells you exactly the number of input values that have to be provided to execute the Take Action command.

CLI example

This example executes the action command named "URL Add" of type UM on the managed system LSC334ASFSdp:UAGENT00. –c provides the input values required by the action command to execute.

```
tacmd executeaction -n "URL Add" -t um -m LSC334ASFSdp:UAGENT00
       -c value=http://w3.ibm.com,w3,1000,105,object
```

This example executes the action command named "Sample Linux Kill Process" from a Windows environment to terminate running process 6383 on a Linux endpoint. This command also requests that the standard output, standard error, and return code be captured with the response file having section separators added, and the file saved in the `c:\temp` directory:

```
tacmd executeaction -m LINUX:LZ -n "Sample Linux Kill Process" -t lz -c value="6383"
-e -o -r -l -p c:\temp
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd executecommand

Description

Use the **tacmd executecommand** command to execute the system command provided in the given command.

The hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. If the Tivoli Enterprise Monitoring Agent component is at the IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later level, all the agents installed in the same `CANDLEHOME` directory at the endpoint are capable of handing this command. For this command, the specified system cannot be an i5/OS or z/OS monitoring agent.

Note: This command is intended for executing short CLI based actions/commands on the remote target. It should not be used to execute long running, GUI, or interactive executables.

File names

When either the results file name or the directory location that can be specified by using the -d|--destination, -s|--remotedestination, and -w|--workingdir options contain spaces, you must include double quotation marks around the results file name and directory location. For example, run the following command from a Windows system to list the files in the `C:\Program Files` directory on a Windows system, and copy the resulting result file to `C:\Documents and Settings\response.out` on the local machine where the command was issued:

```
tacmd executecommand -m Primary:WINDOWS:NT -c dir -w "c:\\Program Files" -o -l
-d "C:\\Documents and Settings\\response.out"
```

When working with file and directory names that have nonalphanumeric or special characters (for example, ! @ #, etc), the path and file references for the -s|--source, -d|--destination, and -w|--workingdir options must be surrounded by double quotation marks (" "). However, paths that include an at symbol (@) must be escaped with an at symbol (@). The path user@home@directory is escaped as follows:

```
user@@home@@directory
```

Variable substitution

You can run this command by using an environment variable for the result file name and directory location that can be specified by using the -d|--destination, -s|--remotedestination, and -w|--workingdir options. If used for the -s|--remotedestination or -w|--workingdir options, it is for the specified monitoring agent's managed system rather than the local environment where the command is issued. If used for the -d|--destination option, it is for the local environment where the command is issued.

The environment variable format is @NAME@. The following characters are valid as the first character of any name, or any subsequent character:

- _ (underscore)
- Lower case alphabetic letters
- Upper case alphabetic letters

The following characters are valid as any character in any name except the first:

- - (dash)
- The following numbers, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

The following example runs the command from a UNIX system to list the files in the `C:\IBM\ITM\tmaitm6\logs` directory on a Windows systems. In this example, *CANDLE_HOME* is `c:\IBM\ITM`:

```
./tacmd executecommand -m Primary:WINDOWS:NT -c "dir @CANDLE_HOME@\\tmaitm6\\logs"
-e -o -r -l -v
```

To use this command, the KT1_TEMS_SECURE configuration parameter must be set in the hub monitoring server's configuration file to specify that the hub monitoring server supports this command. After setting the environment variable, you must recycle the hub monitoring server. The default value is no. Specify Y or YES or y or yes if you want to use this command.

Hub server configured with non-default port number

The executecommand, getfile, and putfile commands fail if the HUB TEMS is configured with a non-default port number. You must set the environment variable KDE_TRANSPORT in the Windows command prompt or UNIX Shell before issuing these commands to configure the TACMD to use the non-default port number to connect to the hub monitoring server. See the "KDE_TRANSPORT Structure" section of the "Configuring IBM Tivoli Monitoring components" chapter in the *IBM Tivoli Monitoring: Installation and Setup Guide* for descriptions and examples.

Monitoring Agent's PATH environment variable

The monitoring agent's PATH environment variable might not have the same list of directories as the endpoint system PATH environment variable. As a result, specifying a system command for the -c|--commandstring option not located in the directories defined for the agent's PATH environment variable results in the command failing to execute. In this case, identify the absolute location of the system command, and then specify it along with the system command for the -c|--commandstring option.

For example, on AIX systems, the ping command is located in the `/etc` directory. However, a monitoring agent might not have the `/etc` directory in its PATH environment variable. Issuing the following ping command as shown fails:

```
./tacmd executecommand -m AIX:KUX -c "ping AIX" -o -e -r -l -v

  KUIEXC001I: Content of the response file
TACMD_EXC_AIX_KUX_ping_AIX_20100420093037_60245434.log is:
  ------Command-------
  ping AIX
  ------Command Result------
  127
  ------Standard Error-------
 /usr/bin/ksh: ping:  not found.
  -------Standard Output-------

  KUIEXC000I: Executecommand request was performed successfully.
The return value of the command run on the remote systems is 127
```

However, if the fully qualified path for the ping command binary is specified, the command completes successfully as shown:

```
./tacmd executecommand -m AIX:KUX -c "/etc/ping -c 1 AIX"
-o -e -r -l -v

KUIEXC001I: Content of the response file
TACMD_EXC_AIX_KUX__etc_ping_-c_1_AIX_20100420093428_07551048.log is:
------Command-------
/etc/ping -c 1 AIX
------Command Result-------
0
------Standard Error-------
------Standard Output-------
PING AIX.tivlab.raleigh.ibm.com: (9.42.11.174): 56 data bytes
64 bytes from 9.42.11.174: icmp_seq=0 ttl=255 time=0 ms

----AIX.tivlab.raleigh.ibm.com PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms

KUIEXC000I: Executecommand request was performed successfully.
The return value of the command run on the remote systems is 0
```

For information on identifying the PATH environment variable for the monitoring agent, see the *IBM Tivoli Monitoring Administrator's Guide*, and the section on the Agent Service Interface. Also, examine the ENVFILE section to locate the PATH value.

Escaping backslashes, spaces, and double quotation marks

When defining directories or file names, backslashes, spaces and double quotation marks must be escaped. For example, on a Windows system, if you attempt to list the files in directory `C:\Documents and Settings`, you must double-quote this directory. Because parameters defined for the -c|--commandstring option must be in double quotation marks, you must escape the double quotation marks used for specifying the directory with a space:

```
./tacmd executecommand -m Primary:WINDOWS:NT -c
"dir \"c:\\Documents and Settings\" " -o -l -v
```

CLI syntax

tacmd executecommand
{-m|--system} *SYSTEM*
{-c|--commandstring} *COMMAND_STRING*
[{-w|--workingdir} *REMOTE_WORKING_DIRECTORY***]**
[{-o|--stdout}]
[{-e|--stderr}]
[{-r|--returncode}]
[{-l|--layout}]
[{-t|--timeout} *TIMEOUT***]**
[{-d|--destination} *LOCAL_STD_OUTPUT_ERROR_FILENAME***]**
[{-s|--remotedestination} *REMOTE_STD_OUTPUT_ERROR_FILENAME***]**
[{-f|--force} *FORCE_MODE***]**
[{-v|--view}]

where:

-m|--system
Specifies on which managed system to execute the command. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ). The specified managed system must be a monitoring agent. The

specified system cannot be a Tivoli Enterprise Portal Server managed system. Use the **tacmd listsystems** command to view a list of available systems.

-c|--commandstring
Specifies the command to run. Use double quotation marks for commands with parameters. You must escape back slashes when defining a Windows directory path. See the "Escaping backslashes, spaces, and double quotation marks" section in the preceding command description.

-w|--workingdir
Specifies the working directory that is switched to before executing the command. Environment variables are supported.

When running this command between a UNIX or Linux system and targeting a Windows monitoring agent, you must replace the backslashes with forward slashes in the path definitions for the source parameter. It is best to use forward slashes for tolerance with Windows systems.

If the -w|--workingdir option is not specified, a default directory is used. To determine the default working directory for targeted Windows systems, you can issue a **./tacmd executecommand -m** *ManagedSystem* **-c chdir -o -v** command. For targeted UNIX and Linux systems, issue a **./tacmd executecommand -m** *ManagedSystem* **-c pwd -o -v** command.

-o|--stdout
Requests the standard output from the command to be captured. For section separators, use the -l|--layout option.

-e|--stderr
Requests the standard error from the command to be captured. For section separators, use the -l|--layout option.

-r|--returncode
Requests the return code to be captured. For section separators, use the -l|--layout option.

-l|--layout
Requests the command string executed at the monitoring agent to be captured and adds section separators to the result file.

-t|--timeout TIMEOUT
Specifies the timeout value of the command request. The TIMEOUT is the number of seconds allowed for the specified command to complete. The range is 1 to 32400 seconds. The default is 600 seconds. The command is not stopped if the timeout limit is reached, but the output is not captured.

-d|--destination
Specifies the name of the local file on the system where the command was issued, to which the results file on the endpoint is copied. If the -d|--destination option is not specified, a default file name is given. The default file name has the following format:

`TACMD_EXC_`*`ManagedSystemName_Command_Timestamp`*`_RND8Char.log`

Where:

ManagedSystemName
Specifies the managed system name, and is filtered removing special characters such as colons (:) and substituting them with underscores (_).

Command
Specifies the name of the command to be executed, is filtered removing special characters such as colons (:) and substituting them with underscores (_), and is truncated to 64 characters.

Timestamp
Specifies the timestamp, and is given in the format YYYYMMDDHHMMSS.

RND8Char
A random string composed of eight random numbers.

If you use the -d|--destination option, you must use one or more of the following options: -o|--stdout, -e|--stderr, or -r|--returncode. Supports environment variables, absolute paths, and relative paths. When specifying the destination directory, it must be an existing path.

-s|--remotedestination
Specifies the name of the results file at the endpoint. If the -s|--remotedestination option is not specified, the remote results file is not saved. If you specify a name without a fully qualified path, the `CANDLEHOME/kt1v3depot/pc` directory is used as the destination. If you use the -s|--remotedestination option, you must use one or more of the following options: -o|--stdout, -e|--stderr, -r|--returncode. Supports environment variables, absolute paths, but not relative paths.

When specifying the remote destination, it must be an existing path. When specifying the -c|--commandstring option and if you redirect the command output to a file, do not specify that same file name for this option. You can specify forward slashes instead of back slashes when targeting a Windows endpoint.

-f|--force
Overwrites the local and remote results files if they already exist. FORCE_MODE can have one of the following values: LOCAL, REMOTE, ALL. Specifying LOCAL overwrites the file defined with the –d|--destination option if the file already exists at the local machine where the command was issued. Specifying REMOTE overwrites the file defined with the -s|--remotedestination option on the target endpoint. Specifying ALL overwrites the file defined with the –d|--destination option and the file defined with the -s|--remotedestination option, if either file already exists.

-v|--view
Prints to the screen the contents of the results file. If you use the -v option, you must use one or more of the following options: -o|--stdout, -e|--stderr, -r|--returncode.

CLI example

See the example in the description of this command.

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd exportBundles

Description

Use the **tacmd exportBundles** command to export one or more deployment bundles to the specified export directory for use with software distribution products. You must run the **exportBundles** command locally on a server containing a depot or agent installation image as input. A bundle is the combination of an agent silent installation image as well as any necessary prerequisites and configuration information required to silently install an agent on a remote system. An agent deposits a directory on the monitoring server from which you can deploy agents and maintenance packages to remote systems across your environment. If the current OS user has the correct permissions, it is not necessary to have previously run the **login** command to run the **exportBundles** command.

CLI syntax

ExportBundles command:

tacmd exportBundles
[{-i|--imagePath}*IMAGEPATH***]**
{-e|--exportDir} ***DIRECTORY***
{-o|--outputFormat} **{*LOCAL*|*SPD*|*SPB*}**
{-t|--product} ***PRODUCT***
[{-p|--platform} ***PLATFORM*****]**
[{-v|--version} ***VERSION*** **]**
[{-f|--force}]

ExportBundles command by operating system:

tacmd exportBundles
{-i|--imagePath}*IMAGEPATH*
{-e|--exportDir} ***DIRECTORY***
{-o|--outputFormat} **{*LOCAL*|*SPD*|*SPB*}**
{-t|--product} ***PRODUCT***
{-os|--operatingSystem} ***OPERATING_SYSTEM*****]**
[{-v|--version} ***VERSION*** **]**
[{-f|--force}]

where:

-i |--imagePath
Specifies a directory that contains bundles to be added. The directory you specify should contain files with dsc as the file extension. On UNIX and Linux systems, this is in the `UNIX` directory of the agent media, and on Windows systems, this is in the `Windows\Deploy` directory of the agent media. The command uses the depot as the imagePath if no imagePath is specified. This option is required when specifying with -os option.

-e|--exportDir
Specifies the destination directory for the export operation.

-o|--outputFormat
Specifies whether the command creates a local silent installation bundle, creates a Software Package Definition file, or creates a Software Package Block as output. Specifying LOCAL copies a silent installation bundle out

to the destination directory for general use with software distribution technologies. The `silentInstall.sh` or `silentInstall.bat` script that is available in the destination directory can then be run to install the agent.

Note: You cannot interactively install the exported agent bundle by using the `install.sh` script because many interactive elements have been removed to optimize the bundle for remote transmission and silent execution using software distribution technologies.

Specifying SPD exports a Software Package Definition (SPD) file and other accompanying agent files so that Tivoli Configuration Manager (TCM) or Tivoli Provisioning Manager (TPM) can be used to remotely install the agent. When moving the SPD and associated files to another system, the SOURCE_DIR in the default_variable section of the SPD file needs to be updated to reflect the new directory where the agent files are located.

Specifying SPB creates a Software Package Block so that Tivoli Provisioning Manager (TPM) can be used to remotely install the agent.

-t|--product
Specifies the product code of the product to add. The product value corresponds to the value that is displayed in the Product Code field as a result of running the **viewDepot** or **listBundles** command.

-p|--platform
Specifies the platform code of the products to add. The platform value corresponds to the value that is displayed in the Host Type field as a result of running the **viewDepot** or **listBundles** command.

-v|--version
Specifies the version of the products to add. The version value corresponds to the value that is displayed in the Version field as a result of running the **viewDepot** or **listBundles** command. The command exports the latest version if a version is not specified.

-os|--operatingSystem
Specifies the operating system of the products to add. All binaries available for the operating system and corresponding product code will be added to the bundle. The supported operating system values are: LINUX, LINUX_Z, HP, WINDOWS, SOLARIS or AIX.

-f|--force
Performs actions without asking confirmation.

CLI example

This example creates a Software Package Definition file for the Monitoring Agent for Linux OS with the destination directory of `\bundle` for the export operation and also specifies the directory that contains the bundles to be added:

```
  tacmd exportBundles -o SPD -t lz -e \bundle -i
\CAT\itm62a\tmv620-d7310a-200711132133.xlin
ux1\unix -p li6263
```

This example shows using this command for product code ul, using a -p li6263 option. The exported bundles can be installed on a Linux li6263 machine.

```
  tacmd exportBundles -t ul -p li6263 -i
d:\tms623dev_pkg\tmv623dev-d0280a-201010080653.agents\unix
-o LOCAL -e d:\IBM\exportBundles_dir -f
```

This example shows using this command for product code ul, using a -os LINUX option. The exported bundles can be installed on a supported Linux interp machine.

```
  tacmd exportBundles -t ul -os LINUX  -i
d:\tms623dev_pkg\tmv623dev-d0280a-201010080653.agents\unix
-o LOCAL -e d:\IBM\exportBundles_dir -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd exportCalendarEntries

Description

Use the **tacmd exportCalendarEntries** command to export all the calendar entries available in the Tivoli Enterprise Portal Server to the specified XML file. You can optionally specify one or more calendar entry names to be exported by using the -n|--name|--names option.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **exportCalendarEntries** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd exportCalendarEntries
{-x|--file} *XMLFILE*
[{-u|--username} *TEPS_USER***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPS_HOSTNAME***]**
[{-n|--name|--names} *CALENDAR_ENTRY_NAME...***]**
[{-f|–force}]

where:

-x|--file
Specifies the name of the xml file accessible to the local file system where the calendar entry names are exported. The file name can either be a relative or absolute file name.

-u|--username
Specifies the identifier of the user to authenticate on the Tivoli Enterprise Portal Server.

-w|--password
Specifies the password of the user to authenticate on the Tivoli Enterprise Portal Server.

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname from where the calendar entries are exported.

-n | --name | --names
Specifies the name of the calendar entries to export.

-f | –force
Exports the calendar entries without prompting for confirmation.

CLI example

This example exports all the calendar entries from the Tivoli Enterprise Portal Server on localhost to the specified XML file:

```
tacmd  exportCalendarEntries -x D:\IBM\ITM\BIN\All_Calendarentries.xml
-u sysadmin -w ******
```

This example exports all the calendar entries from the Tivoli Enterprise Portal Server on LEVER2 to the specified XML file:

```
tacmd  exportCalendarEntries -x D:\IBM\ITM\BIN\All_Calendarentries.xml -u sysadmin
-w ***** -s LEVER2
```

This example exports only the calendar entries matching the names time_entries or task_entries to the specified XML file:

```
tacmd  exportCalendarEntries -x D:\IBM\ITM\BIN\BackUp_Calendarentries.xml
-u sysadmin -w ****** -n "time_entries" " task_entries"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd exportNavigator

Description

Use the **tacmd exportNavigator** command to export a Tivoli Enterprise Portal custom navigator view and all workspaces, queries, and situation associations referenced within the custom navigator view from the Tivoli Enterprise Portal Server to an XML file.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **exportNavigator** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd exportNavigator
{-x | --xmlFile} *XMLFILE*
{-n | --navigator} *NAVIGATOR*
[{-u | --username} *TEPS_USER***]**
[{-p | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOSTNAME[:PORT]***]**
[{-o | --navigatorOnly} **]**
[{-f | --force}]

where:

-x | --xmlFile
Specifies the name of the XML file accessible to the local file system where the custom navigator view, workspaces, queries, and situation associations are exported. The file name can either be a relative or absolute file name.

-n | --navigator
Specifies the name of the custom navigator view to export.

-s | --server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
Specifies the identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Custom Navigator Views' object, the 'Modify' permission for the 'Query' object, and the 'Workspace Administration Mode' and 'Workspace Author Mode' permissions for the 'Workspace Administration' object enabled on the server to execute the **exportNavigator** command. For most users, the permissions for these objects are disabled by default. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password of the user to authenticate to the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-o | --navigatorOnly
Specifies that only the custom navigator view will be exported. Workspaces, queries, and situation associations referenced within the custom navigator view will not be exported.

-f | –force
Exports the custom navigator view without confirmation.

CLI example

The following example exports the custom navigator view "Logical" from the Tivoli Enterprise Portal Server on HDCHASDSTC0420 to the file `logicalNavigator.xml`:

```
tacmd exportnavigator -x logicalNavigator.xml -n Logical -s HDCHASDSTC0420
-u sysadmin -p ***** -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd exportQueries

Description

Use the **tacmd exportQueries** command to export one or more Tivoli Enterprise Portal queries from the Tivoli Enterprise Portal Server to an XML file.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **exportQueries** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd exportQueries
{-x | --xmlFile} *XMLFILE*
[{-u | --username} *TEPS_USER***]**
[{-p | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOST[:PORT]***]**
[{-q | --query} *QUERY* **...]**
[{-t | --type} *TYPE* **...]**
[{-e | --exclude}]
[{-f | --force}]

where:

-x | --xmlFile
Specifies the name of the XML file accessible to the local file system where the query definitions are exported. The file name can either be a relative or an absolute file name.

-s | --server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
Specifies the identifier of the user to authenticate to the Tivoli Enterprise Portal Server. The user must have 'Modify' permissions for the 'Query' object enabled on the server to execute the **exportQueries** command. The 'Modify' permission for the 'Query' object is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password of the user to authenticate to the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-q | --query
Specifies the names of one or more queries to display.

-t | --type
Specifies the IBM Tivoli Monitoring product type code. If a 2-character type is entered, the letter 'k' will be prepended automatically to form a 3-character product type code.

Note: For the Monitoring Agent for UNIX OS, you must specify omunx as the product code. For example:

```
./tacmd exportqueries -t omunx -x /temp/test.xml
```

-e | --exclude
Specifies the query names and also product types to be excluded from the operation.

-f | --force
Exports the queries without confirmation.

CLI example

The following example exports all the queries from the Tivoli Enterprise Portal Server on HDCHASDSTC0420 to the file `exportQueries.xml`:

```
tacmd exportqueries -x exportQueries.xml -s HDCHASDSTC0420 -u sysadmin -p ***** -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd exportSitAssociations

Description

Use the **tacmd exportSitAssociations** command to export all situation associations for a Tivoli Enterprise Portal navigator, or optionally, a particular navigator item within the navigator, to an XML file.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd exportSitAssociations** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **exportSitAssociations** command.

CLI syntax

tacmd exportSitAssociations
{-x|--xmlFile} *XML_FILE*
[{-a|--navItem} *NAVIGATOR_ITEM***]**
[{-n|--navigator} *NAVIGATOR_NAME* **]**
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**
[{-f|--force} **]**

where:

-x|--xmlFile
The name of the xml file accessible to the local file system where the situation associations will be exported. The file name can either be a relative or absolute file name.

-a|--navItem
The fully qualified name of the navigator item to export the situation associations from. If this option is not provided, situation associations will be exported for the entire navigator. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the

forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n | --navigator
The name of the navigator view that the navigator item belongs to. By default, the Physical navigator view is used.

-s | --server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Situation' object enabled on the server to execute the **exportSitAssociations** command.

-p | --password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f | --force
Performs the action without prompting for confirmation.

CLI example

The following example exports the situation association, exp_sit_assoc.xml, from the Enterprise/child_logical navigator item that belongs to the Logical navigator:

```
tacmd exportsitassociations -a Enterprise/child_logical -n Logical
-x exp_sit_assoc.xml
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSitAssociation” on page 58

“tacmd deleteSitAssociation” on page 77

“tacmd importSitAssociations” on page 162

“tacmd listSitAssociations” on page 186

Return to Table 1 on page 5.

tacmd exportSysAssignments

Description

Use the **tacmd exportSysAssignments** command to export all managed system assignments for a Tivoli Enterprise Portal navigator, or optionally, a particular navigator item within the navigator, to an XML file.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd exportSysAssignments** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **exportSysAssignments** command.

CLI syntax

tacmd exportSysAssignments
{-x|--xmlFile} *XML_FILE*
[{-a|--navItem} *NAVIGATOR_ITEM***]**
{-n|--navigator} *NAVIGATOR_NAME*
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**
[{-f|--force}]

where:

-x|--xmlFile
The name of the xml file accessible to the local file system where the managed system assignments will be exported. The file name can either be a relative or absolute file name.

-a|--navItem
The fully qualified name of the navigator item to export the managed systems or managed system lists for. If this option is not provided, managed system assignments will be exported for the entire navigator. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n|--navigator
The name of the navigator view to export managed system assignments for. If the -a|--navItem option is provided, the navigator item must belong to the specified navigator.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Custom Navigator Views' object enabled on the server to execute the **exportSysAssignments** command.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f|--force
Performs the action without prompting for confirmation.

CLI example

The following example exports a managed system assignment from the Enterprise/child_logical navigator item that belongs to the Logical navigator to the xml file, exportsysass.xml:

```
tacmd exportsysassignments -a Enterprise/child_logical -n Logical
-x exportsysass.xml
```

Return values

See "Return codes" on page 272

Related commands

"tacmd createSysAssignment" on page 60

"tacmd deleteSysAssignment" on page 78

"tacmd importSysAssignments" on page 164

"tacmd listSysAssignments" on page 188

Return to Table 1 on page 5.

tacmd exportWorkspaces

Description

This command exports one or more Tivoli Enterprise Portal Server workspaces to the file *xml_file*. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **exportWorkspaces** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: If you get an OutOfMemoryError when running this command, you can increase the java JVM memory size by using the TACMD_JVM_MAX_MEMORY variable. Valid values are 256 through to 2048, inclusive of that number is the number of megabytes to allocate for the max heap size.

CLI syntax

tacmd exportWorkspaces
{-x|--xmlFile} *XMLFILE*
{{-w|--workspace} WORKSPACE ... | {-i|--objectid} OBJECT_ID }
[{-t|--type} TYPE ...]
[{-r|--workspaceUser} USERID ...]
[{-e|--exclude}]
[{-q|--queries}]

[{-f|--force}]
[{-u|--username} TEPS_USER]
[{-p|--password} TEPS_PASSWORD]
[{-s|--server} TEPS_HOST[:PORT]]

tacmd exportWorkspaces
{-h|--html}
[{-l|--onehtmlfile}]
[{-w|--workspace} WORKSPACE ...]
[{-i|--objectid} *OBJECT_ID* **]**
[{-t|--type} TYPE ...]
[{-r|--workspaceUser} USERID ...]
[{-e|--exclude}]
[{-f|--force}]
[{-u|--username} TEPS_USER]
[{-p|--password} TEPS_PASSWORD]
[{-s|--server} TEPS_HOST[:PORT]]

where:

-x|--xmlFile
The name of the XML file accessible to the local file system where the workspace definition or definitions will be exported. This is the name of a file that can be created or overwritten. The file name can either be a relative or absolute file name.

-s|--server
Specifies a Tivoli Enterprise Portal Server to use. The host is a 32 or 64 bit IP address or hostname and port is an integer between 1 and 65536. If not specified, host defaults to localhost and port defaults to 15200.

-u|--username
The identifier of the user to authenticate on the remote Tivoli Enterprise Portal Server. The user must have both 'Workspace Administration Mode' and 'Workspace Author Mode' Workspace Administrator permissions enabled on the server to execute the **listworkspaces** command. The 'Workspace Administration Mode' permission is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p|--password
The password of the user to authenticate on the remote Tivoli Enterprise Portal Server. Specify a valid string in the local locale. The software prompts you for the password if you do not specify one.

-w|--workspace
The name or names of the workspaces to display. Specify a string (any character except hyphen (-)) up to a maximum length of 72 characters. If not specified, all workspaces will be exported.

-i|--objectid
The object identifier of the workspaces to export. This switch cannot be used with the -w|--workspace or -e|--exclude options. You can retrieve the workspace object identifier by running the **listworkspaces** command with the -i|--objectid option.

-r|--workspaceUser
A Tivoli Enterprise Portal user ID that one or more Tivoli Enterprise Portal

workspaces are associated with. Specify a string of letters (upper or lower case) or numbers up to a maximum length of 32 characters. If not specified, workspaces are exported for all users. To export only global workspaces, use this option without specifying any Tivoli Enterprise Portal User IDs.

-t|--type
An IBM Tivoli Monitoring application type. If a 2-character type is entered, the letter 'k' will be prefixed automatically to form a 3-character type code. For example, _kib is the type for the Tivoli Enterprise Tivoli Monitor Workspaces. If not specified, all types are exported.

-e|--exclude
Exclude the specified workspace users, application types, and Tivoli Enterprise Portal user IDs from the export operation.

-q|--queries
Exports any queries that are used by the exported workspaces to the XML file.

-h|--html
Exports the workspaces and queries to HTML instead of XML.

-l|--onehtmlfile
Exports the queries and workspaces to a single HTML file.

-f|--force
Perform the export without prompting for confirmation first.

CLI example

This example exports all workspaces on the Tivoli Enterprise Portal Server myteps.ibm.com without any filtering arguments (such as workspace name, user ID, or application type).

Note: A large number (over 500) of workspaces can be displayed and exported.

```
tacmd exportWorkspaces –s http://myteps.ibm.com:15200 –u imasample
                       –p mypassword-x all_workspaces.xml
```

This example exports all workspaces on the Tivoli Enterprise Portal Server myteps.ibm.com without any filtering arguments (such as workspace name, user ID, or application type). The **-f** option is used in this example to perform the export operation without prompting for confirmation.

Note: A large number (over 500) of workspaces are likely be exported.

```
tacmd exportWorkspaces –s http://myteps.ibm.com –u imasample
                       –p mypassword-x all_workspaces.xml -f
```

This example exports all workspaces belonging to the *klz* and *knt* application types on the Tivoli Enterprise Portal Server running on the local computer on port 15200 and filtered by application type.

```
tacmd exportWorkspaces –u imasample –p mypassword–t klz knt
                       -x klz_and_knt_workspaces.xml
```

This example is identical to the one above, except that the server credentials (username and password) were omitted at invocation time, and the user is interactively prompted to enter them.

```
tacmd exportWorkspaces –s myteps.ibm.com –t klz knt
```

This example exports all workspaces belonging to (customized for) the *SYSADMIN* user on the Tivoli Enterprise Portal Server myteps.ibm.com and filtered by username.

Note: In this example no global workspaces are exported.

```
tacmd exportWorkspaces –s myteps.ibm.com –u imasample –p mypassword–r SYSADMIN
```

This example exports only workspaces matching the names *Historical Summarized Availability Daily* or *Historical Summarized Availability Weekly* on the Tivoli Enterprise Portal Server myteps.ibm.com and filtered by workspace name.

```
tacmd exportWorkspaces –s myteps.ibm.com –u imasample –p mypassword
      –w "Historical Summarized Availability Daily"
         "Historical Summarized Availability Weekly"
```

This example exports only workspaces belonging to the *klz* and *knt* application types, workspace names matching the names *Historical Summarized Availability Daily* or *Historical Summarized Availability Weekly* on the Tivoli Enterprise Portal Server myteps.ibm.com, and filtered by both workspace name and application type.

```
tacmd exportWorkspaces –s myteps.ibm.com –u imasample –p mypassword–t klz kux
–w "Historical Summarized Availability Daily"
       "Historical Summarized Availability Weekly"
```

Return values

See “Return codes” on page 272.

Related commands

“tacmd listworkspaces” on page 196

“tacmd importWorkspaces” on page 166

Return to Table 1 on page 5.

tacmd getDeployStatus

Description

Use the **getdeploystatus** command to view the status of the asynchronous agent deployment operations. This command is also available for non-agent bundles. You must log in by using the **login** command before running the **getdeploystatus** command.

Note: When using the **getDeployStatus** command to view the status for a createnode group deployment, it generates the same transaction ID twice.

You can also follow the deployment status using the Deployment Status Summary workspace in the Tivoli Enterprise Portal. For more information about the Deployment Status Summary and Deployment Status Summary by Transaction workspaces, see the *Tivoli Enterprise Portal User's Guide*.

CLI syntax

tacmd getDeployStatus
[{-g|--transactionID} *TRANSID*]

[{-c|--command} *COMMAND*]
[{-h|--hostname} *HOSTNAME*]
[{-p|--platform} *PLATFORM*]
[{-t|--product}]
[[{-f|--failed}]
[{-q|--queued}]
[{-s|--successful}]
[{-r|--retryable}]
[{-o|--overview}]

where:

-g|--transactionID
Specifies the global transaction ID.

-c|--command
Specifies the type of the deployment operation. Acceptable operations are: START, RESTART, STOP, INSTALL, REMOVE, CONFIGURE, UPDATE, CHECKPREREQ, or SETAGENTCONN.

-h|--hostname
Specifies the hostname that the deployment operation(s) will occur on.

-p|--platform
Specifies the platform that the deployment operation(s) will occur on.

-t|--product
Specifies the product type that the deployment operation(s) will occur on.

-f|--failed
The flag to filter the result by the failed transaction.

-q|--queued
The flag to filter the result by the queued transaction.

-s|--successful
The flag to filter the result by the successful transaction.

-r|--retryable
The flag to filter the result by the retryable transaction.

-i|--inprogress
The flag to filter the result by the in progress transaction.

-o|-- overview
Displays an overview of the results based on the specified filter.

CLI example

The following example retrieves the status of all the restartagent command requests made on the server you are logged on to:

```
tacmd getdeploystatus -c RESTART

Transaction ID     : 1213498140711931000187424439 4
Command             : RESTART
Status                  : SUCCESS
Retries                 : 0
TEMS Name          : r096o001
Target Hostname : F50PA2D:f50pa2d:SYB
Platform               :
Product                : OY
Version                :
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd getfile

Description

Use the **getfile** command to transfer a file from a remote managed system to a local destination.

The hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. If the Tivoli Enterprise Monitoring Agent component is at the IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later level, all the agents installed in the same `CANDLEHOME` directory at the endpoint are capable of handing this command. For this command, the specified system cannot be an i5/OS or z/OS monitoring agent.

Note: Do not run more than 10 concurrent getfile, putfile, executeaction, or executecommand operations, in any combination. These 10 concurrent operations apply to both different agents and different physical machines. This command is recommended for transfers of 16 MB or less although not limited to this transfer size.

Transfer file sizes exceeding this limit can require additional response time and IBM Tivoli Monitoring environment consumption. If the getfile, putfile, executeaction and executecommand operations will be executed frequently, monitor the CPU utilization and network activity of the hub monitoring server and remote monitoring servers before and during these operations to ensure that resource consumption is acceptable. If resource consumption is too high, consider reducing the number of concurrent operations and the frequency of the operations.

Note: On the local machine where the command is issued, ensure that system's defined temporary directory has sufficient space to temporarily contain the transferred file. The temporary directory is defined by the %TEMP% or %TMP% environment variable for Windows systems, and is the `/tmp` directory for UNIX and Linux systems.

Hub server configured with non-default port number

The executecommand, getfile, and putfile commands fail if the HUB TEMS is configured with a non-default port number. You must set the environment variable KDE_TRANSPORT in the Windows command prompt or UNIX Shell before issuing these commands to configure the TACMD to use the non-default port number to connect to the hub monitoring server. See the “KDE_TRANSPORT Structure” section of the “Configuring IBM Tivoli Monitoring components” chapter in the *IBM Tivoli Monitoring: Installation and Setup Guide* for descriptions and examples.

Relative and absolute path support at the endpoint

When running this command between a UNIX or Linux system and targeting a Windows monitoring agent, you must replace the backslashes with forward slashes in the path definitions for the -s|--source option. It is best to use forward slashes for tolerance with Windows systems. For example, if you want to run the command from a UNIX system to take the monitor agent's log file in the `C:\IBM\ITM\tmaitm6\logs` directory on a Windows system, use the following command:

```
 ./tacmd getfile -m Primary:WINDOWS:NT
-s C:/IBM/ITM/tmaitm6/logs/WINDOWS_nt_kntcma_4b8c6aef-01.log
-d ./WINDOWS_nt_kntcma_4b8c6aef-01.log -t text
```

File names

When either the remote file's directory or name and the destination file's directory or name contain spaces, you must include double quotation marks around the respective directory and file name. For example, run the following command from a UNIX system to take the monitoring agent's log file in the `C:\Program Files\ITM\tmaitm6\logs` directory.

```
./tacmd getfile -m Primary:WINDOWS:NT
-s "C:/Program Files/ITM/tmaitm6/logs/WINDOWS_nt_kntcma_4b8c7baf-01.log"
-d "/log files/WINDOWS_nt_kntcma_4b8c7baf-01.log" -t text
```

When working with file and directory names that have nonalphanumeric or special characters (for example, ! @ #, etc), the path and file references for either the -s|--source or -d|--destination options must be surrounded by double quotation marks (" "). However, paths that include an at symbol (@) must be escaped with an at symbol (@). The path user@home@directory is escaped as follows:

```
user@@home@@directory
```

Variable substitution

You can run this command by using an environment variable for both the d|--destination and the -s|--source options. If used for the -s|--source option, it is for the specified monitoring agent's managed system rather than the local environment where the command is issued. If used for the d|--destination option, it is for the local environment where the command is issued.

The environment variable format is @NAME@. The following characters are valid as the first character of any name, or any subsequent character:

- _ (underscore)
- Lower case alphabetic letters
- Upper case alphabetic letters

The following characters are valid as any character in any name except the first:

- - (dash)
- The following numbers, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

In the following example, *CANDLEHOME* on the local machine is `/opt/IBM/ITM` and *CANDLE_HOME* on the managed system is `c:\IBM\ITM`:

```
./tacmd getfile -m Primary:WINDOWS:NT
-s @CANDLE_HOME@/tmaitm6/logs/WINDOWS_nt_kntcma_4b8c6aef-01.log
-d @CANDLEHOME@/xfer/WINDOWS_nt_kntcma_4b8c6aef-01.log -t text
```

Note:

1. For monitoring agents running on AIX 6.1 systems as a root user, it is possible to issue a **tacmd getfile** command for files having permission 000.
2. To use this command, the KT1_TEMS_SECURE environment variable must be set in the hub monitoring server's configuration file to specify that the hub monitoring server supports this command. After setting the environment variable, you must recycle the hub monitoring server. The default value is no. Specify Y or YES or y or yes if you want to use this command.

CLI syntax

tacmd getfile
{-m|--system} *SYSTEM*
{-s|--source} *REMOTE_FILE*
{-d|--destination} *LOCAL_FILE*
[{-t|--type} *MODE***]**
[{-f|--force}]

where:

-m|--system
Specifies from which managed system to get the file. This must be a monitoring agent. Use the **listsystems** command to receive a list of which systems are available. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-s|--source
Specifies the remote file name. Environment variables are supported. When specifying the source, it must be an existing path. If the path is not specified, the default path is the *CANDLEHOME*/kt1v3depot/*product_code* directory on the endpoint.

-d|--destination
Specifies the local file name. Environment variables are supported. When specifying the destination, it must be an existing path. If the path is not specified, the default path is relative to where the command is issued.

-t|--type
Specifies the MODE of transfer. MODE can be bin or text. If not specified, the default is bin. Specify text mode if the file is a human readable file, otherwise, specify bin (binary) mode.

-f|--force
Overwrites the local file as specified by the –d|--destination option if it already exists.

CLI example

See the example in the description of this command.

Return values

See Table 8 on page 272.

Related commands

"tacmd putfile" on page 203

Return to Table 1 on page 5.

tacmd help

Description

Use the **tacmd help** command to display the name and short description of all the available CLI commands or to display the complete help for a specified command.

CLI syntax

tacmd help | ? {*command*}

where:

{*command*}

Specifies the command you want detailed help for. The following lists and describes the available commands:

tacmd acceptbaseline
Sets a situation override based on the baseline (situation override) values calculated by using one of several statistical functions for a situation attribute based on historical data from the Tivoli Data Warehouse. This command yields identical calculations to the suggestBaseline command. You can use the acceptBaseline command to calculate and set baseline values with a single command invocation.

tacmd addbundles
Adds component bundles to a deployment depot. This command can only be run from a Tivoli Enterprise Monitoring Server installation with a depot.

tacmd addcalendarentry
Adds a calendar entry to the Tivoli Enterprise Monitoring Server.

tacmd addgroupmember
Adds a group member to the specified group.

tacmd addsdainstalloptions
Adds a version to the Self-Describing Agent (SDA) install option record.

tacmd addsystem
Adds a new system to the Tivoli Enterprise Monitoring Server..

tacmd bulkexportpcy
All or the specified types and policies are exported from Tivoli Enterprise Monitoring Server.

tacmd bulkexportsit
All or the specified types and situations are exported from Tivoli Enterprise Monitoring Server.

tacmd bulkimportpcy
All or the specified types and policy objects are imported to Tivoli Enterprise Monitoring Server.

tacmd bulkimportsit
All or the specified types and situation objects are imported to Tivoli Enterprise Monitoring Server.

tacmd checkprereq
Checks the specified managed system for required prerequisites for a specified agent installation.

tacmd cleanms
Deletes the entries for offline managed systems.

tacmd clearappseedstate
Clear the seed state of an application supports install records.

tacmd cleardeploystatus
Remove entries from the table that stores the status of the asynchronous agent deployment operations.

tacmd configureportalserver
Configure a user-defined portal server data source.

tacmd configuresystem
Updates the configuration of a system.

tacmd createaction
Creates a new Take Action command definition on the monitoring server.

tacmd createeventdest
Creates a new event destination definition on the monitoring server.

tacmd creategroup
Creates a new deployment, bundle, or situation group definitions in the agent deployment tables on the Tivoli Enterprise Monitoring Server.

tacmd createnode
Creates a node and starts an OS agent on it.

tacmd createsit
Creates a situation on the server.

tacmd createsitassociation
Creates one or more situation associations for a Tivoli Enterprise Portal navigator item. Optionally, you can also create one or more managed system or managed system list assignments for the navigator item.

tacmd createsysassignment
Assigns one or more managed systems or managed system lists to a Tivoli Enterprise Portal navigator item.

tacmd createsystemlist
Creates a system list on the server.

tacmd createuser
Creates a new user in the Tivoli Enterprise Portal Server.

tacmd createusergroup
Creates a new user group in the Tivoli Enterprise Portal Server.

tacmd deleteaction
Deletes a Take Action command on the Tivoli Enterprise Monitoring Server.

tacmd deleteappinstallrecs
Deletes application supports install records from the Tivoli Enterprise Monitoring Server.

tacmd deletecalendarentry
Deletes a calendar entry from the Tivoli Enterprise Monitoring Server.

tacmd deleteeventdest
Deletes an event destination server definition from the Tivoli Enterprise Monitoring Server.

tacmd deletegroup
Deletes a group definition and all group members belonging to the group from the deployment tables on the Tivoli Enterprise Monitoring Server.

tacmd deletegroupmember
Deletes a group member definition from the group.

tacmd deleteoverride
Deletes the situation overrides defined for a specified situation on a managed system or list of managed systems.

tacmd deletesdainstalloptions
Deletes a version from a Self-Describing Agent (SDA) install option record.

tacmd deletesdaoptions
Deletes Self-Describing Agent (SDA) option configuration entries.

tacmd deletesdasuspend
Delete the Self-Describing Agent (SDA) Suspend record from the database.

CAUTION:
Do not use the deleteSdaSuspend command unless directed by IBM Software Support.

tacmd deletesit
Deletes a situation from the server.

tacmd deletesitassociation
Dissociates one or more situations from a Tivoli Enterprise Portal navigator item.

tacmd deletesysassignment
Deletes one or more managed system assignments from a Tivoli Enterprise Portal navigator item.

tacmd deletesystemlist
Deletes a system list from the server.

tacmd deleteuser
Deletes a specified user from the Tivoli Enterprise Portal Server.

tacmd deleteusergroup
Deletes a specified user group from the Tivoli Enterprise Portal Server.

tacmd deleteworkspace
Deletes a global or user-customized Tivoli Enterprise Portal workspace from the Tivoli Enterprise Portal Server.

tacmd describesystemtype
Displays the configuration options available for a system.

tacmd editaction
Edits a Take Action command definition on the Tivoli Enterprise Monitoring Server.

tacmd editcalendarentry
Modifies an existing calendar entry definition on the Tivoli Enterprise Monitoring Server.

tacmd editeventdest
Modifies an existing event destination server definition on the Tivoli Enterprise Monitoring Server.

tacmd editgroup
Modifies an existing deployment, bundle, or situation group definition in the agent deployment tables on the Tivoli Enterprise Monitoring Server.

tacmd editgroupmember
Modifies or defines an alias for an existing situation group member, modifies the properties for a deployment group, or modifies the type, version, platform, or properties for a bundle definition.

tacmd editsdainstalloptions
Edits the version of a Self-Describing Agent (SDA) install option record.

tacmd editsdaoptions
Edits Self-Describing Agent (SDA) option configuration entries.

tacmd editsit
Edits a situation definition.

tacmd editsystemlist
Edits a system list definition.

tacmd edituser
Edits a specified user definition.

tacmd editusergroup
Edits a specified user group definition.

tacmd executeaction
Executes the Take Action command.

tacmd executecommand
Executes a command on a managed system.

tacmd exportBundles
Exports one or more deployment bundles to the specified export directory.

tacmd exportcalendarentries
Exports all the calendar entries available in the Tivoli Enterprise Portal Server to the specified XML file.

tacmd exportnavigator
Exports a Tivoli Enterprise Portal custom navigator and all workspaces, queries, and situation associations referenced within the custom navigator from the Tivoli Enterprise Portal Server to an XML file. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd exportqueries
Exports one or more Tivoli Enterprise Portal queries from the Tivoli Enterprise Portal Server to an XML file. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd exportsitassociations
Exports all situation associations for a Tivoli Enterprise Portal navigator, or optionally, a particular navigator item within the navigator, to an XML file.

tacmd exportsysassignments
Exports all managed system assignments for a Tivoli Enterprise Portal navigator, or optionally, a particular navigator item within the navigator, to an XML file.

tacmd exportworkspaces
Exports one or more Tivoli Enterprise Portal workspaces from the Tivoli Enterprise Portal Server to an XML file. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd getdeploystatus
Displays the status of the asynchronous agent deployment operations.

tacmd getfile
Transfers a file from a remote managed system to a local destination.

tacmd help | ? {command}
Displays complete help for a specified command.

tacmd histconfiguregroups
Configures the specified attribute groups for history collection.

tacmd histcreatecollection
Creates the historical data collection of specified attribute groups.

tacmd histdeletecollection
Deletes the historical data collection of specified attribute groups.

tacmd histeditcollection
Edits the historical data collection of specified attribute groups.

tacmd histlistattributegroups
Lists all the attribute groupnames for the specified product that are available for History Configuration.

tacmd histlistcollections
Lists historical data collection for the specified managed system.

tacmd histlistproduct
Lists all the products that are available for History Configuration.

tacmd histstartcollection
Starts the historical data collection of specified attribute groups.

tacmd histstopcollection
Stops the historical data collection of specified attribute groups.

tacmd histunconfiguregroups
Unconfigures the history configuration details of the specified attribute groups.

tacmd histviewattributegroup
Displays the history configuration details of the specified attribute group.

tacmd histviewcollection
Displays historical data collection configuration information for the specified collection.

tacmd importcalendarentries
Imports all the calendar entries available in the specified XML file to the Tivoli Enterprise Portal Server.

tacmd importnavigator
Imports a Tivoli Enterprise Portal custom navigator view, workspaces, queries, and situation associations from an XML file to the Tivoli Enterprise Portal Server. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd importqueries
Imports Tivoli Enterprise Portal queries from an XML file to the Tivoli Enterprise Portal Server. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd importsitassociations
Imports all situation associations from an XML file to the Tivoli Enterprise Portal Server.

tacmd importsysassignments
Imports all managed system assignments from an XML file to the Tivoli Enterprise Portal Server.

tacmd importworkspaces
Imports Tivoli Enterprise Portal workspaces from an XML file to the Tivoli Enterprise Portal Server. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd listaction
Displays a list of Take Action commands.

tacmd listappinstallrecs
Displays the application install records.

tacmd listbundles
Displays details for component bundles not yet added to a

deployment depot. This command can only be run from a Tivoli Enterprise Monitoring Server installation with a depot.

tacmd listcalendarentries
Displays calendar entry name, type and data for each calendar entry definition on the Tivoli Enterprise Monitoring Server.

tacmd listeventdest
Displays the server ID, server name, and server type for every event destination server definition on the server.

tacmd listgroups
Displays the name and bundle type for each group definition on the server.

tacmd listnavigators
Displays a list of Tivoli Enterprise Portal custom navigator views assigned to the specified Tivoli Enterprise Portal user on the Tivoli Enterprise Portal Server. The custom navigator name and description are displayed for each custom navigator. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd listoverrideablesits
Displays a list of override-eligible situations for a given application.

tacmd listoverrides
Displays the situation overrides defined for a specified situation on a managed system or list of managed systems.

tacmd listqueries
Displays a list of Tivoli Enterprise Portal queries on the Tivoli Enterprise Portal Server. You can optionally filter the list by query names or product codes. The query name and product code are displayed for each query. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd listsdainstalloptions
Lists the Self-Describing Agent (SDA) install options records.

tacmd listSdaOptions
Lists Self-Describing Agent (SDA) option configuration entries.

tacmd listSdaStatus
List the Self-Describing Agent (SDA) Enablement information.

tacmd listsit
Lists known situations.

tacmd listsitassociations
Displays a list of all situations associated with or eligible for association with a Tivoli Enterprise Portal navigator item.

tacmd listsitattributes
Displays a list situation attribute names that are eligible for use with dynamic thresholding (override) commands for a given situation. The command distinguishes between attributes that can be used as part of a predicate expression and attributes that can be used as part of a condition expression.

tacmd listsysassignments
Displays a list of managed systems or managed system lists that are assigned to a Tivoli Enterprise Portal navigator item. Optionally, the command can display a list of situations that are eligible for association with the specified navigator item.

tacmd listsystemlist
Displays a list of known system lists.

tacmd listsystems
Displays a list of known systems.

tacmd listtrace
Displays the current RAS1 tracing level on a managed system.

tacmd listusers
Lists all the users or the users belonging to a particular group.

tacmd listusergroups
Lists all existing user groups.

tacmd listworkspaces
Displays a list of Tivoli Enterprise Portal workspaces on the Tivoli Enterprise Portal Server. You can optionally filter the list by workspace names, product codes, or workspace users. The workspace name, product code, and user ID are displayed for each workspace. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

tacmd login
Authenticates a username and password with a hub monitoring server so that a user can execute subsequent commands from the local computer.

tacmd logout
Logs a user off the server.

tacmd managesit
Starts or stops the specified situations in the Tivoli Enterprise Monitoring Server.

tacmd pdcollect
Remotely invoke PDCollect tool and transfer the compressed file to the local computer.

tacmd putfile
Transfers a file from a local source to a remote managed system.

tacmd refreshcatalog
Refresh the catalog file, allows the data server to reread the catalog files and refresh the affinity information.

tacmd refreshTECinfo
Refresh Tivoli Enterprise Console Event Integration Facility configuration or event mapping files.

tacmd removebundles
Removes component bundles from a deployment depot.

tacmd removesystem
Remove one or more instances of an agent or uninstall an agent

from a managed system. By using the bulk deployment option, the command will remove all agents in a deployment and bundle group combination.

tacmd restartagent
Restarts a monitoring agent.

tacmd restartfaileddeploy
Restarts a failed deployment command.

tacmd resumesda
Resumes the Self-Describing Agent (SDA) installation functions.

tacmd setagentconnection
Updates the connection properties and environment variables for the agent.

tacmd setoverride
Defines a situation override for a specified situation on a managed system or list of managed systems.

tacmd settrace
Changes the RAS1 tracing level on a managed system to the specified new value.

tacmd startagent
Starts a monitoring agent.

tacmd stopagent
Stops a monitoring agent.

tacmd suggestbaseline
Calculates a baseline (situation override) value by using one of several statistical functions for a situation attribute based on historical data from the Tivoli Data Warehouse.

tacmd suspendsda
Suspends the Self-Describing Agent (SDA) installation functions.

tacmd tepslogin
Logs on to the Tivoli Enterprise Portal Server.

tacmd tepslogout
Logs off of the Tivoli Enterprise Portal Server.

tacmd updateagent
Updates a monitoring agent to a new version.

tacmd viewaction
Displays a Take Action command definition.

tacmd viewagent
Displays the details and status of a monitoring agent.

tacmd viewcalendarentry
Displays a calendar entry definition.

tacmd viewdepot
Displays the components that you can deploy remotely.

tacmd vieweventdest
Displays all properties for the specified event destination definition on the monitoring server.

tacmd viewgroup
Displays the details of a group definition.

tacmd viewgroupmember
Displays the properties of a specific group member, depending upon the group type of the group member.

tacmd viewnode
Displays the versions and patch levels of the systems that are installed on a node or a group of nodes.

tacmd viewsit
Displays or exports a situation definition.

tacmd viewsystemlist
Displays or exports a system list definition.

tacmd viewuser
Displays the details of a specified user.

tacmd viewusergroup
Displays the details of a specified user group.

CLI example

This command displays the name and short description of all the available CLI commands.

```
tacmd help
```

or

```
tacmd ?
```

This command displays the detailed help for the **addSystem** command.

```
tacmd help addSystem
```

or

```
tacmd ? addSystem
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histconfiguregroups

Description

Use the **tacmd histconfiguregroups** command to configure the provided attribute group or attribute groups with the specified input values. The command configures the collection settings of the default collection for each attribute group unless you specify the -m|--summarizationonly option.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the

histconfiguregroups command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: The -c|--collectioninterval, -l|--collectionlocation and -i|--warehouseinterval options cannot be specified with the -m|--summarizationonly option.

CLI syntax

tacmd histconfiguregroups
{-t|--type} *PRODUCTTYPE*
{-o|--object} *ATTRIBUTEGROUPNAME...*
[{-c|--collectioninterval} *COLLECTIONINTERVAL***]**
[{-l|--collectionlocation} *COLLECTIONLOCATION***]**
[{-i|--warehouseinterval} *WAREHOUSEINTERVAL***]**
[{-d|--summarizationdetails} *SUMMARIZATIONDETAILS***]**
[{-p|--pruningdetails} *PRUNINGDETAILS***]**
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

tacmd histconfiguregroups
{-m|--summarizationonly}
{-t|--type} *PRODUCTTYPE*
{-o|--object} *ATTRIBUTEGROUPNAME...*
[{-d|--summarizationdetails} *SUMMARIZATIONDETAILS***]**
[{-p|--pruningdetails} *PRUNINGDETAILS***]**
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

where:

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken if you do not specify one.

-u|--userid
Specifies the user to authenticate. The software prompts you for the username if you do not specify one. The User ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character. The User ID specified must have Configure History permission.

-w|--password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-t|--type
Specifies the product code or product name to list its attribute groups. Use the **tacmd histlistproduct** command to determine the codes and names.

-o|--object
Specifies the attribute group name(s) to be configured. When multiple attribute groups are specified, each attribute group must be separated by a space. Use the **tacmd histlistattributegroups** command to find the correct values to use.

-m | --summarizationonly
Specifies the flag for configuring only the summarization and pruning settings.

-c | --collectioninterval
Specifies the collection interval. The acceptable inputs are 1m (1 minute), 5m, 15m, 30m, 1h (1 hour), or 1d (1 day). The default value is 15m (15 minutes).

-l | --collectionlocation
Specifies where the data collection is stored. The acceptable inputs are TEMA or TEMS. The default value is TEMA.

-i | --warehouseinterval
Specifies the warehouse interval. The acceptable inputs are 15m (15 minutes), 30m, 1h (1 hour), 12h, 1d (1 day), or off. The default value is 1d (1day).

-d | --summarizationdetails
Specifies the summarization details. If this option is not specified, summarization will be disabled. This option can be given as YQMWDH where Y=Yearly,Q=Quarterly,M=Monthly,W=Weekly,D=Daily, and H=Hourly.

-p | --pruningdetails
Specifies the pruning details. If this option is not specified, pruning will be disabled. This option can be given as Y=1y,Q=1y,M=2m,W=2m,D=1d,H=2d,R=2d where Y=Yearly,Q=Quarterly,M=Monthly,W=Weekly,D=Daily,H=Hourly, and R=Detailed data.

CLI example

This example configure the attribute group “NT Processor” for historical data collection:

```
tacmd histconfiguregroups -s LEVER2 -t "Windows OS" -o "NT Processor "
-c 5m -l TEMS -i 1h  -d YQMWDH -p Y=1y,Q=2y,M=3m,W=4m,D=5d,H=6d,R=7d
-u Administrator  -w tivoli123$
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histcreatecollection

Description

Use the **tacmd histcreatecollection** command to create the given collection by using specified inputs.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histcreatecollection** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: The -o | --object option cannot have multiple values when the -a | --name option has been specified.

CLI syntax

tacmd histcreatecollection
{-a | --name} *COLLECTIONNAME*
[{-e | --description} *DESCRIPTION***]**
{-t | --type} *PRODUCTTYPE*
{-o | --object} *ATTRIBUTEGROUPNAME*
[{-c | --collectioninterval} *COLLECTIONINTERVAL***]**
[{-l | --collectionlocation} *COLLECTIONLOCATION***]**
[{-i | --warehouseinterval} *WAREHOUSEINTERVAL***]**
[{-f | --filter} *FILTERFORMULA***]**
[{-u | --userid} *TEPS_USERID***]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPSHOSTNAME***]**

where:

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is used if you do not specify one.

-u | --userid
Specifies the user to authenticate. The software prompts you for the username if you do not specify one.

-w | --password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-t | --type
Specifies the product code or product name. Use the **tacmd histlistproduct** command to determine the codes and names.

-o | --object
Specifies the attribute group name to be configured. Use the **tacmd histlistattributegroups** command to find the correct values to use.

-a | --name
Specifies the name of collection to create or modify. Do not use the ampersand (&) when specifying this name. The maximum value for this option is 256 bytes.

-e | --description
Specifies the description of the named historical collection to create. The maximum value for this option is 64 bytes.

-c | --collectioninterval
Specifies the collection interval. The acceptable inputs are 1m (1 minute), 5m, 15m, 30m, 1h (1 hour), or 1d (1 day). The default value is 15m (15 minutes).

-l | --collectionlocation
Specifies where the data collection is stored. The acceptable inputs are TEMA or TEMS. The default value is TEMA.

-i | --warehouseinterval
Specifies the warehouse interval. The acceptable inputs are 15m (15 minutes), 30m, 1h (1 hour), 12h, 1d (1 day), or off. The default value is 1d (1 day).

-f | --filter
Specifies the formula used to filter data stored at the binary file. By default no filter is applied if you do not specify one. The following format is for the the base filter formula:

```
*IF CONDITION
```

CONDITION can be one condition or a list of conditions, each separated by an *AND or *OR logical operator and, if needed, grouped in parentheses. The condition can be a function, an attribute, a comparison operator, or a value. The following example is for historical filters:

```
{*VALUE|*SCAN|*STR}
ATTRIBUTE_GROUP.ATTRIBUTE
{*EQ|*NE|*GT|*GE|*LT|*LE|*IN}
VALUE
```

ATTRIBUTE_GROUP and ATTRIBUTE are the names of the table and of the column in the database. Run the **tacmd histviewattributegroup** command with the --verbose option to list the table and column (attribute) names for each attribute group. For more information see "tacmd createSit" on page 54 for an explanation for creating situation formulas.

CLI example

The following example creates a collection called newCollection for the BIX InterChange Server Memory attribute group:

```
tacmd histcreatecollection -a newCollection -t BIX
-o "BIX InterChange Server Memory"
```

The following example creates a collection called newCollection for the AMN TMW DNS Response Time attribute group with a description, a collection interval, a collection location, and a warehouse interval:

```
tacmd histcreatecollection -a newCollection -t AMN -o "AMN TMW DNS Response Time"
-e "Response Time" -c 30m -l TEMS -i 12h
```

The following example creates a collection with a specified warehouse interval of 15 minutes and a collection interval of 30 minutes:

```
tacmd histcreatecollection -a newCollection -t KNT -o "DNS Query" -i 15m -c 30m
```

The following example creates a collection with a formula used to filter the data stored at the binary file:

```
tacmd histcreatecollection -a coll1 -t nt -o "Active Server Pages"
-f "*IF ( ( *VALUE Active_Server_Pages.System_Name *EQ Test ) )"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histdeletecollection

Description

Use the **tacmd histdeletecollection** command to delete the given collection by using specified inputs.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histdeletecollection** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histdeletecollection
{-a|--name} *COLLECTIONNAME*
[{-s|--server} *TEPSHOSTNAME***]**
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**

where:

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is used if you do not specify one.

-u|--userid
Specifies the user to authenticate. The software prompts you for the username if you do not specify one.

-a|--name
Specifies the name or the ID of the collection to remove.

-w|--password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

CLI example

The following example deletes a collection called newCollection:

```
tacmd histdeletecollection -a newCollection
```

The following example shows all of the options for this command:

```
tacmd histdeletecollection -a newCollection -s "ipTEPS" -u "username" -w "password"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histeditcollection

Description

Use the **tacmd histeditcollection** command to edit the given collection by using specified inputs.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histeditcollection** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histeditcollection
{-a | --name} *COLLECTIONNAME*
[{-n | --newname} *DESCRIPTION***]**
[{-e | --description} *DESCRIPTION***]**
[{-c | --collectioninterval} *COLLECTIONINTERVAL***]**
[{-l | --collectionlocation} *COLLECTIONLOCATION***]**
[{-i | --warehouseinterval} *WAREHOUSEINTERVAL***]**
[{-f | --filter} *FILTERFORMULA***]**
[{-s | --server} *TEPSHOSTNAME***]**
[{-u | --userid} *TEPS_USERID***]**
[{-w | --password} *TEPS_PASSWORD***]**

where:

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken if you do not specify one.

-u | --userid
Specifies the user to authenticate. The software prompts you for the username if you do not specify one.

-w | --password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-a | --name
Specifies the name or the ID of the historical collection to edit.

-n | --newname
Specifies the new name to be used for indicated collection. A value of 256 bytes is the maximum value.

-e | --description
Specifies the description of the named historical collection to edit. A value of 64 bytes is the maximum value.

-c | --collectioninterval
Specifies the collection interval. The acceptable inputs are 1m (1 minute), 5m, 15m, 30m, 1h (1 hour), or 1d (1 day). The default value is 15m (15 minutes).

-l|--collectionlocation
Specifies where the data collection is stored. The acceptable inputs are TEMA or TEMS. The default value is TEMA.

-i|--warehouseinterval
Specifies the warehouse interval. The acceptable inputs are 15m (15 minutes), 30m, 1h (1 hour), 12h, 1d (1 day), or off. The default value is 1d (1 day).

-f|--filter
Specifies the formula used to filter data stored at the binary file. You can remove an existing filter for the indicated collection by specifying this option.

CLI example

The following example edits a collection called newCollection:

```
tacmd histeditcollection -a newCollection
```

The following example edits a collection called NewCollection to have a new name, New_Coll, a description, a collection interval of 1 hour, a collection location of TEMS, and a warehouse interval of 12 hours:

```
tacmd histeditcollection -a NewCollection -n New_Coll -e description
-c 1h -l TEMS -i 12h
```

The following example edits a collection with a formula used to filter the data stored at the binary file:

```
tacmd histeditcollection -a coll1
-f "*IF ( ( *VALUE Active_Server_Pages.System_Name *EQ Test ) )"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histlistattributegroups

Description

Use the **tacmd histlistattributegroups** command to list all of the attribute groups for the specified product name. Use the -v|--verbose option to display the names of the collections that are defined for each attribute group. Use the **tacmd histlistcollections** with the -t and -o options to view which collections are started for an attribute group.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histlistattributegroups** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histlistattributegroups
{-t|--type} *PRODUCTTYPE*
[{-v|--verbose}]
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

where:

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken.

-u|--userid
Specifies the user to authenticate. If not specified, you are prompted to give the Tivoli Enterprise Portal Server log in username. The software prompts you for the User ID if you do not specify one. The User ID specified must have Configure History permission.

-w|--password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-t|--type
Specifies the product code or product name to list its attribute groups. Use the **tacmd histlistproduct** command to determine the codes and names.

-v|--verbose
Displays the names of the collections defined for each attribute group. If this argument is specified, the command output display in a linear format instead of a table format.

CLI example

This example lists all attribute groups for the specified product name:

```
   tacmd histlistattributegroups -s LEVER2 –t "Windows OS"
-u Administrator -w tivoli123$
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histlistcollections

Description

Use the **tacmd histlistcollections** command to list all the collections that are started for a managed system, all the collections that are defined for a product, or all that are defined for an attribute group.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the

histlistcollections command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histlistcollections
{-m|--system} *MANAGED_SYSTEM*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

tacmd histlistcollections
{-t|--type} *PRODUCTTYPE*
[{-o|--object} *ATTRIBUTEGROUPNAME...***]**
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

where:

-m|--system
Specifies the managed system for which collections should be listed.

-t|--type
Specifies the product code or product name of the product to which the attribute group specified by -o|--object belongs. Use the **tacmd histlistproduct** command to determine the codes and names.

-o|--object
Specifies the attribute group name for which collections should be listed. Use the **tacmd histlistattributegroups** command to find the correct values to use.

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken if you do not specify one.

-u|--userid
Specifies the user to authenticate. The software prompts you for the username if you do not specify one.

-w|--password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

The output of histlistcollections shows the defined filter, if any. For example:

```
tacmd histlistcollections -t nt -o "Active Server Pages"
KUIHLC001I Validating user credentials...
Collection Name        : coll1
Collection ID          : KNT_ACTSRVPG
Description            :
Product Name           : Windows OS
Attribute Group Name   : Active Server Pages
Collection Location    : TEMA
Status                 : Not started
Filter                 : *IF ( ( *VALUE Active Server Pages.ORIGINNODE *EQ Test ) )
```

CLI example

The following example lists a collection for the LZ product code:

```
histlistcollections -t KLZ
```

The following example lists a collection for the specified managed system name, LZ:

```
histlistcollections -m managedSystemName:LZ
```

The following example lists a collection for the LZ product code and the attribute group, Linux Disk:

```
histlistcollections -t KLZ -o "Linux Disk"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histlistproduct

Description

Use the **tacmd histlistproduct** command to list all of the products available for the historical data collection and configuration feature.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histlistproduct** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histlistproduct
[{-u|--userid} ***TEPS_USERID*****]**
[{-w|--password} ***TEPS_PASSWORD*****]**
[{-s|--server} ***TEPSHOSTNAME*****]**

where:

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken.

-u|--userid
Specifies the user to authenticate. The software prompts you for the User ID if you do not specify one. The User ID specified must have Configure History permission.

-w|--password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

CLI example

This example lists all the products available for historical data collection and configuration:

```
tacmd histlistproduct -s LEVER2 -u Administrator -w tivoli123$
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histstartcollection

Description

Use the **tacmd histstartcollection** command to start the data collection in the warehouse database for specified attribute group name(s).

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histstartcollection** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: To start historical collection for a historical collection group (and therefore the historical collections in the group), use the **tacmd editgroup** command to edit the group's distribution list.

CLI syntax

tacmd histstartcollection
{-t|--type} *PRODUCTTYPE*
{-o|--object} *ATTRIBUTEGROUPNAME...*
{-n|--temsname} *TEMSNAME...*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

tacmd histstartcollection
{-a|--collection} *COLLECTIONNAME...*
{-m|--system} *MSN_OR_MSL...*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

tacmd histstartcollection
{-a|--collection} *COLLECTIONNAME...*
{-n|--temsname} *TEMS_NAME...*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPSHOSTNAME***]**

where:

-a | --collection
Specifies the collection group names to be started.

-m | --system
Specifies the managed systems or managed system groups names to be started.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken.

-u | --userid
Specifies the user to authenticate. The software prompts you for the User ID if you do not specify one. The User ID specified must have Configure History permission.

-w | --password
Specifies the password of the user to authenticate. If not specified, you are prompted to give the password. The software prompts you for the password if you do not specify one.

-t | --type
Specifies the product code or product name to list its attribute groups. Use the **tacmd histlistproduct** command to determine the codes and names.

-o | --object
Specifies the attribute group names for collection to be started. When multiple attribute groups are specified, each attribute group must be separated by a space. Use the **tacmd histlistattributegroups** command to find the correct values to use.

-n | --temsname
Specifies the Tivoli Enterprise Monitoring Server names for collection to be started. If the names are not defined, it will take the Tivoli Enterprise Monitoring Server name where the Tivoli Enterprise Portal Server is connected. If more than one Tivoli Enterprise Monitoring Server exists in the environment, this argument is mandatory. When multiple Tivoli Enterprise Monitoring Servers are specified, each monitoring server must be separated by a space.

CLI example

This example start the historical data collection for the attribute group "NT Processor":

```
tacmd histstartcollection -s LEVER2 -t "Windows OS" -o "NT Processor "
-n HUB_LEVER2 -u Administrator  -w tivoli123$
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histstopcollection

Description

Use the **tacmd histstopcollection** command to stop the given attribute groups using provided inputs for historical data collection.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histstopcollection** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: This command cannot be used to stop historical collections that are started through a historical collection group. The distribution for the historical collection group must be modified to stop collections that are performed through the group. Use the **tacmd editGroup** command to modify the historical collection group.

CLI syntax

tacmd histstopcollection
{-t|--type} *PRODUCTTYPE*
{-o|--object} *ATTRIBUTEGROUPNAME(S)*
{-n|--temsname} *TEMSNAME(S)*
[{-u|--userid} *TEPS_USERID*]
[{-w|--password} *TEPS_PASSWORD*]
[{-s|--server} *TEPSHOSTNAME*]

tacmd histstopcollection
{-a|--collection} *COLLECTIONNAME...*
{-m|--system} *MSN_OR_MSL...*
[{-u|--userid} *TEPS_USERID*]
[{-w|--password} *TEPS_PASSWORD*]
[{-s|--server} *TEPSHOSTNAME*]

tacmd histstopcollection
{-a|--collection} *COLLECTIONNAME...*
{-n|--temsname} *TEMS_NAME...*
[{-u|--userid} *TEPS_USERID*]
[{-w|--password} *TEPS_PASSWORD*]
[{-s|--server} *TEPSHOSTNAME*]

where:

-a|--collection
Specifies the collection group names to be stopped.

-m|--system
Specifies the managed systems or managed system groups names to be stopped.

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken.

-u|--userid
Specifies the user to authenticate. If not specified, you are prompted to give the Tivoli Enterprise Portal Server log in username. The software

prompts you for the User ID if you do not specify one. The User ID specified must have Configure History permission.

-w | --password
Specifies the password of the user to authenticate. If not specified, you are prompted to give the password. The software prompts you for the password if you do not specify one.

-t | --type
Specifies the product code or product name to list its attribute groups. Use the **tacmd histlistproduct** command to determine the codes and names.

-o | --object
Specifies the attribute group names to be stopped. When multiple attribute groups are specified, each attribute group must be separated by a space. Use the **tacmd histlistattributegroups** command to find the correct values to use.

-n | --temsname
Specifies the Tivoli Enterprise Monitoring Server names for collection to be stopped. If the names are not defined, it will take the Tivoli Enterprise Monitoring Server name where the Tivoli Enterprise Portal Server is connected. If more than one Tivoli Enterprise Monitoring Server exists in the environment, this argument is mandatory. When multiple Tivoli Enterprise Monitoring Servers are specified, each monitoring server must be separated by a space.

CLI example

This example stop the historical data collection for the attribute group "NT Processor":

```
   tacmd histstopcollection -s LEVER2 -t "Windows OS" -o "NT Processor "
-n HUB_LEVER2 -u Administrator  -w tivoli123$
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histunconfiguregroups

Description

Use the **tacmd histunconfiguregroups** command to unconfigure the given attribute group or groups. The command unconfigures the collection settings of the default collection for each attribute group unless you specify the -m | --summarizationonly option.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histunconfiguregroups** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histunconfiguregroups
{-t | --type} *PRODUCTCODE*
{-o | --object} *ATTRIBUTEGROUPNAME...*
[{-m | --type} *summarizationonly***]**
[{-u | --userid} *TEPS_USERID***]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPSHOSTNAME***]**

where:

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken.

-u | --userid
Specifies the user to authenticate. The software prompts you for the User ID if you do not specify one. The User ID specified must have Configure History permission.

-m | --summarizationonly
Specifies the flag for configuring only the summarization and pruning settings.

-w | --password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-t | --type
Specifies the product code or product name to list its attribute groups. Use the **tacmd histlistproduct** command to determine the codes and names.

-o | --object
Specifies the attribute group name or names to be unconfigured. When multiple attribute groups are specified, each attribute group must be separated by a space. Use the **tacmd histlistattributegroups** command to find the correct values to use.

CLI example

This example unconfigures the attribute group "NT Processor" for historical data collection:

```
tacmd histunconfiguregroups -s LEVER2 -t "Windows OS" -o "NT Processor "
-u Administrator  -w tivoli123$
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histviewattributegroup

Description

Use the **tacmd histviewattributegroup** command to display the historical configuration information of the specified attribute group. This command is used to display the summarization and pruning settings for the specified attribute group and the names of the collections defined for the specified attribute group. Use the **tacmd histviewcollection** command to view the configuration details for a specific collection.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histviewattributegroup** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histviewattributegroup
{-t | --type} *PRODUCTTYPE*
{-o | --object} *ATTRIBUTEGROUPNAME*
[{-v | --verbose}]
[{-u | --userid} *TEPS_USERID*]
[{-w | --password} *TEPS_PASSWORD*]
[{-s | --server} *TEPSHOSTNAME*]

where:

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken.

-u | --userid
Specifies the user to authenticate. The software prompts you for the User ID if you do not specify one. The User ID specified must have Configure History permission.

-w | --password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-t | --type
Specifies the product code or product name to list its attribute groups. Use the **tacmd histlistproduct** command to determine the codes and names.

-o | --object
Specifies the attribute group name. Use the **tacmd histlistattributegroups** command to find the correct values to use.

-v | --verbose
Displays the table name and the attributes names defined for the specified attribute group.

CLI example

This example displays the historical configuration information of the specified attribute group:

```
tacmd histviewattributegroup -s LEVER2 -t "Windows OS" -o "NT Processor "
-u Administrator  -w tivoli123$
```

This example displays the historical configuration information of the specified attribute group, and also displays the table name and the attributes names defined for the specified attribute group:

```
tacmd histviewattributegroup -t "Windows OS" -o "NT Processor " -v
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd histviewcollection

Description

Use the **tacmd histviewcollection** command to display the configuration information of a specified collection.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **histviewcollection** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd histviewcollection
{-a|--name} *COLLECTIONNAME*
[{-u|--userid} *TEPS_USERID***]**
[{-s|--server} *TEPSHOSTNAME***]**
[{-w|--password} *TEPS_PASSWORD***]**

where:

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname to log on to. By default, the local hostname is taken if you do not specify one.

-u|--userid
Specifies the user to authenticate. The software prompts you for the username if you do not specify one.

-w|--password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-a|--name
Specifies the name or the ID of the collection for which to display historical data collection configuration information.

The output of histviewcollection shows the defined filter defined, if any. For example:

```
tacmd histviewcollection -a coll1
KUIHVC001I Validating user credentials...
Collection Name        : coll1
Collection ID          : KNT_ACTSRVPG
Description            :
Attribute Group Name   : Active Server Pages
Product Name           : Windows OS
Collection Interval    : 15 mins
Collection Location    : TEMA
Warehouse Interval     : 1 day
Status                 : Not started
Filter                 : *IF ( ( *VALUE Active Server Pages.ORIGINNODE *EQ Test ) )
```

CLI example

The following example displays a collection called newCollection:

```
tacmd histviewcollection -a newCollection
```

The following example includes all of the options for this command:

```
tacmd histviewcollection -a NewCollection -s "ipTEPS" -u "username" -w "password"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd importCalendarEntries

Description

Use the **tacmd importCalendarEntries** command to import all the calendar entries available in specified XML file to the Tivoli Enterprise Portal Server. You can optionally specify one or more calendar entry names to be imported by using the -n|--name|--names option.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **importCalendarEntries** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd importCalendarEntries
{-x|--file} *XMLFILE*
[{-u|--userid} *TEPS_USER***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPS_HOSTNAME***]**
[{-n|--name|--names} *CALENDAR_ENTRY_NAME...***]**
[{-f|--force} **]**

where:

-x | --file
Specifies the name of the xml file accessible to the local file system where the calendar entry names are imported from. The file name can either be a relative or absolute file name.

-u | --userid
Specifies the identifier of the user to authenticate on the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password of the user to authenticate on the Tivoli Enterprise Portal Server.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname where the calendar entries has to be imported.

-n | --name | --names
Specifies the name of the calendar entries to import.

-f | --force
Imports the calendar entries without prompting for confirmation.

CLI example

This example imports all the calendar entries from the XML file `All_calendarentries.xml` to the Tivoli Enterprise Portal Server running on the localhost:

```
tacmd  importCalendarEntries -x D:\IBM\ITM\BIN\All_Calendarentries.xml
-u sysadmin -w *****
```

This example imports all the calendar entries from the XML file `All_calendarentries.xml` to the Tivoli Enterprise Portal Server running on LEVER2:

```
tacmd  importCalendarEntries -x D:\IBM\ITM\BIN\All_Calendarentries.xml
-u sysadmin -w ***** -s LEVER2
```

This example imports only the calendar entries matching the names time_entries or task_entries from the XML file `BackUp_Calendarentries.xml` to the Tivoli Enterprise Portal Server running on LEVER2:

```
tacmd  importCalendarEntries -x D:\IBM\ITM\BIN\BackUp_Calendarentries.xml
-u sysadmin -w ****** -n "time_entries" " task_entries"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd importNavigator

Description

Use the **tacmd importNavigator** command to import a Tivoli Enterprise Portal custom navigator view, workspaces, queries, and situation associations from an XML file to the Tivoli Enterprise Portal Server.

When a navigator view is imported, the navigator is not assigned to any users. If the navigator already existed, the previously established user assignments are preserved. The **tacmd editUser** command can be used to assign the logical navigator to a user. Some important notes about using the **tacmd editUser** command:

- The **tacmd editUser** command explicitly assigns navigators, so you must be careful to append the new navigator to the list of navigators already assigned to you, or the other assignments for you will be lost.
- The Tivoli Enterprise Portal client does not receive a refresh event notification when assignments are made through the CLI, so a client restart is required to pick up the changes.
- The default navigator is displayed as the first navigator in the list assigned to the "NavigatorViews" property; in the following example, "VerizonCellular" is the default navigator view when the Tivoli Enterprise Portal client is restarted:

```
tacmd edituser -u sysadmin -w mypassword -i sysadmin
-p NavigatorViews=VerizonCellular,Physical
```

When a navigator is imported, by default all workspaces, workspace links, queries, situation associations, and managed system assignments (distributions) that are defined in the xml file are imported.

After the **tacmd importNavigator** command has been used to import a custom navigator that was exported from another environment, the **tacmd exportWorkspaces** and **tacmd importWorkspaces** commands can be used to import workspaces into the newly imported environment.

Note: If you are importing a workspace that has been viewed in the Tivoli Enterprise Portal previously, you must recycle the Tivoli Enterprise Portal to see the correct version. If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **importNavigator** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if these values are missing.

CLI syntax

tacmd importNavigator
{-x|--xmlFile} *XMLFILE*
[{-u|--username} *TEPS_USER***]**
[{-p|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPS_HOST[:PORT]]*
[{-o|--navigatorOnly}]
[{-f|--force}]

where:

-x|--xmlFile
Specifies the name of the XML file accessible to the local file system where the custom navigator view, workspaces, queries, and situation associations will be imported from. The file name can either be a relative or absolute file name.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
Specifies the identifier of the user to authenticate to the Tivoli Enterprise

Portal Server. The user must have the 'Modify' permission for the 'Custom Navigator Views' object, the 'Modify' permission for the 'Query' object, and the 'Workspace Administration Mode' and 'Workspace Author Mode' permissions for the 'Workspace Administration' object enabled on the server to execute the **importNavigator** command. The permissions for these objects are disabled by default for most users. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password of the user to authenticate to the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-o | --navigatorOnly
Specifies that only the custom navigator view is imported. Workspaces, queries, and situation associations in the XML file will not be imported.

-f | --force
Imports the custom navigator view without confirmation.

CLI example

The following example imports the custom navigator views specified from the file `importNavigator.xml` to the Tivoli Enterprise Portal Server on HDCHASDSTC0420:

```
tacmd importnavigator -x importNavigator.xml -s HDCHASDSTC0420
-u sysadmin -p ***** -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd importQueries

Description

Use the **tacmd importQueries** command to import Tivoli Enterprise Portal queries from an XML file to the Tivoli Enterprise Portal Server.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **importQueries** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd importQueries
{-x | --xmlFile} *XMLFILE*
[{-u | --username} *TEPS_USER***]**
[{-p | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOST[:PORT]***]**
[{-f | --force}]

where:

-x | --xmlFile
Specifies the name of the XML file accessible to the local file system where the query definitions are imported from. The file name can either be a relative or absolute file name.

-s | --server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
Specifies the identifier of the user to authenticate to the Tivoli Enterprise Portal Server. The user must have 'Modify' permissions for the 'Query' object enabled on the server to execute the **importQueries** command. The 'Modify' permission for the 'Query' object is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password of the user to authenticate to the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f | --force
Imports the queries without confirmation.

CLI example

The following example imports all the specified queries from the file `importQueries.xml` to the Tivoli Enterprise Portal Server on HDCHASDSTC0420:

```
tacmd importqueries -x importQueries.xml -s HDCHASDSTC0420 -u sysadmin -p ***** -f
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd importSitAssociations

Description

Use the **tacmd importSitAssociations** command to import all situation associations from an XML file to the Tivoli Enterprise Portal Server.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd importSitAssociations** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **importSitAssociations** command.

CLI syntax

tacmd importSitAssociations
{-x|--xmlFile} *XML_FILE*
[{-a|--navItem} *NAVIGATOR_ITEM***]**
[{-n|--navigator} *NAVIGATOR_NAME* **]**
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**
[{-f|--force} **]**

where:

-x|--xmlFile
The name of the xml file accessible to the local file system where the situation associations will be imported from. The file name can either be a relative or absolute file name.

-a|--navItem
The fully qualified name of the navigator item to import situation associations into. If this option is provided, navigator and navigator node object ID information in the situation association definition will be disregarded, and all situation associations in the XML file will be imported into the specified navigator item. If this option is not provided, situation associations will only be imported into the target navigator view if the object IDs for the navigator items in each situation association definition can be matched with a navigator item object ID for the specified target navigator view. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n|--navigator
The name of the navigator view that the navigator item belongs to. By default, the Physical navigator view is used.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Situation' object enabled on the server to execute the **importSitAssociations** command.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f|--force
Performs the action without prompting for confirmation.

CLI example

The following example imports situation associations to the Enterprise/child_logical navigator item that belongs to the Logical navigator by an xml file, exp_sit_assoc.xml:

```
tacmd importsitassociations -a Enterprise/child_logical -n Logical
-x exp_sit_assoc.xml
```

Return values

See "Return codes" on page 272

Related commands

"tacmd createSitAssociation" on page 58

"tacmd deleteSitAssociation" on page 77

"tacmd exportSitAssociations" on page 120

"tacmd listSitAssociations" on page 186

Return to Table 1 on page 5.

tacmd importSysAssignments

Description

Use the **tacmd importSysAssignments** command to import all managed system assignments from an XML file to the Tivoli Enterprise Portal Server. The command verifies that the system exists in the target Tivoli Monitoring environment. If the system is not in the the target Tivoli Monitoring environment, the command fails. Contrast this logic with that of the **tacmd importSitAssociations** command that you use to import all situation associations.

For the **tacmd importSysAssignments** command, a "system assignment" is a logical relation between a system and a navigator item that is used as the event indicator for situations. If you have no managed systems assigned to this navigator item, no events are displayed for it unless they are part of a roll-up display of events. In addition, the Situation editor will not be available from the menu. As a result, do not import a "system assignment" if the specified managed system does not exist on the target.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd importSysAssignments** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **importSysAssignments** command.

CLI syntax

tacmd importSysAssignments
{-x|--xmlFile} *XML_FILE*
[{-a|--navItem} *NAVIGATOR_ITEM***]**

{-n|--navigator} *NAVIGATOR_NAME*
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**
[{-f|--force}]

where:

-x|--xmlFile
The name of the xml file accessible to the local file system where the managed system assignments will be imported from. The file name can either be a relative or absolute file name.

-a|--navItem
The fully qualified name of the navigator item to import managed system assignments into. If this option is provided, navigator and navigator node object ID information in the situation association definition will be disregarded, and all managed system assignments in the XML file will be imported into the specified navigator item. If this option is not provided, managed system assignments will only be imported into the target navigator view if the object IDs for the navigator items in each situation association definition can be matched with a navigator item object ID for the specified target navigator view. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n|--navigator
The name of the navigator view the managed system assignments will be imported to. The Physical navigator view cannot be used; only custom navigator views are valid.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server. The user must have the 'Modify' permission for the 'Situation' object enabled on the server to execute the **importSysAssignments** command.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-f|--force
Performs the action without prompting for confirmation.

CLI example

The following example imports system assignments to the Enterprise/child_logical navigator item that belongs to the Logical navigator from the xml file, importsysassign.xml:

```
tacmd importsysassignments -a Enterprise/child_logical -n Logical
-x importsysassign.xml
```

Return values

See "Return codes" on page 272

Related commands

"tacmd createSysAssignment" on page 60

"tacmd deleteSysAssignment" on page 78

"tacmd exportSysAssignments" on page 121

"tacmd listSysAssignments" on page 188

Return to Table 1 on page 5.

tacmd importWorkspaces

Description

This command is used to import the Tivoli Enterprise Portal workspaces from an XML file into the Tivoli Enterprise Portal Server. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

Once the **tacmd importNavigator** command has been used to import a custom navigator that was exported from another environment, the **tacmd exportWorkspaces** and **tacmd importWorkspaces** commands can be used to import workspaces into the newly imported environment. Workspaces imported into a custom navigator environment are not visible in the Tivoli Enterprise Portal or Browser Client unless the custom navigator has been imported from the source environment for the workspaces.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **importWorkspaces** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: If you are importing a workspace that has been viewed in the Tivoli Enterprise Portal previously, you must recycle the Tivoli Enterprise Portal to see the correct version.

Note: The XML schema for workspaces was redesigned for the IBM Tivoli Monitoring v6.2 Fix Pack 1 release. Any workspace XML produced by the tacmd exportWorkspaces command before IBM Tivoli Monitoring v6.2 Fix Pack 1 is incompatible with the new format, and cannot be imported by using this command.

CLI syntax

tacmd importWorkspaces
{-x|--xmlFile} XMLFILE
[{-u|--username} TEPS_USER]
[{-p|--password} TEPS_PASSWORD]
[{-s|--server} TEPS_HOST[:PORT]]
[{-q|--queries}]
[{-f|--force}]

where:

-x|--xmlFile
Specifies the name of the XML file accessible to the local file system where the workspace definitions will be imported from. The file name can either be a relative or absolute file name.

-s|--server
Specifies a Tivoli Enterprise Portal Server to use. The *host* is a 32 or 64 bit IP address or hostname and *port* is an integer between 1 and 65536. If not specified, *host* defaults to localhost and *port* defaults to 15200.

-u|--username
Specifies the identifier of the user to authenticate on the Tivoli Enterprise Portal Server. Specify a valid string in the local locale. You must have both 'Workspace Administration Mode' and 'Workspace Author Mode' Workspace Administrator permissions enabled on the server to execute the **listWorkspaces** command. The 'Workspace Administration Mode' permission is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. Specify a valid string in the local locale. The software prompts you for the password if you do not specify one.

-q|--queries
Imports any queries from the XML file to the Tivoli Enterprise Portal Server.

-f|--force
Imports the workspace(s) without confirmation.

CLI example

This example imports workspaces from the file `all_lever_workspaces.xml` to the server located at myteps.ibm.com.

```
tacmd importworkspaces –s myteps.ibm.com –u imasample
                       –p mypassword –x all_lever_workspaces.xml
```

This example is the same scenario as the previous example, except that the force flag is used to suppress confirmation prompts.

```
tacmd importworkspaces –s myteps.ibm.com –u imasample
                       –p mypassword –x all_lever_workspaces.xml -f
```

This example imports workspaces from the file `all_lever_workspaces.xml` to the server located at myteps.ibm.com on port 1996. The user is prompted to enter the server username and password.

```
tacmd importworkspaces –s http://myteps.ibm.com:1996 –x all_lever_workspaces.xml
```

Return values

See “Return codes” on page 272.

Related commands

“tacmd listworkspaces” on page 196

“tacmd exportWorkspaces” on page 123

Return to Table 1 on page 5.

tacmd listAction

Description

Use the **tacmd listAction** command to display the list of the take action commands in the server. The action commands can be optionally filtered out by type. You can filter for a specified system type or for a list of specified system types. You must log in by using the **tacmd login** command before running the **tacmd listAction** command.

CLI syntax

tacmd listAction
[{-t|--type} ***TYPE*****]**

where:

-t|--type
One or more system types. Specifies a two-digit character code of the system type name to list the action.

CLI example

This example lists all the action commands present in the server.

```
tacmd listAction
```

This example lists all the action commands of type NT present in the server.

```
tacmd listAction -t NT
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listappinstallrecs

Description

Use the **tacmd listappinstallrecs** command to list the application support installation records. You can list the application support installation records specific to a monitoring server by product type or by version. You can choose to show

records in an error state only and also to show detailed information. You must log in by using the **tacmd login** command before running the **tacmd listappinstallrecs** command.

The following list describes the State fields for this command:

Install Request (IR)
Indicates a product application request has been initiated.

Install Metadata (IM)
Indicates a product application request is in progress and copying JAR files.

Metadata Complete (MC)
Indicates when finishing copying the JAR files, extracting the JAR files, and autorefresh is complete.

Install Complete (IC)
Indicates when seeding has finished.

MetaData Error (ME)
Indicates a terminal SDA installation error has occurred. The **tacmd listappinstallrecs** command's STATUS codes, depicted in Table 2, indicate either a terminal error (STATE=ME) or a retryable condition other than for the STATE=ME value. All subsequent attempts at reinstalling this product and version will be blocked until further action is taken. To re-attempt the agent installation, clear the SDA error record containing the ME state code using the **tacmd deleteappinstallrecs** command. This command can be run immediately after the SDA installation failure, or after you contact IBM Support for assistance in correcting the installation problem.

Table 2. STATUS codes

Value	Symbol (KFASDM_ST_*)	Explanation
1001	RequestQueued = 1001	Indicates a request queued, waiting for async response.
1002	Shortage	Indicates a memory shortage.
1003	BadArgument	Indicates a bad input argument.
1004	KFASDM_ST_NotThere	Indicates that a file was not found.
1005	SystemError	Indicates an unknown system error.
1006	Duplicate	Indicates that a request for same pc is already in progress.
1007	KT1_Error	Indicates a KT1 error.
1008	SDM_Disabled	Indicates that self-described processing is disabled at the monitoring server.
1009	HUB_NotThere	Indicates that the hub monitoring server is not available.
1010	Shutdown	Indicates that a monitoring server is shutting down.
1011	Manifest_Error	Indicates invalid content in the manifest file.
1012	Wrong_TEMS_Version	Indicates an incorrect monitoring server version.
1013	Nonsupportable_Feature	Indicates a required feature (for example, a new data type) is not supported by the monitoring server.
1014	UnKnown_Error	Indicates an unknown error.
1015	BadArgument_Length	Indicates a bad input argument length.

Table 2. STATUS codes (continued)

Value	Symbol (KFASDM_ST_*)	Explanation
1016	Manual_Install_Update	Indicates a record updated by the manual installation detection process.
1017	Temp_Install_Error	Indicates a temporary installation error; the agent retries the installation request.
1018	Refresh_Catalog_Error	Indicates a TEMS Auto-refresh error in catalog cache processing.
1019	Refresh_Attribute_Error	Indicates a TEMS Auto-refresh error in attribute cache processing
1020	Refresh_KFAOT_Error	Indicates a TEMS Auto-refresh error in OTEA cache processing.
1021	Server_TimedOut	Indicates that time expired at the RTEMS waiting for HUB SDA installation completion.
1022	SeedError	Indicates that a TEMS SDA seeding error occurred.
1023	SDM_NotInitialized	Indicates that a TEMS SDA was not properly initialized at TEMS startup.
1024	Install_Blocked	Indicates that a product SDA installation was blocked by user control.

CLI syntax

tacmd listappinstallrecs
[{-n | --temsname} ***SYSTEMS*****]**
[{-t | --type} ***PRODUCT_TYPE*****]**
[{-v | --version} ***PRODUCT_VERSION*****]**
[{-e | --errors}]
[{-d | --details}]

where:

-n | --temsname
Specifies one or more monitoring server names where the records are retrieved. If this option is not specified, the records are retrieved from all found monitoring servers.

-t | --type
Specifies one or more managed system types (product codes) to be listed. Note that multiple product codes should be space-delimited. For example, UX NT LZ.

-v | --version
Specifies the product version of the records to be listed.
PRODUCT_VERSION must be in the format XXXXXXXX (8 integers). For example, 06230000.

-e | --errors
Specifies showing records in error state only.

-d | --details
Shows a larger table with all the details of the installation records.

CLI example

The following example lists the application support installation records:

```
./tacmd listappinstallrecs

HUB/RTEMS PRODUCT VERSION  GRPID ID  IDVER SEEDSTATE STATE STATUS
RTEMS_1   A4      06300000 5655  TMS 06300000      0
RTEMS_1   HD      06300000 5655  TMS 06300000      0
RTEMS_1   UM      06230200 5655  TMS 06230200      0
```

The following example lists the application support installation records for the specified managed system type (UNIX):

```
./tacmd listappinstallrecs -t UX
HUB/RTEMS    PRODUCT VERSION  GRPID ID  IDVER SEEDSTATE STATE STATUS
HUB_TEMS     UX      06220200 5655  TMS 06220200              0
HUB_TEMS     UX      06230000 5655  TMS 06230000        MC    0
HUB_TEMS     UX      06230000 5655  TPS 06230000        IC    0
HUB_TEMS     UX      06230000 5655  TPW 06230000        IC    0
RTEMS_1      UX      06230000 5655  TMS 06230000        IC    0
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addSdaInstallOptions" on page 23

"tacmd deleteappinstallrecs" on page 67

"tacmd deleteSdaInstallOptions" on page 73

"tacmd deleteSdaOptions" on page 74

"tacmd editSdaInstallOptions" on page 93

"tacmd editSdaOptions" on page 95

"tacmd listSdaInstallOptions" on page 178

"tacmd listSdaOptions" on page 180

"tacmd listSdaStatus" on page 180

Return to Table 1 on page 5.

tacmd listBundles

Description

Use the **listBundles** command to display the details of one or more deployment bundles that are available for deployment to the local deployment depot. This command is also available for non-agent bundles. This command must be run locally from a Tivoli Enterprise Monitoring Server installation containing a depot.

Assuming the current OS user has the proper permissions, it is not necessary for a log in command to have been previously issued to execute this command.

CLI syntax

tacmd listBundles
{-i | --imagePath} *IMAGEPATH*
[{-t | --product | --products} *PRODUCT* **...]**
[{-p | --platform | --platforms} *PLATFORM* **...]**
[{-v | --version | --versions} *VERSION* **...]**

where:

-i | --imagePath
The directory that contains the deployment bundles to be listed.

-t | --product | --products
The product code or codes of the agents to list bundles for. This value corresponds to the value that is displayed in the *Product Code* field that is displayed by the **viewDepot** or **listBundles** command.

-p | --platform | --platforms
The platform code or codes of the products to list bundles for. This value corresponds to the value that is displayed in the *Host Type* field that is displayed by the **viewDepot** or **listBundles** command.

-v | --version | --versions
The version or versions of the bundles to add. This value corresponds to the value that is displayed in the *Version* field that is displayed by the **viewDepot** command

CLI example

This command displays details for all the deployment bundles in the *D:\cdimage\bundles* directory.

```
tacmd listBundles –i D:\cdimage\bundles
```

This command displays details for all the deployment bundles in the */mnt/bundles* directory where the bundle product type is *ux*, the bundle platform is *aix513*, and the bundle version is *060100000*.

```
tacmd listBundles –i /mnt/bundles –t ux –p aix513 –v 060100000
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addBundles" on page 17

"tacmd removeBundles" on page 208

"tacmd viewDepot" on page 243

Return to Table 1 on page 5.

tacmd listCalendarEntries

Description

Use the **tacmd listCalendarEntries** command to list an existing calendar entry on the Tivoli Enterprise Monitoring Server. You must log in by using the **login** command before running the **tacmd listCalendarEntries** command.

CLI syntax

tacmd listCalendarEntries

CLI example

The following example lists the calendar entry's details available in the server:

```
tacmd listCalendarEntries
Name: Clean_Temp
Type: CRON
Data: 30 21 * * SUN
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listEventDest

Description

Use the **tacmd listEventDest** command to display the server ID, name, and type for every event destination definition on the server.

CLI syntax

tacmd listEventDest

CLI example

This example displays the server ID, server name, and server type for every event destination server definition on the server:

```
tacmd listEventDest
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listGroups

Description

Use the **tacmd listGroups** command to display a list of known groups. It can optionally filter the list by type of the groups.

CLI syntax

tacmd listGroups
[-t|--grouptype] DEPLOY|BUNDLE|SITUATION|COLLECTION
[-v|--verbose]

where:

-t|--grouptype
Specifies the type of the groups to be listed. The valid types are DEPLOY, BUNDLE, SITUATION, and COLLECTION.

-v|--verbose
Specifies the description for each group.

CLI example

The following example displays the list of groups available on the server:

```
tacmd listGroups
```

The following example displays the list of all deployment groups available on the server:

```
tacmd listGroups -t DEPLOY
```

Return values

See Table 8 on page 272.

Return to Table 1 on page 5.

tacmd listNavigators

Description

Use the **tacmd listNavigators** command to display a list of Tivoli Enterprise Portal custom navigator views assigned to the specified Tivoli Enterprise Portal user on the Tivoli Enterprise Portal Server. This command displays the custom navigator name and description for each custom navigator view.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listNavigators** command. If you specify values for some, but not all of these options, are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listNavigators
[{-u|--username} *TEPS_USER*]

[{-p | --password} *TEPS_PASSWORD*]
[{-s | --server} *TEPS_HOST [: PORT]*]

where:

-s | --server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
Specifies the identifier of the user to authenticate to the Tivoli Enterprise Portal Server. You must have 'Modify' permissions for the 'Custom Navigator Views' object enabled on the server to execute the **listNavigators** command. The 'Modify' permission for the 'Custom Navigator Views' object is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password of the user to authenticate to the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

CLI example

The following example displays a list of custom navigator views are available for the user "sysadmin" on the Tivoli Enterprise Portal Server at HDCHASDSTC0420:

```
tacmd listNavigators -s HDCHASDSTC0420 -u sysadmin -p *****
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listOverrideableSits

Description

Use the **tacmd listOverrideableSits** command to display a list of override-eligible situations for a given application.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listOverrideableSits** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listOverrideableSits
{-t | --type} *TYPE*
[{-u | --userid} *TEPS_USERID*]
[{-w | --password} *TEPS_PASSWORD*]
[{-h | --tepshostname} *TEPS_HOSTNAME*]

where:

-t | --type
Specifies the application product type code to display override-eligible situations for.

-u | --userid
Specifies the existing user ID to log on to Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

CLI example

This example lists the situations that are eligible for overrides:

```
tacmd listoverrideablesits -u sysadmin -w ******** -t nt
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listOverrides

Description

Use the **tacmd listOverrides** command to display the situation overrides defined for a specified situation on a managed system or list of managed systems.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listOverrides** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listOverrides
{-s | --situation} *SITNAME*
{-m | --system} *SYSTEM | SYSTEM_LIST*
[{-u | --userid} *TEPS_USERID*]
[{-w | --password} *TEPS_PASSWORD*]
[{-h | --tepshostname} *TEPS_HOSTNAME*]
[{-f | --formula}]

where:

-s | --situation
Specifies the situation to display override definitions for. If you include either the & character or the < character in the situation name, you must use quotation marks around the name, for example, "abc&def" or "abc<def".

-m | --system
Specifies the name of the managed system or managed system group to display override definitions for.

-u | --userid
Specifies the existing user ID to log on to the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

-f | --formula
Causes the command to display the predicate and condition attribute names by using the situation formula name (for example, 'NT_Process.Process_Name') instead of the display name (for example, 'Process Name') that is used by the Tivoli Enterprise Portal.

CLI example

This example lists overrides for a managed system, where the overrides have associated calendar entries and key conditions:

```
tacmd listoverrides -u sysadmin -w ******** -s NT_NotesServerProcess
-m Primary:LEVER:NT
```

This example lists overrides for a managed system group:

```
tacmd listoverrides -u sysadmin -w ******** -s NT_Disk_Space_Low -m *NT_SYSTEM
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listQueries

Description

Use the **tacmd listQueries** command to display a list of Tivoli Enterprise Portal queries on the Tivoli Enterprise Portal Server. You can optionally filter the list by query names or product codes. The query name and product code are displayed for each query.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listQueries** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listQueries
[{-u | --username} *TEPS_USER***]**
[{-p | --password} *TEPS_PASSWORD***]**
[{-s | --server} *TEPS_HOST[:PORT]***]**

[{-q|--query} *QUERY* **...]**
[{-t|--type} *TYPE* **...]**
[{-e|--exclude}]

where:

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
Specifies the identifier of the user to authenticate to the Tivoli Enterprise Portal Server. The user must have 'Modify' permissions for the 'Query' object enabled on the server to execute the **importQueries** command. The 'Modify' permission for the 'Query' object is disabled by default for most users. The software prompts you for the username if you do not specify one.

-p|--password
Specifies the password of the user to authenticate to the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

-q|--query
Specifies the names of one or more queries to display.

-t|--type
Specifies an IBM Tivoli Monitoring product type code. If a 2-character type is entered, the letter 'k' will be prepended automatically to form a 3-character product type code.

-e|--exclude
Excludes the specified query names and also product types from the list operation.

CLI example

The following example displays a list of queries on the Tivoli Enterprise Portal Server at HDCHASDSTC0420:

```
tacmd listQueries -s HDCHASDSTC0420 -u sysadmin -p *****
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listSdaInstallOptions

Description

Use the **tacmd listSdaInstallOptions** command to list the product versions configured to be allowed for Self-Describing Agent (SDA) install. You can filter by product type and version. You must log in by using the **tacmd login** command before running the **tacmd listSdaInstallOptions** command.

CLI syntax

tacmd listSdaInstallOptions
[{-t|--type} *PRODUCT_TYPE*...]
[{-v|--version} *VERSION*]

where:

-t|--type
Lists the product versions configured to be allowed for SDA install for one or more managed system types (product types).

-v|--version
Lists the product versions configured to be allowed for SDA install that identify support for the version specified. If multiple versions are listed, they are separated by spaces. The version is an eight-digit identifier in the format *VVRRMMFF*, where *VV* specifies Version, *RR* specifies Release, *MM* specifies Modification, and *FF* specifies PTF Level. For example, the *VVRRMMFF* designation for ITM 623 Fix Pack 2 is 06230200.

CLI example

Run the following command to list all product versions configured to be allowed for SDA install:

```
tacmd listSdaInstallOptions
```

Run the following command to list the product versions configured to be allowed for SDA install for the NT product type:

```
tacmd listSdaInstallOptions -t NT
```

Run the following command to list the product versions configured to be allowed for SDA install for the DEFAULT and the Linux product type:

```
tacmd listSdaInstallOptions -t DEFAULT LZ
```

Run the following command to list the product versions configured to be allowed for SDA install for the ITM623 FP2 Windows and Linux product types:

```
tacmd listSdaInstallOptions -t NT LZ -v 06230200
```

Run the following command to list the product versions configured to be allowed for SDA install for all product types that support ITM623 FP2:

```
tacmd listSdaInstallOptions -v 06230200
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addSdaInstallOptions" on page 23

"tacmd deleteSdaInstallOptions" on page 73

"tacmd editSdaInstallOptions" on page 93

"tacmd listSdaOptions" on page 180

Return to Table 1 on page 5.

tacmd listSdaOptions

Description

Use the **tacmd listSdaOptions** command to list the self-described agent options.

When the application support for a product is applied to the hub monitoring server, commonly called *seeding*, the definitions are added to the hub and these are automatically propagated to any active remote monitoring server. Configuration for the SDA seeding for a product type specifies how the distribution targets for definitions are applied. The option to control the seeding is provided to prevent prior customization from being lost.

Note: You must log in by using the **tacmd login** command before running the **listSdaOptions** command.

CLI syntax

tacmd listSdaOptions [{-t|--type} *PRODUCT_TYPE* **...]**

where:

-t|--type *PRODUCT_TYPE ...*
Specifies one or more managed system types (product codes) to be listed, for example, UX, NT, and LZ. If the parameter is not specified, information for all product types is listed.

Return values

See Table 8 on page 272.

Related commands

"tacmd editSdaOptions" on page 95

"tacmd deleteSdaOptions" on page 74

"tacmd listappinstallrecs" on page 168

"tacmd deleteappinstallrecs" on page 67

Return to Table 1 on page 5.

tacmd listSdaStatus

Description

Use the **tacmd listSdastatus** command to display the SDA Enablement status for a monitoring server. You can display the SDA Enablement status for a list of monitoring servers or for all monitoring servers (default action). The SDA suspend state for the hub monitoring server is also provided if the command is issued to an ITM V6.3.0 (or later) hub monitoring server. The SDA suspend state defines the

SDA activity for all the monitoring servers attached to the hub monitoring server. See the **tacmd suspendSda** and **tacmd resumeSda** commands for more information.

You must log in by using the **tacmd login** command before running the **listSdaStatus** command.

SDA Enablement status is indicated by the values of *HUB/RTEMS name*, *STATE*, and *STATUS*:

HUB/RTEMS name
The SDA Enablement status for the managed server name listed.

STATE
The values for SDA STATE are ON, OFF or ERROR, where

- ON indicates that the server SDA function is active and operational.
- OFF indicates that the server SDA function is not active and not operational. The state might be OFF if the KMS_SDA environment variable has not been set to Y or because the hub monitoring server is not active.
- ERROR indicates that the server is configured for SDA. However, the server cannot perform SDA due to an error. The error might indicate a local error or if the server is a remote monitoring server, an error during synchronization or connection to the hub monitoring server. If the hub monitoring server's SDA Enablement status is ERROR, the remote monitoring server SDA Enablement status will not indicate ERROR, but instead, OFF.

STATUS
See Table 3 to determine SDA STATUS error codes, symbols, explanations, and your response.

Table 3. SDA STATUS error codes

Value	Symbol (KFASDM_CONFIG_*)	Explanation	Response
0	ST_Success	SDA is operational.	None.
1	SDM_InitError	An error occurred while SDA was initializing, updating the SDA_STATE record, or while retrieving or processing the Suspend record.	Contact Tivoli customer support.
2	ITM_HOME_ Missing	Indicates a non-MVS error only: on Windows - CANDLE_HOME undefined, on UNIX - CANDLEHOME undefined (possibly an installation error)	Contact Tivoli customer support.
3	MANIFEST_PATH_ Missing	The TEMS environment variable TEMS_MANIFEST_PATH is undefined or the value is the empty string.	This variable is defined by the installation program. If the configuration file has been edited, verify that the variable still exists and identifies a valid directory. If the predefined variable was not altered, contact Tivoli customer support.

Table 3. SDA STATUS error codes (continued)

Value	Symbol (KFASDM_CONFIG_*)	Explanation	Response
4	MANIFEST_PATH_ TooLong	The path name specified through the TEMS_MANIFEST_PATH environment variable exceeds 512 characters.	Redefine the manifest directory at a location whose path name is less than or equal to 512 characters.
5	MANIFEST_PATH_ NotFound	The directory specified through the TEMS_MANIFEST_PATH environment variable was not found.	This variable is defined by the installation program. If the configuration file has been edited, verify that the value identifies a valid directory. If the predefined variable was not altered, contact Tivoli customer support.
6	SDM_ENABLED_ UnknownValue	The value assigned to the KMS_SDA environment variable is not valid.	Assign a value of either Y|y (enabled) or N|n (disabled).
7	DISTREQMGR_ InitFailed	For remote TEMS only: the creation of the Distributed Request Monitor thread failed.	Contact Tivoli customer support.
8	NOTIFICATIONMGR_ InitFailed	The creation of the thread to monitor the hub's TAPPLOGT table failed. In the case of a remote TEMS, this error can also occur if the creation of the thread to monitor the remote TEMS TAPPLOGT table failed.	Contact Tivoli customer support.
9	REQUESTMGRTHREAD_ NotCreated	The creation of the Workload Manager thread failed.	Contact Tivoli customer support.
10	HUB_SDA_Disabled	For remote TEMS only: SDA has been disabled at the remote TEMS because SDA is disabled at the hub.	This is the default configuration in which the TEMS is installed. To activate SDA, enable it at the hub.
11	HUB_SDA_Error	For remote TEMS only: SDA has been disabled at the remote TEMS because SDA functions at the hub are inoperative.	Run the **listSdaStatus -n *HUB** command to determine the nature of the error at the hub. Then refer to this table for the appropriate response.
12	HUB_SDA_ Unknown	For remote TEMS only: SDA has been disabled because the state of SDA at the hub is unknown> This error might indicate a communication failure with the hub.	Once the remote TEMS re-establishes the connection to the hub, the error should be resolved. If not, contact Tivoli customer support.
13	HUB_SDA_ CommFailed	Unused.	Not applicable.
14	BROADCASTMGR_ InitFailed	The creation of the Broadcast Manager thread failed.	Contact Tivoli customer support.
15	BROADCASTMGRTHREAD_ NotCreated	Unused.	Not applicable.
16	SDA_No	The environment variable KMS_SDA has been set to N.	None.
17	CMS_FTO_ Configured	For ITM v6.2.3 Fix Pack 1 or later hub only: SDA has been disabled because FTO is enabled on the hub.	Prior to ITM v6.2.3 Fix Pack 1, FTO and SDA were mutually exclusive. Select one or the other or upgrade the hub to ITM v6.2.3 Fix Pack 1 or later.

CLI syntax

tacmd listSdaStatus
[{-n |--temsname} TEMS...]

where:

-n|--temsname

Specify one or more monitoring server names to display SDA Enablement status. If this option is not specified, the SDA Enablement status is displayed for all online monitoring servers that are able to perform SDA.

If a monitoring server specified by the –n|--temsname option is not online, is not SDA capable, or is not registered with the HUB, then a warning is displayed.

CLI example

The following example lists the SDA Enablement status for a specific set of monitoring servers.

Note: The ITM V6.3.0 hub monitoring server's output includes the SDA suspend state only if the monitoring server is suspended.

```
tacmd listsdastatus –n REMOTE_1 REMOTE_2 REMOTE_3

KUILSS200W: REMOTE_2 is not online and SDA enablement cannot be provided.

HUB/RTEMS   STATE    STATUS
REMOTE_1    ON       0
REMOTE_3    OFF      16
```

The following example lists the SDA Enablement status for a set of specified monitoring servers. In this example, assume that the second remote monitoring server is REMOTE_B. The error message reflects the incorrect name listed in the command.

```
tacmd listsdastatus –n REMOTE_A REMOTEB

KUILSS202W: TEMS name : REMOTEB is not connected to this HUB. Verify the input
is correct and retry.

HUB/RTEMS   STATE   STATUS
REMOTE_A    ON      0
```

The following example lists the SDA Enablement for a set of monitoring servers that are SDA capable and online. The acting hub monitoring server (ITM V6.3.0) is HUB_A. HUB_A is configured for FTO. HUB_B is the FTO peer. The current suspend state is ON. The environment includes several remote monitoring servers including versions above V6.3.0. The uplevel version remote monitoring server has a new SDA failure STATUS.

```
tacmd listsdastatus

KUILSS203I:  SDA functions are suspended.
.

HUB/RTEMS     STATE     STATUS
HUB_A         ON        0
```

```
HUB_B          ON        0
RTEMS_6.3.0    ON        0
RTEMS_6.2.3    ON        0
RTEMS_6.3.1    ERROR     18
```

Related commands

"tacmd suspendSda" on page 236

"tacmd listappinstallrecs" on page 168

"tacmd resumeSda" on page 216

Return to Table 1 on page 5.

tacmd listSit

Description

The **tacmd listSit** command lists the defined situations on the hub monitoring server. You can optionally filter the list for those distributed to a particular managed system or managed system type. You must log in by using the **login** command before running the **listsit** command.

Note:

1. The correct name to use in commands for the Unix Logs agent is "Unix Logs". "Monitoring agent for Unix Logs" has been superseded.
2. You cannot use this command to list UADVISOR situations.

CLI syntax

tacmd listSit
[{-d|--delim} ***DELIM***
[{-n|--noheader}] | [{-l|--linear}]

tacmd listSit
[{-d|--delim} DELIM]
[{-n|--noheader}] | [{-l|--linear}]
{-m|--system} ***SYSTEM***

tacmd listSit
[{-d|--delim} ***DELIM*****]**
[{-n|--noheader}] | [{-l|--linear}]
{-t|--type} ***TYPE***

where:

-m|--system|--systems
Specifies a managed system name and restricts the list of situations to those distributed to the managed system or managed systems specified. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-t|--type|--types
Specifies a managed system type and restricts the situation list to those of

the specified managed system type name or names. Use **tacmd listsystemlist** to obtain a list of valid managed system types.

-d|--delim
Separates the fields with the delimiter. You can specify a delimiter character of a comma (,), colon (:), semicolon (;), asterisk (*), hashmark (#), dollar ($), exclamation (!), or tilde (~).

-l|--linear
Specifies to display in linear format with long situation name.

-n|--noheader
Excludes a header line from the list.

CLI example

This command lists all situations on the monitoring server separated by commas and without a header:

```
tacmd listSit –d "," -n
```

The result is:

```
Candle Management Server,QOMEGAMON_ONLINE
Generic Configuration,NonPrimeShift
Generic Configuration,PrimeShift
Generic Configuration,Weekday
Windows NT,NT_Available_Bytes_Critical
Windows NT,NT_Available_Bytes_Warning
Windows NT,NT_Bottleneck_Disk
```

This command lists all the Universal Database situations:

```
tacmd listSit –-type "Universal Database"
```

The result is:

```
TYPE            NAME
Universal Database UDB_Database_Lock_Warning
Universal Database UDB_Status_Warning
```

This command displays the situation Name, Type and Full Name in linear format:

```
tacmd listsit -t "Windows OS" –l
```

The result is:

```
Name     : NT_Process_Memo48C503654ED3AE16
Type     : Windows OS
Full Name: NT Process Memory Critical

Name     : NT_Logical_Disk48C8FF4D78379BB6
Type     : Windows OS
Full Name: NT Logical Disk Space Warning
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSit” on page 54

“tacmd deleteSit” on page 76

"tacmd editSit" on page 98

"tacmd viewSit" on page 247

Return to Table 1 on page 5.

tacmd listSitAssociations

Description

Use the **tacmd listSitAssociations** command to display a list of all situations associated with or eligible for association with a Tivoli Enterprise Portal navigator item.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd listSitAssociations** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **listSitAssociations** command.

CLI syntax

tacmd listSitAssociations
{-a|--navItem} *NAVIGATOR_ITEM*
[{-n|--navigator} *NAVIGATOR_NAME* **]**
[{-e|--eligible}]
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**

where:

-a|--navItem
The fully qualified name of the navigator item to display situation associations for. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n|--navigator
The name of the navigator view that the navigator item belongs to. By default, the Physical navigator view is used.

-e|--eligible
Display the names of all situations that are eligible for association with the navigator item. If this option is not provided, the command displays only the situations that are currently associated with the navigator item.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u | --username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server.

-p | --password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

CLI example

The following example displays the names of all situations that are eligible for association with the Enterprise/child_logical navigator item:

```
tacmd listsitassociations -a Enterprise/child_logical -n Logical -e
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSitAssociation” on page 58

“tacmd deleteSitAssociation” on page 77

“tacmd exportSitAssociations” on page 120

“tacmd importSitAssociations” on page 162

Return to Table 1 on page 5.

tacmd listSitAttributes

Description

The **tacmd listSitAttributes** command lists the attribute names that are eligible for use with dynamic thresholding (override) commands for a given situation. The command distinguishes between attributes that can be used as part of a predicate expression and attributes that can be used as part of a condition expression.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listSitAttributes** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: The total number of characters used in all the expression overrides defined for a situation should not exceed 4000 bytes. The actual size requirement for a single override varies depending on the names and values of the key columns and the override expression. In one case the limit might be 25 or, in a simpler case, it might be higher. The symptom of exceeding the 4000-byte limit is that the overrides do not work and the monitoring server trace log shows an "exceeds limit 4000" override error.

CLI syntax

tacmd listSitAttributes
{-s | --situation} *SITNAME*
[{-u | --userid} *TEPS_USERID*]
[{-w | --password} *TEPS_PASSWORD*]
[{-h | --tepshostname} *TEPS_HOSTNAME*]

where:

-s | --situation
Specifies the situation to display attribute names for. If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-u | --userid
Specifies the existing User ID to log on to the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

CLI example

This example lists the attributes eligible to be used for overrides and as a key or condition:

```
tacmd listsitattributes -u sysadmin -w ******** -s NT_NotesServerProcess
```

Return values

See “Return codes” on page 272

Related commands

Return to Table 1 on page 5.

tacmd listSysAssignments

Description

Use the **tacmd listSysAssignments** command to display a list of managed systems or managed system lists that are assigned to a Tivoli Enterprise Portal navigator item.

If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd listSysAssignments** command. If you specify values for some, but not all of these options, you might be prompted to specify these values.

You must log in by using the **tepsLogin** command before running the **listSysAssignments** command.

CLI syntax

tacmd listSysAssignments
{-a|--navItem} *NAVIGATOR_ITEM*
{-n|--navigator} *NAVIGATOR_NAME*
[{-s|--server} *TEPS_HOSTNAME[:PORT]* **]**
[{-u|--username} *TEPS_USER* **]**
[{-p|--password} *TEPS_PASSWORD* **]**

where:

-a|--navItem
The fully qualified name of the navigator item to display assigned managed systems or managed system lists for. List the navigator item starting with the root node of the navigator view to the navigator item, separating each navigator node with a forward slash character (/). For example, `"Enterprise/Windows Systems/MYHOST"`. As in the example, on Windows systems, you must put double quotation marks around the name of the navigator item if it contains a space. If the navigator item contains the forward slash character (for example, `Trunk/TreeBranch1/TreeBranch2/TreeBranch3`), a sequence of two consecutive forward slash characters will serve as an escape sequence, as in, `Trunk/TreeBranch1/TreeBranch2/Tree//Branch//3`.

-n|--navigator
The name of the navigator view that the navigator item belongs to.

-s|--server
Specifies which Tivoli Enterprise Portal Server to use.

-u|--username
The identifier of the user to authenticate on the Tivoli Enterprise Portal Server.

-p|--password
The password of the user to authenticate on the Tivoli Enterprise Portal Server. The software prompts you for the password if you do not specify one.

CLI example

The following example displays a list of managed systems or managed system lists that are assigned to the Enterprise/child_logical navigator item:

```
tacmd listsysassignments -a Enterprise/child_logical  -n Logical
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSysAssignment” on page 60

“tacmd deleteSysAssignment” on page 78

“tacmd exportSysAssignments” on page 121

“tacmd importSysAssignments” on page 164

Return to Table 1 on page 5.

tacmd listsystemlist

Description

This command lists the available managed system groups. You can filter for a specified managed system type or for a list of specified managed system types.

Note: The correct name to use in commands for the Unix Logs agent is "Unix Logs". "Monitoring agent for Unix Logs" has been superseded.

CLI syntax

tacmd listsystemlist
[{-d|--delim} ***DELIM*****]**
[{-t|--type|--types} ***TYPE*****]**

where:

-t|--type|--types
One or more managed system types. Specify a string for the managed system type name or its associated 2-character code. The string might consist of letters (upper or lower case), numbers, underscores (_), slashes (/), left parenthesis "(", right parenthesis ")", or spaces ( ). If not specified, list all the managed system groups available.

-d|--delim
Use this string to separate the fields. You can specify a delimiter character of a comma (,), colon (:), semicolon (;), asterisk (*), hashmark (#), dollar ($), exclamation (!), or tilde (~). If not specified, use one or more tabs to separate the columns so they line up.

CLI example

This example lists the managed system group catalog.

```
tacmd listsystemlist
```

Return values

See “Return codes” on page 272.

Related commands

“tacmd createsystemlist” on page 62

“tacmd editsystemlist” on page 100

“tacmd deletesystemlist” on page 80

“tacmd viewsystemlist” on page 248

Return to Table 1 on page 5.

tacmd listSystems

Description

Use the **tacmd listSystems** command to display a list of known managed systems, optionally filtering for the list by node, server name, and product codes. The managed system name, product code, version, and status are displayed for each managed system. When detail option is provided, a new field, **IBM**, is added in the output display that states 'Yes' if the managed system is a standard IBM managed system, and states 'No' in the case of a non-standard IBM managed system.

Note: Extended version information for every agent might not always be available. When this happens, the last two digits of the version displayed are represented as "XX". This occurs for subnode agents or when agents are not enabled for Agent Deploy support.

CLI syntax

tacmd listSystems
[{-n|--node} *MANAGED-OS***]**
[{-t|--type|--types} *TYPE* **...]**
[{-ns|--nonstandard} *NON-STANDARD* **]**
[{-d|--detail} *DETAIL* **]**

tacmd listSystems
[{-t|--type|--types} *TYPE* **...]**
[{-s|--server|--servers} *SERVER* **...]**
[{-ns|--nonstandard} *NON-STANDARD* **]**
[{-d|--detail} *DETAIL* **]**

where:

-n|--node
Identifies the node, the directory on monitoring system where the OS agent is installed, for which you want to list the agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, `stone.ibm.com:LZ` is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-t|--type|--types
Identifies one or more *product type* codes for which to filter. The value of *type* is scoped to the *node* level.

-s|--server|--servers
The identifiers of one or more servers to filter for. Each server must connect to the hub server you are logged on to.

-ns|--nonstandard
Identifiers of the non-standard IBM managed system to filter for.

-d|--detail
Displays the output in detail.

CLI example

This command lists all of the systems in your enterprise.

```
tacmd listSystems
```

This command lists all of the systems in your enterprise with the product code *UM* (universal agent systems).

```
tacmd listSystems –t UM
```

This command lists all of the systems in your enterprise with the product code *NT* (NT nodes or operating system agents). This command is effective for listing all of the NT nodes in your enterprise.

```
tacmd listSystems –t NT
```

This command lists all of the systems in your enterprise with product codes *NT*, *LZ*, or *UX* (NT operating system agents). This command is effective for listing all of the nodes in your enterprise.

```
tacmd listSystems –t NT UX LZ
```

This command lists all of the systems on node *Primary:STONE:NT* with the product code *UM* (universal agent).

```
tacmd listSystems –n Primary:STONE:NT –t UM
```

This command lists all of the systems on remote monitoring server *REMOTE_TEMS5* with the product code *UM* (universal agent).

```
tacmd listSystems –s REMOTE_TEMS5 –t UM
```

Return values

See Table 8 on page 272.

Related commands

"tacmd viewAgent" on page 242

"tacmd viewNode" on page 246

Return to Table 1 on page 5.

tacmd listtrace

Description

Use the **listtrace** command to query the RAS1 logging level on a remote managed system. You must log in by using the **tacmd login** command before running the **tacmd listtrace** command. Each **tacmd listtrace** command returns either the current value of the RAS1 trace variable or a status message providing diagnostic information, if an error occurred.

Note that all ITM processes in the RAS1 command flow must be at ITM v6.2.3 Fix Pack 2 (or higher), as only those releases support the RAS1 Dynamic Trace Facility. For example, if a **tacmd listtrace** request must travel from hub to remote monitoring server to agent, all three components must be at ITM v6.2.3 Fix Pack 2 (or higher) or the request will fail.

CLI syntax

tacmd listtrace
{-m|--system} *SYSTEM*
{-p|--property} *PROPERTY*

where:

-m|--system
Specifies to which managed system to send the listtrace command.

-p|--property
Identifies the RAS1 trace property to query. Valid values include KBB_RAS1, KDC_DEBUG, KDE_DEBUG, KDH_DEBUG, KLX_DEBUG, and KBS_DEBUG. Note that KLX_DEBUG is only active on z/OS and Windows platforms. KBS_DEBUG is only active if 1) it has been set as an environment variable on the managed system, or 2) it was dynamically activated with a previous **tacmd settrace** command.

CLI example

This example displays the current KBB_RAS1 value of an agent on SystemA:

```
tacmd listtrace -m SystemA:Agent1 -p KBB_RAS1
```

This example displays whether debug tracing of the KDH component is active on the HUB.

```
tacmd listtrace -m *HUB -p KDH_DEBUG
```

Note: A hub monitoring system can be specified using either its CMS_NODEID value or as *HUB.

Return values

See Table 8 on page 272.

Related commands

"tacmd settrace" on page 224

Return to Table 1 on page 5.

tacmd listUsers

Description

Use the **tacmd listUsers** command to list all the available users or users belonging to a particular group. To list users, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listUsers** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listUsers
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-g|--gid} *GROUPID***]**
[{-s|--server} *TEPS_HOSTNAME***]**

where:

-g|--gid
Specifies the Group ID of an existing group. When you specify this option, the listUsers command shows all users that belong to the group. The Group ID must not contain any blank spaces characters in it. Its maximum allowed length is 32 characters, and it must begin with "_" or " * ".

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname where the users exists. If not specified, the users belonging to the group ID are listed from the local Tivoli Enterprise Portal Server.

-u|--userid
Specifies the existing user ID to log in to Tivoli Enterprise Portal Server. The software prompts you for the User ID if you do not specify one.

-w|--password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

CLI example

This example lists all the users on the server.

```
tacmd listUsers –u sysadmin –w "tivoli123" –s HDCHASDSTC0219
```

This example lists all the users belongs to the group *ADMINISTRATOR.

```
tacmd listUsers –g *ADMINISTRATOR –u sysadmin –w "tivoli123"
–s HDCHASDSTC0219
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listUserGroups

Description

Use the **tacmd listUserGroups** command to list all the available user groups. To list groups, the log in user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listUserGroups** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listUserGroups
[{-u|--userid} *TEPS_USERID* **]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-i|--id} *USERID***]**
[{-s|--server} *TEPS_HOSTNAME***]**

where:

-i|--id
Specifies the user ID for which the assigned groups have to be listed. The User ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character.

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname where the groups exist. If not specified, the groups are listed from the local Tivoli Enterprise Portal Server.

-u|--userid
Specifies an existing User ID to log in to Tivoli Enterprise Portal. The software prompts you for the User ID if you do not specify one.

-w|--password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

CLI example

This example lists the entire user group available on the server.

```
tacmd listUserGroups –u sysadmin –w "tivoli123" –s HDCHASDSTC0219
```

This example lists the group that the user TESTUSER belongs to.

```
tacmd listUserGroups –i TESTUSER –u sysadmin –w "tivoli123"
–s HDCHASDSTC0219
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd listworkspaces

Description

This command displays a list of the Tivoli Enterprise Portal Server workspaces on the Tivoli Enterprise Portal Server. The workspace name, product code, and user ID are displayed for each workspace. You can optionally filter the list by workspace names, product codes, or workspace users. This command can only be run from a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal Desktop Client installation.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **listworkspaces** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd listworkspaces
[{-w|--workspace} ***WORKSPACE*** **...]**
[{-t|--type} ***TYPE*** **...]**
[{-r|--workspaceUser} ***USERID*** **...]**
[{-i|--objectid}]
[{-e|--exclude}]
[{-u|--username} ***TEPS_USER*****]**
[{-p|--password} ***TESP_PASSWORD*****]**
[{-s|--server} ***TEPS_HOST*****[:*****PORT*****]]**

where:

-w|--workspace
The name or names of the workspaces to list. Specify a string (any character except hyphen (-)) up to a maximum length of 72 characters. If not specified, all workspaces are displayed.

-t|--type
An IBM Tivoli Monitoring application type. If a 2-character type is entered, the letter 'k' will be prefixed automatically to form a 3-character type code. For example, _kib is the type for the Tivoli Enterprise Tivoli Monitor Workspaces. If not specified, all types are exported.

-r|--workspaceUser
A Tivoli Enterprise Portal user ID that one or more Tivoli Enterprise Portal workspaces are associated with. Specify a string of letters (upper or lower case) or numbers up to a maximum length of 32 characters. If not specified, workspaces are displayed for all users. To list only global workspaces, use this option without specifying any Tivoli Enterprise Portal User IDs.

-i|--objectid
Displays the object identifier of each workspace in the workspace results.

-e|--exclude
Excludes the specified workspace users, application types, and Tivoli Enterprise Portal User IDs from the list operation.

-u|--username
The name of the user to authenticate on the remote Tivoli Enterprise Portal

Server. Specify a valid string in the local locale. The software prompts you for the password if you do not specify one.

-p|--password
The password of the user to authenticate on the remote Tivoli Enterprise Portal Server. Specify a valid string in the local locale. The software prompts you for the password if you do not specify one.

-s|--server
Specifies a Tivoli Enterprise Portal Server to use. The *host* is a 32 or 64 bit IP address or hostname and *port* is an integer between 1 and 65536. If not specified, *host* defaults to localhost and *port* defaults to 15200.

CLI example

This example displays all workspaces on the Tivoli Enterprise Portal Server myteps.ibm.com without any filtering arguments (such as workspace name, user ID, or application type).

Note: A large number (over 500) of results are likely to be displayed.

```
tacmd listworkspaces –s myteps.ibm.com –u imasample –p mypassword
```

This example displays all workspaces belonging to the *klz* and *knt* application types on the Tivoli Enterprise Portal Server running on the local computer on port 15200 and filtered by application type.

```
tacmd listworkspaces –u imasample –p mypassword–t klz knt
```

This example is identical to the one above, except that the portal server credentials (username and password) were omitted at invocation time, and the user is interactively prompted to enter them.

```
tacmd listworkspaces –t klz knt
```

This example displays all workspaces belonging to the *SYSADMIN* user on the Tivoli Enterprise Portal Server myteps.ibm.com and filtered by username.

Note: In this example no global workspaces are displayed.

```
tacmd listworkspaces –s http://myteps.ibm.com –u imasample –p mypassword
                     –r SYSADMIN
```

This example displays only workspaces matching the names *Historical Summarized Availability Daily* or *Historical Summarized Availability Weekly* on the Tivoli Enterprise Portal Server myteps.ibm.com and filtered by workspace name.

```
tacmd listworkspaces –s myteps.ibm.com –u imasample –p mypassword
                     –w "Historical Summarized Availability Daily"
                        "Historical Summarized Availability Weekly"
```

This example displays only workspaces belonging to the *klz* and *knt* application types, workspace names matching the names *Historical Summarized Availability Daily* or *Historical Summarized Availability Weekly* on the Tivoli Enterprise Portal Server myteps.ibm.com on port 1996, and filtered by both workspace name and application type.

```
tacmd listworkspaces –s myteps.ibm.com:1996 –u imasample –p mypassword–t klz kux
                  –w "Historical Summarized Availability Daily"
                        "Historical Summarized Availability Weekly"
```

Return values

See “Return codes” on page 272.

Related commands

“tacmd importWorkspaces” on page 166

“tacmd exportWorkspaces” on page 123

Return to Table 1 on page 5.

tacmd login

Description

Use the **tacmd login** command to authenticate with a hub monitoring server so that you can execute subsequent commands from the local computer.

Note: Different warning messages might be returned by this command if the command is run more than once. This command enables you to specify the protocol (http or https), and to default to the correct protocol and server port. The command's connection PROTOCOL defaults to https and then to http if a connection cannot be established using https. Therefore the connection depends on which protocol the Tivoli Enterprise Monitoring Server is running on. If it connects to http, a warning message displays.

Note: You cannot run the **tacmd login** command against a Hot Standby (FTO) mirror hub monitoring server while that server is acting as the mirror.

CLI syntax

tacmd login {-s | --server} {[*PROTOCOL://]HOST[:PORT*]}
{-u | --username} *USERNAME*
{-p | --password} *PASSWORD*
[{-t | --timeout} *TIMEOUT*]

Standard input option:

tacmd login
{-stdin | --stdin}

where:

-s | --server
Specifies the hostname of the Tivoli Enterprise Monitoring Server to log on to.

-u | --username
Specifies the user to authenticate. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password of the user to authenticate. The software prompts you for the password if you do not specify one.

-t | --timeout
Specifies the maximum number of minutes that can elapse between

invocations of **tacmd** before the user is denied access. The default timeout is 15 minutes. The maximum timeout is 1440 minutes (24 hours).

-stdin|--stdin
Indicates that all command-line parameters are processed from standard input (in the same command-line format) instead of being parsed from the command-line arguments.

CLI example

This command logs on to the Tivoli Enterprise Monitoring Server on *pebble.ibm.com* with the user ID, *administrator*, the password, *mypassword*, and a log in expiration time of *1440* minutes.

```
tacmd login –s pebble.ibm.com –u administrator –p mypassword –t 1440
```

Return values

See Table 8 on page 272.

"tacmd logout"

Related commands

Return to Table 1 on page 5.

tacmd logout

Description

Use the **tacmd logout** command to log out of the monitoring server and disable the security token created by the **tacmd login** command.

CLI syntax

tacmd logout

CLI example

```
tacmd logout
```

Return values

See Table 8 on page 272.

Related commands

"tacmd login" on page 198

Return to Table 1 on page 5.

tacmd managesit

Description

Use the **tacmd managesit** command to start or stop situations in the Tivoli Enterprise Monitoring Server. You can specify one or more situations using the -s|--situation option. You can also specify a managed system type by using the

-t | --type option to start or stop all the situations of the specified type. The user can also optionally specify the Tivoli Enterprise Monitoring Server names on which the situations have to be started or stopped by using the -n | --temsname option. You must log in by using the **tacmd login** command before running the **tacmd managesit** command.

CLI syntax

tacmd managesit
{-s | --situation}*SITUATIONNAMES*
{-o | --option}*START | STOP*
[{-n | --temsname}*TEMSNAME*]

tacmd managesit
{-t | --type} *TYPE*
{-o | --option} *START | STOP*
[{-n | --temsname}*TEMSNAME*]

where:

-s | --situation
Name or names of the situations to be started or stopped. If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-t | --type
Specifies the two-digit character code for a system type. All the situations of this type will be started or stopped.

-o | --option
Specifies whether the situation is to be started or stopped on the Tivoli Enterprise Monitoring Server. The allowed values for this option are START and STOP.

-n | --temsname
Specifies the names of the Tivoli Enterprise Monitoring Servers on which the situations have to be started or stopped. If not specified, the situations are started or stopped on the HUB Monitoring Server by default.

CLI example

This example is used to start the situations specified by the -s option. The situations NT_System_File_Critical NT_Service_Error are started on the Tivoli Enterprise Monitoring Server.

```
tacmd managesit -s NT_System_File_Critical  NT_Service_Error -o start
```

This example is used to stop all the situations of type NT on the specified Tivoli Enterprise Monitoring Server.

```
tacmd managesit –t NT –o stop –n REMOTE_HDCHASDSTC0061
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd pdcollect

Description

Use the **tacmd pdcollect** command to remotely execute the pdcollect script in the specified host computer and transfer the resultant pdcollect file to the local computer. If no host is specified, the local host is run. This command can collect eWas log files and the *CANDLEHOME*/config/.ConfigData directory.

Note: If this command terminates abnormally, the temporary folder remains and needs to be cleaned manually.

CLI syntax

tacmd pdcollect
[{-s | --server} [{smb | ssh | rexec | rsh}://] *HOST* [:*PORT*]]
[{-u | --username} *USERNAME*]
[{-p | --password} *PASSWORD*]
[{-c | --candlehome} *CANDLEHOME*]
[{-d | --directory} *LOCAL_TAR_DIRECTORY*]
[{-o | --options} *PD_OPTION...*]
[{-a | --archivename} FILENAME]
[{-f | --force}]
[{-m | --migrate-export}]

where:

-s | --server
Identifies the server on which to execute the pdcollect script. Optionally, a specific connection protocol and a port can be specified. By default, all supported protocols are attempted until a connection is successfully established on one of them. The only valid protocols for this argument are SMB | SSH | REXEC | RSH.

Note: -u and -p options are required if -s option is specified with a value different from localhost.

-u | --username
A valid user log in ID on the specified host. The software prompts you for the username if you do not specify one. The user name is not required if the command is executed locally.

Note: -u and -p options are required if -s option is specified with a value different from localhost.

-p | --password
The password for the specified username. The software prompts you for the password if you do not specify one. The password is not required if the command is executed locally.

Note: -u and -p options are required if -s option is specified with a value different from localhost.

-d | --directory
The desired location to put the transferred file in the local computer. If not specified, it is placed in the local temp folder. The *CANDLEHOME* directory should not be used as the local directory path.

-c | --candlehome
The candlehome directory used to execute the pdcollect script. If not specified, the path "C:\IBM\ITM" is used as *candlehome* in Windows systems and "/opt/ibm/itm" is used in UNIX systems. Provide the directory path with double quotation marks.

-o | --options
The options to change default command behavior can be any of the following options separated by spaces, in any order or case:

noapp Specifies that application event log data is not to be collected.

nosec Specifies that security event log data is not to be collected.

nosys Specifies that system event log data is not to be collected.

noevent
Specifies that no event log data is to be collected. Equivalent to specifying noapp, nosec and nosys.

nohist Specifies that history files are not to be collected.

nologs
Specifies that IBM Tivoli Monitoring log and trace files are not to be collected. If nologs is specified, neither audit logs are collected.

noprompt
Specifies that all interactions with the user are suppressed.

noaudit
Specifies that IBM Tivoli Monitoring audit log files are not to be collected.

Note: The options noapp, nosec, nosys, and noevent apply only if the target machine is Windows.

-a | --archivename
The desired archive file name. Specify the file name without the extension. If the archive file is needed for a PMR, the suggested archive name is the PMR number associated with the logs being collected. Example: 62064,499,000. If not specified, the default archive name is pdcollect-HOSTNAME. This -a | --archivename option does not support specifying the file path. To specify the destination of the transferred file, the -d | --directory option must be used in conjunction with the -a | --archivename option.

-f | --force
Specifies that all interaction with you is suppressed. If not specified, you are prompted before archiving the collected files.

-m | --migrate-export
Executes the migrate-export script before collecting data.

CLI example

This example is used to invoke the pdcollect script on the remote computer demerzel, transfer the resultant tar file to the local computer, and place it under the "c:\demerzel" directory.

```
   tacmd pdcollect -s demerzel.tivlab.austin.ibm.com -u root -p bug2app3r
-c /opt/IBM/ITM -d "c:\demerzel" -o nosys noevent
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd putfile

Description

Use the **putfile** command to transfer a file from a local source to a remote managed system.

Note: Do not run more than 10 concurrent getfile, putfile, executeaction, or executecommand operations, in any combination. These 10 concurrent operations apply to both different agents and different physical machines. This command is recommended for transfers of 16 MB or less although not limited to this transfer size.

Transfer file sizes exceeding this limit can require additional response time and IBM Tivoli Monitoring environment consumption. If the getfile, putfile, executeaction and executecommand operations will be executed frequently, monitor the CPU utilization and network activity of the hub monitoring server and remote monitoring servers before and during these operations to ensure that resource consumption is acceptable. If resource consumption is too high, consider reducing the number of concurrent operations and the frequency of the operations.

Note: On the destination endpoint machine, ensure that system's defined temporary directory has sufficient space to temporarily contain the transferred file. The temporary directory is defined by the %TEMP% or %TMP% environment variable for Windows systems, and is the `/tmp` directory for UNIX and Linux systems.

The hub monitoring server, the targeted monitoring agents, and any remote monitoring servers to which the targeted agents are connected must be at IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later. If the Tivoli Enterprise Monitoring Agent component is at the IBM Tivoli Monitoring v6.2.2 Fix Pack 2 or later level, all the agents installed in the same `CANDLEHOME` directory at the endpoint are capable of handing this command. For this command, the specified system cannot be an i5/OS or z/OS monitoring agent.

Hub server configured with non-default port number

The executecommand, getfile, and putfile commands fail if the HUB TEMS is configured with a non-default port number. You must set the environment variable KDE_TRANSPORT in the Windows command prompt or UNIX Shell before issuing these commands to configure the TACMD to use the non-default port number to connect to the hub monitoring server. See the “KDE_TRANSPORT Structure” section of the “Configuring IBM Tivoli Monitoring components” chapter in the *IBM Tivoli Monitoring: Installation and Setup Guide* for descriptions and examples.

Relative and absolute path support at the endpoint

When running this command between a UNIX or Linux system and targeting a Windows monitoring agent, you must replace the backslashes with forward slashes in the path definitions for the -d|--destination option. It is best to use forward slashes for tolerance with Windows systems. For example, if you want to run the command from a UNIX system to place the monitor agent's configuration file in the C:\IBM\ITM\tmaitm6 directory on a Windows system, use the following command:

```
./tacmd putfile -m Primary:WINDOWS:NT -s ./kntenv -d C:/IBM/ITM/tmaitm6/kntenv
-t text
```

File names

When either the remote file's directory or name and the destination file's directory or name contain spaces, you must include double quotation marks around the respective directory and file name. For example, run the following command from a UNIX system to place the monitoring agent's configuration file in the C:\Program Files\ITM\tmaitm6 directory

```
./tacmd putfile -m Primary:WINDOWS:NT -s /opt/IBM/ITM/kntenv
-d "C:/Program Files/ITM/tmaitm6/kntenv" -t text
```

When working with file and directory names that have nonalphanumeric or special characters (for example, ! @ #, etc), the path and file references for either the -s|--source or -d|--destination option must be surrounded by double quotation marks (" "). However, paths that include an at symbol (@) must be escaped with an at symbol (@). The path user@home@directory is escaped as follows:

```
user@@home@@directory
```

Variable substitution

You can run this command by using an environment variable for both the -d|--destination and the -s|--source options. If used for the -d|--destination option, it is for the specified monitoring agent's managed system rather than the local environment where the command is issued. If used for the -s|--source option, it is for the local environment where the command is issued.

The environment variable format is @NAME@. The following characters are valid as the first character of any name, or any subsequent character:

- _ (underscore)
- Lower case alphabetic letters
- Upper case alphabetic letters

The following characters are valid as any character in any name except the first:

- - (dash)
- The following numbers, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

In the following example, *CANDLEHOME* on the local machine is /opt/IBM/ITM and *CANDLE_HOME* on the managed system is c:\IBM\ITM:

```
./tacmd putfile -m Primary:WINDOWS:NT -s @CANDLEHOME@/kntenv
-d @CANDLE_HOME@/tmaitm6/kntenv -t text
```

Note:

1. For monitoring agents running on AIX 6.1 systems as a root user, it is possible to issue a **tacmd putfile** command for files having permission 000.

2. To use this command, the KT1_TEMS_SECURE configuration parameter must be set in the hub monitoring server's configuration file to specify that the hub monitoring server supports this command. After setting the environment variable, you must recycle the hub monitoring server. The default value is no. Specify Y or YES or y or yes if you want to use this command.

CLI syntax

tacmd putfile

{-m|--system} *SYSTEM*
{-s|--source} *LOCAL_FILE*
{-d|--destination} *REMOTE_FILE*
[{-t|--type} *MODE***]**
[{-f|--force}]

where:

-m|--system
Specifies to which managed system to put the file. This must be a monitoring agent. Use the **listsystems** command to receive a list of which systems are available. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-s|--source
Specifies the local file name. Environment variables are supported. When specifying the source option, it must be an existing path. If the path is not specified, the default path is relative to where the command is issued.

-d|--destination
Specifies the remote file name. Environment variables are supported. When specifying the destination option, it must be an existing path. If the path is not specified, the default path is the *CANDLEHOME*/kt1v3depot/*product_code* directory on the endpoint

-t|--type
Specifies the MODE of transfer. MODE can be bin or text. If not specified, the default is bin. Specify text mode if the file is a human readable file, otherwise, specify bin (binary) mode.

-f|--force
Overwrites the remote file as specified by –d|--destination option if it already exists.

CLI example

See the example in the description of this command.

Return values

See Table 8 on page 272.

Related commands

"tacmd getfile" on page 128

Return to Table 1 on page 5.

tacmd refreshCatalog

Description

Use the **tacmd refreshCatalog** command to update the catalog file and refresh affinity information. This command allows the data server to reread the catalog files; therefore, it eliminates the need for a Tivoli Enterprise Monitoring Server recycle after support files are installed. Attribute files, however, are not reread, and Tivoli Enterprise Monitoring Server processing will not process this data from the agent correctly. In most cases, a Tivoli Enterprise Monitoring Server recycle is required after product seeding.

The **refreshCatalog** command updates the catalog file and refreshes the affinity information on the Tivoli Enterprise Monitoring Server to which it connected. Use **-s|--server** option to update the catalog files and refresh affinity information on a remote Tivoli Enterprise Monitoring Server. If the corresponding attribute file is manually deployed along with the catalog member the refresh catalog is intended for, then IBM Tivoli Monitoring based situations are expected to work. However, the following items still require a recycle of the Tivoli Enterprise Monitoring Server:

- OMEGAMON® Tivoli Event Adapter forwarding
- Data warehousing
- Simple Object Access Protocol (SOAP) requests

You must log in by using the **login** command before running the **refreshcatalog** command.

CLI syntax

tacmd refreshCatalog [{-s|--server} *TEMSNAME*]

where:

-s|--server
Specify a Tivoli Enterprise Monitoring Server name where the catalog file needs to be updated and affinity information needs to be refreshed. When this option is not specified, it operates against the Tivoli Enterprise Monitoring Server into which the tacmd is logged.

CLI example

This example updates the catalog file on the Tivoli Enterprise Monitoring Server into which the tacmd command is logged.

```
tacmd refreshCatalog
```

This example updates the catalog file on the remote Tivoli Enterprise Monitoring Server "REMOTE_LEVER2".

```
tacmd refreshCatalog -s REMOTE_LEVER2
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd refreshTECinfo

Description

Use the **tacmd refreshTECinfo** command to trigger the Event Forwarder to reprocess any updated event destinations, EIF configurations, and custom event mapping files without recycling the HUB Tivoli Enterprise Monitoring Server.

CLI syntax

tacmd refreshTECinfo {-t|--type} {eif|maps|attr|all}

where:

-t|--type
Specifies the type of info to refresh.

eif Refresh Tivoli Enterprise Console Event Integration Facility configuration files only (event destinations and EIF configuration file changes).

maps Refresh event mapping files only.

attr Refresh new and updated attribute files only.

all Refresh the Tivoli Enterprise Console Event Integration Facility configuration, event mapping files, and attribute files.

CLI example

This example triggers the reprocessing of the Tivoli Enterprise Console Event Integration Facility configuration without recycling the Tivoli Enterprise Monitoring Service:

```
tacmd refreshtecinfo -t eif
```

This example triggers the reprocessing of the event mapping files by the TEC Event Forwarder without recycling the Tivoli Enterprise Monitoring Service:

```
tacmd refreshtecinfo -t maps
```

This example triggers the reprocessing of new and updated attribute files only without recycling the Tivoli Enterprise Monitoring Service:

```
tacmd refreshtecinfo -t attr
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd removeBundles

Description

Use the **removeBundles** command to remove one or more deployment bundles from the local deployment depot.

This command must be run locally on a monitoring server containing a depot.

CLI syntax

tacmd removeBundles
{-i | --imagePath} *IMAGEPATH*
[{-t | --product | --products} *PRODUCT* **...]**
[{-p | --platform | --platforms} *PLATFORM* **...]**
[{-v | --version | --versions} *VERSION* **...]**
[{-f | --force }]

where:

-i | --imagePath
Specifies the directory on the agent install image media that contains the deployment bundles to be removed.

-t | --product | --products
Specifies the product code or codes of the products to remove. This value corresponds to the value that is displayed in the *Product Code* field that is displayed by the **viewDepot** or **listBundles** command.

-p | --platform | --platforms
The platform code or codes of the products to remove. This value corresponds to the value that is displayed in the *Host Type* field that is displayed by the **viewDepot** or **listBundles** command.

-v | --version | --versions
The version or versions of the products to remove. This value corresponds to the value that is displayed in the *Version* field that is displayed by the **viewDepot** command.

-f | --force
Removes any matching deployment bundles from the depot without prompting for confirmation first.

CLI example

This command removes all of the deployment bundles in the *D:\cdimage\bundles* directory from the local deployment depot.

```
tacmd removeBundles –i D:\cdimage\bundles
```

This command removes all of the deployment bundles in the */mnt/bundles* directory from the local deployment depot where the bundle product type is *ux*, the bundle platform is *aix513*, and the bundle version is *060100000*.

```
C:\>tacmd removeBundles -i
C:\IBM\ITM\cms\Depot\Packages\WINNT\KUX\060100000
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addBundles" on page 17

"tacmd listBundles" on page 171

"tacmd viewDepot" on page 243

Return to Table 1 on page 5.

tacmd removeSystem

Description

The **removeSystem** command removes one or more instances of an agent or uninstalls an agent from a managed system. By using the bulk deployment option, the agents specified in the bundle group are removed on the managed systems specified in the deployment group. This command is also available for non-agent bundles.

The no execute option is intended to allow the user to determine which managed systems are uninstalled on specific nodes.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Removing single IBM Tivoli Monitoring Agent, MSN option:

tacmd removeSystem
{-m|--system} ***SYSTEM*** **...**
[{-f|--force }]

Removing single IBM Tivoli Monitoring Agent, Node and Product Type option:

tacmd removeSystem
{-t|--type} ***TYPE***
{-n|--node} ***MANAGED-OS***
[{-p|--property|--properties} ***SECTION.NAME=VALUE*** **...]**
[{-f|--force}]

Removing System Service Monitors agent:

tacmd removeSystem
{-h|--host} [{smb|ssh|rexec|rsh}://]]
{-u|--username} ***USERNAME***
{-w|--password} ***PASSWORD***
[{-d|--dir|--directory} ***INSTALLDIR*****]**
[{-f|--force}]

Removing one or more System Service Monitors patches from a System Service Monitors agent:

tacmd removeSystem
{-h|--host} [{smb|ssh|rexec|rsh}://] ***HOST [:PORT]***

{-l | --patchlist} *PATCH_LIST*
[{-p | --property | --properties} *SECTION.NAME=VALUE* **...]**
[{-d | --dir | --directory} *INSTALLDIR***]**
[{-f | --force}]

Removing IBM Tivoli Monitoring agent in bulk:

tacmd removeSystem
{-g | --deploygroup}*DEPLOY_GROUP_NAME*
{-b | --bundlegroup} *BUNDLE_GROUP_NAME*
[{-f | --force} | {-x | --noexecute}]

Removing a System Service Monitors agent in bulk:

tacmd removeSystem
{-g | --deploygroup}*DEPLOY_GROUP_NAME*
{-b | --bundlegroup} *BUNDLE_GROUP_NAME*
[{-u | --username} *USERNAME***]**
[{-w | --password} *PASSWORD***]**
[{-s | --serverlist} *SERVER_LIST***]**
[{-f | --force} | {-x | --noexecute}]

Removing one or more System Service Monitors patches from a System Service Monitors agent in bulk:

tacmd removeSystem
{-g | --deploygroup}*DEPLOY_GROUP_NAME*
{-b | --bundlegroup} *BUNDLE_GROUP_NAME*
[{-s | --serverlist} *SERVER_LIST***]**
[{-f | --force} | {-x | --noexecute}]

where:

-m | --system
Specifies the management system name to uninstall.

-t | --type
Specifies the type of agent to uninstall.

-n | --node
Identifies the node on which the agent type will be uninstalled.

-h | --host
Specifies the host from which to remove the System Service Monitors agent. Optionally, a specific connection protocol and a port can be specified.

-u | --username
Specifies a valid user log in ID on the specified host.

-w | --password
Specifies the password for the specified username.

-p | --property | --properties
Specifies one or more SECTION.NAME=VALUE pairs that identify configuration properties, where SECTION specifies the configuration section containing the configuration property, NAME specifies the name of the configuration property, and VALUE specifies the property value.

Specify the instance name of the system to be configured through the INSTANCE property for systems that can have multiple instances.

-s | --serverlist
Specifies one or more server names separated by spaces.

-d | --dir | --directory
Specifies the location on the specified host where the agent is installed, if the agent is not installed in the default location. This location must be specified as a directory, in absolute path format.

-l | --patchlist
List of one or more patch names (separated by spaces) that will be removed from the System Service Monitors agent.

-g | --deploygroup
Specifies the name of the deployment group in which the agents in the bundle group will be uninstalled.

-b | --bundlegroup
Specifies the name of the bundle group containing the agents that will be uninstalled from the managed systems in the deployment group.

-f | --force
Removes the specified system without asking for confirmation.

-x | --noexecute
Causes the command to display which managed systems will be removed.

CLI example

The following command removes the specified LA1 patch on the specified host "lever2":

```
tacmd removesystem -h smb://lever2 -l LA1 -f
```

The following command removes all UA (Universal Agent) type agents from the managed node Primary:HDCHASDSTC0540:NT:

```
tacmd removesystem -t UM -n Primary:HDCHASDSTC0540:NT
```

The following command removes a System Service Monitors agent from the specified host:

```
tacmd removesystem -h HDCHASDSTC0540 -u root -w ****** -d /ibm/itm/
```

The following command removes IBM Tivoli Monitoring agents in bulk:

```
tacmd removesystem -g UnixGroup -b ULBundle
```

The following command removes System Service Monitors agents in bulk on all specified servers:

```
tacmd removesystem -g UnixGroup -b ULBundle -u root -w ******
-s HDCHASDSTC0540 HDCHASDSTC0452 HDCHASDSTC0061 -f
```

The following command removes an instance of the UNIX Log Agent (KUL):

```
tacmd removeSystem -m nc118215_FP5:KUL
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd restartAgent

Description

Use the **tacmd restartAgent** command to restart the specified agents or the agents for the specified managed systems, locally or remotely, if they are not running. The *OS agent* must be running on the local computer before issuing this command for a non-OS agent. You must have permission to restart a service on an operating system to restart an agent. You can restart all agents of one or more specified types on a specified node remotely by running the **restartagent** command with the -t|--type and -n|--node options.

If you have the authority to restart agents and you specify only the agent type, you do not need to log in to restart an agent on a local computer. When you run the **restartagent** command on a local system, use the -t|--type option and do not use the -n|--node or -m|--system options.

When you specify only an agent type, all agents of that type are restarted on the local computer.

To restart an *OS agent*, you *must* issue the command on the local computer where the agent is installed.

By using the bulk deployment option, the agents specified in the bundle group are restarted on the managed systems specified in the deployment group.

The no execute option is intended to allow the user to determine which managed systems will be restarted.

Note: You cannot use this command to restart a non-default Universal Agent instance that you created manually. Use the **itmcmd agent** command with the **-p** option instead to restart a non-default Universal Agent instance.

Note: If you attempt to restart the portal server using this command, you will receive a failure message. To restart the portal server, you must use either the Manage Tivoli Enterprise Services or the associated command line, for example, **itmcmd server**.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Restarting IBM Tivoli Monitoring agent, remote execution path:

tacmd restartAgent
{-m|--system} ***SYSTEM*...**
[{-f|--force}]

Restarting IBM Tivoli Monitoring agent, remote execution path:

tacmd restartAgent
{-n|--node} *MANAGED-OS*
{-t|--type} *TYPE* **...**
[{-f|--force}]

Restarting IBM Tivoli Monitoring agent, local execution path:

tacmd restartAgent
{-t|--type} *TYPE* **...**
[{-f|--force}]

Restarting System Service Monitors agent, remote execution path:

tacmd restartAgent
{-h|--host} [{smb|ssh|rexec|rsh}://] *HOST [:PORT]*
{-u|--username} *USERNAME*
{-w|--password} *PASSWORD*
[{-p|--property|--properties} *SECTION.NAME=VALUE* **...]**
[{-d|--dir|--directory} *INSTALLDIR* **]**
[{-f|--force}]

Restarting IBM Tivoli Monitoring agent, bulk remote execution path:

tacmd restartAgent
{-g|--deploygroup *DEPLOY_GROUP_NAME***}**
{-b|--bundlegroup *BUNDLE_GROUP_NAME***}**
[[{-f|--force} | {-x|--noexecute}]

Restarting System Service Monitors agent, bulk remote execution path:

tacmd restartAgent
{-g|--deploygroup *DEPLOY_GROUP_NAME***}**
{-b|--bundlegroup *BUNDLE_GROUP_NAME***}**
[{-u|--username} *USERNAME* **]**
[{-w|--password} *PASSWORD* **]**
[{-s|--serverlist} *SERVER_LIST* **]**
[[{-f|--force} | {-x|--noexecute}]

where:

-m|--system
Specifies a managed system on which to restart the agents.

-f|--force
Restarts the specified agents without confirmation.

-t|--type
Specifies one or more agents or agent instances to restart. The value of *type* is scoped to the *node* level.

-n|--node
Specifies the node on which to restart the agent. The node is the installation directory for all agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed.

-h | --host
Specifies the host on which to restart the System Service Monitors agent. Optionally, a specific connection protocol and a port can be specified.

-u | --username
Specifies a valid user log in ID on the specified host.

-w | --password
Specifies the password for the specified username.

-p | --property | --properties
Specifies one or more SECTION.NAME=VALUE pairs that identify configuration properties, where SECTION specifies the configuration section containing the configuration property, NAME specifies the name of the configuration property, and VALUE specifies the property value. Specify the instance name of the system to be configured through the INSTANCE property for systems that can have multiple instances.

-s | --serverlist
Specifies one or more server names separated by spaces.

-d | --dir | --directory
Specifies the location on the specified host where the agent is installed, if the agent is not installed in the default location. This location must be specified as a directory, in absolute path format.

-g | --deploygroup
Specifies the name of the deployment group to which the agents in the bundle group will be restarted.

-b | --bundlegroup
Specifies the name of the bundle group containing the agent(s) which will be restarted on the managed system(s) in the deployment group.

-x | --noexecute
Causes the command to display which managed systems will be restarted.

CLI example

This command restarts the *Universal Agent* agent with name *stone:UA*.

```
tacmd restartAgent -m stone:UA
```

This command restarts all *UM* agents on the node *Primary:STONE:NT*.

```
tacmd restartAgent –n Primary:STONE:NT –t UM
```

This command restarts all *NT* agents on the local system.

```
tacmd restartAgent –t NT
```

The following command restarts a System Service Monitors agent:

```
tacmd restartagent -h stone.tivlab.raleigh.ibm.com -u administrator
-w ****** -d D:\ibm\itm
```

The following example is for bulk execution. The command restarts IBM Tivoli Monitoring agents:

```
tacmd  restartagent -g WindowsGroup -b NTBundle
```

Return values

See Table 8 on page 272.

Related commands

"tacmd stopAgent" on page 229

"tacmd startAgent" on page 226

Return to Table 1 on page 5.

tacmd restartFailedDeploy

Description

Use the **tacmd restartFailedDeploy** command to restart failed deployments. Use this command to restart all the failed entries in the status table or filter the table entries to restart from all status table entries to a specific deployment operation.

You must log in by using the **login** command before running the **restartFailedDeploy** command.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Restarting all the failed entries in the status table:

tacmd restartFailedDeploy {-a|--all}

Restarting specific failed deployment operations in the status table:

tacmd restartFailedDeploy
[{-g|--transactionID} *TRANSID*]
[{-c|--command} *COMMAND*]
[{-h|--hostname} *HOSTNAME*]
[{-p|--platform} *PLATFORM*]
[{-v|--version} *VERSION*]
[{-t|--product}]
[{-a|--all}]

where:

-g|--transactionID
Specifies the global transaction ID. This filter maps to the value shown in the Transaction ID row from the **tacmd getdeploystatus** command.

-c|--command
Specifies the type of the deployment operation. The following operations are acceptable:

- START
- RESTART
- STOP
- INSTALL
- REMOVE
- CONFIGURE
- UPDATE

- CHECKPREREQ
- SETAGENTCONN

-h | --hostname
Specifies the target host filter that is used to select the entries to restart from the status table. This filter maps to the value shown in the Target Hostname row from the **tacmd getdeploystatus** command.

-p | --platform
Specifies the platform filter that is used to select the entries to restart from the status table. This filter maps to the value shown in the Platform row from the **tacmd getdeploystatus** command.

-v | --version
Specifies the version filter that is used to select the entries to restart from the status table. This filter maps to the value shown in the Version row from the **tacmd getdeploystatus** command.

-t | --product
Specifies the product type filter that is used to select the entries to restart from the status table. This filter maps to the value shown in the Product row from the **tacmd getdeploystatus** command.

-a | --all
Restarts all the failed entries in the Remote Deploy status table.

CLI example

This command restarts all the failed entries in the Remote Deploy status table:

```
tacmd restartFailedDeploy -a
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd resumeSda

Description

Use the **tacmd resumeSda** command to resume Self-Describing Agent (SDA) installation without recycling the hub monitoring server. You must log in by using the **login** command before running the **resumeSda** command.

For a hub monitoring server v6.3 or later, the **tacmd resumeSda** command and **tacmd suspendSda** command are available at the hub, but not at the downlevel remote monitoring server (for example, v6.2.3 Fix Pack 1). For a hub monitoring server v6.2.3 Fix Pack 1 and remote monitoring server v6.3 or later, the **tacmd resumeSda** command and **tacmd suspendSda** command are not available at either monitoring server.

Note: You cannot run the **tacmd resumeSda** command against any remote monitoring server or Hot Standby (FTO) mirror hub monitoring server while that server is acting as the mirror.

CLI syntax

tacmd resumeSda
[{-f|--force }]

where:

-f|--force
Resumes Self-Describing Agent (SDA) installation without prompting for confirmation first.

CLI example

This command resumes Self-Describing Agent (SDA) installation without prompting for confirmation first:

```
tacmd resumeSda -f
```

Return values

See Table 8 on page 272.

Related commands

"tacmd deleteSdaSuspend" on page 75

"tacmd suspendSda" on page 236

Return to Table 1 on page 5.

tacmd setAgentConnection

Description

Use the **tacmd setAgentConnection** command to edit connection properties and environment variables of agents running on the target node. Target agents with updated connection settings or environment variables are restarted following the update. Specify the configuration data through the parameter pair `SECTION.NAME=VALUE`. Using the **Bulk Deploy** option, the agents specified in the bundle group are configured on the managed systems specified in the deployment group.Use the no execute option to determine which configuration properties are used to configure which managed systems. When using this command to set or modify an environment variable, ensure that the value you assign to the variable is correct. An incorrect value assignment might impact the agent behavior and possibly prevent the agent from starting.

When specifying configuration properties, specify the individual property keywords and avoid using a URL syntax.

```
[{IP.PIPE|IP.SPIPE}://[HOSTNAME][:PORT]
```

While specifying a URL syntax is allowed, you might not set the additional parameters which are required for the configuration specification. When any of the protocol properties are changed for an agent, those same protocol properties should be changed for all the other agents on that target machine. If the connection information for all the agents is not the same, then you might encounter problems with managing or collecting metrics from the agents. Additionally, the parameters that are passed to the configuration utility are not validated for compatibility or

effectiveness when applied. Therefore it is important to thoroughly understand the changes that are intended and the parameter values that you specify for the command.

CLI syntax

Configuring a single monitoring agent:

tacmd setAgentConnection
{-t|--type} ***TYPE***
{-n|--node} ***MANAGED-OS***
[{-p|--property} ***NAME=VALUE*** **...** **]**
[{-e|--environment} ***NAME=VALUE*** **...** **]**
[{-a|--allagents} **]**
[{-f|--force} **]**

Note: You must specify at least one **-p** or **-e** parameter.

Bulk monitoring agent connection properties update:

tacmd setAgentConnection
{-g|--deploygroup DEPLOY_GROUP_NAME}
{-b|--bundlegroup BUNDLE_GROUP_NAME}
[{-e|--environment} ***NAME=VALUE*** **...** **]**
[{-a|--allagents} **]**
[{-x|--noexecute}]
[{-f|--force} **]**

where:

-t|--type
Specifies the type of agent to add to the monitoring system.

-n|--node
Identifies the node, or monitoring system, to which you want to add the agent. A node is identified by the managed operating system that it contains.

-p|--property

Specifies one or more agent connection properties to be updated for the agent. The connection properties are specified with `NAME=VALUE` pairs, where NAME specifies the name of the connection property and VALUE specifies the property value. If more than one connection property is specified, each `NAME=VALUE` pair should be separated by a space. Valid properties for a monitoring agent include ENCRYPT, KEY, IP_PIPE, IP_SPIPE, PROTOCOL, PROTOCOL1, PROTOCOL2, PROTOCOL3, PORT, SERVER, SNA_NETNAME, SNA_LOGMODE, SNA_LUNAME, SNA_TPNAME, BACKUP, BSERVER, BPROTOCOL, BPROTOCOL1, BPROTOCOL2, BPROTOCOL3, BSNA_NETNAME, BSNA_LOGMODE, BSNA_LUNAME, BSNA_TPNAME, FOLDER, BPORT, and BIND_TO_NIC.

The following table describes the parameter keywords most frequently used to change an agent's connection parameters and identifies the additional parameters to specify when changing parameters.

Parameter keyword	Expected value	Also specify these additional parameter keywords...
1 KEY	The encryption key to encrypt your local passwords and IDs. The value must be exactly 32 characters. If it is short it will be padded with "9"'s. If it is long it will be truncated.	
2 PROTOCOL1	Highest precedent protocol for TEMS communication. (IP.PIPE,IP.SPIPE)	7, one of 5 or 6
3 PROTOCOL2	Second highest precedent protocol for TEMS communication. (IP.PIPE,IP.SPIPE)	7, one of 5 or 6
4 PROTOCOL3	Third highest procedent protocol for TEMS communication. (IP.PIPE,IP.SPIPE)	7, one of 5 or 6
5 IP_PIPE_PORT	Port number used by IP.PIPE protocol to connect to Primary TEMS	7, one of 2, 3 or 4
6 IP_SPIPE_PORT	Port number used by IP.SPIPE protocol to connect to Primary TEMS	7, one of 2, 3 or 4
7 SERVER	Hostname or IPv4 address of the Primary TEMS to connect to.	2, one of 5 or 6
8 BACKUP	Does a secondary TEMS connection exist or are you adding one? (Y or N)	9, 10, one of 13 or 14
9 BSERVER	Hostname or IPv4 address of the Secondary TEMS to connect to	8, 10, one of 13 or 14
10 BPROTOCOL1	Highest precedent protocol for Secondary TEMS communication. (IP.PIPE,IP.SPIPE)	8, 9, one of 13 or 14
11 BPROTOCOL2	Second highest precedent protocol for Secondary TEMS communication. (IP.PIPE,IP.SPIPE)	8, 9, one of 13 or 14
12 BPROTOCOL3	Third highest procedent protocol for Secondary TEMS communication. (IP.PIPE,IP.SPIPE)	8, 9, one of 13 or 14
13 BIP_PIPE_PORT	Port number used by IP.PIPE protocol to connect to Secondary TEMS	8, 9, one of 10, 11 or 12
14 BIP_SPIPE_PORT	Port number used by IP.SPIPE protocol to connect to Secondary TEMS	8, 9, one of 10, 11 or 12

Parameter keyword	Expected value	Also specify these additional parameter keywords...
15 BIND_TO_NIC	IPv4 address for dual network host adapter cards	

-e | --environment
Specifies one or more `NAME=VALUE` pairs that identify environment variables to update, where NAME specifies the name of the environment variable, and VALUE specifies the value to be assigned. If more than one environment variable is specified, each `NAME=VALUE` pair should be separated by a space. Valid environment variables for a monitoring agent are CMS_MSGBASE, CTIRA_HEARTBEAT, CTIRA_HOSTNAME, CTIRA_MAX_RECONNECT_TRIES, CTIRA_NCSLISTEN, CTIRA_NODETYPE, CTIRA_OS_INFO, CTIRA_PRODUCT_SEP, CTIRA_RECONNECT_WAIT, CTIRA_REFLEX_ATOMIC, CTIRA_REFLEX_TARGET, CTIRA_SIT_CLEAN, CTIRA_SIT_FILE, CTIRA_SIT_MGR, CTIRA_SUBSYSTEM_ID, CTIRA_SYSTEM_NAME, IRA_DUMP_DATA, ITM_BINARCH, KHD_HISTRETENTION, TEMA_SDA, and KBB_SHOW_NFS.

Note: If you update the environment variable CTIRA_HOSTNAME, you should use the **tacmd cleanMS** command to delete the entries for offline managed systems from the Tivoli Enterprise Monitoring Server before running other remote deploy commands to the target system. Use the **tacmd listSystems** command to display the list of managed systems.

-a | --allagents
Specifies the connection settings properties to update for all the agents on the node.

-g | --deploygroup
Specifies the name of the deployment group to which the agents in the bundle group will be deployed.

-b | --bundlegroup
Specifies the name of the bundle group containing the agents that will be deployed to the managed systems in the deployment group.

-x | --noexecute
Specifies which bundles to deploy to which managed systems.

-f | --force
Specifies the command to run without prompting for confirmation.

CLI example (see Note)

In the following example, the agent will connect to newTEMS as its server when it is restarted after the configuration. In addition, the CTIRA_HOSTNAME and CTIRA_HEARBEAT environment variables will be updated. When you run the **tacmd listSystems** command after the **tacmd setAgentCommand** command completes, the results indicate the agent connected to newTEMS with CTIRA_HOSTNAME of aix526 and the original entry with an inactive (N) status. You can delete the inactive entry using the **tacmd cleanms** command. If there are other agents installed on the node, use the -a | --allagents option to connect them to the new server (newTEMS).

```
tacmd setagentconnection -t ux -n amsaix75:KUX -p SERVER=newTEMS
-e CTIRA_HOSTNAME=aix526 CTIRA_HEARTBEAT=9
```

In the following example, you assign a backup server (BSERVER=YYYYYYY) with IP.PIPE protocol for agent on node XXXXXX.

```
tacmd setagentconnection -t nt -n <Primary:XXXXXX:NT>
-p BACKUP=Y BSERVER=YYYYYYYY BPROTOCOL=IP.PIPE
```

Changing the protocol setting for the Primary TEMS from IP.PIPE to IP.SPIPE

```
tacmd setagentconnection -n Primary:endpoint:NT -a -p SERVER=PrimaryTEMS
PROTOCOL1=IP.SPIPE IP_SPIPE_PORT=3600
```

Changing or adding a Backup TEMS for all the Agents

```
tacmd setagentconnection -n Primary:endpoint:NT -a -p BSERVER=SecondaryTEMS
BPROTOCOL1=IP.PIPE BIP_PIPE_PORT=1918 BACKUP=Y
```

Swapping the Primary TEMS with the Backup TEMS

```
tacmd setagentconnection -n endpoint:UX -a -p SERVER=BackupRTEMS
PROTOCOL1=IP.PIPE IP_PIPE_PORT=1918 BSERVER=PrimaryRTEMS
BPROTOCOL1=IP.PIPE BIP_PIPE_PORT=1918 BACKUP=Y
```

Removing a Backup TEMS

```
tacmd setagentconnection -n endpoint:UX -a -p SERVER=PrimaryRTEMS PROTOCOL1=IP.PIPE
IP_PIPE_PORT=1918 BACKUP=N
```

Note: Specifying an IP.PIPE and IP.SPIPE value for Unix and Linux is not currently supported. To change the designated SERVER for these platforms, specify only the SERVER keyword in your parameter list. The currently set values for IP.PIPE and IP.SPIPE will remain in effect if you specify only the SERVER keyword.

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd setOverride

Description

Use the **tacmd setOverride** command to define a situation override for a specified situation on a managed system or list of managed systems.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **setOverride** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: The total number of characters used in all the expression overrides defined for a situation should not exceed 4000 bytes. The actual size requirement for a single override varies depending on the names and values of the key columns and the override expression. In one case the limit might be 25 or, in a simpler case, it might be higher. The symptom of exceeding the 4000-byte limit is that the overrides do not work and the monitoring server trace log shows an "exceeds limit 4000" override error.

CLI syntax

tacmd setOverride

{-s | --situation} *SITNAME*
{-m | --system} *SYSTEM | SYSTEM_LIST*
{-p | --predicate} *PREDICATE ...*
[{-u | --userid} *TEPS_USERID***]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-h | --tepshostname} *TEPS_HOSTNAME***]**
[{-c | --calendarentry} *CALENDAR_ENTRY***]**
[{-t | --inlinecal} *INLINE_CAL_ENTRY***]**
[{-k | --key} *KEY_CONDITION ...***]**
[{-f | --force}]

tacmd setOverride

{-x | --xmlfile} *XMLFILE*
[{-u | --userid} *TEPS_USERID***]**
[{-w | --password} *TEPS_PASSWORD***]**
[{-h | --tepshostname} *TEPS_HOSTNAME***]**
[{-f | --force}]

where:

-s | --situation
Specifies the situation to set the override for. If you include either the & character or the < character in the situation name, you must include quotation marks around the name, for example, "abc&def" or "abc<def".

-m | --system
Specifies the name of the managed system or managed system group to set the override for. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-c | --calendarentry
Specifies the name of the calendar entry that defines the time period when the override is active. The situation override is always active if you do not enter a calendar name.

-t | --inlinecal
Specifies the name of the Hourly Schedule entry that defines the time period when the override is active. The situation override is always active if you do not enter a Hourly Schedule name. For the INLINE_CAL_ENTRY variable, use the [HHmm,HHmm] format, where HH is for hours in 00-23 notation and mm stands for minutes.

-p | --predicate
Specifies the situation formula predicate or predicates to override. Predicates must be enclosed in double quotation marks and entered in the format "ATTRIBUTE OPERATOR VALUE" with spaces between ATTRIBUTE, OPERATOR, and VALUE. The predicate OPERATOR must be one of the following: "EQ", "NE", "GT", "LT", "GE", or "LE".

The attribute can be entered by using either the attribute name or the display name for the attribute. Run the **tacmd listSitAttributes -s** ***SITNAME*** command to view the eligible attribute names for the situation.

-k | --key
Specifies the key condition or key conditions that must be met in order for

the override to apply. Each key condition must be enclosed in double quotation marks and entered in the format "ATTRIBUTE VALUE" with spaces between ATTRIBUTE and VALUE. The key condition OPERATOR is restricted to the value "EQ".

The attribute can be entered by using either the attribute name or the display name for the attribute. Run the **tacmd listSitAttributes -s** ***SITNAME*** command to view the eligible key condition attribute names for the situation.

-x | --xmlfile
Specifies the name and location of the xml file where the situation override definition is located. This file can be produced by the **tacmd suggestBaseline** command.

-u | --userid
Specifies the existing User ID to log on to the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

-f | --force
Sets the override without prompting for confirmation.

CLI example

This example sets overrides by using an XML file produced by the **suggestBaseline** command, using the force prompt to suppress the confirmation prompt:

```
tacmd setoverride --userid sysadmin --password ********
--xmlfile cpubaseline.xml --force
```

This example sets an override for a managed system group that only overrides the situation threshold value:

```
tacmd setoverride -u sysadmin -w ******** -s NT_NotesServerProcess
-m *NT_SYSTEM -p "% Processor Time GE 27"
```

This example sets an override for a managed system group, where the override has an associated calendar entry and key condition:

```
tacmd setoverride -u sysadmin -w ******** -s NT_NotesServerProcess
-m *NT_SYSTEM -p "% Processor Time GE 40" -c Weekday
-k "Binary Path EQ C:\Notes\NotesServer\nserver.exe"
```

This example sets an override for a managed system, where the override has an associated calendar entry:

```
tacmd setoverride -u sysadmin -w ******** -s NT_NotesServerProcess
-m Primary:LEVER:NT -p "% Processor Time GE 40" -c Weekend
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd settrace

Description

Use the **settrace** command to modiy the RAS1 logging level to the requested value on a remote managed system. Each **tacmd settrace** command returns a status message indicating whether the command completed successfully. If an error occurred, a status message provides diagnostic information. You could also use the **tacmd listtrace** command to verify that the requested logging level is now in effect. Before modifying the RAS1 logging level with the **tacmd settrace** command, display the current value with the **tacmd listtrace** command to determine if there are any active diagnostic traces that should be preserved in the **tacmd settrace** command. You must log in by using the **tacmd login** command before running the **tacmd settrace** command.

Note that all ITM processes in the RAS1 command flow must be at ITM v6.2.3 Fix Pack 2 (or higher), as only those releases support the RAS1 Dynamic Trace Facility. For example, if a **tacmd settrace** request must travel from hub to remote monitoring server to agent, all three processes must be at ITM v6.2.3 Fix Pack 2 (or higher) or the request will fail. In addition, if you want to disable the Dynamic Trace Facility, set the environment variable KBB_DISABLE_DYNAMIC_TRACE=N. With this setting, the **tacmd settrace** command will be ignored.

CLI syntax

Use this format to change the current tracing level:

tacmd settrace
{-m|--system} *SYSTEM*
{-p|--property} *PROPERTY*
{-o|--option} *OPTION*
{-d|--description} *DESCRIPTION*

Use this format to restore the original value by undoing any tracing changes that have been made:

tacmd settrace
{-m|--system} *SYSTEM*
{-p|--property} *PROPERTY*
{-r|--restore} *RESTORE*

where:

-m|--system
Specifies to which managed system to send the command.

-p|--property
Identifies the RAS1 trace property to set. Valid values include KBB_RAS1, KDC_DEBUG, KDE_DEBUG, KDH_DEBUG, KLX_DEBUG, and KBS_DEBUG. Note that KLX_DEBUG is only active on z/OS and Windows platforms.

-o|--option
Identifies the trace options to set for the -p property. K*xx*_DEBUG trace properties support one-character values: **Y** for Yes, **N** for Normal, *I* for Inhibit (i.e., NONE), **V** for Verbose, **S** for State, **T** for Trace, **D** for Detail, **M** for Maximum, and **A** for All. KBB_RAS1 supports more complex options.

For help with KBB_RAS1, refer to "Setting traces" in the *IBM Tivoli Monitoring Troubleshooting Guide*, or use the settings provided by your IBM Software Support representative. If a KBB_RAS1 string contains any embedded blanks, the entire string must be enclosed in double quotes.

-d | --description
Specifies an optional description text string to help identify the reason for changing the trace variable. If a description string contains any embedded blanks, the entire string must be enclosed in double quotes.

-r | --restore
Specifies that the RAS1 trace property should be restored back to the original value it had at product startup. If the -r flag is set, the -o and -d flags are ignored.

CLI example

This example modifies the RAS1 logging level for managed system AIX_RTEMS. Note that the new value of the RAS1 variable, containing embedded blanks, is enclosed in double quotes.

```
tacmd settrace -m AIX_RTEMS -p KBB_RAS1-o "ERROR (UNIT:kfasd ALL)"
```

This example activates debug tracing of the KDC component on the HUB. An optional description text string, "PMR 12345", is specified so that when the tracing change takes effect, the HUB's RAS1 log message will include the "PMR 12345" string to help explain why the debug tracing was activated.

```
tacmd settrace -m *HUB -p KDC_DEBUG -o Y -d "PMR 12345"
```

In this example, the KBB_RAS1 setting of an agent on SystemA is restored to its original value. This command would typically be issued after diagnostic trace data had been collected from the agent and it was no longer necessary to continue running with dynamic tracing active.

```
tacmd settrace -m SystemA:Agent1 -p KBB_RAS1 -r
```

Note: A hub monitoring system can be specified using either its CMS_NODEID value or as *HUB.

Note: In the case of an ITM process that spawns other child processes, the RAS1 tracing level of the child processes will not be altered by the **tacmd settrace** command. For example, if you run the **tacmd settrace** command on a UNIX OS agent, the RAS1 tracing level of the agent's child processes, such as kux_vmstat, ifstat, or stat_deamon, will remain the same. Only the kuxagent parent process will be affected by the **tacmd settrace** command.

Return values

See Table 8 on page 272.

Related commands

“tacmd listtrace” on page 192

Return to Table 1 on page 5.

tacmd startAgent

Description

Use the **tacmd startAgent** command to start the given agent or agents for the given managed systems, locally or remotely, if they are not running. You can start all agents of one or more specified types on a specified node remotely by running the **startagent** command with the -t|--type and -n|--node options.

If you have the authority to start agents and you specify only the agent type, you do not need to log in to start an agent on a local computer. When you run the **startagent** command on a local system, use the -t|--type option and do not use the -n|--node or -m|--system options.

When you specify only an agent type, all agents of that type are started on the local computer.

To start an *OS agent*, you *must* issue the command on the local computer where the agent is installed.

By using the bulk deployment option, the agents specified in the bundle group are started on the managed systems specified in the deployment group.

The no execute option is intended to allow the user to determine which managed systems will be started.

Note:

1. If you have made changes to the agent configuration file on a UNIX computer, use the **itmcmd agent** command with the **-c** option to start the agent instead of this command. By using the **-c** option with **itmcmd agent** preserves any changes that you have made to the configuration. The **tacmd startAgent** command does not preserve the changes.
2. You cannot use this command to start a non-default Universal Agent instance that you created manually. Use the **itmcmd agent** command with the **-p** option instead to start a non-default Universal Agent instance.
3. If you attempt to start the portal server using this command, you will receive a failure message. To start the portal server, you must use either the Manage Tivoli Enterprise Services or the associated command line, for example, **itmcmd server**.
4. Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Starting IBM Tivoli Monitoring agent, remote execution path:

tacmd startAgent
{-m|--system} ***SYSTEM*** **...**
[{-f|--force}]

Starting IBM Tivoli Monitoring agent, remote execution path:

tacmd startAgent
{-n|--node} ***MANAGED-OS***
{-t|--type} ***TYPE*** **...**
[{-f|--force}]

Starting IBM Tivoli Monitoring agent, local execution path:

tacmd startAgent
{-t|--type} ***TYPE*** **...**
[{-f|--force}]

Starting System Service Monitors agent, remote execution path:

tacmd startAgent
{-h|--host} [{smb|ssh|rexec|rsh}://] ***HOST [:PORT]***
{-u|--username} ***USERNAME***
{-w|--password} ***PASSWORD***
[{-p|--property|--properties} ***SECTION.NAME=VALUE*** **...]**
[{-d|--dir|--directory} ***INSTALLDIR*****]**
[{-f|--force}]

Starting IBM Tivoli Monitoring agent, bulk remote execution path:

tacmd startAgent
{-g|--deploygroup ***DEPLOY_GROUP_NAME*****}**
{-b|--bundlegroup ***BUNDLE_GROUP_NAME*****}**
[[{-f|--force} | {-x|--noexecute}]

Starting System Service Monitors agent, bulk remote execution path:

tacmd startAgent
{-g|--deploygroup ***DEPLOY_GROUP_NAME*****}**
{-b|--bundlegroup ***BUNDLE_GROUP_NAME*****}**
[{-u|--username} ***USERNAME*** **]**
[{-w|--password} ***PASSWORD*** **]**
[{-s|--serverlist} ***SERVER_LIST*** **]**
[[{-f|--force} | {-x|--noexecute}]

where:

-m|--system
Specifies a managed system on which to start the agents.

-f|--force
Starts the specified agents without confirmation.

-t|--type
Specifies one or more agents or agent instances to start. The value of *type* is scoped to the *node* level.

-n|--node
Specifies the node on the computer where you want to start an agent. The node is the installation directory for all agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed.

-h | --host
Specifies the host on which to start the System Service Monitors agent. Optionally, a specific connection protocol and a port can be specified.

-u | --username
A valid user log in ID on the specified host.

-w | --password
The password for the specified username.

-p | --property | --properties
Specifies one or more SECTION.NAME=VALUE pairs that identify configuration properties, where SECTION specifies the configuration section containing the configuration property, NAME specifies the name of the configuration property, and VALUE specifies the property value. Specify the instance name of the system to be configured through the INSTANCE property for systems that can have multiple instances.

-s | --serverlist
Specifies one or more server names separated by spaces.

-d | --dir | --directory
Specifies the location on the specified host where the agent is installed, if the agent is not installed in the default location. This location must be specified as a directory, in absolute path format.

-g | --deploygroup
Specifies the name of the deployment group to which the agents in the bundle group will be started.

-b | --bundlegroup
Specifies the name of the bundle group containing the agent(s) which will be started on the managed system(s) in the deployment group.

-x | --noexecute
Causes the command to display which managed systems will be started.

CLI example

This command starts the *Universal Agent* agent with the name *stone:UA*.

```
tacmd startAgent –m stone:UA
```

This command starts all *UM* agents on the node *Primary:STONE:NT*.

```
tacmd startAgent –n Primary:STONE:NT –t UM
```

The following command starts all *NT* agents on the local system:

```
tacmd startAgent –t NT
```

The following command starts a System Service Monitors agent:

```
tacmd startagent -h stone.tivlab.raleigh.ibm.com -u adminstrator
-w ****** -d D:\ibm\itm
```

The following is an example for bulk execution. The command starts IBM Tivoli Monitoring agents:

```
tacmd  startagent -g WindowsGroup -b NTBundle
```

Return values

See Table 8 on page 272.

Related commands

"tacmd stopAgent"

"tacmd restartAgent" on page 212

Return to Table 1 on page 5.

tacmd stopAgent

Description

Use the **tacmd stopAgent** command to stop the given agent or agents for the given managed systems. The *OS agent* must be running on the local computer before issuing this command for a non-OS agent.

If you have the authority to start agents and you specify only the agent type, you do not need to log in to stop an agent on a local computer. When you run the **stopagent** command on a local system, use the -t|--type option and do not use the -n|--node or -m|--system options.

When you specify only an agent type, all agents of that type are stopped on the local computer.

To stop an *OS agent*, you *can* issue the command on the local computer where the agent is installed.

By using the bulk deployment option, the agents specified in the bundle group are stopped on the managed systems specified in the deployment group.

The no execute option is intended to allow the user to determine which managed systems are stopped.

Note: You cannot use this command to stop a non-default Universal Agent instance that you created manually. Use the **itmcmd agent** command with the **-p** option instead to stop a non-default Universal Agent instance.

Note: If you attempt to stop the portal server using this command, you will receive a failure message. To stop the portal server, you must use either the Manage Tivoli Enterprise Services or the associated command line, for example, **itmcmd server**.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Stopping IBM Tivoli Monitoring agent, remote execution path:

tacmd stopAgent
{-m|--system} ***SYSTEM*** **...**
[{-f|--force}]

Stopping IBM Tivoli Monitoring agent, remote execution path:

tacmd stopAgent

{-n|--node} *MANAGED-OS*
{-t|--type} *TYPE* **...**
[{-f|--force}]

Stopping IBM Tivoli Monitoring agent, local execution path:

tacmd stopAgent

{-t|--type} *TYPE* **...**
[{-f|--force}]

Stopping System Service Monitors agent, remote execution path:

tacmd stopAgent

{-h|--host} [{smb|ssh|rexec|rsh}://] *HOST [:PORT]*
{-u|--username} *USERNAME*
{-w|--password} *PASSWORD*
[{-p|--property|--properties} *SECTION.NAME=VALUE* **...]**
[{-f|--force}]

Stopping IBM Tivoli Monitoring agent, bulk remote execution path:

tacmd stopAgent

{-g|--deploygroup *DEPLOY_GROUP_NAME***}**
{-b|--bundlegroup *BUNDLE_GROUP_NAME***}**
[{-f|--force} | {-x|--noexecute}]

Stopping System Service Monitors agent, bulk remote execution path:

tacmd stopAgent

{-g|--deploygroup *DEPLOY_GROUP_NAME***}**
{-b|--bundlegroup *BUNDLE_GROUP_NAME***}**
[{-u|--username} *USERNAME* **]**
[{-w|--password} *PASSWORD***]**
[{-s|--serverlist} *SERVER_LIST* **]**
[{-d|--dir|--directory} INSTALLDIR]
[{-f|--force} | {-x|--noexecute}]

where:

-m|--system
Specifies a managed system on which to stop the agents.

-f|--force
Stops the specified agents without confirmation.

-t|--type
Specifies one or more agents or agent instances to stop. The value of *type* is scoped to the *node* level.

-n|--node
Specifies the node on the computer where you want the agent to be stopped. A node is identified by the managed operating system that it contains; it is the installation directory for all agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed.

-h | --host
Specifies the host on which to stop the System Service Monitors agent. Optionally, a specific connection protocol and a port can be specified.

-u | --username
A valid user log in ID on the specified host.

-w | --password
The password for the specified username.

-p | --property | --properties
Specifies one or more SECTION.NAME=VALUE pairs that identify configuration properties, where SECTION specifies the configuration section containing the configuration property, NAME specifies the name of the configuration property, and VALUE specifies the property value. Specify the instance name of the system to be configured through the INSTANCE property for systems that can have multiple instances.

-s | --serverlist
Specifies one or more server names separated by spaces.

-d | --dir | --directory
Specifies the location on the specified host where the agent is installed, if the agent is not installed in the default location. This location must be specified as a directory, in absolute path format.

-g | --deploygroup
Specifies the name of the deployment group to which the agents in the bundle group will be stopped.

-b | --bundlegroup
Specifies the name of the bundle group containing the agent(s) which will be stopped on the managed system(s) in the deployment group.

-x | --noexecute
Causes the command to display which managed systems will be stopped.

CLI example

The following command stops the *Universal Agent* agent with the name *stone:UA*:

```
tacmd stopAgent –m stone:UA
```

The following command stops all *UM* agents on the node *Primary:STONE:NT*:

```
tacmd stopAgent –n Primary:STONE:NT –t UM
```

The following command stops all *NT* agents on the local system:

```
tacmd stopAgent –t NT
```

The following command stops a System Service Monitors agent:

```
tacmd stopagent -h stone.tivlab.raleigh.ibm.com -u adminstrator
-w ****** -d D:\ibm\itm
```

The following is an example for bulk execution. The command stops IBM Tivoli Monitoring agents:

```
tacmd  stopagent -g WindowsGroup -b NTBundle
```

Return values

See Table 8 on page 272.

Related commands

"tacmd restartAgent" on page 212

"tacmd startAgent" on page 226

Return to Table 1 on page 5.

tacmd suggestBaseline

Description

Use the **tacmd suggestBaseline** command to calculate a baseline (situation override) value by using one of several statistical functions for a situation attribute based on historical data from the Tivoli Data Warehouse.

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **tacmd suggestBaseline** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

Note: The managed system specified with the -m|--system option must be online to run the command.

Note: For a managed system group, the overrides are only applied to members of the list that are override-eligible. Overrides are not distributed to ineligible managed systems.

Note: The total number of characters used in all the expression overrides defined for a situation should not exceed 4000 bytes. The actual size requirement for a single override varies depending on the names and values of the key columns and the override expression. In one case the limit might be 25 or, in a simpler case, it might be higher. The symptom of exceeding the 4000-byte limit is that the overrides do not work and the monitoring server trace log shows an "exceeds limit 4000" override error.

CLI syntax

tacmd suggestBaseline
{-s|--situation} *SITNAME*
{-m|--system} *SYSTEM|SYSTEM_LIST*
{-p|--predicate} *PREDICATE*
{-f|--function} *STATISTICAL_FUNCTION*
{-d|--startdata} *START_TIMESTAMP*
{-e|--enddata} *END_TIMESTAMP*
[{-k|--key} *KEY_CONDITION* **...]**
[{-x|--xmlfile} *XMLFILE***]**
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-h|--tepshostname} *TEPS_HOSTNAME***]**
[{-c|--calendarentry} *CALENDAR_ENTRY***]**
[{-t|--inlinecal} *INLINE_CAL_ENTRY...***]**

where:

-s | --situation

Specifies the situation to calculate the baseline value and set the overrides for. If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-m | --system

The name of the managed system or managed system group to calculate the baseline value and set the overrides for. Historical data results from the warehouse used for statistical calculations are restricted to values recorded for the managed system or managed systems specified. Valid values include letters (upper or lower case), numbers, periods (.), at symbols (@), dollar signs ($), asterisks (*), number signs (#), underscores (_), colons (:) or blanks ( ).

-c | --calendarentry

Specifies the name of the calendar entry that defines the time period when the override is active. If one or more calendar entries are entered, historical data results from the warehouse are filtered such that only the results that fall within each calendar entry are used to calculate the baseline value. A separate baseline value is calculated for each calendar entry.

-t | --inlinecal

Specifies the name of the Hourly Schedule entry that defines the time period when the override is active. The situation override is always active if you do not enter a Hourly Schedule name. For the INLINE_CAL_ENTRY variable, use the [HHmm,HHmm] format, where HH is for hours in 00-23 notation and mm stands for minutes.

If one or more Hourly Schedule intervals are entered, historical data results from the warehouse are filtered such that only the results that fall within each Hourly Schedule interval are used to calculate the baseline value. A separate baseline value is calculated for each Hourly Schedule interval.

-d | --startdata

Specifies the starting time from which historical data from the warehouse will be used. Historical results queried from the warehouse are bounded by the start and end times. The start time value is specified as a timestamp in the format CYYMMDDHHmmSS or CYYMMDDHHmmSSsss, where:

- C=the century identifier (use 1 for year 2000 and later, 0 for earlier)
- YY=the year (for example, '08' for 2008)
- MM=the month (for example, '01' for January, or '12' for December)
- DD=the day of the month (for example, '06' for the 6th, or '31' for the 31st)
- HH=the hour of the day (for example, '08' for 8 A.M. or '17' for 5 P.M.)
- mm=the minute of the hour (for example, '00' for 'on the hour', '30', and so on.)
- SS=the second (for example, '01' for one second past the minute)
- sss=milliseconds (for example, '500' for half a second). This value is optional.

-e | --enddata

Specifies the ending time from which historical data from the warehouse will be used. Historical results queried from the warehouse are bounded by the start and end times. The end time value is specified as a timestamp in the format CYYMMDDHHmmSS or CYYMMDDHHmmSSsss, where:

- C=the century identifier (use 1 for year 2000 and later, 0 for earlier)

- YY=the year (for example, '08' for 2008)
- MM=the month (for example, '01' for January, or '12' for December)
- DD=the day of the month (for example, '06' for the 6th, or '31' for the 31st)
- HH=the hour of the day (for example, '08' for 8 A.M. or '17' for 5 P.M.)
- mm=the minute of the hour (for example, '00' for 'on the hour', '30', and so on.)
- SS=the second (for example, '01' for one second past the minute)
- sss=milliseconds (for example, '500' for half a second). This value is optional.

-f | --function

Specifies the statistical function which is used to calculate baseline values for the historical data queried from the warehouse. The statistical function is specified in the format:

```
{ mode | percent NUM | avg[{+|-}NUM] | min[{+|-}NUM] | max[{+|-}NUM] }
```

where:

```
min[{+|-}NUM] : minimum value +/- NUM percent of the value
max[{+|-}NUM] : maximum value +/- NUM percent of the value
avg[{+|-}NUM] : average value +/- NUM standard deviations
percent NUM   : value for the NUM percentile
mode          : most frequently observed value
```

When the mode calculation yields multiple results and an output xml file has been specified by using the -x | --xmlfile option, the first result will be used by the **acceptBaseline** command for the purposes of setting the override value in the xml file.

-p | --predicate

Specifies the situation formula predicate for which the baseline value is calculated. The predicate must be enclosed in double quotation marks and entered in the format "ATTRIBUTE OPERATOR VALUE" with spaces between ATTRIBUTE, OPERATOR, and VALUE. The predicate OPERATOR must be one of the following: "EQ", "NE", "GT", "LT", "GE", or "LE". Historical data results from the warehouse used for statistical calculations is restricted to values recorded for the attribute specified by this predicate.

The attribute can be entered by using either the formula name or the display name for the attribute. Run the **tacmd listSitAttributes -s *SITNAME*** command to view the eligible attribute names for the situation.

-k | --key

Specifies the key condition or key conditions restricting the predicate attribute for which the baseline value is calculated. Each key condition must be enclosed in double quotation marks and entered in the format "ATTRIBUTE OPERATOR VALUE" with spaces between ATTRIBUTE, OPERATOR, and VALUE. The key condition OPERATOR is restricted to the value "EQ". Historical data results from the warehouse used for statistical calculations is restricted to values recorded for the predicate attribute where all of the key conditions (where ATTRIBUTE equals VALUE) are satisfied.

The key condition attribute name can be entered by using either the formula name or the display name for the attribute. Run the **tacmd listSitAttributes -s *SITNAME*** command to view the eligible key condition attribute names for the situation.

-x | --xmlfile
Specifies the name and location of the xml file where the situation override definitions for each suggested baseline value is persisted. This xml file can be used as input by the **tacmd setOverride** command.

Overrides set by using this xml file only apply for the situation and managed systems specified. If calendar entries are specified, the overrides only apply during the specified calendar entries. If calendar entries are not specified, the override applies for all time periods. If key conditions are entered, the overrides only apply when the (optional) key conditions are met.

-u | --userid
Specifies the existing User ID to log on to the Tivoli Enterprise Portal Server.

-w | --password
Specifies the password for user authentication.

-h | --tepshostname
Specifies the Tivoli Enterprise Portal Server hostname.

CLI example

This example calculates suggested baseline values by using the average value plus 1 standard deviation for managed system Primary:LEVER:NT for the NT_NotesServerProcess situation, where the "Binary Path" attribute value is equal to "`C:\Notes\NotesServer\nserver.exe`" . Baseline values for the calendar entries WeekdayMorning and WeekdayAfternoon are calculated by using metrics stored in the Tivoli Data Warehouse between 5:59 a.m. July 28th, 2008, and 1 a.m. August 29th, 2008. Suggested baseline values are written to the xml file `cpubaseline.xml` in the local execution directory:

```
tacmd suggestbaseline --userid sysadmin --password ********
--system Primary:LEVER:NT --situation NT_NotesServerProcess
--predicate "% Processor Time GE 50" --function AVG+1
--startdata 1080728055900 --enddata 1080829010000
--key "Binary Path EQ C:\Notes\NotesServer\nserver.exe"
--calendarentry WeekdayMorning WeekdayAfternoon
```

This example calculates suggested baseline values by using 10% below the minimum value for managed system Primary:LEVER:NT for the NT_Disk_Space_Low situation, where the "Logical Disk Name" attribute value is equal to "C:" . The baseline value is calculated by using metrics stored in the Tivoli Data Warehouse between 5:59 a.m. July 28th, 2008, and 1 a.m. August 29th, 2008:

```
tacmd suggestbaseline --userid sysadmin --password ********
--system Primary:LEVER:NT --situation NT_Disk_Space_Low
--predicate "% Free LE 15" --function MIN-10 --startdata 1080728055900
--enddata 1080829010000 --key "Logical Disk Name EQ C:"
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd suspendSda

Description

Use the **tacmd suspendSda** command to suspend Self-Describing Agent (SDA) installation without recycling the hub monitoring server. While SDA installation is suspended, no SDA installations will occur. All previously defined SDA product version installation configurations set by the **tacmd addsdainstalloptions** command are ignored. You must log in by using the **tacmd login** command before running the **tacmd suspendSda** command.

For a hub monitoring server v6.3 or later, the **tacmd resumeSda** command and **tacmd suspendSda** command are available at the hub, but not at the downlevel remote monitoring server (for example, v6.2.3 Fix Pack 1). For a hub monitoring server v6.2.3 Fix Pack 1 and remote monitoring server v6.3 or later, the **tacmd resumeSda** command and **tacmd suspendSda** command are not available at either monitoring server.

Note: You cannot run the **tacmd suspendSda** command against any remote monitoring server or Hot Standby (FTO) mirror hub monitoring server while that server is acting as the mirror.

CLI syntax

tacmd suspendSda
[{-f | --force }]

where:

-f | --force
Suspends Self-Describing Agent (SDA) installation without prompting for confirmation first.

CLI example

This command suspends Self-Describing Agent (SDA) installation without prompting for confirmation first:

```
tacmd suspendSda -f
```

Return values

See Table 8 on page 272.

Related commands

"tacmd deleteSdaSuspend" on page 75

"tacmd resumeSda" on page 216

Return to Table 1 on page 5.

tacmd tepsLogin

Description

Use the **tacmd tepsLogin** command to log on to the Tivoli Enterprise Portal Server. For ITM v6.2.3 or later, the IBM HTTP Server (IHS) serves as the default HTTP server at port 15200. However, the portal server is still compatible with the KDH HTTP server at port 1920.

CLI syntax

tacmd tepsLogin
{-s | --server} TEPS_HOSTNAME
{-u | --username} USERNAME
{-p | --password} TEPS_PASSWORD
[{-t | --timeout} TIMEOUT]

Standard input option:

tacmd tepsLogin
{-stdin | --stdin}

where:

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname to connect to. Use either -s <TEPS_HOSTNAME> or -s http://<TEPS_HOSTNAME>

-u | --username
Specifies the username to log on to the Tivoli Enterprise Portal Server. The software prompts you for the username if you do not specify one.

-p | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-t | --timeout
Specifies the maximum number of minutes that can elapse between invocations of the **tacmd tepsLogin** command before the user is denied access to the Teps credentials file. The default timeout is 15 minutes. The maximum timeout is 1440 minutes (24 hours).

-stdin | --stdin
Indicates that all command-line parameters are processed from standard input (in the same command-line format) instead of being parsed from the command-line arguments.

CLI example

The following example shows how to log in on the command line with a specific username. The user is prompted to enter a password. The log in is valid for 24 hours (1,440 minutes):

```
C:\IBM\ITM\bin>tacmd tepslogin -s lever -u sysadmin -t 1440

 Password?
 KUICTL001I Validating user credentials...

KUIC00007I: User sysadmin logged on to server on http://lever:15200.
```

The following example shows how to log in with only the server and time specified. The user is prompted to enter a username and a password. The log in is valid for 1 hour (60 minutes).

```
C:\IBM\ITM\bin>tacmd tepslogin -s lever  -t 60

 Username?  sysadmin

 Password?
 KUICTL001I Validating user credentials...

KUIC00007I: User sysadmin logged on to server on http://lever:15200.
```

The following example shows how to log on to the KDH HTTP server at port 1920.

```
tacmd tepslogin -s ip-address:1920 -u sysadmin -p mypwd
```

Related commands

Attention: A variety of commands that communicate with the Tivoli Enterprise Portal Server are authenticated using one of two methods:

- Use the **tacmd tepsLogin** command with -u, -p, and -s options to autenticate to the Tivoli Enterprise Portal Server. Then use the related command (for example, the **tacmd createUser** command or the **tacmd listworkspaces** command), or
- Use the related command you want to use (for example, the **tacmd createUser** command or the **tacmd listworkspaces** command). However, you should specify the -u, -p, and -s options. If you don't specify the -s option, but you specify the -w and -u options for a command that communicates with the Tivoli Enterprise Portal Server, then the command uses the default value for the hostname (-s localhost:15200).

Return to Table 1 on page 5.

tacmd tepsLogout

Description

Use the **tacmd tepsLogout** command to disable the security token created by the **tacmd tepslogin** command.

CLI syntax

tacmd tepsLogout

CLI example

The following example shows how to log out:

```
C:\IBM\ITM\bin>tacmd tepslogout

KUIC01002I: sysadmin logged off of the http://lever:15200 server.
```

Related commands

Return to Table 1 on page 5.

tacmd updateAgent

Description

Use the **tacmd updateAgent** command to install an agent or non-agent bundle update on a specified managed system. Updating agents involves stopping any that are running, applying the changes, and restarting them. Updating non-agent bundles transfers the bundle to the endpoint and executes the user defined install command. If a version is not specified, the agent is updated to the latest version. This command is also available for non-agent bundles.

By using the bulk deployment option, the agents specified in the bundle group are updated on the managed systems specified in the deployment group.

The no execute option is intended to allow the user to determine which bundles will be updated to which managed systems.

Note: If the agent to be updated is configured to have a different name than the OS Agent hostname, then the tacmd updateagent process is unable to determine the target platform attributes and thus the process fails. If an agent has a different system name than the OS Agent, the tacmd updateagent process cannot be used.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

Updating a single IBM Tivoli Monitoring agent:

tacmd updateAgent
{-t|--type} *TYPE*
{-n|--node} *MANAGED-OS*
[{-v|--version} *VERSION***]**
[{-o|--option} *OPTIONSNAME=VALUE* **...]**
[{-k|--securegroup} *ITMGROUP***]**
[{-f|--force}]

Updating a single non-agent bundle:

tacmd updateAgent
{-t|--type} *TYPE*
{-n|--node} *MANAGED-OS*
[{-v|--version} *VERSION***]**
[{-p|--property|--properties} *SECTION.NAME=VALUE* **...]**
[{-f|--force}]

Updating a single System Service Monitors agent:

tacmd updateAgent
{-h|--host} *HOST[:PORT]*
{-l|--patchlist} *PATCH_LIST*
[{-p|--property|--properties} *SECTION.NAME=VALUE* **...]**
[{-f|--force}]

Updating bulk IBM Tivoli Monitoring or System Service Monitors agent:

tacmd updateAgent

{-g|--deploygroup} *DEPLOY_GROUP_NAME*
{-b--bundlegroup}*BUNDLE_GROUP_NAME*
[{-v|--version} *VERSION***]**
[{-o|--option} *OPTIONSNAME=VALUE* **...]**
[{-k|--securegroup} *ITMGROUP***]**
[{-f|--force} | **{-x|--noexecute}]**

where:

-t|--type
Specifies the type of agent to update.

-n|--node
Identifies the node on the computer that has the agent that you want to update. A node is identified by the managed operating system that it contains. The node is the installation directory for all agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed.

-v|--version
Specifies the version of the agent to switch to. You must specify the *version* in the format: *vvrrmmfff* where *vv* = version number; *rr* = release number; *mm* = modification number (fix pack number); and *fff* = interim fix number. You cannot use this command to revert an agent to a previous version of that agent.

-f|--force
Performs actions without asking confirmation.

-h|--host
Specifies the host on which the System Service Monitors agent to update resides.

-l|--patchlist
Specifies the list of one or more patch names separated by spaces that will be installed on the System Service Monitors agent.

-p|--property|--properties
Specifies one or more *section.name=value* pairs that identify configuration properties to update, where *section* specifies the configuration section containing the configuration property, *name* specifies the name of the configuration property, and *value* specifies the property value. Specify the instance name of the system to be configured via the *instance* property for a systems that can have multiple instances.

-o|--option
One or more configuration parameters that can be used to customize the operation of this program. The valid options are: COLLECTALL, EXECPREREQCHECK, IGNOREPREREQCHECK. The values are to be specified in KEY=VALUE format.

-k|--securegroup
Specifies the ITMGROUP. This option is only valid for UNIX and Linux nodes.

-g|--deploygroup
Specifies the name of the deployment group to which the agents in the bundle group will be updated.

-b | --bundlegroup
Specifies the name of the bundle group containing the agent(s) which will be updated on the managed system(s) in the deployment group.

-x | --noexecute
Causes the command to display which bundles will be updated to which managed systems.

CLI example

The following command updates an agent at version 6, release 3.0, with no interim fixes:

```
tacmd updateAgent -t UX -n itmserver:KUX -v 06300000
```

The following command updates System Service Monitors agents with Fix Pack 1:

```
tacmd updateAgent -h smb://emerald.raleigh.ibm.com  -l "Fixpack 1"
-p SNMPPORT=16002 -f
```

Return values

See Table 8 on page 272.

Related commands

"tacmd addBundles" on page 17

"tacmd addSystem" on page 24

Return to Table 1 on page 5.

tacmd viewAction

Description

Use the **tacmd viewAction** command to display the details of a take action. You must log in by using the **tacmd login** command before running the **tacmd viewAction** command.

CLI syntax

tacmd viewAction {-n | --name} *ACTIONNAME*
[{-t | --type} *TYPE*]
[{-d | --detailtextname} *TYPEDESC*]

where:

-n | --name
The name of the action to be viewed.

-t | --type
Application type name. Specify a two-digit character code of the system type name to view the action.

-d | --detailtextname
Application detail text name. Specify detail text of the system type name to view the action.

Note: The -d|--detailtextname option should be used along with the -t|--type option to filter the action having the same name and type.

CLI example

This example displays the details of the action "New Action" of type NT.

```
tacmd viewaction –n "New Action" –t NT
```

Return values

See Table 8 on page 272.

Return to Table 1 on page 5.

tacmd viewAgent

Description

Use the **tacmd viewAgent** command to display the details of the given agent or the agent for a given managed system. Details include whether the agent is running or not, the agent version, and all of the configuration data for the agent. Configuration data is not displayed for universal agents or OS agents.

Note: Remote Deployment is not supported in environments with a z/OS Tivoli Enterprise Monitoring Server. This restriction includes this command.

CLI syntax

tacmd viewAgent
{-m|--system} ***SYSTEM*** **...**

tacmd viewAgent
{-n|--node} ***MANAGED-OS***
{-t|--type} ***TYPE*** **...**

where:

-m|--system
Specifies the managed system for which you want to view agent status.

-t|--type
Specifies one or more agents to view. The value of *type* is scoped to the *node* level.

-n|--node
Specifies the node on which the agents you want to view are installed. The node is the installation directory for all agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed.

CLI example

The following command displays the details for the *UM* agent with the name *stone:UA*:

```
tacmd viewAgent –m stone:UA
```

The following command displays the details for all *UM* agents on the node *Primary:STONE:NT*:

```
tacmd viewAgent –n Primary:STONE:NT –t UM
```

Return values

See Table 8 on page 272.

Related commands

"tacmd viewNode" on page 246

Return to Table 1 on page 5.

tacmd viewCalendarEntry

Description

Use the **tacmd viewCalendarEntry** command to display the information of a calendar entry. You must log in by using the **login** command before running the **tacmd viewCalendarEntry** command.

CLI syntax

tacmd viewCalendarEntry
{-n|--name} *CALENDAR_ENTRY_NAME*

where:

-n|--name
Specifies the name of the calendar entry.

CLI example

The following example displays information for the Clean_Temp calendar entry:

```
tacmd viewCalendarEntry -n "Clean_Temp"
Name        : Clean_Temp
Type        : CRON
Description : Clean Temporary directory on weekend
Data         : 30 21 * * SUN
Info         :
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd viewDepot

Description

Use the **tacmd viewDepot** command to display the types of agents you can install from the deployment depot on the server which you are logged on to or the specified remote server. This command is also available for non-agent bundles.

CLI syntax

tacmd viewDepot
[{-j|--depot} *DEPOT*]
[{-t|--type} *TYPE*]
[{ -v|--version} *VERSION*]
[{-p|--platform} *PLATFORM*]

where:

-j|--depot
Specifies the name of the remote server that hosts the depot when you are logged on to the hub monitoring server.

-t|--type
Specifies the product type for filtering.

-v|--version
Specifies the version for filtering.

-p|--platform
Specifies the platform for filtering.

CLI example

The following command displays the contents of the deployment depot on the Tivoli Enterprise Monitoring Server you logged on to by using the **tacmd login** command:

```
tacmd viewDepot
```

The following command displays the contents of the deployment depot on the remote monitoring server, *REMOTE_ROCK*, which connects to the hub monitoring server. You must log on to the hub monitoring server before running this command:

```
tacmd viewDepot -j REMOTE_ROCK
```

Return values

See Table 8 on page 272.

Related commands

"tacmd listBundles" on page 171

"tacmd addBundles" on page 17

"tacmd removeBundles" on page 208

Return to Table 1 on page 5.

tacmd viewEventDest

Description

Use the **tacmd viewEventDest** command to display all properties for the specified event destination definition on the server.

CLI syntax

tacmd viewEventDest {-i|--id|–serverID} ID...

where:

-i|---id|–serverID
Identifies the Server Destination ID of the event destination record to display. The value must be a value between 0 and 999, inclusive.

CLI example

This example displays all properties for the specified event destination definition on the server:

```
tacmd viewEventDest -i 123
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd viewgroup

Description

Use the **tacmd viewgroup** command to displays details of the specified group. You must log in by using the **login** command before running the **viewgroup** command.

CLI syntax

tacmd viewgroup
{-g|--group} *GROUPNAME*
{-t|--groupType} *GROUPTYPE*
[-v|--verbose]

where:

-g|--group
Specifies the name of the group to be viewed.

-t|--groupType
Specifies the type of group to be viewed. Acceptable type names are DEPLOY, BUNDLE, SITUATION, or COLLECTION.

-v|--verbose
Specifies the verbose option used to display members detail information that is dependant upon the group type. For example, if -t BUNDLE, then the command displays information of all the bundle members.

CLI example

The following example displays the details of the deployment group "NewWindowsDeployGroup":

```
tacmd viewGroup -g NewWindowsDeployGroup -t DEPLOY
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd viewgroupmember

Description

Use the **tacmd viewgroupmember** command to displays the details of the specified group member. You must log in by using the **login** command before running the **viewgroupmember** command.

CLI syntax

tacmd viewgroupmember
{-g|--group} *GROUPNAME*
{-m|--member} *MEMBERNAME*
{-t|--grouptype} *DEPLOY|BUNDLE|SITUATION|COLLECTION*

where:

-g|--group
Specifies the name of the group that owns the group member to be displayed.

-m|--member
Specifies the name of the member to be displayed.

-t|--grouptype
Specifies the group member type name. Acceptable type names are DEPLOY, BUNDLE, SITUATION, or COLLECTION.

CLI example

This example displays the deployment member w099o002.tivlab.raleigh.ibm.com details that belong to the group NewWindowsDeployGroup:

```
tacmd viewGroupMember -g NewWindowsDeployGroup -t DEPLOY
-m w099o002.tivlab.raleigh.ibm.com
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd viewNode

Description

Use the **tacmd viewNode** command to display the versions and patch levels of the managed systems that are installed on a node or a group of nodes.

CLI syntax

tacmd viewNode {{-n|--node} *MANAGED-OS* **|**
{-l|--managedSystemList} *MANAGED_SYSTEM_LIST* **}**

where:

-n|--node
Specifies the node to display. A node is identified by the managed operating system that it contains. The node is the installation directory for all agents. The name of a node includes the computer where the OS agent is installed and the product code for the OS agent. For example, stone.ibm.com:LZ is the name of the node on computer stone.ibm.com, which has a Linux OS agent installed. The **-n** option is mutually exclusive with **-l**, but one or the other must be specified.

-l|--managedSystemList
Specifies the managed system group to display. Specify a managed system containing nodes to display all of the nodes in the managed system group. A node is identified by the managed operating system that it contains. The **-l** option is mutually exclusive with **-n**, but one or the other must be specified.

CLI example

The following command displays the components installed on the managed system named *icarus.austin.ibm.com*.

```
tacmd viewNode -n icarus.austin.ibm.com:Lz
```

Return values

See Table 8 on page 272.

Related commands

"tacmd viewAgent" on page 242

Return to Table 1 on page 5.

tacmd viewSit

Description

Use the **viewSit** command to display the configuration of a situation in your monitored environment or to save the configuration in an export file.

When you issue this command to view the properties of situations, many situations display K*XX*:*XXXX* for the description field. The Tivoli Enterprise Portal maps the K*XX*:*XXXX* value to some descriptions before display, but the CLI displays the data, as such, from the TSITDESC table. The tacmd CLI interface can only display the contents of the situation as defined or displayed within the Situation Editor. The desired substitution takes place as a result of the situation firing; whereas that CLI displays the situation as described above.

Note: You cannot use this command to view UADVISOR situations.

CLI syntax

tacmd viewSit {-s|--situation} *SITNAME*
[{-e|--export} [*FILENAME*]]

where:

-s|--situation
Specifies the name of the situation to display or export. If you include either the & character or the < character in the situation name, you must quote the name, for example, "abc&def" or "abc<def".

-e|--export
Exports the situation definition to a file of the name specified. If you specify a full path, the destination directory must exist.

CLI example

This example displays the definition for the situation named *CalcMonitor* and exports the details to a file named *CalcMonitor.sit*, which can then be used to create a new situation based on the original.

```
tacmd viewSit --situation CalcMonitor --export CalcMonitorOut.sit
```

Return values

See “Return codes” on page 272

Related commands

“tacmd createSit” on page 54

“tacmd deleteSit” on page 76

“tacmd editSit” on page 98

“tacmd listSit” on page 184

“tacmd viewDepot” on page 243

Return to Table 1 on page 5.

tacmd viewsystemlist

Description

This command displays the configuration of a managed system group or saves it in an export file.

CLI syntax

tacmd viewsystemlist {-l|--list } *LIST*
[{-e|--export} [*FILENAME*]]
[{-a|--availableSytem}]

where:

-l|--list
Name of the managed system group to be viewed or exported. Specify a string of letters (upper or lower case), numbers, underscores (_), or asterisks (*) up to a maximum length of 32 characters.

-e|--export
Export the managed system group definition to the specified export stream (file). The name of a file can be created or overwritten. If *filename* is not specified, the manage system list will be redirected to the standard output stream.

-a|--availableSytem
List all the available managed systems on the server.

CLI example

This example displays the details of one of the catalog entries.

```
tacmd viewsystemlist –l Test_All_Managed_Systems
```

This example displays the details of a new manages system list.

```
tacmd viewsystemlist –l testList1
```

This example exports the managed system group testList1 to the specified file `apache_httpd.xml`.

```
tacmd viewsystemlist –l testList1 –e apache_httpd.xml
```

Return values

See “Return codes” on page 272.

Related commands

“tacmd createsystemlist” on page 62

“tacmd editsystemlist” on page 100

“tacmd deletesystemlist” on page 80

“tacmd listsystemlist” on page 190

Return to Table 1 on page 5.

tacmd viewUser

Description

Use the **tacmd viewUser** command to display the details of a specified user. To view users, the logged on user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the

viewUser command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd viewUser
{-i|--id} *USERID*
[{-u|--userid} *TEPS_USERID***]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPS_HOSTNAME***]**
[{-p|--permissions} **|**
{-a|--applications} **|**
{-v|--views} **|**
{-o|--memberof}]

where:

-i|--id
Specifies the user ID for which users are to be listed. The user ID must not contain any blank space characters in it, and its maximum allowed length is 10 characters and it must not begin with '*' or '_' character.

-u|--userid
Specifies the existing user ID to log on to Tivoli Enterprise Portal Server. The software prompts you for the user ID if you do not specify one.

-w|--password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s|--server
Specifies the Tivoli Enterprise Portal Server hostname from where the users are to be listed. If not specified, the users belonging to the group ID will be listed from the local Tivoli Enterprise Portal Server.

-p|--permissions
Displays permissions for the given user. The permissions inherited from group are prefixed with '#' and read only permissions will have 'I' appended to the permission value. The permissions are listed by realm (KFW is the default realm).

-a|--applications
Displays assigned and available applications for a given user. The applications inherited from group will be prefixed with '#'.

-v|--views
Displays assigned and available navigator views for the given user.

-o|--memberof
Displays the group names in which the user is member of.

Note: The -v, -p -a, and -o options are mutually exclusive. If you enter more than one, you will receive a message that the second option entered is repeating. For example:

```
C:\IBM\ITM\bin>tacmd viewuser -u sysadmin -w mypassword -a -v

   KUIC02022E: The command did not complete because -v option is repeating.
```

CLI example

This example displays permission related details for the user TESTUSER for the server HDCHASDSTC0219.

```
tacmd viewUser –i TESTUSER –u sysadmin –w "tivoli123" –s HDCHASDSTC0219 –p
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

tacmd viewUserGroup

Description

Use the **tacmd viewUserGroup** command to display details of the specified user group. To list group, the logged on user must have the following permissions on the Tivoli Enterprise Portal:

- User Administration -> Login
- User Administration -> View
- User Administration -> Modify

Note: If you want to use the current **tacmd tepsLogin** values for username, password, and server hostname, do not enter any of these options for the **viewUserGroup** command. If you specify values for some, but not all of these options, you are prompted to specify the username and password if they are missing.

CLI syntax

tacmd viewUserGroup **{-g|--gid}** *GROUPID*
[{-u|--userid} *TEPS_USERID* **]**
[{-w|--password} *TEPS_PASSWORD***]**
[{-s|--server} *TEPS_HOSTNAME*
[{-p|--permissions} **|**
{-a|--applications} **|**
{-v|--views} **|**
{-o|--memberof} **|**
{-b|--members}]

where:

-g|--gid
Specifies the new Group ID to be created. The Group ID must not contain any blank spaces characters in it. Its maximum allowed length is 32 characters, and it must begin with "_" or " * ".

-u|--userid
Specifies an existing user ID to log on to the Tivoli Enterprise Portal Server. The software prompts you for the user ID if you do not specify one.

-w | --password
Specifies the password for user authentication. The software prompts you for the password if you do not specify one.

-s | --server
Specifies the Tivoli Enterprise Portal Server hostname from where the user details are to be viewed. If not specified, the user details will be viewed from the local Tivoli Enterprise Portal Server.

-p | --permissions
Displays permissions for the given group. The permissions inherited from the group are prefixed with '#' and read only permissions will have 'I' appended to the permission value. The permissions are listed by realm (KFW is the default realm).

-a | --applications
Displays assigned and available applications for a given group. The applications inherited from the group will be prefixed with '#'.

-v | --views
Displays assigned and available navigator views for the given user group.

-o | --memberof
Displays the group names of which the group is member.

-b | --members
Displays the users and groups who belong to this group.

CLI example

This example displays application details belongs to the group *ADMINISTRATOR.

```
   tacmd viewUserGroup –g *ADMINISTRATOR –u sysadmin –w "tivoli123"
–s HDCHASDSTC0219 –v
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 1 on page 5.

Configuration options and properties

Description

For the **createNode**, **createGroup**, **addGroupMember**, and **editGroupMember** commands there are a number of configuration options and properties to be specified.

IBM Tivoli Monitoring OS Agents

For the **createNode** command options (–o | --option | --options), you can specify one or more of the following configuration options to customize the operation of the **createNode** command. The values are to be specified in key=value format. Some valid options are: VERSION, AUTOCLEAN, AGENT, JLOG_APPENDING, JLOG_SET_FILE_DIR, JLOG_SET_FILE_NAME, JLOG_SET_MAX_FILES, JLOG_SET_MAX_FILE_SIZE, ENV_env_variable_name.

Table 4. Options for the createNode command

Option name	Description
KEYFILE	The full path to a Secure Shell private key file that, when specified, is used to authenticate with the specified remote computer (which *must* already have been configured to accept the private key). Use **KEYFILE** when the Secure Shell protocol is, or might be, in use, and you want to use non-login based authentication.
PASSPHRASE	Use this option in conjunction with the **KEYFILE** option to enable the specification of the passphrase (if any) that was used to encrypt the private key file. If you use the actual passphrase in this command, that passphrase is visible in the process table for the current computer, where anyone can access it. If access to the computer is restricted to only trusted personnel, this is not an issue. However, if you need to provide the passphrase but do not want to publish the passphrase, instead of using the passphrase as the option value, specify the path to a secured local file that contains the passphrase.
TIMEOUT	A positive integer that indicates the maximum amount of time, in seconds, to wait for the current node creation to complete. If the node creation exceeds this time limit, then the node creation operation ends, and an error message is displayed. The default value is 1800 seconds (30 minutes). **Note:** In the event that a timeout is triggered, the node creation on the remote host might continue and complete successfully. However, if the timeout is exceeded, **createNode** does not wait for results to be returned from the remote host and reports a failure.
CONNECT_TIMEOUT	A positive integer that indicates the maximum amount of time, in seconds, to wait for a successful connection through each communications protocol being used to connect to the specified remote computer. The default value is 60 seconds.
TEMP=temp_dir	Defines a temporary directory to use during the deployment of the agent. By default, the **tacmd createNode** command uses the `/tmp` directory on UNIX systems and `%TEMP%` on Windows systems. Use the TEMP=temp_dir option if your `/tmp` directory does not have enough space to accommodate the temporary files needed for agent deployment. This option is not available for SSM agents.

For the **createNode**, **createGroup**, **addGroupMember**, and **editGroupMember** properties (--p | --property | --properties), you can specify the following properties:

Table 5. Valid properties for all operating systems unless specified

Property name in key format defined for createNode command	**Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands**	**Description**
ENCRYPT	KDY.ENCRYPT	A Boolean flag that indicates whether or not to use SSL encryption for node communication with the monitoring server. By default, encryption is enabled for all communications. Accepted values are NO and YES.
KEY	KDY.KEY	The text to be used to encrypt the SSL communications. This value *must* match the same value that was specified for the monitoring server encryption.
IP.PIPE	KDY.IP_PIPE	Specifiy the KDC_PARTITION Name for the IP.PIPE protocol.
IP.SPIPE	KDY.IP_SPIPE	Specifiy the KDC_PARTITION Name for the IP.SPIPE protocol.
IP.PIPE.PORT	KDY.IP_PIPE_PORT	This is for the IP_PIPE port.
IP.SPIPE.PORT	KDY.IP_SPIPE_PORT	This is for IP_SPIPE port.
PROTOCOL	KDY.PROTOCOL	Specifies the primary communications protocol used between the node and the monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA. **Note:** If both PROTOCOL or PROTOCOL1 and SERVER protocol information are specified, the SERVER protocol is used as the primary communications protocol and PROTOCOL or PROTOCOL1 information is ignored.

Table 5. Valid properties for all operating systems unless specified (continued)

Property name in key format defined for createNode command	Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
PROTOCOL1	KDY.PROTOCOL1	Specifies the primary communications protocol used between the node and the monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA. **Note:** If both PROTOCOL or PROTOCOL1 and SERVER protocol information are specified, the SERVER protocol is used as the primary communications protocol and PROTOCOL or PROTOCOL1 information is ignored.
PROTOCOL2	KDY.PROTOCOL2	Specifies a secondary communications protocol to use between the node and the monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA.
PROTOCOL3	KDY.PROTOCOL3	Specifies a third communications protocol to use between the node and the monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA.
PORT	KDY.PORT	This is for the TCPIP Port.

Table 5. Valid properties for all operating systems unless specified (continued)

Property name in key format defined for createNode command	Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
SERVER	KDY.SERVER	Enables you to specify a specific monitoring server. The default value is the monitoring server from which you are running the command. This property accepts an optional URL-style format, which enables you to specify the primary communication protocol and port number: `[{IP.UDP|IP.PIPE|IP.SPIPE|SNA}://][HOSTNAME][:PORT]` **Note:** If both PROTOCOL or PROTOCOL1 and SERVER protocol information are specified, the SERVER protocol is used as the primary communications protocol and PROTOCOL or PROTOCOL1 information is ignored.
SNA_NETNAME	KDY.SNA_NETNAME	The Systems Network Architecture primary network name.
SNA_LOGMODE	KDY.SNA_LOGMODE	The Systems Network Architecture one to eight character log mode name.
SNA_LUNAME	KDY.SNA_LUNAME	The Systems Network Architecture primary logical unit name.
SNA_TPNAME	KDY.SNA_TPNAME	The Systems Network Architecture transaction program specification. Accepted values are: SNASOCKETS (default), KDCLLBD, KDTMSNAP.QAUTOMON.
BACKUP	KDY.BACKUP	A Boolean flag used to indicate whether or not to use a secondary (back up) monitoring server for this node. By default, no backup is specified. Accepted values are NO and YES.

Table 5. Valid properties for all operating systems unless specified (continued)

Property name in key format defined for createNode command	Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
BSERVER	KDY.BSERVER	Specifies a backup monitoring server. **Note:** If both BPROTOCOL or BPROTOCOL1 and BSERVER protocol information are specified, the BSERVER protocol is used as the primary communications protocol and BPROTOCOL or BPROTOCOL1 information is ignored.
BPROTOCOL	KDY.BPROTOCOL	Specifies the primary communications protocol used between the node and the secondary monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA. **Note:** If both BPROTOCOL or BPROTOCOL1 and BSERVER protocol information are specified, the BSERVER protocol is used as the primary communications protocol and BPROTOCOL or BPROTOCOL1 information is ignored.
BPROTOCOL1	KDY.BPROTOCOL1	Specifies the primary communications protocol used between the node and the secondary monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA. **Note:** If both BPROTOCOL or BPROTOCOL1 and BSERVER protocol information are specified, the BSERVER protocol is used as the primary communications protocol and BPROTOCOL or BPROTOCOL1 information is ignored.
BPROTOCOL2	KDY.BPROTOCOL2	Specifies a secondary communications protocol to use between the node and the secondary monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA.

Table 5. Valid properties for all operating systems unless specified (continued)

Property name in key format defined for createNode command	Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
BPROTOCOL3	KDY.BPROTOCOL3	Specifies a secondary communications protocol to use between the node and the secondary monitoring server. Accepted values are: IP.UDP, IP.PIPE, IP.SPIPE, SNA.
BPORT	KDY.BPORT	This is for the TCPIP Port for backup server.
BIP.PIPE.PORT	KDY.BIP_PIPE_PORT	This is for the IP.PIPE port number for the backup server.
BIP.SPIPE.PORT	KDY.BIP_SPIPE_PORT	This is for the IP.SPIPE port number of the backup server.
BSNA_NETNAME	KDY.BSNA_NETNAME	The Systems Network Architecture primary network name for the secondary monitoring server.
BSNA_LOGMODE	KDY.BSNA_LOGMODE	The Systems Network Architecture one to eight character log mode name for the secondary monitoring server.
BSNA_LUNAME	KDY.BSNA_LUNAME	The Systems Network Architecture primary logical unit name for the secondary monitoring server.
BSNA_TPNAME	KDY.BSNA_TPNAME	The Systems Network Architecture transaction program specification for the secondary monitoring server. Accepted values are: SNASOCKETS (default), KDCLLBD, KDTMSNAP.QAUTOMON.
FOLDER	KDY.FOLDER	For Windows systems only. The name of the folder under which to place the node components. This is the folder name that is displayed in the Start menu.

Table 5. Valid properties for all operating systems unless specified (continued)

Property name in key format defined for createNode command	Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
createNode -d | --dir | --directory command-line option	KDYRXA.INSTALLDIR	This is the same as the -d | --dir | --directory option for createNode - Specifies the location on the specified host where the agent is installed. This location must be specified as a directory in absolute path format.
BIND.TO.NIC	KDY.BIND_TO_NIC	For Linux and UNIX systems only, indicates a specific IP address for the node to bind to on multi-homed computers.

For the **createGroup**, **addGroupMember**, and **editGroupMember** commands properties (-p | --property | --properties), the following properties must be specified in SECTION.KEY=value format:

Table 6. Valid properties for the OS agents

Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
KDYRXA.KEYFILE	The full path to a Secure Shell private key file that, when specified, is used to authenticate with the specified remote computer (which *must* already have been configured to accept the private key). Use **KEYFILE** when the Secure Shell protocol is, or might be, in use, and you want to use non-login based authentication.
KDYRXA.PASSPHRASE	Use this option in conjunction with the **KEYFILE** option to enable the specification of the passphrase (if any) that was used to encrypt the private key file. If you use the actual passphrase in this command, that passphrase is visible in the process table for the current computer, where anyone can access it. If access to the computer is restricted to only trusted personnel, this is not an issue. However, if you need to provide the passphrase but do not want to publish the passphrase, instead of using the passphrase as the option value, specify the path to a secured local file that contains the passphrase.

Table 6. Valid properties for the OS agents (continued)

Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	**Description**
KDYRXA.TIMEOUT	A positive integer that indicates the maximum amount of time, in seconds, to wait for the current node creation to complete. If the node creation exceeds this time limit, then the node creation operation ends, and an error message is displayed. The default value is 1800 seconds (30 minutes). **Note:** In the event that a timeout is triggered, the node creation on the remote host might continue and complete successfully. However, if the timeout is exceeded, the command does not wait for results to be returned from the remote host and reports a failure.
KDYRXA.CONNECT_TIMEOUT	A positive integer that indicates the maximum amount of time, in seconds, to wait for a successful connection through each communications protocol being used to connect to the specified remote computer. The default value is 60 seconds.
KDYRXA.TEMP	Defines a temporary directory to use during the deployment of the agent. Use the TEMP option if your `/tmp` directory does not have enough space to accommodate the temporary files needed for agent deployment.
KDYRXA.VERSION	Corresponds to the value that is displayed in the *Version* field that is displayed by the **viewDepot** command.
KDYRXA.AUTOCLEAN	Determines if the deployment bundle is deleted from the local computer (the target host of the deployment) after installation. The defaults value is YES.
KDYRXA.JLOG_APPENDING	Specifies whether entries should be appended to an existing file, or if a new log file should be created. The default value is "false".
KDYRXA.JLOG_SET_FILE_DIR	Specifies the directory where log files will be written on the server. The value defaults to `%CANDLE_HOME%\logs` on Windows computers, and `$CANDLEHOME/logs` on Unix and Linux computers.
KDYRXA.JLOG_SET_FILE_NAME	Specifies the name of the log file on the server. The default value is `trace_cn.log`.
KDYRXA.JLOG_SET_MAX_FILES	Specifies the maximum number of log files to use. The default value is 3.
KDYRXA.JLOG_SET_MAX_FILE_SIZE	Specifies the maximum size for each log file. The default value is 10,024 kilobytes (10 megabytes).

Table 6. Valid properties for the OS agents (continued)

Property name in SECTION.KEY format defined for createGroup, addGroupMember, and editGroupMember commands	Description
KDYRXA.ENV_env_variable_name	Allows the user to set an environment variable on the remote (target) environment during execution.

For information on RXA connection properties, see "RXA connection properties" on page 264.

System Services Monitors agents

For the **createNode** command, the valid properties for a System Service Monitors agent include: SVCUSERNAME, SVCPASSWORD, SNMPPORT, SNMPCOMMUNITY, COEXIST, OVERWRITE, SERVER_GUI, MS_SNMP_OVERRIDE, DISABLE_SNMPV2, V3AUTHPROTOCOL, V3AUTHPASSWORD, V3PRIVPROTOCOL, CORE_ONLY, V3PRIVPASSWORD, MANUAL_SERVICE, CLUSTER_INST, CLUSTER_GROUP, CORE_CONFIG_DISK, AGENTLOG, BYPASS_RECONFIG, AGENTLOGSIZE, SNMPTRAPVER, CONFIGDIR, INST_CONSOLE. The values are to be specified in *key=value* format. Values can differ per system.

Table 7. Valid properties for the System Service Monitor agents

Property name in *key* format defined for createNode, updateAgent, configureSystem, and removeSystem commands	Property name in SECTION.KEY format required for createGroup, addGroupMember, and editGroupMember commands	Description
SVCUSERNAME	SSMCONFIG.SVCUSERNAME	For Windows systems, this property sets the username used to register the Netcool/SSM service. If you specify this switch, you must also specify the SvcUserPassword parameter. On host computers with Active Directory installed (Windows 2000 domain controllers) this parameter is ignored and the local system account is used instead. The default: local system account.
SVCPASSWORD	SSMCONFIG.SVCPASSWORD	For Windows systems, this property, sets the password used to register the Netcool/SSM service. If you specify this parameter, you must also specify the SvcUser switch. On host computers with Active Directory installed (Windows 2000 domain controllers) this parameter is ignored and the local system account is used instead.
SNMPPORT	SSMCONFIG.SNMPPORT	The UDP port on which Netcool/SM Configuration listens for incoming SNMP requests. This port must be a number in the range 1-65535 inclusive and must not be in use by any other application.

Table 7. Valid properties for the System Service Monitor agents (continued)

Property name in *key* format defined for createNode, updateAgent, configureSystem, and removeSystem commands	**Property name in SECTION.KEY format required for createGroup, addGroupMember, and editGroupMember commands**	**Description**
SNMPCOMMUNITY	SSMCONFIG.SNMPCOMMUNITY	The community string for SNMP v1 and v2.
OVERWRITE	SSMCONFIG.OVERWRITE	For Windows systems, this property specifies that the current Netcool/Service Monitor Configuration installation is retained in the new installation: Y - configuration not retained, N - configuration retained.
SERVER_GUI	SSMCONFIG.SERVER_GUI	For Windows systems, this property installs the Netcool/SSM agent GUI components on server computers. These components (the MIB Explorer, Service Controller, and Desktop Help) are usually only installed on desktop computers: Y - Netcool/SSM agent GUI components are installed on server computers, N - no Netcool/SSM agent GUI components are installed on server computers.
MS_SNMP_OVERRIDE	SSMCONFIG.MS_SNMP_OVERRIDE	For Windows systems, if OverrideSNMP is specified, the installer stops the Microsoft SNMP service, sets its startup mode to manual, installs the Netcool/Service Monitor Configuration service and registers the Netcool/Service Monitor Configuration service in automatic startup mode. If OverrideSNMP is not specified, the installer leaves the Microsoft SNMP service running and registers the Netcool/ Service Monitor Configuration service in manual startup mode.
DISABLE_SNMPV1	SSMCONFIG.DISABLE_SNMPV1	Set to enable Netcool/SM Configuration to respond to SNMP v1 requests.
DISABLE_SNMPV2	SSMCONFIG.DISABLE_SNMPV2	Set to enable Netcool/SM Configuration to respond to SNMP v2 requests.
V3User	SSMCONFIG.V3User	Sets the username used by Netcool/SM Configuration in SNMPv3 communication.
V3AUTHPROTOCOL	SSMCONFIG.V3AUTHPROTOCOL	Sets the method of authentication used for the SNMP v3 protocol. Valid values are: NONE - No authentication, SHA - Secure Hash Algorithm, and MD5 - MD5 message digest algorithm. If you select a value other than NONE, you must also specify a password.

Table 7. Valid properties for the System Service Monitor agents (continued)

Property name in *key* format defined for createNode, updateAgent, configureSystem, and removeSystem commands	**Property name in SECTION.KEY format required for createGroup, addGroupMember, and editGroupMember commands**	**Description**
V3AUTHPASSWORD	SSMCONFIG.V3AUTHPASSWORD	Sets the password used in SNMPv3 authentication. This password must be at least 8 characters in length and can contain spaces.
V3PRIVPROTOCOL	SSMCONFIG.V3PRIVPROTOCOL	Sets the method used for encrypting SNMP v3 protocol messages. Valid values are: NONE - no encryption used and DES - Data Encryption Standard algorithm. If you set this parameter to a value other than NONE, you must also specify a password.
V3PRIVPASSWORD	SSMCONFIG.V3PRIVPASSWORD	Sets the password used in SNMPv3 encryption. This password must be at least 8 characters in length and can contain spaces.
CORE_ONLY	SSMCONFIG.CORE_ONLY	For Windows systems, this property sets the selection of components installed to core agent files only.
MANUAL_SERVICE	SSMCONFIG.MANUAL_SERVICE	For Windows systems, this property registers Netcool/SSM service in manual mode and does not start it after installation.
CLUSTER_INST	SSMCONFIG.CLUSTER_INST	For Windows systems, this property selects the cluster-aware installation mode for installing Netcool/SSM on a Windows server cluster node.
CLUSTER_GROUP	SSMCONFIG.CLUSTER_GROUP	For Windows systems, this property specifies the name of the cluster group on which to install Netcool/SSM.
CORE_CONFIG_DISK	SSMCONFIG.CORE_CONFIG_DISK	For Windows systems, this property specifies the name of the physical disk resource on which to install Netcool/SSM core configuration files.
BYPASS_RECONFIG	SSMCONFIG.BYPASS_RECONFIG	For Windows systems, this property bypasses the installer configuration dialogs during an upgrade.
AGENTLOG	SSMCONFIG.AGENTLOG	For UNIX systems, this property specifies the path and file name of the agent log file. The default: `agent.log`.
AGENTLOGSIZE	SSMCONFIG.AGENTLOGSIZE	For UNIX systems, this property specifies the maximum size of the agent log file (in bytes). The default: 1000000.
SNMPTRAPVER	SSMCONFIG.SNMPTRAPVER	For UNIX systems, this property specifies the format of notifications sent by the agent: 1- SNMP v1 Trap or 2 - SNMPv2 Trap. The default: 1.

Table 7. Valid properties for the System Service Monitor agents (continued)

Property name in *key* format defined for createNode, updateAgent, configureSystem, and removeSystem commands	**Property name in SECTION.KEY format required for createGroup, addGroupMember, and editGroupMember commands**	**Description**
CONFIGDIR	SSMCONFIG.CONFIGDIR	For UNIX systems, this property sets the agent configuration directory. The default: `/opt/netcool/ssm/config`.
COEXIST	SSMCONFIG.COEXIST	For UNIX systems, this property specifies whether any existing Netcool/SSM installation in any path other than that indicated by INST_PATH is removed during installation. If you want to install Netcool/SSM in the same directory as the existing installation, specify n. This option is intended for testing multiple installations on one computer. The default: y.
INST_CONSOLE	SSMCONFIG.INST_CONSOLE	For UNIX systems, this property selects whether the command console is installed: y - Command console installed and n - Command console not installed. The default: y.
SERVER	SSMCONFIG.SERVER	Specify the hostname of the Tivoli Enterprise Management Systems to run the command. If the server name specified is a remote monitoring server, the request is routed to the remote monitoring server to run. However, if the remote monitoring server is currently not connected to the hub monitoring server, the request is run at the hub monitoring server. The "TEMS Name" field of the output from the **tacmd getdeploystatus** command shows the monitoring server that the command is running.

Note: In order for all the standalone commands to work, you must provide the SECTION.KEY for the property to route the request to the remote monitoring server.

For information on RXA connection properties, see “RXA connection properties.”

RXA connection properties

The following RXA connection properties are valid for use with the **addGroupMember**, **createGroup**, and **editGroupMember** commands for group deployments with OS agents or System Service Monitor agents:

Property name	**Description**
KDYRXA.RXAUSERNAME	A valid user log in ID on the deployment target computer.
KDYRXA.RXAPASSWORD	The password for the user ID specified by KDYRXA.RXAUSERNAME.

Property name	Description
KDYRXA.SERVERLIST	One or more monitoring server names, separated by spaces, from which the group deployment operations should be issued.
KDYRXA.RXAPROTOCOL	Optionally identifies the RXA connection protocol to use. By default, all supported protocols are attempted until a connection is successfully established on one of them.
KDYRXA.RXAPORT	Optionally identifies the port number to be used to attempt to establish an RXA connection with the remote host. If not provided, the default port is used for the appropriate RXA protocol.

kincinfo

Description

Use the **kincinfo** command to validate your installation. With this command, you can display a range of installation information. In addition, you can save the installation information in a file or the debug information in a file.

Typing **kincinfo -?** displays this help:

```
kincinfo [-d] [-e] [-e product] [-g] [-i] [-l filename] [-o] [-r] [-t]
[-t product] [-v filename] [-x] [-z]
-d          Displays a list of installed products, which can be parsed.
-e          Displays the product name, version, build information and
application support propagation status.
-e product  Displays the product name, version, build information and
application support propagation status for a specific product code.
-g          Turns message globalization on (globalization is off by default).
-i          Lists the inventory.
-l filename Saves the output in the file.
-o          Displays a list of running agents with an additional "Configured"
column.
-r          Displays a list of running agents.
-t          Displays the product name, version, build information, and
installation date for all of the products installed in the installation directory.
-t product  Displays the product name, version, build information, and
installation date for a specific product code.
-v filename Saves debug information in the file.
-x          Displays information about non-agent bundles.
-z          Adds provisional information to -t, -d, and -i output.
```

CLI syntax

kincinfo
[-d]
[-e *product***]**
[-g]
[-i]
[-l *filename***]**
[-o]
[-r]
[-t]
[-t *product***]**

[-v *filename***]**
[-x]
[-z]

where:

-d Displays a list of installed products, which can be parsed.

-e *product*
Displays the product name, version, build information and application support propagation status (the "SDA STATUS" column) for installed agents. Use the kincinfo -e *product* command with the product code of the particular agent or the kincinfo -e command to list the information for all of the installed agents.

-g Turns message globalization on (globalization is off by default).

-i Lists the inventory in English.

-l *filename*
Saves the output in the file.

-o Displays a list of running agents with an additional 'Configured' column.

-r Displays a list of running agents.

-t Displays the product name, version, build information, and installation date for all of the products installed in the installation directory. See the *IBM Tivoli Monitoring Installation and Setup Guide* for information on arch abbreviation.

You can also use this option to review the installed support of self-described agents, which is displayed in a table. The following example shows the table for the r2 agent:

```
************* Thursday, May 20, 2010 3:23:29 PM *************
User       : Administrator         Group     : NA
Host Name  : NC045161              Installer : Ver: 062300000
CandleHome : C:\IBM\ITM
Installitm : C:\IBM\ITM\InstallITM
************************************************************
...Product Inventory

PC  PRODUCT DESC                                    PLAT   VER
BUILD         INSTALL DATE

R2  Agentless Monitoring for Windows Operating Sy  WINNT  06.23.00.00
201004121113  20100520 1459

PC  SELF-DESCRIBED APPLICATION SUPPORT PACKAGE     PLAT   APP VER

R2  Agentless Monitoring for Wind - r2tms_support  CMS    06.23.00.00
R2  Agentless Monitoring for Wind - r2tps_support  CNS    06.23.00.00
R2  Agentless Monitoring for Wind - r2tpw_support  XEB    06.23.00.00

PC  APPLICATION SUPPORT DESC                       PLAT   APP VER
BUILD         INSTALL DATE

R2  Agentless Monitoring for Windows Operating Sy  WICMS  06.23.00.00
201004121113  20100520 1512
```

Note: The **kincinfo -t** option displays a report of components description using FIXED widths of columns. For component description columns, the maximum length of the description text is limited to 45 characters (to ensure spaces between columns, total width of description column is 47

characters). If text exceeds the limit, it is trimmed. To avoid this condition, use **kincinfo -d** which will display an unformatted report.

-t *product*
Displays the product name, version, build information, and installation date for a specific product code.

-v *filename*
Saves verbose debug information in the file.

-x Displays information about non-agent bundles.

-z Adds provisional information to -t, -d, and -i output.

CLI example

The following example shows all installed products:

```
kincinfo -i
```

The following is the output of this example:

```
************ Tuesday, April 12, 2011 11:37:18 AM ************
User       : Administrator         Group     : NA
Host Name  : NC045161              Installer : Ver: 062300000
CandleHome : C:\IBM\ITM
Installitm : C:\IBM\ITM\InstallITM
************************************************************
...Product Inventory

IN        Windows Install Component
          WINNT Version: 06.23.00.00 Build: 201103301522

IN        TEP Desktop Windows Install Component
          WINNT Version: 06.23.00.00 Build: 201103301522

IN        TEMS Windows Install Component
          WINNT Version: 06.23.00.00 Build: 201103301522

IN        TEP Server Windows Install Component
          WINNT Version: 06.23.00.00 Build: 201103301522

IN        ITM 6.x Agent Install Component
          WINNT Version: 06.23.00.00 Build: 201103301522

IN        ITM 6.x Agent Install Component Extensions
          WINNT Version: 06.23.00.00 Build: 201103301522

A4        Monitoring Agent for i5/OS
          WINNT Version: 06.23.00.00 Build: 10891

A4        Monitoring Agent for i5/OS
          WINNT Version: 06.23.00.00 Build: 10891

A4        Monitoring Agent for i5/OS
          WINNT Version: 06.23.00.00 Build: 10891

A4        Monitoring Agent for i5/OS
          WINNT Version: 06.23.00.00 Build: 10891

AC        32/64 Bit Agent Compatibility Package
          WIX64 Version: 06.23.00.00 Build: 201103301522

CJ        Tivoli Enterprise Portal Desktop Client
          WINNT Version: 06.23.00.00 Build: d1089a

CQ        Tivoli Enterprise Portal Server
```

```
          WINNT Version: 06.23.00.00 Build: d1089a

CW        Tivoli Enterprise Portal Browser Client
          WINNT Version: 06.23.00.00 Build: d1089a

GL        Tivoli Enterprise Monitoring Agent Framework
          WIX64 Version: 06.23.00.00 Build: d1089a

GL        Tivoli Enterprise Monitoring Agent Framework
          WINNT Version: 06.23.00.00 Build: d1089a

GS        IBM GSKit Security Interface
          WIX64 Version: 07.40.27.00 Build: d1088a

GS        IBM GSKit Security Interface
          WINNT Version: 07.40.27.00 Build: d1088a

HD        Warehouse Proxy Agent
          WINNT Version: 06.23.00.00 Build: d1089a

HD        Warehouse Proxy
          WINNT Version: 06.23.00.00 Build: d1089a

HD        Warehouse Proxy
          WINNT Version: 06.23.00.00 Build: d1089a

HD        Warehouse Proxy
          WINNT Version: 06.23.00.00 Build: d1089a

HD        Warehouse Proxy
          WINNT Version: 06.23.00.00 Build: d1089a

IT        TEC GUI Integration
          WINNT Version: 06.23.00.00 Build: d1088a

IT        TEC GUI Integration
          WINNT Version: 06.23.00.00 Build: d1088a

IT        TEC GUI Integration
          WINNT Version: 06.23.00.00 Build: d1088a

IU        IBM HTTP Server
          WINNT Version: 07.00.00.00 Build: d1088a

IW        Tivoli Enterprise Portal Server Extensions
          WINNT Version: 07.00.15.00 Build: d1088a

JM        Embedded JVM
          WINNT Version: 05.12.01.00 Build: e1088a

KF        IBM Eclipse Help Server
          WINNT Version: 06.23.00.00 Build: d1088a

LZ        Monitoring Agent for Linux OS
          WINNT Version: 06.23.00.00 Build: 10891

LZ        Monitoring Agent for Linux OS
          WINNT Version: 06.23.00.00 Build: 10891

LZ        Monitoring Agent for Linux OS
          WINNT Version: 06.23.00.00 Build: 10891

LZ        Monitoring Agent for Linux OS
          WINNT Version: 06.23.00.00 Build: 10891

MS        Tivoli Enterprise Monitoring Server
          WINNT Version: 06.23.00.00 Build: d1089a
```

```
NT        Monitoring Agent for Windows OS
          WINNT Version: 06.23.00.00 Build: 10891

NT        Monitoring Agent for Windows OS
          WINNT Version: 06.23.00.00 Build: 10891

NT        Monitoring Agent for Windows OS
          WINNT Version: 06.23.00.00 Build: 10891

NT        Monitoring Agent for Windows OS
          WINNT Version: 06.23.00.00 Build: 10891

PA        Tivoli Performance Analyzer
          WINNT Version: 06.23.00.00 Build: 10881

PA        Tivoli Performance Analyzer
          WINNT Version: 06.23.00.00 Build: 10881

PA        Tivoli Performance Analyzer
          WINNT Version: 06.23.00.00 Build: 10881

PA        Tivoli Performance Analyzer
          WINNT Version: 06.23.00.00 Build: 10881

PA        Tivoli Performance Analyzer
          WINNT Version: 06.23.00.00 Build: 10881

R2        Agentless Monitoring for Windows Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221538

R2        Agentless Monitoring for Windows Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221538

R2        Agentless Monitoring for Windows Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221538

R2        Agentless Monitoring for Windows Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221538

R3        Agentless Monitoring for AIX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221541

R3        Agentless Monitoring for AIX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221541

R3        Agentless Monitoring for AIX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221541

R3        Agentless Monitoring for AIX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221541

R4        Agentless Monitoring for Linux Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221543

R4        Agentless Monitoring for Linux Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221543

R4        Agentless Monitoring for Linux Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221543

R4        Agentless Monitoring for Linux Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221543

R5        Agentless Monitoring for HP-UX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221545
```

```
R5        Agentless Monitoring for HP-UX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221545

R5        Agentless Monitoring for HP-UX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221545

R5        Agentless Monitoring for HP-UX Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221545

R6        Agentless Monitoring for Solaris Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221547

R6        Agentless Monitoring for Solaris Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221547

R6        Agentless Monitoring for Solaris Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221547

R6        Agentless Monitoring for Solaris Operating Systems
          WINNT Version: 06.23.00.00 Build: 201103221547

SY        Summarization and Pruning Agent
          WINNT Version: 06.23.00.00 Build: d1053a

SY        Summarization and Pruning Agent
          WINNT Version: 06.23.00.00 Build: d1053a

SY        Summarization and Pruning Agent
          WINNT Version: 06.23.00.00 Build: d1053a

SY        Summarization and Pruning Agent
          WINNT Version: 06.23.00.00 Build: d1053a

SY        Summarization and Pruning Agent
          WINNT Version: 06.23.00.00 Build: d1053a

T1        File Transfer Enablement
          WINNT Version: 07.30.00.00 Build: 201000000000

TM        IBM Tivoli Monitoring 5.x Endpoint Support
          WINNT Version: 06.23.00.00 Build: d1088a

TM        IBM Tivoli Monitoring 5.x Endpoint Support
          WINNT Version: 06.23.00.00 Build: d1088a

UE        Tivoli Enterprise Services User Interface Extensions
          WINNT Version: 06.23.00.00 Build: d1089a

UI        Tivoli Enterprise Services User Interface
          WINNT Version: 06.23.00.00 Build: 201103301522

UL        Monitoring Agent for UNIX Logs
          WINNT Version: 06.23.00.00 Build: 10811

UL        Monitoring Agent for UNIX Logs
          WINNT Version: 06.23.00.00 Build: 10811

UL        Monitoring Agent for UNIX Logs
          WINNT Version: 06.23.00.00 Build: 10811

UL        Monitoring Agent for UNIX Logs
          WINNT Version: 06.23.00.00 Build: 10811

UM        Universal Agent
          WINNT Version: 06.23.00.00 Build: d1082a

UM        Universal Agent
```

```
          WINNT Version: 06.23.00.00 Build: d1082a

UM        Universal Agent
          WINNT Version: 06.23.00.00 Build: d1082a

UM        Universal Agent
          WINNT Version: 06.23.00.00 Build: d1082a

UX        Monitoring Agent for UNIX OS
          WINNT Version: 06.23.00.00 Build: 10881

UX        Monitoring Agent for UNIX OS
          WINNT Version: 06.23.00.00 Build: 10881

UX        Monitoring Agent for UNIX OS
          WINNT Version: 06.23.00.00 Build: 10881

UX        Monitoring Agent for UNIX OS
          WINNT Version: 06.23.00.00 Build: 10881
```

The following example shows the product name, version, build information and application support propagation status (the "SDA STATUS" column) for installed agents.:

```
kincinfo -e
```

The following is the output of this example:

```
************ Tuesday, March 22, 2011 2:47:26 PM *************
User       : Administrator         Group     : NA
Host Name  : NC045158              Installer : Ver: 062300000
CandleHome : C:\IBM\ITM
Installitm : C:\IBM\ITM\InstallITM
************************************************************
...Application support propagation


PC  PRODUCT DESC                     PLAT   VER          BUILD    SDA STATUS

HD  Warehouse Proxy Agent            WINNT  06.23.00.00  d1076a   Enabled
PA  Tivoli Performance Analyzer      WINNT  06.23.00.00  10611    Disabled
SY  Summarization and Pruning Agent  WINNT  06.23.00.00  d1053a   Enabled
```

Related commands

Return to Table 1 on page 5.

KinCli.exe command

Description

Use the **KinCli.exe** command to generate response files for specified functions.

CLI syntax

KinCli.exe
{**-***FUNCTION*}
{**-a**| **--***AGENT_CODE*}
[{**-i**| **--***INSTANCE_NAME*}
{**-p** | **--***IMAGE_FILE_PATH*}
{**-o**| **--** *OUTPUT_PATH*}]
[{**-Lfile**}]

where:

-*FUNCTION*
Specifies the function name, including STARTAGENT, STOPAGENT, RESTARTAGENT, CONFIGURERESOURCE, ADDRESOURCE, and GENERATERESPONSE.

-a | --*AGENT_CODE*
Specifies the type of agent.

-i | --*INSTANCE_NAME*
Specifies the optional instance name of the agent type. This option is required for the GENERATERESPONSE function.

-p | --*IMAGE_FILE_PATH*
Specifies the image file path for the ADDRESOURCE function.

-o | --*OUTPUT_PATH*
Specifies the output path for the GENERATERESPONSE function.

-Lfile Specifies the log file name.

CLI example

The following example generates a response file for the Windows OS agent with the instance name of clump. The response file is called `silent_install_knt.txt` and is saved in the *`install_dir`*`/out` directory.

```
   KinCli.exe -GENERATERESPONSE -a knt -i clump -o install_dir/out
-L"silent_install_knt.txt"
```

Related commands

Return to Table 1 on page 5.

Return codes

The following table lists the return codes for the **tacmd** commands.

Table 8. Return codes for tacmd CLI commands

Code	Category	Description
0	Success	Indicates that the command was successful.
1	Syntax Error or Help	Indicates either that the help command was given or that the syntax used was incorrect.
2	No Permission	Indicates that the user does not have permission to issue the command.
3	Version Mismatch	Indicates that the version of the server is not what was expected.
4	Communication Error	Indicates that an error occurred in the communications with the server.
5	Timeout	Indicates that an operation waiting for data did not receive it within the time it was expected.
6	Input Error	Indicates that the input to the command was not what was expected.
7	Server Exception	Indicates that an error occurred on the server that caused the command to fail.

Table 8. Return codes for tacmd CLI commands (continued)

Code	Category	Description
8	Command Error	Indicates that an internal error occurred while executing the command.
9	Invalid Object	Indicates that a specified object does not exist.
10	Duplicate Object	Indicates that the object is a duplicate of an already existing object.
11	Partial Success	Indicates that the command was partially successful.
15	Password too long	Indicates that the password is longer than 16 characters, which is the maximum password length
19	Situation not present	Indicates that no historical situation is available on the server.
100	Deploy Queued	Indicates that the deployment operation is queued.
101	Deploy in Progress	Indicates that the deployment operation is in progress.
102	Deploy Retryable	Indicates that the deployment operation is retryable.
103	Deploy Failed	Indicates that the deployment operation failed.
1001	Request Queued	Indicates a request queued, waiting for async response.
1002	Shortage	Indicates a memory shortage.
1003	Bad Argument	Indicates a bad input argument.
1004	Not There	Indicates that a file was not found.
1005	System Error	Indicates an unknown system error.
1006	Duplicate	Indicates that a request for same pc is already in progress.
1007	KT1 Error	Indicates a KT1 error.
1008	SDM Disabled	Indicates that self-described processing is disabled at the monitoring server.
1009	HUB Not There	Indicates that the hub monitoring server is not available.
1010	Shutdown	Indicates that a monitoring server is shutting down.
1011	Manifest Error	Indicates invalid content in the manifest file.
1012	Wrong TEMS Version	Indicates an incorrect monitoring server version.
1013	Nonsupportable Feature	Indicates a required feature (for example, a new data type) is not supported by the monitoring server.
1014	Unknown Error	Indicates an unknown error.
1015	Bad Argument Length	Indicates a bad input argument length.
1016	Manual Install Update	Indicates a record updated by the manual installation detection process.
1017	Temp Install Error	Indicates a temporary installation error; the agent retries the installation request.

Chapter 3. itmcmd commands

You can run the itmcmd commands only on UNIX monitoring servers.

Table 9. itmcmd commands

Command	Description
"cinfo"	View information for your monitoring server, including inventory of installed IBM Tivoli products, configuration settings, installed CD versions, and a list of running IBM Tivoli processes.
"itmcmd agent" on page 279	Start and stop a monitoring agent.
"itmcmd audit" on page 283	Manage the size and number log files.
"itmcmd config" on page 284	Configure or reconfigure the execution environment for an agent or server.
"itmcmd dbagent" on page 287	Start and stop a distributed database monitoring agent,
"itmcmd dbconfig" on page 288	Configure the execution environment for the distributed database agent.
"itmcmd execute" on page 289	Run a user script or command when its execution requires the same environment settings as for a particular IBM Tivoli product.
"itmcmd history" on page 290	Manage the rolloff of history data into delimited text files.
"itmcmd manage" on page 292	Start, stop, and configure monitoring components.
"itmcmd resp" on page 293	Creates silent response file.
"itmcmd server" on page 294	Start and stop monitoring servers.
"itmcmd support" on page 295	Add agent-specific information to the monitoring server.
"SetPerm" on page 296	Set file permissions to ensure that the permissions were set properly during the installation procedure.
"tmsdla" on page 297	Generate the XML output file in the `installdir\CNPS\tmsdla` subdirectory on the portal server.

Note: If you export the CANDLEHOME environment variable to your current session, many of the installation and configuration commands do not require that CANDLEHOME be passed to them (usually by using the -h CLI option).

cinfo

Description

Use the **cinfo** command to view the following information for your monitoring server:

- An inventory of installed IBM Tivoli products
- The configuration settings for products
- The configuration settings for products in the context of the actual variables used by the installation program
- A list of running IBM Tivoli processes (such as agents or monitoring server)
- A validated list of running IBM Tivoli processes, after first performing an update on the tracking database, to remove stale PIDs (processes logged as "running" but not found when attempting to verify by using the **ps** command)
- The product name, version, build information, and installation date for all of the products installed in the installation directory

The command can be run in several ways.

Typing **cinfo** enables this menu:

```
-- CINFO Menu --
1) Show products installed in this CandleHome
2) Show which products are currently running
3) Show configuration settings
4) Exit CINFO
```

The command can also be run without a menu, so the three numbered options above can be invoked as:

```
cinfo -i
cinfo -r
cinfo -c all or cinfo -c <pc>
```

Typing **cinfo -?** displays this help:

```
cinfo [-h candle_directory] [-b] [-c product] [-d] [-e] [-e product] [-g] [-i] [-o]
[-o product] [-p product] [-r] [-R] [-s product] [-t] [-t product]
-b Dumps an inventory of installed products with build information
-c <product> Displays configuration prompts and configuration values
-d Dumps an inventory of installed products
-e or -e <product> Displays the product name, version, build information and
application support propagation status for installed agents
-g Enable globalization
-i Displays an inventory of installed products
-o or -o <product> Lists configured instances
-p <product> Shows associated platform codes for the specified product
-r Shows running processes
-R Shows running processes, after updating a tracking database
-s <product> Displays configuration parameters and settings
-t or -t <product> Displays the product name, version, build information and
installation date
```

CLI syntax

cinfo [-h *install_dir*]
{[-b] |
[-c {*pc***|all}]** |
[-d] |
[-e [*pc***]]** |
[-g] |
[-i] |
[-o [*pc***]]** |
[-p {*pc***}]** |
[-r] |

[-R] |
[-s {*pc*|all**}] |**
[-t [*pc*]] }

where:

-h *install_dir*
Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

-b Dumps an inventory of installed products with build information.

-c {*pc*} Lists configuration prompts and configuratioin values for all components (by default) or for a specific component (identified by product code). If you use the product code, you can only enter one code.

Note: When you are running cinfo -c cq and are asked whether "you are using DB2, Oracle, or None for Warehouse? [1=DB2, 2=Oracle, 3=None]", JDBC=Oracle and None=Debby.

-d Dumps an inventory of installed products.

-e [*pc*] Displays the product name, version, build information and application support propagation status (the “SDA STATUS” column) for installed agents. Use the cinfo -e [*pc*] command with the product code of the particular agent or the cinfo -e command to list the information for all of the installed agents.

-g Enables globalization.

-i Shows an inventory of installed components. Lines starting with an architecture string indicate that the set of binaries for that component are installed. The following table specifies the strings at the beginning of the line(s) and their implication.

Table 10. String at beginning of lines and their implication

String at Start of Line	Description
tms	Application support for the Tivoli Enterprise Monitoring Server has been installed
tpa	Application support for the Tivoli Performance Analyzer has been installed
tpd	Application support for the Desktop Client has been installed
tps	Application support for the Tivoli Enterprise Portal Server has been installed
tpw	Application support for the Tivoli Enterprise Portal has been installed

-o [*pc*] Lists configured instances for the product, if specified, or all of the products, if not specified.

-p {*pc*} Shows associated platform codes for the product, if specified, or all of the products, if not specified.

-r Shows the running processes.

-R Shows the running processes, after updating a tracking database.

All started processes started and stopped by IBM Tivoli Monitoring commands are logged in a tracking database that does not automatically update itself if the process abnormally ends or is stopped without using an IBM Tivoli Monitoring command (for example, by using the UNIX **kill** command directly on the process). The **-r** option shows even these defunct processes in its report. In contrast, the **-R** option updates the tracking database before reporting the results. This results in a cleaner report, but permanently erases the history of processes normally kept in the tracking database.

The output of the **-R** option looks the same as the **-r** option, but any "process not running" messages are absent.

-s {*pc* | **all**}
Shows the configuration parameters and settings for either a single component (identified by using the product code) or for all installed components.

-t [*pc*] Displays the product name, version, build information, and installation date for all of the products installed in the installation directory. See the *IBM Tivoli Monitoring Installation and Setup Guide* for information on arch abbreviations.

CLI example

The following example shows all installed products:

```
cinfo -i
```

The following is the output of this example:

```
*********** Tue Mar 22 13:48:17 CET 2011 ******************
User: root Groups: root bin daemon sys adm disk wheel
Host name : NC045192     Installer Lvl:06.23.00.00
CandleHome: /opt/IBM/ITM
***********************************************************
...Product inventory

a4      Monitoring Agent for i5/OS
         tms     Version: 06.23.00.00

ax      IBM Tivoli Monitoring Shared Libraries
         li6263  Version: 06.23.00.00

gs      IBM GSKit Security Interface
         li6243  Version: 07.40.27.00

hd      Warehouse Proxy
         tms     Version: 06.23.00.00

(...)
```

The following example shows the running processes:

```
cinfo -r
```

The following is the output of this example:

```
*********** Tue Mar 22 13:52:37 CET 2011 ******************
User: root Groups: root bin daemon sys adm disk wheel
Host name : NC045192     Installer Lvl:06.23.00.00
CandleHome: /opt/IBM/ITM
***********************************************************
Host      Prod  PID    Owner  Start  ID    ..Status
nc045192  ms    22072  root   13:52  TEMS  ...running
```

The following example shows the configuration settings for the Universal Agent:

```
cinfo -c um
```

The following is the output of this command:

```
*********** Tue Mar 22 13:58:32 CET 2011 ******************
User: root Groups: root bin daemon sys adm disk wheel
Host name : NC045192     Installer Lvl:06.23.00.00
CandleHome: /opt/IBM/ITM
************************************************************

Configuration    Setting
um  default      "Network Protocol 3 " = none
um  default      "Data Provider" = ASFS
um  default      "Secondary TEMS IP.PIPE Port Number" = 1918
um  default      "Secondary TEMS IP.SPIPE Port Number" = 3660
um  default      "Secondary TEMS Log Mode" = LOGMODE
um  default      "Secondary TEMS LU Name" = LUNAME
um  default      "Secondary TEMS Net Name" = CANDLE
um  default      "Secondary TEMS IP Port Number" = 1918
um  default      "IP.PIPE Port Number" = 1918
um  default      "IP.SPIPE Port Number" = 3660
um  default      "Log Mode" = LOGMODE
um  default      "LU Name" = LUNAME
um  default      "Net Name" = CANDLE
um  default      "IP Port Number" = 1918
um  li6263       "Network Protocol 3 " = none
um  li6263       "Will this agent connect to a TEMS? [1=YES, 2=NO]" = 1
um  li6263       "Data Provider" = ASFS
um  li6263       "Configure connection for a secondary TEMS? [1=YES, 2=NO]" = 2
um  li6263       "TEMS Host Name" = nc045192
um  li6263       "Secondary TEMS IP.PIPE Port Number" = 1918
um  li6263       "Secondary TEMS IP.SPIPE Port Number" = 3660
um  li6263       "Secondary TEMS Log Mode" = LOGMODE
um  li6263       "Secondary TEMS LU Name" = LUNAME
um  li6263       "Secondary TEMS Net Name" = CANDLE
um  li6263       "Secondary TEMS IP Port Number" = 1918
um  li6263       "IP.PIPE Port Number" = 1918
um  li6263       "IP.SPIPE Port Number" = 3660
um  li6263       "Log Mode" = LOGMODE
um  li6263       "LU Name" = LUNAME
um  li6263       "Net Name" = CANDLE
um  li6263       "IP Port Number" = 1918
um  li6263       "Enter Optional Primary Network Name or 0 for "none"" = none
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

itmcmd agent

Description

Use the **itmcmd agent** command to start and stop a monitoring agent. You can start or stop one agent, all agents, or multiple agents. You cannot start or stop multiple instance agents on one command. You can also start the portal server and portal desktop client by using this command.

You must run the **itmcmd agent** command on the architecture for which the agent is installed.

To start or stop agents for distributed database agent, see “itmcmd dbagent” on page 287. However, the **itmcmd agent** command can start and stop agents for distributed databases, although it cannot select monitors for individual database servers or activate debugging options.

Note: The monitoring agent can run using a non-root user ID on UNIX and Linux systems. This can be done by running the **itmcmd agent start** command while logged in as a non-root user. If the agent was running using a non-root user ID, and then the **itmcmd agent start** is run using the root user ID, then the monitoring agent subsequently runs as the root user. To confirm the user ID that the monitoring agent is using, run the following command:

```
itm_install/bin/cinfo -r
```

If the installation is not permissioned properly, then you might be unable to restart the agent as a non-root user ID after it has been run as the root user ID. To prevent this problem, ensure that the **secureMain lock** command with the -g option has been previously run. See the "Securing your IBM Tivoli Monitoring installation on Linux or UNIX" appendix in the *IBM Tivoli Monitoring Installation and Setup Guide* for further details.

If the agent is running as root, and that is not the desired user ID, then use the following steps to restart the agent:

1. Log in as root.
2. Run the **itmcmd agent stop** command.
3. Log in (or 'su') to the user ID that you want the agent to run as.
4. Run the **itmcmd agent start** command.

If the agent was running as root because of a system reboot, then complete the following steps so that the appropriate user ID is used the next time the system is rebooted. Editing the startup file is no longer supported. Instead you must modify the `config/kcirunas.cfg` file and then run `bin/UpdateAutoRun.sh`:

1. Edit *install_dir*`/config/kcirunas.cfg`.
2. Add a section, after the `<agent>` line, to specify the agent or agent instance that you want to start as a specific user ID. To specify the user ID to start a non-instance agent, or to start all instances of an agent, use the following syntax:

   ```
   <productCode>product_code</productCode>
   <default>
      <user>user_name</user>
   </default>
   ```

   To specify different user IDs to start different instances of an agent, use the following syntax:

   ```
   <productCode>product_code</productCode>
   <instance>
      <name>instance_name1</name>
      <user>user_name</user>
   </instance>
   <instance>
      <name>instance_name2</name>
      <user>user_name</user>
   </instance>
   ```

Where:

product_code
2-character product code for the agent, for example, *ux* for the Unix OS monitoring agent.

user_name
Name of the user to run the agent as.

instance_name1
Name of an instance.

instance_name2
Name of another instance.

Examples:

For the Unix OS monitoring agent, to run as itmuser:

```
<productCode>ux</productCode>
<default>
   <user>itmuser</user>
</default>
```

For the DB2 monitoring agent instances to run as the instance owner IDs:

```
<productCode>ud</productCode>
<instance>
   <name>db2inst1</name>
   <user>db2inst1</user>
</instance>
<instance>
   <name>db2inst2</name>
   <user>db2inst2</user>
</instance>
```

For the Websphere MQ monitoring agent instances to all run as the mqm user ID, and for the default instance to not be started:

```
<productCode>mq</productCode>
<default>
   <user>mqm</user>
</default>
<instance>
   <name>None</name>
   <autoStart>no</autoStart>
</instance>
```

3. Repeat step 2 for each agent or agent instance that you want to start as a specific user ID.
4. Save the file.
5. Run *install_dir*/bin/UpdateAutoRun.sh as root user.

CLI syntax

itmcmd agent
[-h ***install_dir*** **]**
[-f]
[-l]
[-o ***instance*** **]**
[-p ***option*** **]**
[-c]
[-n]
[-m]
{start|stop} {*pc* **...|all}**

where:

-h *install_dir*
(optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

-f (optional) Starts or stops an agent without user confirmation.

Note: When stopping an agent, this option must be entered before the stop option or you will receive an error. For example:

```
itmcmd agent -p INST1 -f stop um
```

-l (optional) Deletes the log file associated with the monitoring agent that is being stopped. By default, the log file is saved when the monitoring agent is stopped.

-o *instance*
(optional) Identifies a database instance to start or stop. You must use this option if you are starting or stopping a DB2® agent.

-p *option*
(optional) Identifies a Universal Agent instance to start or stop. Use this option when you are starting or stopping a non-default instance of the Universal Agent.

-c (optional) Indicates that the configuration file used on agent startup should not be updated or regenerated. By default, this file is updated each time the agent is started.

-n (optional) When specified, indicates that the PID is not checked. If the PID of the initial agent is killed, then another process takes that PID.

-m (optional) If specified, indicates the creation of multiple instances.

start | **stop** {*pc ...* | **all**}
Indicates to start or stop the monitoring agent. You can start or stop one or more agents by using the product codes (for example, specifying `lz um` starts the Linux monitoring agent and the Universal Agent). To start or stop all agents on the computer, use the **all** option.

See "cinfo" on page 275 to identify the product code for an agent or component.

CLI example

The following example starts the Universal Agent:

```
itmcmd agent start um
```

The following example stops a non-default instance (inst1) of the Universal Agent:

```
itmcmd agent -p INST1 stop um
```

The following example starts the portal server:

```
itmcmd agent start cq
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Note: The log file for the agent session is always saved, regardless of whether the agent was stopped by using the **itmcmd agent** command or any other means, unless you use the **-l** option when you run the **itmcmd agent** command. Additionally, when the agent is stopped by using the **itmcmd agent** command, the log file for that session ends with the following message:

```
"** Process terminated by user **"
```

Related commands

"itmcmd server" on page 294

"cinfo" on page 275 (to determine the product codes for agents and components)

Return to Table 9 on page 275.

itmcmd audit

Description

Use **itmcmd audit** command to manage the size and number log files. These logs are located in `/staging2/candle-m8//logs/`.

Depending on the number of products you install and the amount of activity on your system, managing the size and number of log files in your *install_dir* directory can be critical. The **itmcmd audit -l** command enables you to remove or truncate log files.

The **itmcmd audit -l** command only takes action on those log files that are stored in the *install_dir*/*logs* subdirectory for the *install_dir* directory in which it is run.

The **itmcmd audit** command can consume time and resource. For best results, run this command during off hours. Run the **itmcmd audit** command to trim or delete files only when the agent is not running.

Note: This command does not audit the installation log file.

CLI syntax

itmcmd audit -l age [*logdays***]**
[-h *$install_dir*]
[-c]

itmcmd audit -l size [*logsize***]**
[-h *$install_dir*]
[-c]

/itmcmd audit -l both [*logdays logsize***]**
[-h *$install_dir*]
[-c]

where:

-l (required) Runs the log management function. Use one of the following arguments:

age [*logdays***]**
Removes all log files that are older than a specified number of days. The age is determined by the last modification date. The default age is 30 days.

size [*logsize***]**
Trims log files to a specified number of bytes. The oldest entries are removed first. The default size is 1024 bytes.

both [*logdays logsize***]**
First removes log files older than the specified number of days, then trims the remaining files to the specified size.

-h *$install_dir*
(optional) Specifies the *$install_dir*, if it is not defined for the current environment.

-c (optional) Displays diagnostic messages from the command to the screen.

CLI example

The following example removes all log files older than 20 days:

```
itmcmd audit -l age 20
```

Return values

An exit status of 0 indicates that the command executed successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

itmcmd config

Description

Use the **itmcmd config** command to configure or reconfigure the execution environment for an agent or server, including the following IBM Tivoli Monitoring items:

- The IP port that the hub monitoring server uses to listen for requests
- The hosts that can run a product
- The location of the hub monitoring server in the network
- The monitoring server an agent connects to
- Whether a monitoring server is a hub or remote server

You can only configure one product at a time. If you reconfigure a monitoring server, you must stop and restart that monitoring server before the changes take effect.

The **itmcmd config** command prompts for input for the required options. Scripts are located in the *install_dir*/bin directory where *install_dir* is the directory into which you installed IBM Tivoli Monitoring.

On UNIX or Linux only, add the same variable and location to the `kbbenv.ini` file located in *itm_installdir*`/config/kbbenv.ini`. If you do not add the variable to the `kbbenv.ini` file, it is deleted from the `KBBENV` file the next time the monitoring server is re-configured.

Note: When using the **itmcmd config** command to configure either the Warehouse Proxy Agent or the Summarization and Pruning agent, note that the command line does not have a validation mechanism like the GUI does.

Note: When using the –A hd options, this command can be used on only UNIX or AIX systems.

CLI syntax

Use the following syntax to configure a monitoring server:

itmcmd config -S
-t *tems_name*
[-h *install_dir***]**
[-p *silent_response_file***]**
[-r]
[-y]
[-a *arch***]**

itmcmd config -S
-t *tems_name*
-u
pc
[-h *install_dir***]**

itmcmd config -S
[-t *tems_name***]**
[-h *install_dir***]**
[-g]

Use the following syntax to configure a monitoring agent:

itmcmd config -A
[-h *install_dir***]**
[-p *silent_response_file***]**
[-r]
[-y]
[-a *arch***]**
[-t *agent_host_name***]**
pc

itmcmd config -A
[-h *install_dir***]**
[-p *silent_response_file***]**
[-r]
[-y]
[-a *arch***]**
[-o *instance_name***]**
pc

```
itmcmd config -A
              [-h install_dir]
              [-g]
              pc
```

Use the following syntax to configure the Tivoli Enterprise Monitoring Server Automation Server:

```
itmcmd config -A as
```

where:

as Indicates the product code of the Tivoli Enterprise Monitoring Automation Server.

Use the following syntax to configure an Oracle agent:

```
itmcmd config -A
              or
              [-h install_dir]
              [-o "servername,userid,pwd"]
```

Use the following syntax to configure a Warehouse Proxy agent:

```
itmcmd config -A hd
```

Use the following syntax to configure a Summarization and Pruning agent:

```
itmcmd config -A sy
```

where:

-S Indicates that you are configuring a monitoring server.

-A Indicates that you are configuring a monitoring agent.

-h *install_dir*
(Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

-a *arch* Specifies the architecture where *arch* is one of the abbreviations used to indicate architecture.

This parameter enables you to configure an agent and a monitoring server for an architecture other than the one that you are on. For example, if you are on AIX and want to configure for Solaris computer, then this option is required. Otherwise the default is the computer you are on. (Optional)

-u Adds application support (catalog and attribute files) to a monitoring server for agents that were not installed or for non-UNIX-based agents. If you specify the **-u** option, you must also specify the product code (*pc*) for the agent or agents. Only used with the **-S** option.

-t *tems_name*
The name of the monitoring server. (Required)

-o *instance_name*
The instance name for the agent that you want to start.

-g Displays the configuration settings.

-p Configures the component by using a silent response file.

-r Rebuilds the configuration. This option is only available for the Tivoli Enterprise Monitoring Server, the Tivoli Enterprise Portal Server or the Tivoli Enterprise Portal Desktop Client and cannot be used with the -p option. Using the -y option with the -r option reconfigures in a silent mode that does not require any confirmations.

pc The product code for the agent or component that you want to configure.

-y This option is only used with the -r option. Using the -y option with the -r option reconfigures in a silent mode that does not require any confirmations.

CLI example

The following example configures the monitoring server ms1:

```
itmcmd config -S -t hub_ms1
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

itmcmd dbagent

Description

Use the **itmcmd dbagent** command to start and stop a distributed database monitoring agent, including the Monitoring Agent for Sybase and the Monitoring Agent for Oracle.

To start other monitoring agents, see "itmcmd agent" on page 279.

CLI syntax

itmcmd dbagent
[-h *install_dir***]**
[-d *trace_option***]**
[-s *server* **...]**
{start|stop} [*pc [,pc]* **...]**

where:

-d Enables diagnostic reporting for one or all monitored database tables. Enables debug tracing for the following items:

Table Turns on KBB_RAS1 tracing for table (korxxxx, kraxxxx). Table names are case-insensitive. You can use ksh wildcards (but not regexp).

debug Turns on collector and agent internal tracing through -dddd.

d Fine tunes internal tracing level: -d, -dd, -ddd, -dddd, -ddddd (debug or ddd's also change col.out to wrap after 100000 lines, and keep col.ou[1-9])

all *,debug

ALL ddddd + all possible agent KBB_RAS1: (UNIT:K ALL)

Note: Any form of tracing also turns on KBB_RAS1 (UNIT:KDD ALL).

-h (Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

-s Starts monitoring only for the specified servers.

Note: This is not the same as the Safe Mode **-s** option available on some commands.

start | stop
Starts or stops the specified agent.

You can specify the product code of the agent you want to take action on. If you have installed agents for more than one kind of database, Oracle and Sybase for example, you can specify the product code for the database type whose agent you want action taken upon. You can specify multiple arguments separated by commas. The default is that **itmcmd dbagent** applies to all.

CLI example

The following example starts all database monitoring agents on the computer:

```
itmcmd dbagent start
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

"itmcmd agent" on page 279

"cinfo" on page 275 (to determine product codes)

Return to Table 9 on page 275.

itmcmd dbconfig

Description

Use the **itmcmd dbconfig** command to configures the execution environment for a distributed database agent.

CLI syntax

itmcmd dbconfig [-d] [-l|-L] [-V] [-h *install_dir***] [-s** *server* **| -i** *ID* **| -p** *password***] [***product_code***]**

where:

-d Enables debug tracing.

-l Indicates a new log.

-L Indicates no log.

-V Indicates no verify.

-h The name of the top-level directory in which you installed the monitoring agent.

-s The name of the server.

-i The user-defined ID for the server.

-p The user-defined password for the server.

product_code
The product code.

Related commands

Return to Table 9 on page 275.

itmcmd execute

Description

Use the **itmcmd execute** command to run a user script or command when its execution requires the same environment settings as for a particular IBM Tivoli product. The **itmcmd execute** command does this by building the necessary environment settings for the intended script or command and then combining them into a temporary shell script before running it.

The process is similar to how the **itmcmd agent** command processes an agent startup, but unlike **itmcmd agent**, the **itmcmd execute** command does not spawn a subshell to run the script before deleting the script. Instead, **itmcmd execute** "sources" the temporary shell script in `$install_dir/config/pc.sh` so that the environment settings become available to the current shell, from which the user command is then also run as the last instruction.

You must run **itmcmd execute** on the platform architecture for which the agent is installed. To use this command, make sure that you are in the correct directory:

```
cd $install_dir/bin
```

where *$install_dir* is the location where you installed your IBM Tivoli software.

CLI syntax

itmcmd execute
[-h $*install_dir***]**
[-k]
***pc* [*command*]**

where:

-h (Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on a *$install_dir* other than the *$install_dir* in the current system.

-k (Optional) Specifies that the temporary script created to run the user command is not to be deleted on completion. The name of the created script is displayed when the command is run.

pc **[***command***]**
Identifies the component (by product code) for which to run the command. The *command* argument is optional and specifies the fully qualified path to the script for which you want to build environment settings. To set the agent environment settings for the current console ksh shell, do not use the *command* argument.

CLI example

This example runs a script on the Tivoli Enterprise Portal Server to merge agent help into the Tivoli Enterprise Portal base help system.

```
itmcmd execute cq helpmerg.sh
```

Return values

An exit status of 0 indicates that the command executed successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

itmcmd history

Description

Use the **itmcmd history** command to manage the rolloff of history data into delimited text files.

CLI syntax

itmcmd history -h

itmcmd history -C
[-h [*install_dir*]]
[-L *nnn*[K|Kb|M|Mb]]
[-t *masks* ...]
[-D *delim*]
[-H|+H]
[-N *n*]
[-i *instance*|-p *cms_name*]
[-x]
pc

itmcmd history -A ?

itmcmd history -A [*n*|**0**|**c***cms_name*]
[-h [*install_dir*]]
[-W *days*]
[-L *nnn*[**K**|**Kb**|**M**|**Mb**]]
[-t masks*,etc]
[-D *delim*]
[-H|**+H]**
[-N *n*]
[-i *instance*|**-p** *cms_name*]
[-x]
pc

where:

-C Identifies this as an immediate one time conversion call.

-A Identifies this as a history conversion call. The default is to run the conversion now.

n Automatically run specified number of times per day. Value must be -A n, where n is 1-24, the number of runs per day, rounded up to the nearest divisor of 24. For example, -A 7 means run every three hours.

0 Cancels all automatic runs for tables specified.

c*ms_name*
Lists automatic collection status for all tables.

-W *days*
Day of the week (0=Sunday, 1=Monday, and so on.). This can be a comma-delimited list of numbers or ranges. For example, -W 1,3-5 means Monday, Wednesday, Thursday, and Friday. The default is Monday through Saturday (1-6).

-H|**+H**
Select the type of column headers desired.

-H Exclude column headers. Default is **attribute**.

+H Include group (long table) names in column headers. Format is **Group_desc.Attribute**. Default is **attribute** only.

-L Only converts files whose size is over a specified number of Kb/Mb (suffix can be any of none, K, Kb, M, Mb with none defaulting to Kb).

-h Override for the value of the *$candlehome* variable.

-t List of tables or mask patterns delimited by commas, colons, or blanks. If the pattern has embedded blanks, it must be surrounded with quotation marks.

-D Output delimiter to use. Default=tab character. Quote or escape blank: -D ' '

-N Keep generation 0-n of output (default 9).

-i *instance* | **-p** *cms_name*
Directs the program to process historical data collected by the specified agent instance or the specified Tivoli Enterprise Monitoring Server instead of the agent.

-i *instance*
For agent instances (those not using the default queue manager).

Directs the program to process historical data collected by the specified agent instance. For example, -i qm1 specifies the instance named **qm1**

-p *cms_name*
Directs the program to process historical data collected by the specified Tivoli Enterprise Monitoring Server instead of the agent.

Note: A product code of **ms** must be used with this option. The default action is to process data collected by prod_code agent.

-x Exclude SAMPLES and INTERVAL attributes in the output file.

pc Two-character product code of the agent from which historical data is to be converted. Refer to the appendix on product codes in the*IBM Tivoli Monitoring Installation and Setup Guide* for more information.

CLI example

Use **itmcmd history** to schedule automatic conversions by the UNIX *cron* facility. To schedule a basic automatic conversion, type the following at the command prompt:

```
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
```

Return values

An exit status of 0 indicates that the command executed successfully. An exit status of 2 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

itmcmd manage

Description

Use the **itmcmd manage** command to start Manage Tivoli Enterprise Monitoring Services on a UNIX or Linux computer. You can start, stop, and configure monitoring components in Manage Tivoli Enterprise Monitoring Services.

Note: In the Manage Tivoli Enterprise Monitoring Services tool, there is an option to Edit host specific configuration (right-click the monitoring agent and select Configure). This option should only be used at the direction of IBM support.

CLI syntax

itmcmd manage [-h *install_dir***]**

where:

-h (Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to use a take action command on an IBM Tivoli Monitoring installation directory other than the one in the current system.

CLI example

The following example starts Manage Tivoli Enterprise Monitoring Services:

```
itmcmd manage
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

itmcmd resp

Description

Use the **itmcmd resp** command to create a silent response file that can be used to either install or deploy similar agents across your environment. The automatic generation of response files does not apply to multi-instance agents or to server components. The agent must be successfully installed and configured before generating the response file.

CLI syntax

itmcmd resp [-d *directory***]** *pc*

where:

directory
Name of the directory where you want the generated files stored. The default is *itm_installdir*/response.

pc Product code for the agent whose configuration parameters you want saved.

CLI example

The following command creates the silent response file for the Monitoring Agent for Windows OS:

```
itmcmd resp itm_installdir/response nt
```

Related commands

Return to Table 9 on page 275.

itmcmd server

Description

Use the **itmcmd server** command to start and stop monitoring servers that are defined in directories under the *install_dir*/tables subdirectory. You must run the **itmcmd server** command from the host computer.

CLI syntax

itmcmd server
[-h *install_dir***]**
[-l]
[-n]
{start|stop} *tems_name*

where:

-h (Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

-l Deletes the log file associated with the monitoring server that is being stopped. By default, the log files is saved when the monitoring server stops.

-n When specified, indicates that the PID is not checked. If the PID of the initial agent is killed, then another process takes that PID.

start|stop
Starts or stops the specified monitoring server.

CLI example

The following command stops the hub_ms1 monitoring server:

```
itmcmd server stop hub_ms1
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

The monitoring server stop message might be displayed before the processes have completely stopped. It might take a minute for the processes to completely disappear, based on the system load.

Note: When the monitoring server stops normally, the log file for that session is saved. Use the **-l** option to delete the log files.

Related commands

"itmcmd agent" on page 279

Return to Table 9 on page 275.

itmcmd support

Description

Use the **itmcmd support** command to add (seed) application support to the monitoring server. Note that you now have the option during the Tivoli Enterprise Monitoring Server installation or upgrade process to seed support automatically. Beginning with agents based on Tivoli Monitoring V6.2.2, you are prompted to perform the seeding during the installation process. You only need to run the **itmcmd support** command if you elected not to perform the seeding during installation or upgrade.

When manually adding agent-specific information to the monitoring server, you need to run this command once during the initial installation of the monitoring server to add data for the components installed from the same installation CD. Whenever you add a new monitoring agent type to your monitoring environment, run the **itmcmd support** command again on the monitoring server to add the new agent information to the monitoring server.

Note:

1. Before you can run the **itmcmd support** command, you must start the monitoring server. See "itmcmd server" on page 294 for details.
2. Add application support only for agent components, not for other installed components such as the portal desktop client.
3. After you add the application support to the monitoring server, stop it and restart it.
4. If you are installing a backup monitoring server, see the *IBM Tivoli Monitoring Installation and Setup Guide* for information about adding application support.

CLI syntax

itmcmd support -t *tems_name*
[-h *install_dir***]**
[-m]
[-f {install | upgrade}]
[-s {NEW | ALL | NONE}]
pc ...

where:

-t Identifies the monitoring server. (Required)

pc ... One or more product codes for the components for which you want to add application support. To display the product codes for agents installed on this computer, run the **cinfo** command. See "cinfo" on page 275 for more information.

-h (Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

-m Copies application support files to a backup monitoring server without adding them. Use this option only when you are configuring a backup monitoring server.

-f Overrides automatic selection of support file. The install option ensures

that the file used during the pristine installation is used. The upgrade option ensures that the file used during the upgrade is used.

-s Identifies which agent support packages have the default distribution list added. The following values are supported:

- NEW - For new agent support packages, the default distribution list is added.
- ALL - For all agent support packages, the default distribution list is added. Not every situation has a default distribution list setting when installing application support for first time or reinstalling application support.
- NONE - The default distribution list is not added for any of the agent support packages.

CLI example

The following example adds application support to the hub_ms1 monitoring server for the agents installed from the IBM Tivoli Monitoring installation CD:

```
itmcmd support -t hub_ms1 a4 lz nt sy tm ul um ux
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

"itmcmd server" on page 294

"cinfo" on page 275 (to determine the product codes for agents)

Return to Table 9 on page 275.

SetPerm

Description

Use the **SetPerm** command to set file permissions to ensure that the permissions were set properly during the installation procedure. To run this command, you must be logged in to the UNIX computer as root.

When you run the **SetPerm** command, a product selection list is displayed. This list is sorted and contains the run architectures within each product description. From the list of installed products, enter a valid number or numbers separated by commas or spaces.

CLI syntax

SetPerm [-s]
[-h *install_dir*]

where:

-s (Optional) Used to set security validation on selected monitoring servers.

-h (Optional) Identifies the installation directory if it is not the one in which the script is located.

Also use this option to take action on an IBM Tivoli Monitoring installation directory other than the one in the current system.

CLI example

The following example starts the **SetPerm** utility:

```
SetPerm -s
```

Return values

An exit status of 0 indicates that the command ran successfully. An exit status greater than 0 indicates that there was a failure in the process.

Related commands

Return to Table 9 on page 275.

tmsdla

Description

Use the **tmsdla** command to scan your monitored environment to identify the managed systems and the resources that they are monitoring. You can then feed this information, as an XML output file, into the Change and Configuration Management Database (CCMDB), Tivoli Application Dependency Discovery Manager (TADDM), or Tivoli Business Service Manager (TBSM).

When the **tmsdla** command is launched, it gathers information by querying the hub Tivoli Enterprise Monitoring Server for all managed systems. If an agent provides a discovery library adapter template file, the **tmsdla** command uses the queries in the template file to map monitored resources to Common Data Model resources. See the agent-specific user guides to determine whether an agent supplies a template file and for mapping information between the agent's monitored resources and Common Data Model resources.

The monitoring servers and the Tivoli Enterprise Portal Server must be running before you can launch the **tmsdla** command. By default, any managed systems that are not online are not included in the output file.

The **tmsdla** command is located in the `installdir\CNPS` directory on Windows. On Linux or UNIX systems, use the **itmcmd execute** command to run the tmsdla.sh script. The output XML is generated in the `installdir\CNPS` directory on Windows and in the `installdir/<interp>/cq/bin/tmsdla` directory on Linux or UNIX systems. The TMS DLA also creates an output file with the `.xml.original` extension which contains the TMS DLA output before any relationships are removed. Removed relationships are written to the `tmsdla.log` file. For examples of scenarios where relationships might be removed from the output file, see "OS agent dependency" in the *IBM Tivoli Administrator's Guide*.

Note: On UNIX or Linux systems, write permission to `/tmp` is required to execute the **tmsdla** command.

CLI syntax

```
tmsdla [ [-? | -h] [-d directory] [-f outputFilename] [-l] [-m listOfManagedSystems]
         [-o] [-p portNumber] [-r] [-s] [-t threadCount] [-w waitTime] ]
```

where:

-? | -h Displays the syntax help information.

-d Specify the template directory location.

-f Specify the resulting output file name.

-l Discovers logical views.

-m Specify the list of managed systems.

The list is double quote delimited and follows this syntax:

```
"os_msys1, os_apptype1, [msys1, apptype1] ~ [os_msys2, os_apptype2,
[msys2, apptype2]] ~ .. ~ [os_msysN, os_apptypeN, [msysN, apptypeN]]"
```

-o Force processing of offline managed systems.

-p Specify the portal server's port number if not the default value of 1920. The port number is included in the output book and used by TADDM or TBSM to generate the URL to launch to the Tivoli Enterprise Portal.

-r Generate a refresh-type output XML file. After you import a refresh-type output file into TADDM, the objects for any managed systems that are offline (for example, for maintenance operations) and their monitored resources are removed from the TADDM database. Similar results occur when you import a refresh-type output file into TBSM or CCMDB. If you do not specify this option, a create-type output XML file is generated that contains only online managed systems and the resources that they are monitoring. When you import a create-type output XML file into TADDM, TBSM, or CCMDB, managed systems and monitored resources are added or updated but not deleted.

-s Suppress generation of the `.original` file from cleanup process.

-t Specify the number of threads to use.

-w Specify the number of seconds to wait for query to be serviced by agent before timing out. Use this option if monitoring agents might not be able to service queries in a reasonable time due to a heavy load on the queried system. **Default value**: 120 seconds.

Minimum value: 50 seconds. Values lower than 50 are ignored and the default value is used.

Maximum value: 600 seconds.

Note: No return codes are provided on the completion of the book to alert you if there was a timeout or missing agent data. To determine if you need to set a higher value than the default, analyze the book to ensure that all agents have responded.

CLI example

The following example starts the **tmsdla** utility on Linux or UNIX systems and generates a create-type output file:

```
installdir/bin/itmcmd execute cq "tmsdla.sh"
```

The following example starts the **tmsdla** utility on Windows systems and generates a create-type output file:

```
installdir\CNPS\tmsdla.bat
```

Related commands

Return to Table 9 on page 275.

Chapter 4. tivcmd commands

You can run the tivcmd commands to create and work with authorization policies that control access to resources displayed in monitoring dashboards in the IBM Dashboard Application Services Hub. The tivcmd commands send requests to the Tivoli Authorization Policy Server application which maintains the master set of authorization policies. For more information on authorization policies and assigning policy administrators, see "Using role-based authorization policies" in the *ITM Administrator's Guide*.

To use the tivcmd commands, you must install the tivcmd Command Line Interface for Authorization Policy component on the system where you intend to run the commands. The Tivoli Authorization Policy Server component must be installed into the IBM Dashboard Application Services Hub. For more details, see the *ITM Installation and Setup Guide*.

The tivcmd command is installed into the `<install_dir>/bin` directory where <install_dir> is the directory where you installed the tivcmd Command Line Interface for Authorization Policy component. Within a Windows environment, the command name is tivcmd.exe. Within a UNIX or Linux environment, the command name is tivcmd.

Before you can use the tivcmd commands to create and work with authorization policies, a user with role administration permissions must use the **tivcmd addtorole** command to assign you to a role that has permissions to perform operations on roles. The first authorization policy administrator is specified when the Tivoli Authorization Policy Server is installed. That user is assigned to the predefined RoleAdministrator role and has permission to use all of the tivcmd commands. That user can assign other users permission to create and work with authorization policies by adding the users or their user group to the RoleAdminstrator role or to a custom role with similar permissions.

For examples of using the tivcmd commands to create roles, grant permissions to roles, and assign users to roles, see “Managing roles and permissions” on page 304. For the permissions that are required to run each tivcmd command, see “tivcmd grant” on page 311.

Table 11. tivcmd commands

Command	Description
“tivcmd addtorole” on page 306	Assign users or groups to a role
“tivcmd copyrole” on page 307	Copy an existing role to create a role
“tivcmd createrole” on page 308	Create a role
“tivcmd deleterole” on page 309	Delete a role
“tivcmd exclude” on page 310	Prevent a role from accessing a resource
“tivcmd grant” on page 311	Assign permission to a role
“tivcmd help” on page 315	Display the name and short description of all the available CLI commands or display the complete help for a specified command

Table 11. tivcmd commands (continued)

Command	Description
"tivcmd listdomains" on page 316	List all the IBM Tivoli Monitoring domains that are known to the Authorization Policy Server
"tivcmd listobjecttypes" on page 317	List all valid object types and corresponding valid operations
"tivcmd listresourcetypes" on page 318	List all valid resource types
"tivcmd listroles" on page 318	List all the role names, as well as their corresponding descriptions and permissions
"tivcmd login" on page 321	Authenticate with the IBM Dashboard Application Services Hub where the Authorization Policy Server is installed so that you can run subsequent tivcmd commands from the local system
"tivcmd logout" on page 322	Disable the security token that is created by the **tivcmd login** command
"tivcmd removefromrole" on page 323	Remove users or groups from a role
"tivcmd revoke" on page 324	Remove the permission previously granted to a role using the **tivcmd grant** command or to remove the permission previously excluded from a role using the **tivcmd exclude** command

Command format

All Authorization Policy Server CLI commands have a common format. The format is:

```
tivcmd command [-arg1 argvalue1 -arg2 argvalue2 ...]
```

where:

command
Specifies the command to issue (for example, createrole or listroles).

-arg*n* Specifies an argument identifier.

-argvalue*n*
Specifies the argument value for the associated argument identifier.

Note: Some commands support arguments that contain spaces or special characters.

You can either provide the command arguments on the command line, through standard input, or from an input file. The tivcmd help command description shows how to provide arguments using standard input. For information on providing arguments in an input file, see "Input files for tivcmd commands" on page 303.

On Windows systems:

- If an argument value contains a space, the entire argument value must be enclosed in double quotation marks (for example, **-n** "My role name").
- If an argument value contains special characters, such as & | < > ^, which can be used as general command line operating system directives, then the argument

value must be surrounded by double quotation marks to successfully run the command (for example, **-n** "My role name").

On UNIX systems:

- If an argument value contains a space, the entire argument value must be enclosed in double quotation marks or quotation marks (for example, -dd "My domain description" or -dd 'My domain description').
- If an argument value contains special characters to the bash shell (for example, $! '#*), perform one of the following actions:
  - Precede the special character with an escape character (\ ) (for example, -n rolename\!).
  - Enclose the argument value in quotation marks (for example, -n 'My domain!'). Note that shell expansions are performed within double quotation marks, but not within quotation marks.

Input files for tivcmd commands

Description

The tivcmd commands allow you to specify all of the command-line options within an input file.

Note: The following syntax is available for all tivcmd commands.

CLI syntax

tivcmd subcommand inputfile

where:

subcommand
Specifies the command name, such as grant or exclude.

input file
Specifies a fully qualified path to the text file that contains the applicable command-line switches for the command. Each command-line switch or command-line argument must be listed on a separate line. Encode the file in the native code page.

CLI example

This example of the command-line input for the grant command:

```
tivcmd grant -n myUNIXOperator -y attributegroup -t
managedsystem -r machine1:KUX machine2:KUX machine3:KUX machine4:KUX -p view
```

can be entered by using the following input file:

```
tivcmd grant c:\temp\grant_cli.txt
```

where grant_cli.txt contains the following lines of code:

```
-n
myUNIXOperator
-y
attributegroup
-t
managedsystem
-r
machine1:KUX
```

```
machine2:KUX
machine3:KUX
machine4:KUX
-p
view
```

Managing roles and permissions

Description

When managing roles, consider the role type: core or custom. The Authorization Policy Server offers a set of predefined roles, called core roles. Core roles cannot be modified or changed. Core roles include:

RoleAdministrator
The main security admin role with the authority to manage all roles and policies.

PolicyDistributor
The role with permission to download policies from the Authorization Policy Server. When authorization policy enforcement is enabled in Tivoli Enterprise Portal Server, the portal server administrator configures connection parameters for the Authorization Policy Server. These parameters include a user ID that must be assigned to this core role or to a custom role that has the same permission.

LinuxOperator
A role that has attribute group and event viewing permission for all Linux agents.

UNIXOperator
A role that has attribute group and event viewing permission for all UNIX agents.

VCenterOperator
A role that has access to all VMWARE Virtual Centers and ESX Servers.

WindowsOperator
A role that has attribute group and event viewing permission for all Windows agents.

You might determine that the core roles are sufficient to meet your needs. If not, a user who is assigned to the RoleAdministrator role can run the **tivcmd createrole** command to define additional roles to better associate users with their IBM Tivoli Monitoring access requirements. These additional roles are referred to as custom roles.

The following examples describe how to use the tivcmd commands to manage roles and their permissions.

CLI example

Use this example to assign the core RoleAdminstrator role to user Chris so that Chris can use all tivcmd commands:

1. Log in to the Authorization Policy Server as a user who is already assigned to the RoleAdministrator role, such as tipadmin.
   ```
   tivcmd login -s localhost -u tipadmin -p <tipadmin_password>
   ```
2. Assign RoleAdministrator to user Chris.
   ```
   tivcmd addtorole -n RoleAdministrator -u cn=Chris,ou=users,ou=SWG,o=IBM,c=US
   ```

Use this example to create a role that is called myUNIXOperator with permission to view all situation events and monitoring data for all managed systems in the WesternRegionUNIXSystems managed system group:

1. Log in to the Authorization Policy Server as a user who is assigned to the RoleAdministrator role or to a custom role that has permissions to perform operations on roles.

   ```
   tivcmd login -s localhost -u tipadmin -p <tipadmin_password>
   ```

2. Create the role called myUNIXOperator.

   ```
   tivcmd createrole -n myUNIXOperator -d "Role to view any events and
   attribute groups for the UNIX OS agents in the western region"
   ```

3. Add permissions to the role myUNIXOperator.

   ```
   tivcmd grant -n myUNIXOperator -y attributegroup -t managedsystemgroup
   -r "WesternRegionUNIXSystems" -p view
   tivcmd grant -n myUNIXOperator -y event -t managedsystemgroup
   -r "WesternRegionUNIXSystems" -p view
   ```

Use this example to create a role that is called myRoleAdmin with permission to create and work with roles if you do not want to assign users to the RoleAdministrator core role:

1. Log in to the Authorization Policy Server as a user who is assigned to the RoleAdministrator role or to a custom role that has permissions to perform operations on roles.

   ```
   tivcmd login -s localhost -u tipadmin -p <tipadmin_password>
   ```

2. Create the role called myRoleAdmin.

   ```
   tivcmd createrole -n myRoleAdmin -d "Role to manage permissions"
   ```

3. Add permissions to the role myRoleAdmin.

   ```
   tivcmd grant -n myRoleAdmin -y role -t rolegroup -r default -p
   create delete modify view viewAll
   ```

Use this example to assign myRoleAdmin role to user Chris:

1. Log in to the Authorization Policy Server with the RoleAdministrator core role assigned to user tipadmin.

   ```
   tivcmd login -s localhost -u tipadmin -p <tipadmin_password>
   ```

2. Assign myRoleAdmin to user Chris.

   ```
   tivcmd addtorole -n myRoleAdmin -u cn=Chris,ou=users,ou=SWG,o=IBM,c=US
   ```

Use this example for user Chris to create a role that is called myLinuxOperator with permission to view any event and attribute group for the EasternRegionLinuxSystems:

1. Log in to the Authorization Policy Server by using user Chris.

   ```
   tivcmd login -s localhost -u Chris -p <Chris_password>
   ```

2. Create the role called myLinuxOperator.

   ```
   tivcmd createrole -n myLinuxOperator -d "Role to view any events and
   attribute groups for the EasternRegionLinuxSystems managed system group"
   ```

3. Add permissions to the role myLinuxOperator.

   ```
   tivcmd grant -n myLinuxOperator -y attributegroup -t managedsystemgroup -r
   "EasternRegionLinuxSystems" -p view
   tivcmd grant -n myLinuxOperator -y event -t managedsystemgroup -r
   "EasternRegionLinuxSystems" -p view
   ```

4. Assign the myLinuxOperator role to the EasternLinuxOperators user group.

   ```
   tivcmd addtorole --rolename myLinuxOperator
   --groups gid=EasternLinuxOperators,cn=itm,o=ibm
   ```

Related commands

Return to Table 11 on page 301.

Determining the unique name of users and user groups

When you use the **tivcmd addtorole** command to assign a user or user group to a role, you must specify the unique name (also called the distinguished name) of the user or user group in the LDAP user registry.
`cn=user1,ou=users,ou=SWG,o=IBM,c=US` is an example of a user name.
`cn=group1,ou=groups,ou=SWG,o=IBM,c=US` is an example of a group name. If you have been assigned the iscadmins and administrator roles for IBM Dashboard Application Services Hub, you can use the WebSphere Administrator Console to see unique names for users and groups in your environment by performing the following steps:

1. Open the WebSphere Administrative Console. By default the URL is `https://<hostname>:16316/ibm/console` where <hostname> is the hostname of the computer where the IBM Dashboard Application Services Hub and the Tivoli Authorization Policy Server are installed.
2. Select **Users and Groups**.
3. Under **Manage Users or Managed Groups**, the known users are listed.
4. When assigning users or user groups to roles, use the value listed under **Unique Name**.

tivcmd addtorole

Description

Use the **tivcmd addtorole** command to assign users or groups to a role. The user name and group name are the unique names that are used in the LDAP user repository. `cn=user1,ou=users,ou=SWG,o=IBM,c=US` is an example of a user name. `cn=group1,ou=groups,ou=SWG,o=IBM,c=US` is an example of a group name. For additional information, see "Determining the unique name of users and user groups."

You must log in by using the **login** command before you run the **addtorole** command.

CLI syntax

tivcmd addtorole
{-n|--rolename} *ROLENAME*
{-u|--users} *USERS*

tivcmd addtorole
{ -n|--rolename} *ROLENAME*
{-g|--groups} *GROUPS*

tivcmd addtorole
{ -n|--rolename} *ROLENAME*
{-u|--users} *USERS*
{-g|--groups} *GROUPS*

where:

-n | --rolename
Specifies an existing role name to which to assign users or groups.

-u | --users
Specifies one or more users to be assigned the role. When you specify more than one user, separate the names by blanks.

-g | --groups
Specifies one or more groups to be assigned the role. When you specify more than one group, separate the names by blanks.

CLI example

This example grants user bob and user mary to the "West Coast WAS administrator" role:

```
tivcmd addtorole -n "West Coast WAS administrator"
-u cn=bob,ou=users,ou=SWG,o=IBM,c=US cn=mary,
ou=users,ou=SWG,o=IBM,c=US
```

This example assigns WASGroup1 and "West Coast WAS" groups to the "West Coast WAS administrator" role:

```
tivcmd addtorole -n "West Coast WAS administrator"
-g cn=WASGroup1,ou=groups,ou=SWG,o=IBM,c=US "cn=West Coast WAS,
ou=groups,ou=SWG,o=IBM,c=US"

KDQATR002I: Groups [cn=WASGroup1,ou=groups,ou=SWG,o=IBM,c=US
"cn=West Coast WAS,ou=groups,ou=SWG,o=IBM,c=US"] have been added
to role [West Coast WAS administrator].
```

Related commands

Return to Table 11 on page 301.

tivcmd copyrole

Description

Use the **tivcmd copyrole** command to copy an existing role to create a new role. Copying a core role results in a blank description for the new role unless overridden with the **-d** option. You can also run this command to copy the permissions only, without copying the user and group membership of the existing role.

You must log in by using the **login** command before you run the **copyrole** command.

CLI syntax

tivcmd copyrole
{-f | --fromrolename} *FROMROLENAME*
{-t | --torolename} *TOROLENAME*
[{-d | --description} *DESCRIPTION***]**
[{-p | --permissionsonly}]

where:

-f | --fromrolename (required)
Specifies the name of the role to be copied. The role name is case-sensitive and must exist.

-t | --torolename (required)
Specifies the name of the role to be created with the copied data. The role name is case-sensitive and must not exist.

-d | --description
Specifies the description for the new role name. If this option is not provided, the description of the original role name is used. However, if the *from* role names is one of the core role names, the new role will have a blank description unless this option is specified.

-p | --permissiononly
Indicates that only permissions are copied to the new role. User and group membership are not copied. Use this option when you move a role from one environment to another that does not already include the same set of users and groups.

CLI example

This example renames the MyAdministrator role to the MyUSAdministrator role:

```
tivcmd copyrole -f MyAdministrator -t MyUSAdministrator
tivcmd deleterole -n MyAdministrator
```

This example creates a role that has the same permission as the MyLinuxAdministrator role:

```
tivcmd copyrole -f MyLinuxAdministrator -t MyUSLinuxAdministrator -p
```

Related commands

Return to Table 11 on page 301.

tivcmd createrole

Description

Use the **tivcmd createrole** command to create a role.

You must log in by using the **login** command before you run the **createrole** command.

CLI syntax

tivcmd createrole
{-n | --rolename} *ROLENAME*
[{-d | --description}] *DESCRIPTION* **]**

where:

-n | --rolename
Specifies the role name that you want to create.

-d | --description
Specifies the role description.

CLI example

This example creates a West Coast WAS administrator role to manage all the West Coast WAS machines:

```
tivcmd createrole -n "West Coast WAS administrator"
-d "Administrator Role to manage the West Coast WAS systems"

KDQACR001I Role [West Coast WAS administrator] has been created successfully.
```

Related commands

Return to Table 11 on page 301.

tivcmd deleterole

Description

Use the **tivcmd deleterole** command to delete a role. User and group membership that is associated with the role are also removed.Before deleting a role, determine which users and groups will be affected. To display users and groups that are associated with this role, run the **tivcmd listroles** command with the **-n** and **-m** options. See the **tivcmd listroles** command description for more information.

You must log in using the **login** command before you run the **tivcmd deleterole** command.

CLI syntax

tivcmd deleterole
{-n|--rolename} *ROLENAME*
[{-f|--force }]

where:

-n|--rolename (required)
Specifies the role name that you want to delete.

-f|--force
Runs the deleterole operation without user confirmation.

CLI example

The following example deletes the role "West Coast WAS Administrator":

```
tivcmd deleterole -n "West Coast WAS Administrator"
KDQADR002I Are you sure you want to delete this role?
User and group membership to this role will also be removed.
Enter Y for yes or N for no:
Y

KDQADR001I Role [West Coast WAS administrator] has been deleted successfully.
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 11 on page 301.

tivcmd exclude

Description

Use the **tivcmd exclude** command to prevent a role from accessing a resource. You can prevent a role from accessing a particular managed system even if the role has access to the managed system group that contains the managed system.

You must log in by using the **login** command before you run the **exclude** command.

Note: The **tivcmd exclude** command does not issue an error message if you attempt to run the same command again.

When you exclude permission for a managed system, the permissions listed in the table below are automatically created in the policy store maintained by the Authorization Policy Server. These permissions prevent a user from performing any operation on the managed system.

Table 12. Permissions automatically created by the exclude command

Object Type	Resource types	Resources	Authorized operations and descriptions
attribute group	managedsystem	One or more managed system names	view: Exclude permission to view monitoring data from the specified managed systems.
event	managedsystem	One or more managed system names	view: Exclude permission to view the list of situation events for the specified managed systems. **Note:** The attributegroup object type must be used to exclude permission to view situation event details.

CLI syntax

tivcmd exclude
{-n | --rolename} *ROLENAME*
{-t | --resourcetype} *RESOURCETYPE*
{-r | --resources} *RESOURCES*
[{-d | --domain} *DOMAIN***]**

tivcmd exclude
{-n | --rolename} *ROLENAME*
{-t | --resourcetype} *RESOURCETYPE*
{-i | --inputfile} *INPUTFILE*
[{-d | --domain} *DOMAIN***]**

where:

-n | --rolename
Specifies the role name which should be excluded access to one or more resources. The role name is case-sensitive and must exist.

-t | --resourcetype
Specifies the type of resources for which the role does not have access. The resource type name is not case-sensitive. Only the managedsystem resource type is supported for the **tivcmd exclude** command.

-r|--resources
Specifies the managed systems that the role is not allowed to access. The resource name is case-sensitive. One or more resource names can be specified as a list of values that are separated by blanks. This option cannot be specified with the **-i|--inputfile** option.

-i|--inputfile
Specifies the fully qualified input file name that contains one or more managed systems that the role is not allowed to access. Specify one managed system name per line. Encode the file in the native code page. This option cannot be specified with the **-r|--resources** option.

-d|--domain
Specifies the domain name that is excluded access. In a multi-hub monitoring server environment, use this option to exclude access to resources in a specific IBM Tivoli Monitoring domain. If this option is not specified, the permission applies to all domains and a value of *any* is saved as the domain name. By default, the domain name is `itm.<Hub TEMS name>` unless the domain name was overridden when you configured the Tivoli Enterprise Portal Server for the domain.

CLI example

Currently the "Unix Administrator" role is allowed to view attribute groups for the *UNIX_ALL managed system group. However, the role should not be allowed to view attribute groups for the important:KUX managed system, which is a member of the *UNIX_ALL group. This example excludes access to the important:KUX managed system:

```
tivcmd exclude -n "Unix Administrator" -t managedsystem -r
"important:KUX"
```

Related commands

Return to Table 11 on page 301.

tivcmd grant

Description

Use the **tivcmd grant** command to assign permission to a role.

You must log in by using the **login** command before you run the **grant** command.

Note: The **tivcmd grant** command does not issue an error message if you attempt to run the same command again. The permissions that have already been granted remain the same.

The following table shows the valid values for the object types, resource types, and resources that are used in the grant command.

Table 13. Valid values for the object types, resource types, and resources

Object Type	Resource types	Resources	Authorized operations and descriptions
attribute group	managedsystemgroup	One or more managed system group names	view: Grant permission to view monitoring data from all managed systems that are members of the specified managed system groups.
	managedsystem	One or more managed system names	view: Grant permission to view monitoring data from the specified managed systems.
event	managedsystemgroup	One or more managed system group names	view: Grant permission to view the list of situation events for all managed systems that are members of the specified managed system groups. **Note:** The attributegroup object type must be used to grant permission to view situation event details.
	managedsystem	One or more managed system names	view: Grant permission to view the list of situation events for the specified managed systems. **Note:** The attributegroup object type must be used to grant permission to view situation event details.

Table 13. Valid values for the object types, resource types, and resources (continued)

Object Type	Resource types	Resources	Authorized operations and descriptions
role	rolegroup	default	create: Grant permission to create roles using the tivcmd createrole and copyrole commands.
			delete: Grant permission to delete roles using the tivcmd deleterole command.
			modify: Grant permission to modify roles using the tivcmd grant, exclude, revoke, addtorole, and removefromrole commands.
			view: Allow a user who has logged in with the tivcmd login command to run the tivcmd listroles command to view their own roles and permissions. **Note:** If you want users who are not authorization policy administrators to be able to view their roles and permissions but not roles and permissions for other users, create a new role and grant the role this permission. Then assign non-administrative users or user groups to the new role.
			viewAll: Grant permission to view all roles and permissions using the tivcmd listroles command.
			distribute: Grant permission to download the authorization policies from the Authorization Policy Server. **Note:** The user that is specified when configuring the Authorization Policy Server connection information for the Tivoli Enterprise Portal Server must be assigned a role with this permission. The permission allows the portal server to download the authorization policies so that they can be enforced in the dashboard data provider. The PolicyDistributor core role has this permission. You can also create a custom role with this permission.

CLI syntax

tivcmd grant

{-n|--rolename} ***ROLENAME***
{-t|--resourcetype} ***RESOURCETYPE***
{-r|--resources} ***RESOURCES***
{-y|--objecttype} ***OBJECTTYPE***
{-p|--operations} ***OPERATIONS***
[{-d|--domain} ***DOMAIN*****]**

tivcmd grant

{-n|--rolename} ***ROLENAME***
{-t|--resourcetype} ***RESOURCETYPE***
{-i|--inputfile} ***INPUTFILE***
{-y|--objecttype} ***OBJECTTYPE***
{-p|--operations} ***OPERATIONS***
[{-d|--domain} ***DOMAIN*****]**

where:

-n|--rolename
Specifies the role name that is granted access. The role name is case-sensitive and must exist.

-t|--resourcetype
Specifies the type of resources that is granted access. The resource type name is not case-sensitive. The **tivcmd listresourcetypes** command can be used to get a list of the valid resource types.

-r|--resources
Specifies the resources of the specified type that are granted access. The resource name is case-sensitive. One or more resources can be specified as a list of values that are separated by blanks. This option cannot be specified with the **-i|--inputfile** option.

-i|--inputfile
Specifies the fully qualified input file name that contains one or more resources (one resource name per line). Encode the file in the native code page. This option cannot be specified with the **-r|--resources** option.

-y|--objecttype
Specifies the type of object that can be accessed for the specified resources. The object type name is not case-sensitive. Use the **tivcmd listobjecttypes** command to display a list of the valid object types.

-p|--operations
Specifies the operations that are allowed for the specified object type. The operation name is not case-sensitive. One or more operations can be specified as a list of values that are separated by blanks. Use the **tivcmd listobjecttypes -s** command to display a list of the valid operations for each valid object type.

-d|--domain
Specifies the domain name that is granted access. In a multi-hub monitoring server environment, use this option to authorize access to resources in a specific IBM Tivoli Monitoring domain. If this option is not specified, the permission applies to all domains and a value of *any* is saved as the domain name. By default, the domain name is `itm.<Hub TEMS name>` unless the domain name was overridden when you configure the Tivoli Enterprise Portal Server for the domain. Use the **tivcmd listdomains** command to list the domains that were specified when you work with policies or that have a connection defined in IBM Dashboard Application Service Hub to a domain's dashboard data provider.

CLI example

This example grants the "UNIX Administrator" role the ability to view all attribute groups for the managed system group name that is called *ALL_UNIX:

```
tivcmd grant -n "UNIX Administrator" -t managedsystemgroup -r "*ALL_UNIX"
-y attributegroup -p view
```

This example grants the "West Coast WAS Administrator" role the ability to view all attribute groups for the managed system group name called "West_Coast_WAS_Systems":

```
tivcmd grant -n "West Coast WAS Administrator" -t managedsystemgroup
-r "West_Coast_WAS_Systems"  -y attributegroup -p view
```

Related commands

Return to Table 11 on page 301.

tivcmd help

Description

Use the **tivcmd help** command to display the name and short description of all the available CLI commands or to display the complete help for a specified command. It also shows how to process command options from standard input or from an input file.

CLI syntax

tivcmd help | ? {*command*}

where:

{*command*}

Specifies the command for which you want detailed help. The following lists and describes the available commands:

tivcmd help | ? {command}
Displays complete help for a specified command.

tivcmd {command} -stdin|--stdin
Indicates that all command-line parameters are processed from standard input instead of being parsed from the command-line arguments.

tivcmd {command} {filename}
Indicates that all command-line parameters are processed from the specified file instead of being parsed from the command-line arguments. Each parameter and parameter value is on a separate line in the file.

where:

{command}
Specifies the command to run or display help for.

{filename}
Specifies the file that contains the command arguments.

The available tivcmd commands:

tivcmd addtorole
To add users or groups to a role.

tivcmd copyrole
To copy an existing role to create a role.

tivcmd createrole
To create a role.

tivcmd deleterole
To delete a role.

tivcmd exclude
To prevent a role from accessing a resource.

tivcmd grant
To assign a permission to a role.

tivcmd listdomains
To list all the IBM Tivoli Monitoring domains that are known to the Authorization Policy Server.

tivcmd listobjecttypes
To list all valid object types and corresponding valid operations.

tivcmd listresourcetypes
To list all valid resource types.

tivcmd listroles
To list all the role names, as well as their corresponding descriptions and permissions.

tivcmd login
To authenticate a user name and password with a server, so that a user can run subsequent commands from the local machine.

tivcmd logout
To disable the security token that is created by the **tivcmd login** command.

tivcmd removefromrole
To remove users or groups from a role.

tivcmd revoke
To remove the permission that was granted to a role with the **tivcmd grant** command or to remove the permission that was excluded from a role with the **tivcmd exclude** command.

CLI example

This command displays the name and short description of all the available CLI commands.

```
tivcmd help

or

tivcmd ?
```

This command displays the detailed help for the **listroles** command.

```
tivcmd help listroles

or

tivcmd ? listroles
```

Related commands

Return to Table 11 on page 301.

tivcmd listdomains

Description

Use the **tivcmd listdomains** command to list all the IBM Tivoli Monitoring domains that are known to the Authorization Policy Server. The list includes any domains for which there is an IBM Tivoli Monitoring dashboard data provider connection that is configured in the IBM Dashboard Application Services Hub

where the Authorization Policy Server is deployed. This list also includes any domain names that are defined for any existing grant and exclude permissions. If a connection is defined for the domain, the **tivcmd listdomains** command also displays the connection ID and name.

You must log in by using the **login** command before you run the **listdomains** command.

CLI syntax

```
tivcmd listdomains
```

CLI example

This example lists all of the IBM Tivoli Monitoring domains that are known to the Authorization Policy Server:

```
tivcmd listdomains
```

Related commands

Return to Table 11 on page 301.

tivcmd listobjecttypes

Description

Use the **tivcmd listobjecttypes** command to list all valid object types and corresponding valid operations.

You must log in by using the **login** command before you run the **listobjecttypes** command.

CLI syntax

```
tivcmd listobjecttypes [{-s|--showoperations}]
```

where:

-s|--showoperations
Specifies that the valid operations for each object type are to be listed.

CLI example

This example lists all the object types that are supported by the Authorization Policy Server:

```
tivcmd listobjecttypes

attributegroup
 Description: A monitoring agent table that includes one or more attributes.
event
 Description: An occurrence of a condition that can be detected by a monitoring
 situation.
role
 Description: A set of permissions for users and groups who perform similar
business or functional tasks.
```

This example lists all the object types and corresponding operations:

```
tivcmd listobjecttypes -s
attribute group
 Description:
A monitoring agent table that includes one or more attributes.
 Operations: {view}
event
 Description:
An occurrence of a condition that can be detected by a monitoring situation.
Operations:{view}
role
 Description: A set of permissions for users and groups who perform similar
business or functional tasks.
 Operations: {create, delete, distribute, modify, view, viewall}
```

Related commands

Return to Table 11 on page 301.

tivcmd listresourcetypes

Description

Use the **tivcmd listresourcetypes** command to list all valid resource types.

You must log in by using the **login** command before you run the **listresourcetypes** command.

CLI syntax

tivcmd listresourcetypes

CLI example

This example lists the supported resource types for Authorization Policy Server:

```
tivcmd listresourcetypes

managedsystem
  Description: An operating system, subsystem or application agent instance.
managedsystemgroup
  Description: A collection of managed systems.
rolegroup
  Description: A collection of roles and permissions associated with
an authorization container.
```

Related commands

Return to Table 11 on page 301.

tivcmd listroles

Description

Use the **tivcmd listroles** command to list all the role names, as well as their corresponding descriptions and permissions. You can filter the results by current login user, user name, group name, role name or resource type and name. The user name and group name are the unique names that are used in the user group repository. cn=user1,ou=users,ou=SWG,o=IBM,c=US is an example of a user name. cn=group1,ou=groups,ou=SWG,o=IBM,c=US is an example of a group name.

You must log in by using the **login** command before you run the **listroles** command.

CLI syntax

tivcmd listroles
[{-s|--showdescription}]

tivcmd listroles
{-l|--loginuser}
[{-s|--showdescription}]
[{-p|--showpermissions}]

tivcmd listroles
{-u|--username} *USERNAME*
[{-s|--showdescription}]
[{-p|--showpermissions}]

tivcmd listroles
{-g|--groupname} *GROUPNAME*
[{-s|--showdescription}]
[{-p|--showpermissions}]

tivcmd listroles
{-n|--rolename} *ROLENAME*
[{-m|--showmembership}]
[{-s|--showdescription}]
[{-p|--showpermissions}]

tivcmd listroles
{-t|--resourcetype} *RESOURCETYPE*
{-r|--resource} *RESOURCE*
[{-m|--showmembership}]
[{-s|--showdescription}]
[{-p|--showpermissions}]

where:

-g|--groupname
Lists all the roles that are assigned to the specific group. This option is mutually exclusive from the **-u|--username** or **-n|--rolename** or **-t|--resourcetype** or **-r|--resource** options.

-l|--loginuser
List roles that are assigned to the user specified with the **tivcmd login** command.

-m|--showmembership
Specifies that the users and groups that are assigned to the specified role are to be listed. This option is only valid with the **-n|--rolename** and **-t|--resourcetype** and **-r|--resource** option.

-n|--rolename
Specifies the role name to be listed. This option cannot be specified with the **-u|--username** or **-g|--groupname** or **-t|--resourcetype** or **-r|--resource** options.

-p|--showpermissions
Display the permissions that are assigned to the role. If this option is specified with the **-t|resourcetype** and **-r|resources** options, only permissions for the specified resourcetype and resources are displayed.

-u|--username
List all the roles that are assigned to the specified user. This option cannot be specified with the **-g|--groupname** or **-n|--rolename** or **-t|--resourcetype** or **-r|--resource** options.

-s|--showdescription
Specifies that the description for each role is to be listed.

-t|--resourcetype
List the roles that are assigned a permission for the specified resource type and the resource name that is provided with the **-r|--resource** option. The resource type name is not case-sensitive. Use the **tivcmd listresourcetypes** command to list the valid resource types.

-r|--resource
List the roles that are assigned a permission for the specified resource name and the resource type that is provided with the **-t|--resourcetype** option. The resource name is case-sensitive.

CLI example

This example lists all the roles that are defined:

```
tivcmd listroles

LinuxOperator
PolicyDistributor
RoleAdministrator
UNIXOperator
VCenterOperator
WindowsOperator
```

This example lists all the roles and their corresponding descriptions:

```
tivcmd listroles -s

LinuxOperator
  Description: A Core role that provides attributegroup and event viewing
for all Linux systems.
  Type: Core
PolicyDistributor
  Description: A Core role that provides distribution permissions.
  Type: Core
RoleAdministrator
  Description: A Core role that provides all permissions.
  Type: Core
UNIXOperator
  Description: A Core role that provides attributegroup and event viewing
for all UNIX systems.
  Type: Core
VCenterOperator
  Description: A Core role that provides attributegroup and event viewing
for all VCenter and any ESX servers hosted by
the VCenter.
  Type: Core
WindowsOperator
  Description: A Core role that provides attributegroup and event viewing
for all Windows systems.
  Type: Core
```

This example lists all the users and groups that are assigned the role "West Coast WAS administrator":

```
tivcmd listroles -n "West Coast WAS administrator" -m -s

West Coast WAS administrator
   Description:Role to manage the West Coast WAS systems
   Users:  "cn=bob,ou=users,ou=SWG,o=IBM,c=US"
"cn=mary,ou=users,ou=SWG,o=IBM,c=US"
   Groups: "cn=WASGroup1,ou=groups,ou=SWG,o=IBM,c=US"
"cn=West Coast WAS,ou=groups,ou=SWG,o=IBM,c=US"
```

This example lists which roles have access to the resource "*ALL_UNIX" of resource type managesystemgroup:

```
tivcmd listroles -t managesystemgroup -r "*ALL_UNIX"

  UNIXOperator
  UNIX Administrators
```

This example lists which users and groups have access to the resource "*ALL_UNIX" of resource type managesystemgroup and what permissions they have on the resource:

```
tivcmd listroles -t managedsystemgroup -r "*ALL_UNIX" -m -p
UNIXOperator
   Users:  "cn=user1,ou=users,ou=SWG,o=IBM,c=US"
cn=user2,ou=users,ou=SWG,o=IBM,c=US"
   Groups: "cn=UnixOperatorGroup,ou=groups,ou=SWG,o=IBM,c=US"
   Permissions:
    Domain: any
    Object Type: event
    Granted Operations: {view}

    Domain: any
    Object Type: attributegroup
    Granted Operations: {view}

UNIX Administrator
   Users: "cn=admin1,ou=users,ou=SWG,o=IBM,c=US cn=admin2,
ou=users,ou=SWG,o=IBM,c=US"
   Groups:  "cn=UnixAdministratorGroup,ou=groups,ou=SWG,o=IBM,c=US"
   Permissions:
    Domain: any
    Object Type: event
    Granted Operations: {view}

    Domain: any
    Object Type: attributegroup
    Granted Operations: {view}
```

Related commands

Return to Table 11 on page 301.

tivcmd login

Description

Use the **tivcmd login** command to authenticate with the IBM Dashboard Application Services Hub where the Authorization Policy Server is installed so that you can run subsequent tivcmd commands from the local system.

CLI syntax

tivcmd login
[{-s|--server}
{[PROTOCOL://]HOST|(HOST)[:PORT][/CONTEXTROOT]}]
[{-u|--users} ***USERNAME*****]**
[{-p|--password} ***PASSWORD*****]**
[{-t|--timeout} ***TIMEOUT*****]**

where:

-s|--server
Specifies a server to log in to, based on the IBM Dashboard Application Services Hub format: [PROTOCOL://]HOST|(HOST)[:PORT][/CONTEXTROOT]. The specification for the **--server** parameter includes, at a minimum, the host. The protocol, port, and context root are optional. Enclosing the host in parentheses is optional, unless you specify an IPv6 address. The IPv6 address format requires enclosing the host in parentheses. If you do not specify the protocol, the value 'https' is assumed. If 'https' does not work, 'https6', 'http', and 'http6' are tried, in that order. If you do not specify the port, and the protocol is:
- 'https' or 'https6', the value '16311' is assumed
- 'http' or 'http6', a value of '16310' is assumed

If you do not specify the context root, the value of 'ibm/tivoli/rest' is assumed. If the entire **--server** option is not specified, the host is assumed to be 'localhost' and the system attempts to connect to it using the various protocols.

-u|--username
Specifies a user name that can log in to the server.

-p|--password
Specifies the password for the user name.

-t|--timeout
Specifies the maximum number of minutes that can elapse between invocations of tivcmd before the user must use the **tivcmd login** command again. The default timeout is 15 minutes. The maximum timeout is 1440 minutes (24 hours).

CLI example

This example logs in to the server myserver to run a command:

```
tivcmd  login -s myserver
```

Related commands

Return to Table 11 on page 301.

tivcmd logout

Description

Use the **tivcmd logout** command to disable the security token that is created by the **tivcmd login** command. No additional tivcmd commands can be issued until another **tivcmd login** command is issued. To prevent unauthorized use of your tivcmd session, log out as soon as you have finished running tivcmd.

CLI syntax

tivcmd logout

CLI example

This example disables the security token that is created by the **tivcmd login** command:

```
tivcmd logout
```

Related commands

Return to Table 11 on page 301.

tivcmd removefromrole

Description

Use the **tivcmd removefromrole** command to remove users or groups from a role. The user name and group name are the unique names that are used in the user group repository. cn=user1,ou=users,ou=SWG,o=IBM,c=US is an example of a user name. cn=group1,ou=groups,ou=SWG,o=IBM,c=US is an example of a group name.

You must log in by using the **login** command before you run the **removefromrole** command.

CLI syntax

tivcmd removefromrole
{-n|--rolename} *ROLENAME*
{-u|--users} *USERS*

tivcmd removefromrole
{ -n|--rolename} *ROLENAME*
{-g|--groups} *GROUPS*

tivcmd removefromrole
{ -n|--rolename} *ROLENAME*
{-u|--users} *USERS*
{-g|--groups} *GROUPS*

where:

-n|--rolename
Specifies an existing role name from which to remove users or groups.

-u|--users
Specifies one or more users to be removed from the role. If you specify more than one user, separate the names by blanks.

-g|--groups
Specifies one or more groups to be removed from the role. If you specify more than one group, separate the names by blanks.

CLI example

This example removes user bob and user mary from the "West Coast WAS administrator" role:

```
tivcmd removefromrole -n "West Coast WAS administrator"
-u cn=bob,ou=users,ou=SWG,o=IBM,c=US cn=mary,ou=users,ou=SWG,o=IBM,c=US

KDQATR001I: Users [cn=bob,ou=users,ou=SWG,o=IBM,c=US
cn=mary,ou=users,ou=SWG,o=IBM,c=US] have been removed
from role [West Coast WAS administrator].
```

This example removes WASGroup1 and "West Coast WAS" groups from the "West Coast WAS administrator" role:

```
tivcmd removefromrole -n "West Coast WAS administrator"
-g cn=WASGroup1,ou=groups,ou=SWG,o=IBM,c=US
"cn=West Coast WAS,ou=groups,ou=SWG,o=IBM,c=US"

KDQATR002I: Groups [cn=WASGroup1,ou=groups,ou=SWG,o=IBM,c=US
"cn=West Coast WAS,ou=groups,ou=SWG,o=IBM,c=US"] have been
removed from role [West Coast WAS administrator].
```

Related commands

Return to Table 11 on page 301.

tivcmd revoke

Description

Use the **tivcmd revoke** command to remove the permission previously granted to a role using the **tivcmd grant** command or to remove the permission previously excluded from a role using the **tivcmd exclude** command.

You must log in using the login command before running the **tivcmd revoke** command.

The following table shows the valid values for the object types, resource types, and resources that are used in the revoke command when revoking a granted permission.

Table 14. Valid values for the object types, resource types, and resources when revoking a granted permission

Object Type	Resource types	Resources	Authorized operations and descriptions
attribute group	managedsystemgroup	One or more managed system group names	view: Revoke permission to view monitoring data from all managed systems that are members of the specified managed system groups.
	managedsystem	One or more managed system names	view: Revoke permission to view monitoring data from the specified managed systems.

Table 14. Valid values for the object types, resource types, and resources when revoking a granted permission (continued)

Object Type	Resource types	Resources	Authorized operations and descriptions
event	managedsystemgroup	One or more managed system group names	view: Revoke permission to view the list of situation events for all managed systems that are members of the specified managed system groups. **Note:** The attributegroup object type must be used to exclude permission to view situation event details.
	managedsystem	One or more managed system names	view: Revoke permission to view the list of situation events for the specified managed systems. **Note:** The attributegroup object type must be used to exclude permission to view situation event details.
role	rolegroup	default	create: Revoke permission to create roles using the tivcmd createrole and copyrole commands.
			delete: Revoke permission to delete roles using the tivcmd deleterole command.
			modify: Revoke permission to modify roles using the tivcmd grant, exclude, revoke, addtorole, and removefromrole commands.
			view: Revoke permission for a user who has logged in using the tivcmd command to use the tivcmd listroles command to view their own roles and permissions.
			viewAll: Revoke permission to view all roles and permissions using the tivcmd listroles command.
			distribute: Revoke permission to download the authorization policies from the Authorization Policy Server.

When you revoke an exclude permission for a managed system, the permissions listed in the table below are automatically removed from the policy store maintained by the Authorization Policy Server.

Table 15. Permissions automatically removed when an exclude permission is revoked

Object Type	Resource types	Resources	Authorized operations and descriptions
attribute group	managedsystem	One or more managed system names	view: Exclude permission to view monitoring data from the specified managed systems.
event	managedsystem	One or more managed system names	view: Exclude permission to view the list of situation events for the specified managed systems. **Note:** The attributegroup object type must be used to exclude permission to view situation event details.

CLI syntax

tivcmd revoke
{-n|--rolename} *ROLENAME*
{-t|--resourcetype} *RESOURCETYPE*
{-r|--resources} *RESOURCES*
[{-d|--domain} *DOMAIN***]**
{-e|--excludecommand}

tivcmd revoke
{-n|--rolename} *ROLENAME*
{-t|--resourcetype} *RESOURCETYPE*
{-r|--resources} *RESOURCES*
{-y|--objecttype} *OBJECTTYPE*
{-p|--operations} *OPERATIONS*
[{-d|--domain} *DOMAIN***]**
{-g--grantcommand}

tivcmd revoke
{-n|--rolename} *ROLENAME*
{-t|--resourcetype} *RESOURCETYPE*
{-i|--inputfile} *INPUTFILE*
[{-d|--domain} *DOMAIN***]**
{-e|--excludecommand}

tivcmd revoke
{-n|--rolename} *ROLENAME*
{-t|--resourcetype} *RESOURCETYPE*
{-i|--inputfile} *INPUTFILE*
{-y|--objecttype} *OBJECTTYPE*
{-p|--operations} *OPERATIONS*
[{-d|--domain} *DOMAIN***]**
{-g--grantcommand}

where:

-n|--rolename (required)
Specifies the role name from which you want to remove permission. The role name is case-sensitive and must exist.

-t|--resourcetype (required)
Specifies the type of resource for which to remove a permission. The resource type name is not case-sensitive.

-r|--resources (required)
Specifies the resources for which to remove a permission. The resource name is case-sensitive. Specify one or more resource names as a list of values that are separated by blanks. This option cannot be specified with the -i|--inputfile option.

-i|--inputfile
Specifies the fully qualified input file name that contains one or more resource names (one resource name per line). Encode this file in the native code page. This option cannot be specified with the -r|--resources option.

-y|--objecttype (required)
Specifies the type of object for which to remove a permission. The object type name is not case-sensitive.

--p|--operations (required)
Specifies the operations that were granted for the specified objects. The operation name is not case-sensitive. Specify one or more operations as a list of values that are separated by blanks.

-d|--domain
Specifies the domain name used with the grant or exclude command. In a multi-hub monitoring server environment, use this option to remove permissions for resources in a specific IBM Tivoli Monitoring domain. If this option is not specified, the permission applies to all domains and a value of *any* is saved as the domain name. By default, the domain name is `itm.<Hub TEMS name>` unless the domain name was overridden when configuring the Tivoli Enterprise Portal Server for the domain. Use the **tivcmd listdomains** command to list the domains that were specified when you work with policies or that have a connection defined in IBM Dashboard Application Service Hub to a domain's dashboard data provider.

-e|--excludecommand
Specifies whether to remove an exclude permission. This option is mutually exclusive from the -g option. You must specify either the **-e** or **-g** option for this command.

-g|--grantcommand
Specifies whether to remove a grant permission. This option is mutually exclusive from the **-e** option. You must specify either the **-e** or **-g** option for this command.

CLI example

In this example, the "UNIX Administrators" role was granted the ability to view all attribute groups for the managed system group name called *ALL_UNIX. To grant permission to the "UNIX Administrators" role, run the following command:

```
tivcmd grant -n "UNIX Administrators" -t managedsystemgroup -r "*ALL_UNIX" -y
attributegroup -p view

KDQCGR001I Role [UNIX Administrators] has been granted [view] access
to resource [*ALL_UNIX] of resource type [managedsystemgroup] of object type
[attributegroup] for the domain [any].
```

To revoke the grant permission from the "UNIX Administrators" role, run the following command:

```
tivcmd revoke -n "UNIX Administrators" -t managedsystemgroup -r "*ALL_UNIX"
-y attributegroup -p view -g

KDQARK001I The grant permission for resource [*ALL_UNIX] and resource type
[managedsystemgroup] of object type [attributegroup] for [view]
operations has been revoked for role [UNIX Administrators]
for the domain [any].
```

In this example, the "Unix Administrator" role was excluded permission to view attribute groups for the WestCoastServer managed system. To eliminate this restriction for the "Unix Administrator" role, run the following command:

```
tivcmd exclude -n "Unix Administrator" -t managedsystemgroup -r
"WestCoastServer"

KDQAEX001I Role [Unix Administrator] has been excluded permission
for the [view] operation on object type [event attributegroup] and resource
[WestCoastServer] of resource type [managedsystem] in [any] domain.
```

To revoke the excluded permission for the "Unix Administrator" role, run the following command:

```
tivcmd revoke -n "Unix Administrator" -t managedsystem -r
"WestCoastServer" -e

KDQARK002I The exclude permission for resource [WestCoastServer] and
resource type [managedsystem] of object type [event attributegroup] for [view]
operations has been revoked for role [Unix Administrator] for in the domain [any].
```

Return values

See Table 8 on page 272.

Related commands

Return to Table 11 on page 301.

Chapter 5. Tivoli Enterprise Console commands

You can run the Tivoli Enterprise Console commands on the Tivoli Enterprise Console event server to configure the event synchronization between IBM Tivoli Monitoring and Tivoli Enterprise Console.

Table 16. Tivoli Enterprise Console commands

Command	Description
"sitconfig.sh command"	Set or change the configuration of the event synchronization.
"sitconfsvruser.sh command" on page 331	Add, update, or delete monitoring server information for event synchronization.
"upg_sentry_baroc.pl script" on page 332	Update the Sentry2_0_Base class file with additional integration attributes for the situation events received from IBM Tivoli Monitoring.
"upg_tec_baroc.pl script" on page 333	Add the TEC_Generic class to the tec.baroc file in the specified rule base.
"wrules_check" on page 333	Assess the impact on an existing set of rules whenever BAROC event classes designs are changed

Run these commands from the *$BINDIR*/TME/TEC/OM_TEC/bin directory (where *$BINDIR* is the location of the Tivoli Management Framework installation).

When running these commands, if you are specifying fully qualified paths, use a forward slash (/) for all operating systems, including Windows.

sitconfig.sh command

Description

Use the **sitconfig.sh** command to set or change the configuration of the event synchronization. You can use this command to initially create the configuration settings or to update existing settings.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process. See the *IBM Tivoli Monitoring Installation and Setup Guide* for information.

CLI syntax

sitconfig.sh add fileName=*config_filename*
fileSize=*size*
fileNumber=*number*
fileLocation=*path*
pollingInterval=*seconds*
crcBytecount=*count*
cmsSoapUrl=*url*
bufferFlushRate=*rate*
logLevel=*level*

sitconfig.sh update fileName=*config_filename*
[fileSize=*size***]**
[fileNumber=*number***]**
[fileLocation=*path***]**
[pollingInterval=*seconds***]**
[crcBytecount=*count***]**
[cmsSoapUrl=*url***]**
[bufferFlushRate=*rate***]**
[logLevel=*level***]**

sitconfig.sh

where:

add Create the configuration file. The default name is `situpdate.conf`.

update
Updates the existing specified configuration file.

refresh
Reads the situation timeouts file (`sit_timeouts.conf`) and loads the situation timeouts into the TEC rule.

fileName = *config_filename*
The name of the configuration file for event synchronization. `situpdate.conf` is the default file name.

fileSize=*size*
Specify this option to set and change the maximum size, in bytes, for any one event cache file. The minimum (and default) value is 50000. Do not use commas when specifying this value (use 50000 instead of 50,000).

fileNumber=*number*
Specify this option to set and change the maximum number of event caches files permitted at any given time. The minimum value is 2, while the default value is 10. When this value is reached, the oldest file is deleted to make room for a new file.

fileLocation=*path*
Specify this option if you want to set and change the location on the event server where event cache files are located. The default locations are as follows:

- On Windows: `C:\tmp\TME\TEC\OM_TEC\persistence`.
- On UNIX: `/var/TME/TEC/OM_TEC/persistence`

pollingInterval=*seconds*
Specify this option to set and change the polling interval, in seconds. The minimum value is 1, while the default value is 3. This is the number of seconds that the Situation Update Forwarder process sleeps when there are no updates to process.

crcBytecount=*count*
Specify this option to set and change the number of bytes that the long running process will use when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) is 50.

cmsSoapUrl=*url*
Specify this option to set and change the URL for the Service Oriented Architecture Protocol Server configured on the computer where the monitoring server is running. The default value is cms/soap. This value is

used to create the URL to which Tivoli Enterprise Console sends event information. For example, `http://hostname:port///cms/soap`, where *hostname* is the hostname of the monitoring server and *port* is the port.

bufferFlushRate=*rate*
Specify this option to set and change the maximum number of event updates sent to the monitoring server at one time. The minimum (and default) value is 100 events.

logLevel=*level*
Specify this option to set and change the level of information for event synchronization that is logged. You have the following choices:

- low (default)
- med
- verbose

CLI example

The following example changes the trace level for the event synchronization to medium:

```
sitconfig.sh update fileName=situpdate.conf logLevel=med
```

Related commands

Return to Table 16 on page 329.

sitconfsvruser.sh command

Description

Use the **sitconfsvruser.sh** command to add, update, or delete monitoring server information for event synchronization.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process. See the *IBM Tivoli Monitoring Installation and Setup Guide* for information.

CLI syntax

sitconfsvruser.sh add
serverid=*server*
userid=*user*
password=*password*

sitconfsvruser.sh update
serverid=*server*
userid=*user***>**
password=*password*

sitconfsvruser.sh delete
serverid=*server*

where:

add Adds a new monitoring server to the list of monitoring servers that forward events to Tivoli Enterprise Console.

update
Modifies the user ID or password for an existing monitoring server.

delete Removes a monitoring server from the list of monitoring servers that forward events to Tivoli Enterprise Console.

serverid=*server*
The fully qualified hostname of the monitoring server. Should be equivalent to what the Tivoli Enterprise Monitoring Server sends as the cms_hostname attribute in an event.

userid=*user*
The user ID to access the computer where the monitoring server is running.

password=*password*
The password to access the computer.

CLI example

The following example adds the itm17 monitoring server:

```
sitconfsvruser.sh add serverid=itm17.ibm.com userid=admin password=acc3ssing
```

Related commands

Return to Table 16 on page 329.

upg_sentry_baroc.pl script

Description

Use the **upg_sentry_baroc.pl** script to update the Sentry2_0_Base class file with additional integration attributes for the situation events received from IBM Tivoli Monitoring.

If you specify a rule base that does not contain the Sentry2_0_Base class, no changes are made.

Use this script only when you are manually upgrading your rule base after installing the event synchronization.

CLI syntax

upg_sentry_baroc.pl [*rb_name* **[***rb_path***]]**

where:

rb_name
Specifies the rule base that you want to update. If you do not specify a rule base, all existing rule bases are updated.

rb_path
The path to the rule base specified with the *rb_name* option. This path is optional.

CLI example

The following example updates the Sentry2_0_Base class in the `Sentry.baroc` file of the itmsynch_rb rule base:

```
upg_sentry_baroc.pl itmsynch_rb
```

Related commands

Return to Table 16 on page 329.

upg_tec_baroc.pl script

Description

Use the **upg_tec_baroc.pl** script to add the TEC_Generic class to the `tec.baroc` file in the specified rule base. This class is required for event synchronization.

If the rule base already contains the TEC_Generic class in the `tec.baroc` file, no changes are made.

Use this script only when you are manually upgrading a rule base after installing the event synchronization.

CLI syntax

upg_tec_baroc.pl *rb_name*

where:

rb_name
The name of the rule base to upgrade. This name is required.

CLI example

The following example adds the TEC_Generic class to the `tec.baroc` file of the itmsynch_rb rule base:

```
upg_tec_baroc.pl itmsynch_rb
```

Related commands

Return to Table 16 on page 329.

wrules_check

Description

The **wrules_check** command provides you with the ability to assess the impact on an existing set of rules whenever BAROC event classes designs are changed. Use this command to verify which rules can be impacted by these event class definition changes.

The Rules check utility is shipped with IBM Tivoli Monitoring and is installed as part of Tivoli Enterprise Console Event Synchronization. This utility is installed in the `$BINDIR/TME/TEC/OM_TEC/bin` directory. It does not require any specific directory configuration if the required privileges for access to the input files and output files are granted.

To run the Rules Check command you must have:

- Read access to the `*.rls` and `*.baroc` files that will be used as inputs.
- Write access to the output that is used to store the results of the check.

- 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.

CLI syntax

wrules_check -h

wrules_check -v

wrules_check ***class*[*,attribute,attribute...*]** **[*:classN,attributeN,...,attributeN*]**
[-rd ***rules_directory*]**
[-cd ***baroc_classes_directory*]**
[-of ***output_file*]**

wrules_check -f ***class_file***
[-rd ***rules_directory*]**
[-cd ***baroc_classes_directory*]**
[-of ***output_file*]**

where:

-h Displays this help and exits.

-v Displays the rules check utility version and exits.

-f Used to specify a file containing classes and class attributes to be checked. This file has the following format:

```
class a[,attribute,...,attribute] [;class z[attributeZ,...,attributeZ]]
```

-rd Used to specify a directory containing the rulesets (`*.rls`) to be checked. If not provided, the `TEC_RULES` subdirectory for the actively loaded rule base is used by default.

-cd Used to specify a directory containing the BAROC event class definitions files (`*.baroc`) to be used as input. If not provided the `TEC_CLASSES` subdirectory for the actively loaded rule base is used by default.

-of Used to specify the name of the output file.

Format of the output file

```
Parsing <rule_base_dir>/<rulefile_1.rls>

rules impacted by class:
CLASS_1: <rule_1_within rulefile_1,rls>, ...,
   <rule_n_within rulefile_1.rls_n>
   <attribute_name_1>:<rule_1_within rulefile_1.rls>
      ...
   ...
CLASS_n:<rule_1_within rulefile.rls>
   <attribute_name_1>:<rule_1_within rulefile_1.rls>...
   <attribute_name_n>:<rule_1_within rulefile_1.rls>,...
   <rule_n_within rulefile_1.rls_1>
*****************************************************************
  ...
  ...
*****************************************************************
Parsing <rule_base_dir>/<rulefile_n.rls>
CLASS_1: <rule_1_within rulefile_n,rls>, ...,
   <rule_n_within rulefile_n.rls_n>
   <attribute_name_1>:<rule_1_within rulefile_n.rls>
      ...
*****************************************************************
```

```
          multiple inheritance sample output
*****************************************************************
<file_1 baroc>has the following classes with multiple inheritance:
CLASS_1
...
CLASS_n
*****************************************************************
...
...
*****************************************************************
<file_n baroc>has the following classes with multiple inheritance:
 CLASS_1
...
*****************************************************************
        outside operator sample output
*****************************************************************
rules impacted by outside operator:
CLASS_1:<rule_1_within rulefile_n.rls>, ...,<rule_n_within
   rulefile_n.rls> ...
CLASS_n:<rule_1_within rulefile_n.rls>
```

CLI example

The following is an example for the wrules_check command:

```
wrules_check -rd C:\temp\itmg5\TEC_RULES -cd C:\temp\itmg5\TEC_CLASSES EVENT
```

Return values

The results of the output file from the above command are:

```
Parsing C:\temp\itmg5\TEC_RULES\
..
..

rules impacted by class:
DB2_Down_Status:
    status:lower_itm,redo_db2_was
DB2_High_ApplicationAgent_TotSystemCpuTime:
    status:lower_itm,redo_db2_was
DB2_High_ApplicationAgents_Workload:

************************************************************

Parsing C:\temp\itmg5\TEC_RULES\

rules impacted by class:
TEC_Heartbeat_missed:
    severity:heartbeat_missed

rules impacted by outside operator:
Omegamon_Base:process_sit_events_only
Sentry2_0_Base:process_sit_events_only
TEC_Generic:process_sit_events_only

************************************************************
```

Related commands

Return to Table 16 on page 329.

Chapter 6. Tivoli Netcool/OMNIbus commands

You can run the sitconf and sitconfuser commands on the Tivoli Netcool/OMNIbus ObjectServer to configure the event synchronization between IBM Tivoli Monitoring and Tivoli Netcool/OMNIbus.

Table 17. Tivoli Netcool/OMNIbus commands

Command	Description
"sitconf"	Set or change the configuration of the event synchronization.
"sitconfuser" on page 339	Add, update, or delete monitoring server information for event synchronization.

Run these commands from the *Event_Sync_Install_Dir*/bin directory, where *Event_Sync_Install_Dir* is the directory where event synchronization was installed.

sitconf

Description

Use the **sitconf.sh** command on UNIX systems and the **sitconf.cmd** command on Windows systems to set or change the configuration of the event synchronization. You can use this command to initially create the configuration settings or to update existing settings.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process. See the *IBM Tivoli Monitoring Installation and Setup Guide* for information.

CLI syntax

sitconf.sh add fileName=*config_filename*
fileSize=*size*
fileNumber=*number*
fileLocation=*path*
pollingInterval=*seconds*
crcBytecount=*count*
cmsSoapUrl=*url*
bufferFlushRate=*rate*
logLevel=*level*
pathc=*path_to_conf_file*
type=*OMNIBUS*

sitconf.sh update fileName=*config_filename*
[fileSize=*size***]**
[fileNumber=*number***]**
[fileLocation=*path***]**
[pollingInterval=*seconds***]**
[crcBytecount=*count***]**
[cmsSoapUrl=*url***]**

[bufferFlushRate=*rate*]
[logLevel=*level*]
pathc=*path_to_conf_file*
type=*OMNIBUS*

sitconf.sh refresh
pathc=path_to_conf_file
type=OMNIBUS

where:

add Create the configuration file. The default name is `situpdate.conf`.

update
Updates the existing specified configuration file.

refresh
Refreshes event synchronization to use the latest situation timeout configuration parameter in the `sit_timeouts.conf` file. To perform the refresh of event synchronization, the `errorevent.conf` file must also be configured. See the topic "Configuring error event flow to OMNIbus (optional)" in the *IBM Tivoli Monitoring Installation and Setup Guide* for more details on configuring the `errorevent.conf` file.

fileName = *config_filename*
The name of the configuration file for event synchronization. `situpdate.conf` is the default file name.

fileSize=*size*
Specify this option to set and change the maximum size, in bytes, for any one event cache file. The minimum (and default) value is 50000. Do not use commas when specifying this value (use 50000 instead of 50,000).

fileNumber=*number*
Specify this option to set and change the maximum number of event caches files permitted at any given time. The minimum value is 2, while the default value is 10. When this value is reached, the oldest file is deleted to make room for a new file.

fileLocation=*path*
Specify this option if you want to set and change the location on the event server where event cache files are located. The default locations are as follows:

- On Windows: *`Event_Sync_Install_Dir`*`/persistence`
- On UNIX: *`Event_Sync_Install_Dir`*`\persistence`

Where *`Event_Sync_Install_Dir`* is the directory chosen when event synchronization was installed.

pollingInterval=*seconds*
Specify this option to set and change the polling interval, in seconds. The minimum value is 1, while the default value is 3. This is the number of seconds that the Situation Update Forwarder process sleeps when there are no updates to process.

crcBytecount=*count*
Specify this option to set and change the number of bytes that the long running process will use when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) is 50.

cmsSoapUrl=*url*
Specify this option to set and change the URL for the Service Oriented Architecture Protocol Server configured on the computer where the monitoring server is running. The default value is cms/soap. This value is used to create the URL to which Tivoli Netcool/OMNIbus sends event information. For example, http://*hostname*:*port*///cms/soap, where *hostname* is the hostname of the monitoring server and *port* is the port.

bufferFlushRate=*rate*
Specify this option to set and change the maximum number of event updates sent to the monitoring server at one time. The minimum (and default) value is 100 events.

logLevel=*level*
Specify this option to set and change the level of information for event synchronization that is logged. You have the following choices:

- low (default)
- med
- verbose

pathc=
The location where the configuration file will be placed.

type=OMNIBUS
Required because this command is specific to Tivoli Netcool/OMNIbus.

CLI example

The following example changes the trace level for the event synchronization to medium:

```
sitconf.sh update fileName=situpdate.conf logLevel=med
pathc=/opt/IBM/SitForwarder/etc type=OMNIBUS
```

Related commands

Return to Table 17 on page 337.

sitconfuser

Description

Use the **sitconfuser.sh** command on UNIX systems and the **sitconfuser.cmd** command on Windows systems to add, update, or delete monitoring server information for event synchronization.

After you change the configuration of the event synchronization, you must manually stop and restart the Situation Update Forwarder process. See the *IBM Tivoli Monitoring Installation and Setup Guide* for information.

CLI syntax

sitconfuser.sh add
serverid=*server*
userid=*user*
password=*password*
pathc=*path_to_conf_file*
type=*OMNIBUS*

sitconfuser.sh update
serverid=*server*
userid=*user***>**
password=*password*
pathc=*path_to_conf_file*
type=*OMNIBUS*

sitconfuser.sh delete
serverid=*server*
pathc=*path_to_conf_file*
type=*OMNIBUS*

where:

add Adds a new monitoring server to the list of monitoring servers that forward events to Tivoli Netcool/OMNIbus.

update
Modifies the user ID or password for an existing monitoring server.

delete Removes a monitoring server from the list of monitoring servers that forward events to Tivoli Netcool/OMNIbus.

serverid=*server*
The fully qualified hostname of the monitoring server. Should be equivalent to what the Tivoli Enterprise Monitoring Server sends as the cms_hostname attribute in an event.

userid=*user*
The user ID to access the computer where the monitoring server is running.

password=*password*
The password to access the computer where the monitoring server is running. Note that the password is required. If your monitoring server is not configured to authenticate SOAP users, specify any non-blank string for the password.

pathc=
The location where the configuration file will be placed.Specify the path to the *<Event_Sync_Install_Dir>*/etc directory where *<Event_Sync_Install_Dir>* is the directory where the IBM Tivoli Monitoring Event Synchronization component is installed.

type=OMNIBUS
Required because this command is specific to Tivoli Netcool/OMNIbus.

CLI example

The following example adds the itm17 monitoring server:

```
sitconfuser.sh add serverid=itm17.ibm.com userid=admin password=acc3ssing
pathc=/opt/IBM/SitForwarder/etc type=OMNIBUS
```

Related commands

Return to Table 17 on page 337.

Documentation library

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

These publications are listed in the following categories:

- IBM Tivoli Monitoring library
- Related publications

For information about accessing and using the publications, select **Using the publications** in the **Contents** pane of the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp.

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

IBM Tivoli Monitoring library

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

- *Quick Start Guide*

  Introduces the components of IBM Tivoli Monitoring.
- *Installation and Setup Guide*, SC22-5445

  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.
- *High Availability Guide for Distributed Systems*, SC22-5455

  Gives instructions for several methods of ensuring the availability of the IBM Tivoli Monitoring components.
- *IBM Tivoli zEnterprise Monitoring Agent Installation and Configuration Guide*, SC14-7358

  Provides instructions for installing and configuring Tivoli zEnterprise monitoring agent components on Windows, Linux, and UNIX systems. Also includes migration and backup information, Enterprise Common Collector troubleshooting, Hardware Management Console configuration, and use of the command line interface or APIs to customize the collector. This guide complements the *Tivoli zEnterprise Monitoring Agent User's Guide*.
- *Administrator's Guide*, SC22-5446

  Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.
- *Command Reference*, SC22-5448

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

- *Messages*, SC22-5450

  Lists and explains messages generated by all IBM Tivoli Monitoring components and by z/OS-based Tivoli Management Services components (such as Tivoli Enterprise Monitoring Server on z/OS and TMS:Engine).
- *Troubleshooting Guide*, GC22-5449

  Provides information to help you troubleshoot problems with the software.
- 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*, SC22-5447

  Complements the Tivoli Enterprise Portal online help. The guide provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.
- *Agent Builder User's Guide*, SC32-1921

  Explains how to use the Agent Builder for creating monitoring agents and their installation packages, and for adding functions to existing agents.
- *Performance Analyzer User's Guide*, SC27-4004

  Explains how to use the Performance Analyzer to understand resource consumption trends, identify problems, resolve problems more quickly, and predict and avoid future problems.
- *IBM Tivoli zEnterprise Monitoring Agent User's Guide*, SC14-7359

  Complements the Tivoli zEnterprise monitoring agent online help. The guide provides reference information about the interface, usage scenarios, agent troubleshooting information, and information about Tivoli Common Reporting reports. This guide complements the *Tivoli zEnterprise Monitoring Agent Installation and Configuration Guide*.

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*, SC22-5451
  - *UNIX OS Agent User's Guide*, SC22-5452
  - *Linux OS Agent User's Guide*, SC22-5453
  - *IBM i Agent User's Guide*, SC22-5454
- 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*, SC22-5457

- *Warehouse Proxy Agent User's Guide*, SC22-5456
- 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:
  - *Tivoli Log File Agent User's Guide*, SC14-7484
  - *Systems Director base Agent User's Guide*, SC27-2872

Related publications

For information about related products and publications select **OMEGAMON XE shared publications** or other entries in the **Contents** pane of the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp .

Other sources of documentation

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

- Service Management Connect (SMC)

  For introductory information about SMC, see IBM Service Management Connect (http://www.ibm.com/developerworks/servicemanagement).

  For information about Tivoli products, see the Application Performance Management community on SMC at IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm).

  Connect, learn, and share with Service Management professionals. Get access to developers and product support technical experts who provide their perspectives and expertise. Using SMC, you can:
  - Become involved with transparent development, an ongoing, open engagement between external users and developers of Tivoli products where you can access early designs, sprint demos, product roadmaps, and pre-release code.
  - Connect one-on-one with the experts to collaborate and network about Tivoli and Integrated Service Management.
  - Benefit from the expertise and experience of others using blogs.
  - Collaborate with the broader user community using wikis and forums.
- Tivoli wikis

  IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm) includes a list of relevant Tivoli wikis that offer best practices and scenarios for using Tivoli products, white papers contributed by IBM employees, and content created by customers and business partners.

  Two of these wikis are of particular relevance to IBM Tivoli Monitoring:
  - The IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Monitoring/page/Home) provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Management products.

  - The Tivoli System z® Monitoring and Application Management Wiki provides information about the OMEGAMON XE products, NetView® for z/OS®, Tivoli Monitoring Agent for z/TPF, and other System z monitoring and application management products.
- IBM Integrated Service Management Library

  http://www.ibm.com/software/brandcatalog/ismlibrary/

  IBM Integrated Service Management Library is an online catalog that contains integration documentation and other downloadable product extensions.
- Redbooks®

  http://www.redbooks.ibm.com/

  IBM Redbooks and Redpapers include information about products from platform and solution perspectives.
- Technotes

  Technotes provide the latest information about known product limitations and workarounds. You can find Technotes through the IBM Software Support Web site at http://www.ibm.com/software/support/.

Support information

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

Online

The following sites contain troubleshooting information:

- Go to the IBM Support Portal (http://www.ibm.com/support/entry/portal/software) and follow the instructions.
- Go to IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm) and select the appropriate wiki.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to IBM Support Assistant (http://www-01.ibm.com/software/support/isa).

Troubleshooting Guide

For more information about resolving problems, see the product's Troubleshooting Guide.

Using IBM Support Assistant

The IBM Support Assistant is a free, stand-alone application that you can install on any workstation. You can then enhance the application by installing product-specific plug-in modules for the IBM products you use.

The IBM Support Assistant saves you the time it takes to search the product, support, and educational resources. The IBM Support Assistant helps you gather support information when you need to open a problem management record (PMR), which you can then use to track the problem.

The product-specific plug-in modules provide you with the following resources:

- Support links
- Education links
- Ability to submit problem management reports

For more information, and to download the IBM Support Assistant, see http://www.ibm.com/software/support/isa. After you download and install the IBM Support Assistant, follow these steps to install the plug-in for your Tivoli product:

1. Start the IBM Support Assistant application.
2. Select **Updater** on the Welcome page.
3. Select **New Properties and Tools** or select the **New Plug-ins** tab (depending on the version of IBM Support Assistant installed).
4. Under **Tivoli**, select your product, and then click **Install**. Be sure to read the license and description.

If your product is not included on the list under **Tivoli**, no plug-in is available yet for the product.

5. Read the license and description, and click **I agree**.
6. Restart the IBM Support Assistant.

Obtaining fixes

A product fix might be available to resolve your problem. To determine which fixes are available for your Tivoli software product, follow these steps:

1. Go to the IBM Software Support website at http://www.ibm.com/software/support.
2. Under **Select a brand and/or product**, select **Tivoli**.

   If you click **Go**, the **Search within all of Tivoli support** section is displayed. If you don't click **Go**, you see the **Select a product** section.
3. Select your product and click **Go**.
4. Under **Download**, click the name of a fix to read its description and, optionally, to download it.

   If there is no **Download** heading for your product, supply a search term, error code, or APAR number in the field provided under **Search Support (this product)**, and click **Search**.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

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

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

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

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

Contacting IBM Software Support

IBM Software Support provides assistance with product defects. The easiest way to obtain that assistance is to open a PMR or ETR directly from the IBM Support Assistant.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere® products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online
  Go to the Passport Advantage website at http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm .

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

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html and click the name of your geographic region for telephone numbers of people who provide support for your location.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Index

A

C

D

E

F

I

K

N

P

R

S

T

U

V

W

Printed in USA

SC22-5448-00