

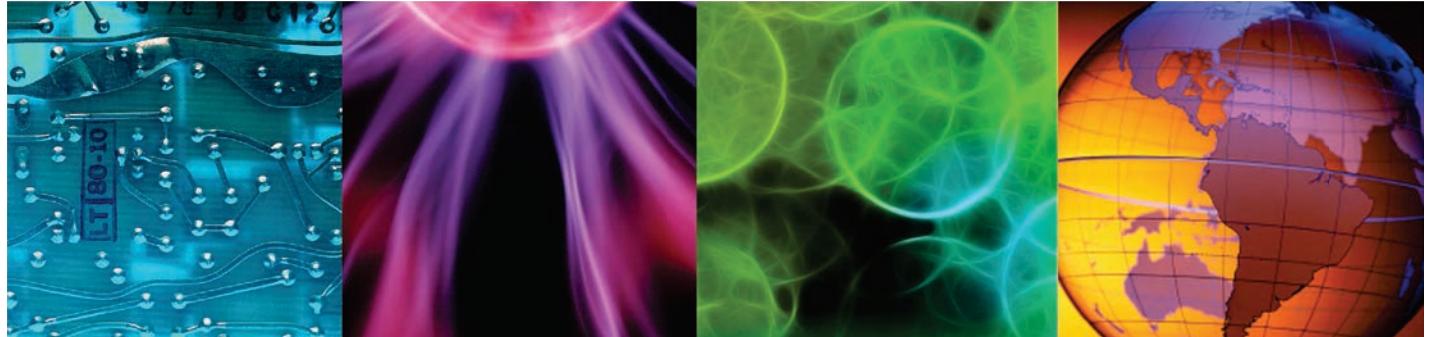


IBM Training

**IBM Tivoli Monitoring 6.3
Advanced Administration
Student Notebook**

Course code TM063 ERC 1.0

November 2013



Cloud & Smarter Infrastructure

All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice.

© Copyright IBM Corp. 2013. All Rights Reserved.

US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, and ibm.com are 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.

Trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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 registered 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 Office of Government Commerce, 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.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.



Contents

About this course	xi
About the student	xii
Course objectives	xiii
Course agenda	xiv
Instructor notes	xvi
1 Queries	1
Objectives	2
Lesson 1. Queries	3
Query overview	4
Product-provided queries	5
Managing queries	6
Accessing the Query editor	7
The Query editor	8
Creating another query	9
Deleting a query	10
Creating a query	11
Possible query data sources	12
Creating an agent query	13
Filters and functions	15
Adding attributes	16
Deleting attributes	17
Query results source	18
Advanced options	21
Advanced options: Group By functions	22
Group By functions: Example	23
Group By functions: Result	24
Lesson 2. Creating a query against external databases	25
Creating a query against ODBC data: Steps	27
Determining the data format (custom database)	28
Creating the ODBC data source in Windows	29
Configuring the portal server	31
Configuring the portal server using tacmd	32
Creating the custom query: Windows	33
Adding a data source on Linux and UNIX	34
Creating the custom query: Linux and UNIX	35
Specifying the SELECT statement	36
Testing the new query	37

Performance considerations	38
Monitoring infrastructure	39
Student exercise	40
Review questions	41
Review answers	42
Summary	43
2 Advanced link topics	44
Objectives	45
Lesson 1. Understanding different types of links	46
Links: Description	47
Links: Sources	48
Links: Simple versus advanced	49
Using a simple link	50
Using an advanced link	51
Link types: Dynamic, absolute, and relative	52
Link types: Absolute links	54
Link types: Relative links	55
Link types: Dynamic links	56
Dynamic links: Example	57
Lesson 2. Defining a simple, absolute link	58
Using the Link wizard to define a new link	59
Entering the link name and description	60
Defining the link target as absolute	61
Selecting the target workspace	62
Issuing the absolute link	63
Lesson 3. Defining a relative link	64
Issuing the relative link	65
Lesson 4. Defining a dynamic link	66
Defining the target selection criteria	67
Issuing the dynamic link	69
Lesson 5. Advanced links	70
New terminology: Link symbols	72
Defining an advanced link: Preparation	74
Launching the Link wizard	75
Assigning expressions to link symbols	76
Modifying expressions	77
Defining the link expression	78
Validating the expression	79
Issuing the advanced link	80
Expression editor: Components	81
Other link symbols: Header and footer	82
Issuing the link: Test the header and footer substitutions	83
Link settings: Link anchor	84
Student exercise	85
Review questions	86
Review answers	87
Summary	88

3 Monitoring your enterprise using advanced situation techniques	89
Objectives	90
Lesson 1. Expert advice	91
Accessing expert advice	93
Specifying expert advice	94
Previewing expert advice	95
Product-provided expert advice definition	96
Creating custom expert advice	97
Accessing expert advice HTML documents	98
Copying and saving a document	100
Modifying the content	101
Copying advice expressions	102
More expert advice techniques	103
Lesson 2. Embedded and correlated situations	105
Embedded situations	107
Embedded situation example: PrimeShift	108
Correlated situations	109
Lesson 3. Creating situations from a model	111
Passing attributes from the view	112
The new situation from current data	113
Using historical data to model situations	114
Situation model editor	115
Applying statistical functions	116
Modifying statistical functions	117
The new situation from historical data	118
Lesson 4. Situation overrides for dynamic thresholds	119
Dynamic thresholds	121
Overriding a situation formula	122
Overriding a situation	123
Applying a calendar schedule	124
Historical data baselines	125
Historical baseline example	126
Student exercise	127
Review questions	128
Review answers	129
Summary	130
4 Agent autonomy	131
Objectives	133
Lesson 1. Agent autonomy introduction	134
Agent autonomy overview	135
Agent autonomy features	136
Four levels of agent autonomy: Level 1	137
Four levels of agent autonomy: Level 2	138
Four levels of agent autonomy: Level 3	139
Four levels of agent autonomy: Level 4	140
Lesson 2. Agent autonomy installation and configuration	141
SNMP MIB and trap types	143

SNMP trap types	144
Simple SNMP trap configuration XML example	146
Another SNMP trap configuration file	147
Agent autonomy messages	148
Lesson 3. Private situations	150
Two ways to create private situations	151
Private situation configuration specification	152
XML elements for private situations	153
Simple private situation example	154
Private situation example with reflex automation	155
Private situation configuration *MISSING support	156
Correct examples of *MISSING support	157
Incorrect examples of *MISSING support	158
Private history	159
Simple private configuration file for historical data collection	160
Lesson 4. Netcool/OMNIbus integration	162
Netcool/OMNIbus integration steps	164
Lesson 5. Agent Service Interface	168
Agent Service Interface features	169
Agent Service Interface requests	170
Agent Service Interface example	171
Agent Service Interface: Agent information	173
Agent Service Interface: Situations	174
Agent Service Interface: History	175
Agent Service Interface: Queries	177
Agent Service Interface: Service Interface request	178
Review questions	180
Review answers	181
Summary	182
5 Working with policies	183
Objectives	184
Lesson 1. The Workflow Editor	185
Accessing the Workflow Editor	186
The Workflow Editor: Content	187
The Workflow Editor pane	188
Edit toolbar	189
View toolbar	190
Policy details	191
Policy distribution	192
Correlation modes	193
Additional options	194
Lesson 2. Creating a policy	195
Situation activities	196
Starting and stopping activities	197
Suspending a policy execution	198
Waiting until a situation is false	199
Take action or write a message activity	200

Take action activity: Attribute substitution	201
Take action activity: More options	202
Take action activity: Continue on failure	203
Connector	204
Emitter activities	205
Extensions	206
Saving changes	207
Lesson 3. Workflow validation	208
Workflow validation: Success	209
Workflow validation: Failure	210
Workflow validation: Policy rules	211
Activating the policy	212
Policy functions: Start and stop	213
Policy usage scenarios: Policy workflow	214
Troubleshooting a policy workflow	215
Student exercises	216
Review questions	217
Review answers	218
Summary	219
6 Agentless monitoring.....	220
Objectives	221
Lesson 1. Agentless monitoring overview	222
Agent or agentless	223
Understanding agent and agentless technology	224
Agentless monitoring architecture	225
Comparing agent and agentless technologies	226
Agentless monitoring for operating systems	227
Agentless monitoring scalability	233
Performance tuning environment variables	234
Agentless management	235
Agentless OS monitoring metric overview	236
Remote node capabilities	237
Agentless monitoring data collection and platforms	238
Lesson 2. Agentless monitoring planning and configuration	239
Windows configuration	240
SNMP configuration	242
SNMP community name configuration	243
SNMP managed system name configuration	244
CIM configuration (1 of 3)	245
Removing the instance on Windows	248
Removing the instance on Linux and UNIX	249
Configuration verification	250
Lesson 3. Troubleshooting agentless monitoring	251
Troubleshooting connectivity	252
Troubleshooting agent log files	253
Troubleshooting tips	254
Student exercise	256

Review questions	257
Review answers	258
Summary	259
7 Agent Management Services.....	260
Objectives	261
Lesson 1. Monitoring the agents	262
Basic functions	263
Results of using Agent Management Services	264
Installation	265
Managing the agents	266
Controlling with the portal user interface	267
Lesson 2. Components and configuration	268
Agent Management Services watchdog component	269
Agent watchdog component	270
CAP directory definition environment variable	271
CAP file	272
CAP file parameters	273
Agents' Management Log	274
Interaction with Tivoli Monitoring components	275
Managing Agent Management Services	276
Watchdog process control	277
Lesson 3. Take Action commands	278
Take Action command example	280
Automated procedure	283
Agent Management Services configuration	285
Agent Management Services verification	286
Lesson 4. Agent Management Services workspace	287
Agents' Management Definitions view	289
Agents' Runtime Status view	291
Agents' Management Status view	293
Agents' Alerts view	295
Student exercises	297
Review questions	298
Review answers	299
Summary	300
8 Managing situation event integration.....	301
Objectives	302
Lesson 1. Integration points to other event repositories	303
Situation event integration	305
Situation event integration using workspaces	306
Common Event Console view	307
Common Event Console: Close or acknowledge events	308
Tivoli Event Console view: Close or acknowledge events	309
Agent autonomy forwards events to Netcool/OMNibus	310
Lesson 2. Situation event integration implementation	311
Installing the event synchronization component	312
Configuring the monitoring server to forward events (Windows)	313

Configuring the monitoring server to forward events (Linux)	314
Event forwarding on z/OS	315
Configuring monitoring server to forward events (z/OS)	317
Event destination creation	318
Editing the situation using the EIF notebook page	319
Lesson 3. Situation event forwarding customization	320
Events forwarded to Netcool/OMNIbus ObjectServer	321
EIF Slot Customization editor	322
Event slot customization features	323
EIF extended slot selection	324
EIF extended slots and base slots	325
Modifying EIF slot values	326
EIF slot Literal value column	327
Map all attributes	329
Customizing the message slot	330
Customizing the message slot: Before	332
Customizing the message slot: After	333
Slot customization window: Problem determination	334
Student exercises	335
Review questions	336
Review answers	337
Summary	338
9 Command-line interfaces.	339
Objectives	340
Lesson 1. Running the tacmd command	341
Basic commands	343
Agent management commands	344
Agent management commands	345
Deployment commands	346
Remote deployment and managing systems	347
Managing remote deployment commands	348
Managing system lists commands	349
Self describing agent commands	350
Working with situations and actions commands	351
Situations and attributes override commands	352
Situation association commands	353
Take Action commands	354
Policy management commands	355
Navigator commands	356
Workspace management commands	357
Query management commands	358
Historical data commands	359
Historical data collection commands	360
Historical data baselines commands	361
Calendar functions commands	362
Administering users with commands	363
Administering users by user group commands	364

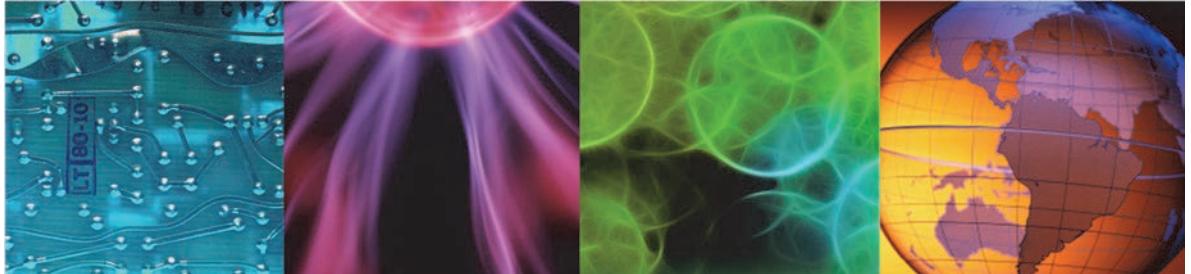
Group management commands	365
Group member commands	366
Event management commands	367
Refreshing Tivoli Enterprise Console and Netcool/OMNIbus information	368
Miscellaneous commands	369
Lesson 2. Issuing the itmcmd commands	371
More itmcmd commands	372
Lesson 3. Managing authorization policies	373
More tivcmd commands	374
Lesson 4. Simple Object Access Protocol (SOAP)	375
Uses of SOAP methods in Tivoli Monitoring	376
List of SOAP methods in Tivoli Monitoring	377
SOAP client command-line utility kshsoap	378
Command results	379
Student exercises	380
Review questions	381
Review answers	382
Summary	383



About this course



IBM Tivoli Monitoring Advanced Administration Student Guide



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM® Tivoli® Monitoring products monitor the performance and availability of distributed operating systems and applications. This two day course provides the skills necessary to administer IBM Tivoli Monitoring and the Tivoli Enterprise Portal. The Tivoli Enterprise Portal is used by IBM Tivoli Monitoring, IBM Tivoli Composite Application Manager, and OMEGAMON XE and DE to manage your enterprise environment. In this course, you will learn about creating queries, links, situations (enterprise and private), autonomous agents, policies, agentless monitoring, and event integration with Netcool®/OMNIbus and Tivoli Enterprise Console. This training class is provided in a classroom environment with multiple opportunities for hands-on lab practice. The scenarios used in this class are based on using IBM Tivoli Monitoring version 6.3.

The lab environment for this course uses the SUSE Linux 11.1 platform.

This course is also available as self-paced virtual classroom (SPVC) course TOS63G.

For information about other related Cloud & Smarter Infrastructure courses, visit the Cloud & Smarter Infrastructure education training paths website:

ibm.com/software/software/tivoli/education/

Details	
Delivery method	Classroom or instructor-led online (ILO) or self-paced (SPVC)
Course level	ERC 1.0 This course is a new course. This course is an update of the following previous courses: TM062: IBM Tivoli Monitoring 6.2.3 Advanced Administration ERC1.0
Product and version	IBM Tivoli Monitoring 6.3
Duration	2 days
Skill level	Intermediate

About the student

This intermediate level course is for Tivoli administrators, technical support personnel, and IT services specialists who are responsible for implementing and administering IBM Tivoli Monitoring.

Before taking this course, make sure that you have the following skills:

- Working knowledge of Linux
- Experience with event management
- Experience with the Tivoli Enterprise Portal
- Working knowledge of Netcool/OMNIbus (suggested but not required)

Course objectives



Course objectives

When you complete this course, you can perform the following tasks:

- Create and manage queries to access data
- Construct links for easy portal navigation
- Build enterprise and private situation events
- Configure and manage agents running in autonomous mode
- Create and manage workflow automation with policies
- Create agentless monitoring
- Use Agent Management Services
- Integrate events with Netcool/OMNIbus
- Use commands effectively

© Copyright IBM Corporation 2013

Course agenda

The course contains the following units:

1. [Queries](#)

This unit introduces the course and covers how to manage the queries to both Tivoli Enterprise Monitoring Agents and ODBC data sources.

These exercises cover how to create and manage queries. Queries are used to retrieve data for views that are contained in workspaces. Typically, a product-provided query can be used for a view. If a product-provided query does not exist for the data that you want to retrieve, you can create the query by using the Query editor.

2. [Advanced link topics](#)

This unit covers the creation of links to simplify navigating the Tivoli Enterprise Portal.

In the prerequisite class, you learned how to create simple links. In this exercise, you learn how to create advanced links by using dynamic and relative links.

3. [Monitoring your enterprise using advanced situation techniques](#)

This unit covers how to monitor the enterprise through expert advice, embedded situation events, correlated situation events, model situations, and dynamic thresholds.

In the prerequisite class, you learned how to create simple situations. In this set of exercises, you learn how to use model situations, embedded situations, correlated situations, and dynamic thresholds.

4. [Agent autonomy](#)

In these exercises, you learn how to send SNMP traps to event receivers without a connection to a monitoring server. This feature has implications in the architecture of your solution, the scalability of IBM Tivoli Monitoring, and the way that the monitoring agents are configured.

These exercises show you the basics of this function.

5. [Working with policies](#)

Agent Management Services are used to monitor the availability of agents and respond automatically if an agent becomes unhealthy or exits unexpectedly.

When you use situation actions, no immediate feedback is available about the success or failure of the system command. The only indication of whether the command is successful is that the situation event is closed when the situation is re-evaluated. You typically need to either receive immediate notification after a problem is solved or issue additional commands. In the exercises for this unit, you issue commands to create a policy to use workflow automation.

6. [Agentless monitoring](#)

This unit covers how to use agentless monitoring in IBM Tivoli Monitoring.

Using agentless monitoring, you can monitor an IT environment from a small set of centralized servers. This type of monitoring is useful for the following reasons:

7. [Agent Management Services](#)

Agent Management Services are used to monitor the availability of agents and respond automatically if an agent becomes unhealthy or exits unexpectedly.

Agent Management Services overview

8. [Managing situation event integration](#)

In these exercises, you configure the Tivoli Enterprise Portal, Tivoli Enterprise Portal Server, and the Tivoli Enterprise Monitoring Server to forward situation events to Netcool/OMNibus. You also customize the Event Integration Facility (EIF) slots that are sent in the situation events.

9. [Command-line interfaces](#)

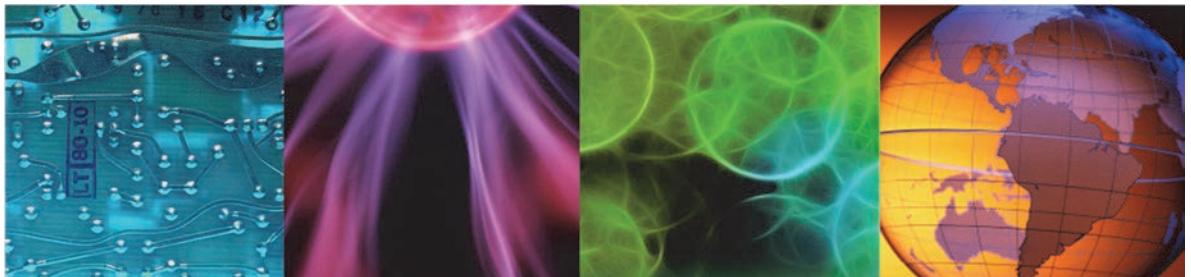
The command-line interfaces (CLIs) are an alternative to using functions in the Tivoli Enterprise Portal user interface. Commands are especially useful when a function is repeated many times. You can use these commands when you deploy the same agent to multiple systems, or you can include them inside a script that is repeatedly run.



1 Queries



1 Queries



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit introduces the course and covers how to manage the queries to both Tivoli Enterprise Monitoring Agents and ODBC data sources.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises



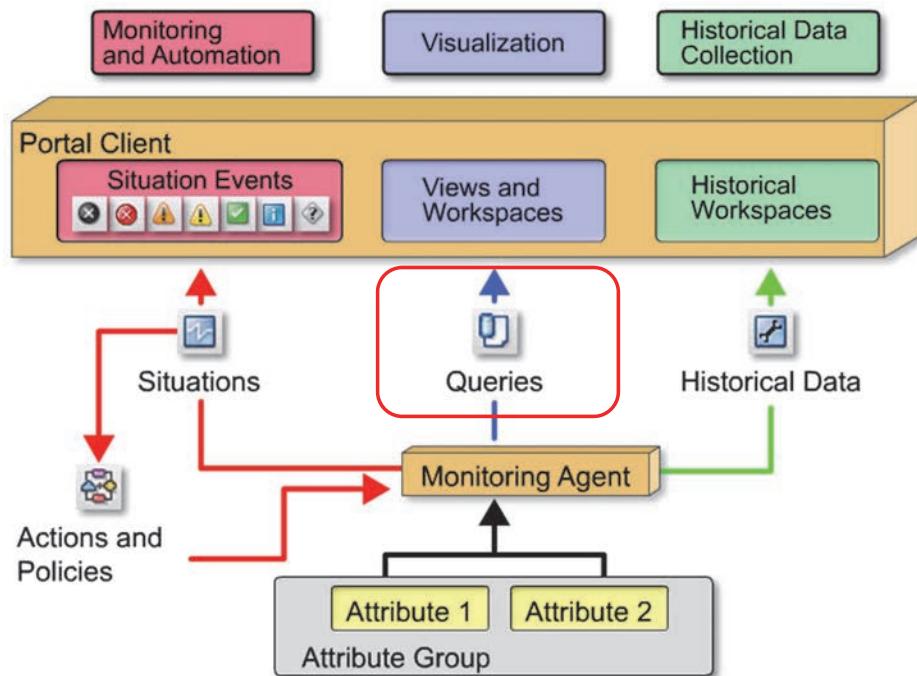
Objectives

When you complete this unit, you can perform the following tasks:

- Create and manage complex queries against Tivoli Enterprise Monitoring Agents
- Build custom queries against DB2 and ODBC data sources to include more monitoring data in your solution

Lesson 1. Queries

Lesson 1: Queries



© Copyright IBM Corporation 2013

3

This diagram shows the relationship between queries, monitoring agents, and attributes, and how they relate to views and workspaces.

What this lesson is about

In this lesson, you learn how to create and manage queries to retrieve and manipulate data from monitoring agents.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Access the query editor.
- Build new queries.
- Use filters in a query.
- Group data retrieved from a query in multiple ways.

Query overview

- A **query** defines what data is requested from a Tivoli Enterprise Monitoring Agent or other data source when a workspace is accessed or refreshed
- Queries are used only for table views and chart views
- Queries are driven by demand from the portal client
 - Initially accessed
 - Workspace refresh

© Copyright IBM Corporation 2013

4

Query overview

All data that is seen in table and chart views is based on queries. Through those queries, you define what data is returned from a data source whenever a workspace is accessed or refreshed. Most agents come with a set of predefined queries that are used to retrieve data from the agent. This data is shown in product-provided workspaces within the Navigator Physical view. In this lesson, you examine how to edit queries and create new ones against various data sources.

Product-provided queries

- IBM provides a set of predefined queries for retrieving all data collected by Tivoli Enterprise Monitoring Agents
- You can control what data is available in the portal client by creating your own queries, which you can model from the product-provided queries
- Queries can include data from Tivoli Enterprise Monitoring Agents or other ODBC data sources
- The portal server uses DB2 UDB, Microsoft SQL Server, or Derby; you can also write queries against those tables
- When you issue a query, you filter at the source

Product-provided queries cannot be modified

© Copyright IBM Corporation 2013

5

Product-provided queries

All Tivoli Enterprise Monitoring Agents come with a set of product-provided queries, which are used to gather all data that each agent collects. Those queries are created by the product developers and cannot be modified. However, you can use the product-provided queries as models to create your own.

Managing queries

- **Managing** queries means adding, modifying, or deleting queries
- The term also includes managing the authority for users to access and modify queries
- Queries are global, and are shared by all users
- Changes to a query affect all views in the enterprise that are based on it, even if the views are used by other users
 - Query changes must be managed carefully

© Copyright IBM Corporation 2013

6

Managing queries

Queries are not saved by user ID.

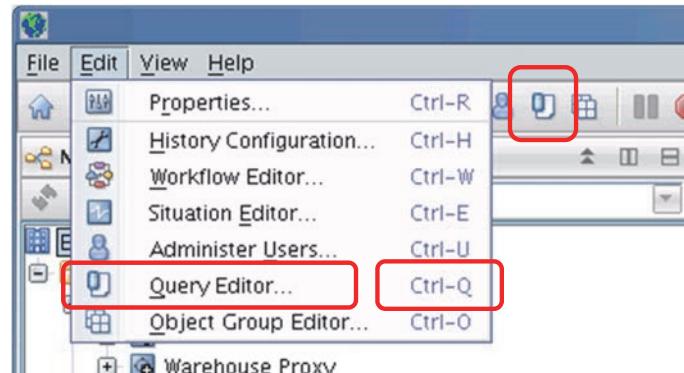


Note: Whenever you change a query, it affects all workspace views that are based on that query. All users that are connected to the same Tivoli Enterprise Portal Server see those changes. Therefore, every user with the permission to modify or create queries can potentially modify the views of other users. For that reason, you must carefully determine who you want to grant access to editing and creating queries.

You can remove the permission to modify, create, or add queries in the user authorities.

Accessing the Query editor

- From the toolbar
- By pressing Ctrl+Q
- From the **Edit** menu
- From the properties page for a view



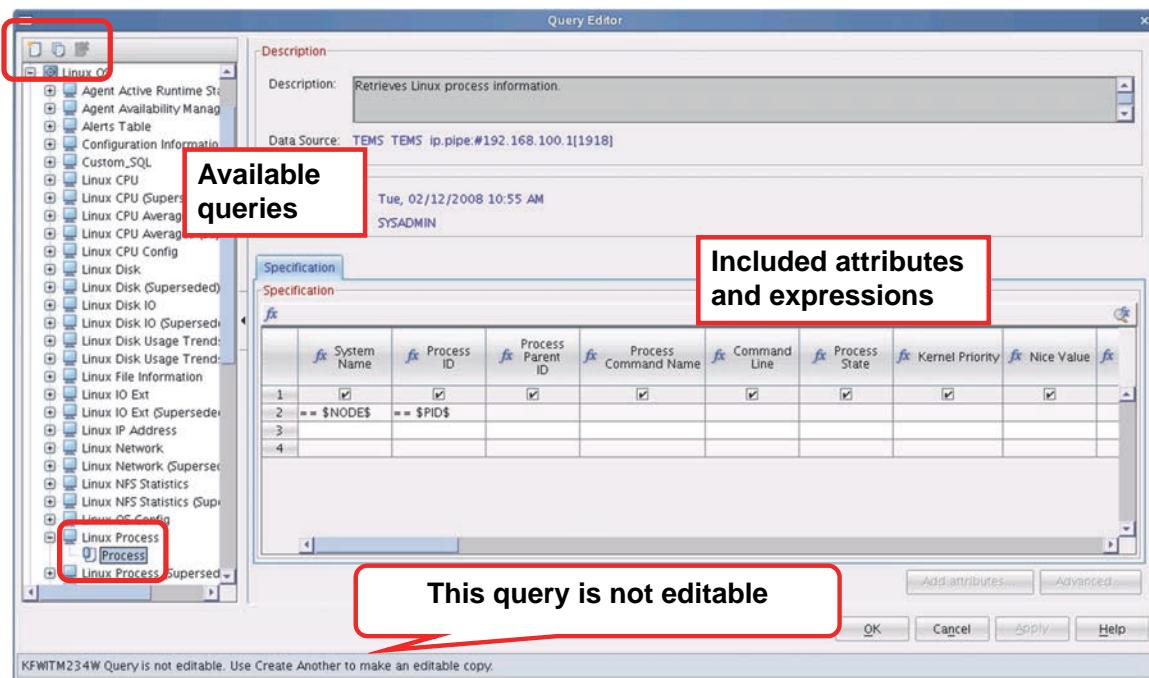
© Copyright IBM Corporation 2013

7

Accessing the Query editor

To access the Query editor, click the toolbar icon, click **Edit > Query**, or press Ctrl+Q. The Query editor can be accessed by anyone with the authority to modify or view queries. If the permissions are set to **view** queries only, the editing buttons are not available, and you cannot make changes to the query specifications.

The Query editor



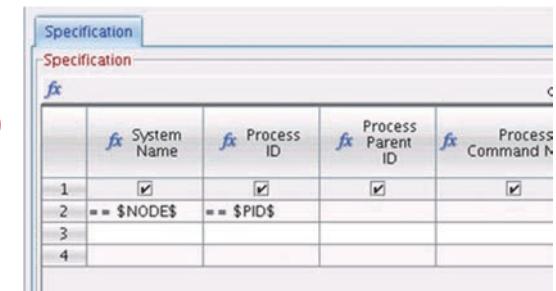
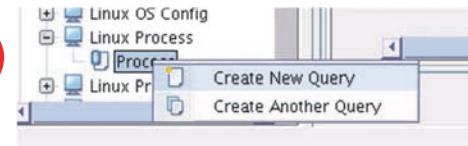
1-8

The Query editor

The list of available queries depends on what product application support was loaded in your Tivoli Enterprise Portal Server. You can access the editor directly from the main toolbar and menu bar in Tivoli Enterprise Portal. You can also access the editor from the view properties. To assign a new query to a view, you open the Query editor.

Creating another query

- The Create Another Query option copies an existing query
- Do not modify queries that can be used by others
- Steps to create another query:
 - Highlight the query that you want to copy
 - Click the **Create Another Query** icon, or right-click the query and select **Create Another Query**
 - Enter a new name
 - Enter the query specifications



© Copyright IBM Corporation 2013

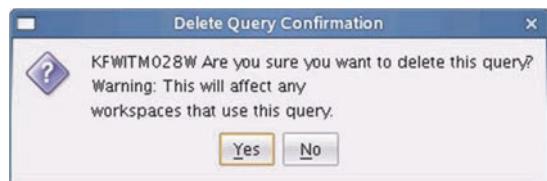
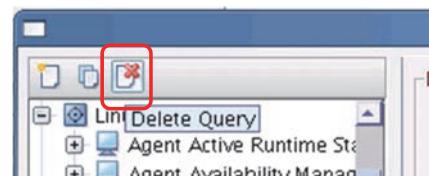
9

Creating another query

Instead of clicking the icon from the toolbar to copy a query, you can right-click the query and select **Create Another Query**. You must specify a new name and select the columns you want to include in your new query. You can apply functions to filter data at the query source.

Deleting a query

- When deleting a query, be aware that all enterprise views that use it are affected, including views of other users
- To delete a query, select the query and click the Delete Query icon
- A Delete Query Confirmation window opens, indicating that this action affects any workspaces that use this query

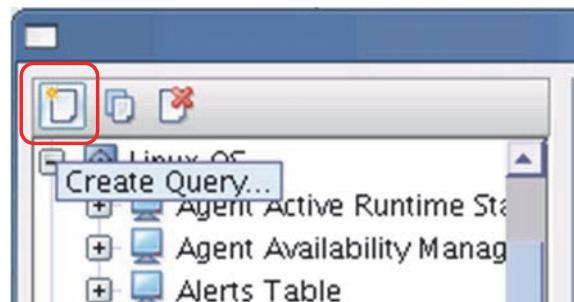


Deleting a query

When deleting queries, remember that the deletion affects all views and workspaces that use this query as a data repository. Queries are not saved with workspaces and are independent of user IDs.

Creating a query

- If possible, copy existing queries
- It might be necessary to create queries against new data sources
- You can create queries against agents, or against any database data source using custom SQL



© Copyright IBM Corporation 2013

11

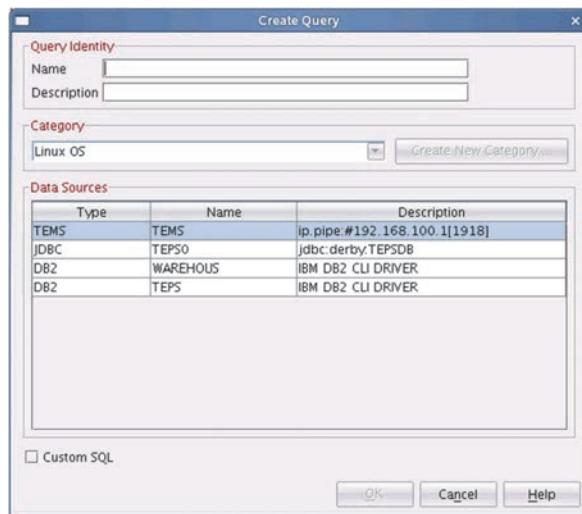
Creating a query

Instead of creating new queries, you can copy existing ones and modify them. Copying prevents problems, particularly when you are learning to creating your own queries.

Possible query data sources

- You can write queries against all data sources listed in the Query editor
- Select the **TEMS** data source to write custom queries against agents
- Select any ODBC data source to include data from that data source

You can create your own custom data sources



© Copyright IBM Corporation 2013

12

Possible query data sources

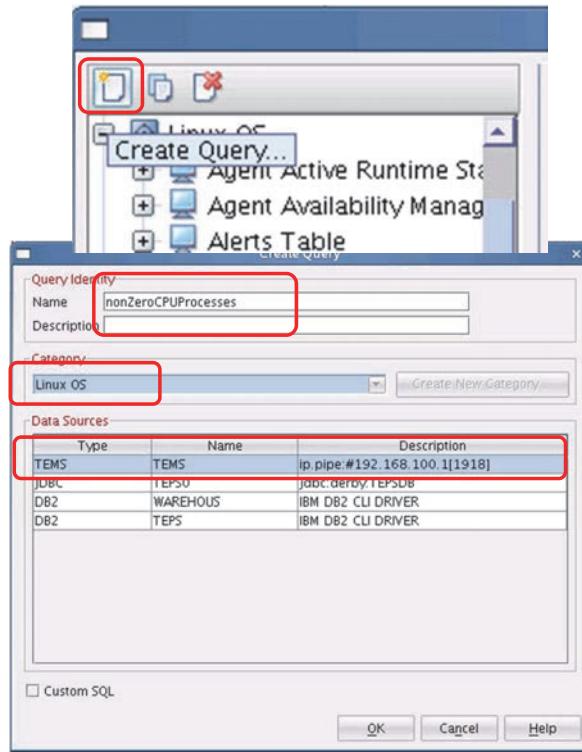
When creating queries against the Tivoli Enterprise Monitoring Server data source, you are generating a query against agents.

For queries you want to write against an ODBC data source, make sure that the Tivoli Enterprise Portal Server is configured to access that data source.

Default sources are the Tivoli Enterprise Portal Server data source and the Tivoli Enterprise Monitoring Server.

Creating an agent query

1. Click the Create Query icon
2. Enter a name for the new query
3. Select the **TEMS** data source
4. Choose the category you want to create the query for
5. Click the **OK** button
You are prompted to select an attribute group and individual attributes for your query



© Copyright IBM Corporation 2013

13

Creating an agent query

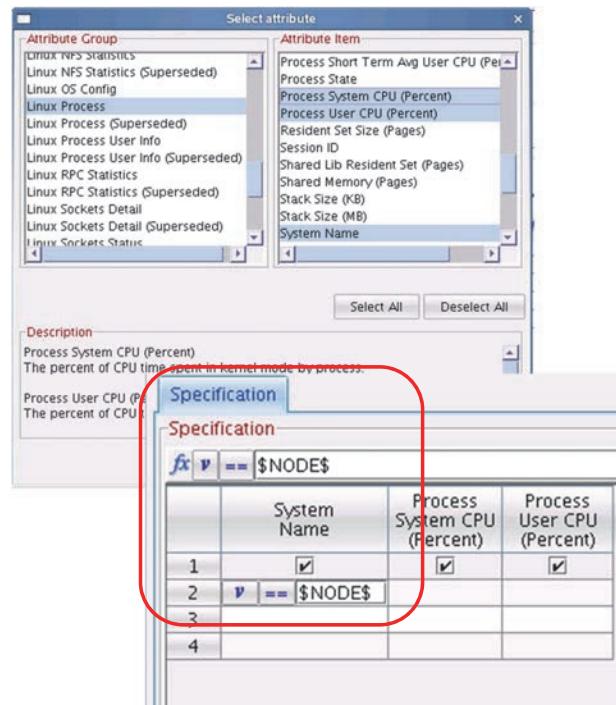
Before you start creating a query, determine the data source that you want to retrieve data from.

Follow these steps to create a query:

1. To create a query, click the **Create Query** icon.
2. Enter a query name.
3. To create a query against an agent, select **TEMS** as the data source.
4. Select a category. The category determines the type of attributes that you can include in your query and the location of the query in the Query editor. New categories cannot be created in Tivoli Enterprise Portal. (The button for this feature is not available.)
5. Click **OK**. The query is now shown in the selected category.

Creating an agent query (continued)

6. Select the attributes that you want to include in the query, including one that identifies the computer
7. Add the symbol **\$NODE\$** to identify the managed system
8. Specify any functions to prefilter data at the agent
9. Click **OK** to save your query



© Copyright IBM Corporation 2013

14

6. Select the attribute group from the Group list, select the attributes to include from the Item list, and click **OK**. After selecting one attribute, you can use Ctrl+click to select multiple attributes or Shift+click to select all attributes from the first selection to this one, or click **Select All**.
7. Add the same symbol expressions to the specification as those used in the predefined query for this attribute group. Usually, this symbol is the **\$NODE\$** symbol in the column that identifies the computer: System Name, Server Name, Originnode or something similar. If you are not sure what the expressions are, click **Apply**, open a predefined query for the attribute group that includes a symbol expression. Make note of the column name and any symbol expressions used in the specification, return to your new query, and add the symbol expression to the appropriate column. The **\$NODE\$** symbol can be used in only one row of the query.
8. You can add functions to filter out unwanted data from the query. Filters are applied by the agent.
9. Clicking **OK** saves the query and copies it to the category that represents the attribute group you selected.

Filters and functions

Use functions to filter attribute values at the data source

The screenshot shows the 'Specification' dialog box. At the top, there are tabs for 'Specification' and 'Specification'. Below the tabs, there is a toolbar with buttons for 'fx' (function), 'v' (value), and '!= 0'. The main area contains a table with four rows. The first three rows have filters applied to the 'Process User CPU (Percent)' column. Row 1 has a checkmark in the 'Process User CPU (Percent)' column. Row 2 has the value '\$NODE\$' in the 'System Name' column and '!= 0.00' in the 'Process User CPU (Percent)' column. Row 3 has a checkmark in the 'Process User CPU (Percent)' column. Row 4 is empty. A red circle highlights the 'Process User CPU (Percent)' column header and the row for node 2.

	System Name	Process System CPU (Percent)	Process User CPU (Percent)
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	$= \$NODE\$$	$\neq 0.00$	$v \neq 0$
3			<input checked="" type="checkbox"/>
4			

© Copyright IBM Corporation 2013

15

Filters and functions

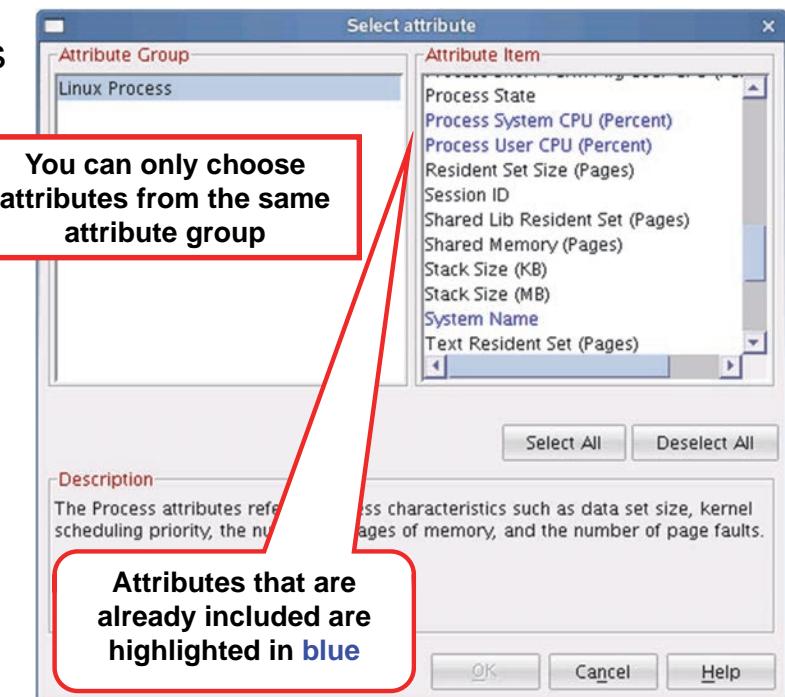
When creating or editing a query against agent attributes, you can include or exclude attributes from being retrieved when the query is issued.

You can also specify functions that filter data at the agent before the data is returned to the Tivoli Enterprise Portal client. Some attributes have a list of specific values, as seen on this slide. String functions can be set to count the group members. Numeric functions have other settings, such as sum, maximum, minimum, average.

To specify a new function, click an empty field in the Attribute column. A menu opens, which you can use to specify a function, operator, and value for that expression.

Adding attributes

You can add attributes to the query at any time by clicking the **Add attributes** button



© Copyright IBM Corporation 2013

16

Adding attributes

You can add the same attribute multiple times to specify more elaborate filter functions. You can add attributes by clicking **Add attributes**.

You can include only attributes from the same attribute group as the ones for which the query was created. Attributes that are included in the query at least once are highlighted in blue.

Deleting attributes

There are two ways of preventing attribute values from being returned by the agent

The screenshot shows the 'Specification' tab of a configuration interface. On the left, a detailed view of a column header 'Process User CPU (Percent)' is shown with a checked checkbox. A context menu is open over this header, with 'Delete' highlighted. A red callout bubble points to this option with the text 'Delete the attribute'. On the right, the main specification table lists several columns: System Name, Process System CPU (Percent), Process User CPU (Percent), Command Line, and Process ID. The first three columns have their checkboxes cleared, while the last two remain checked. A red callout bubble points to the cleared checkboxes with the text 'Exclude the attribute by clearing the check box'. Another red callout bubble points to the same cleared checkboxes with the text 'You can still specify a filter function, which is applied to the collected data record before the data is sent back to the client'.

© Copyright IBM Corporation 2013

17

Deleting attributes

Two options are available to prevent attributes from being returned from the agent:

- When you delete the column completely, the attribute value is not included in the query and is not collected.
- When you clear the check box under a column header, you can still specify a function by which all returned rows are filtered. The attribute itself is not returned to the client.

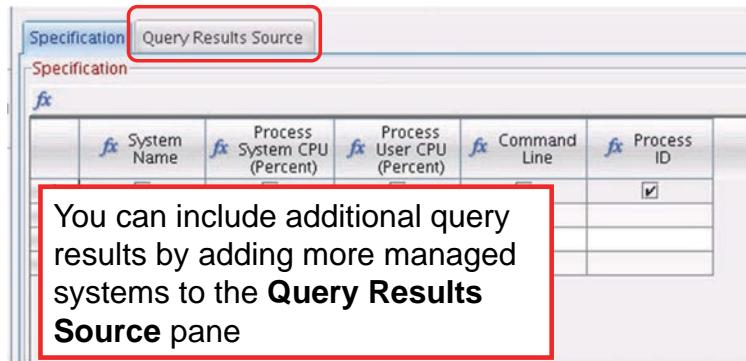
After deleting an attribute, you can add it again as described on the previous slide.

Query results source

- When you assign the query to a view in a workspace, the \$NODE\$ variable is used to substitute the managed system name

System Name	
1	<input checked="" type="checkbox"/>
2	== \$NODE\$
3	
4	

- The **Query Results Source** tab is where you define which source or sources are queried when the query is issued



© Copyright IBM Corporation 2013

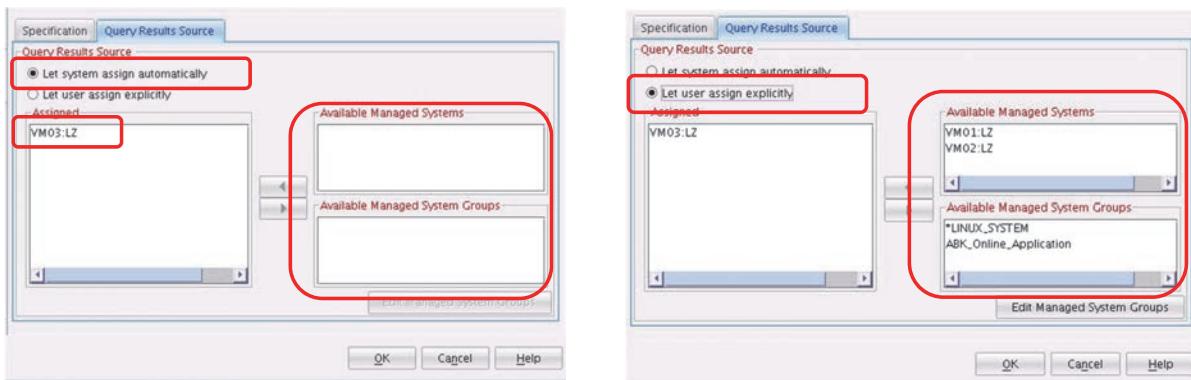
18

Query results source

Most predefined queries include a ==\$NODE\$ function that is used to substitute managed system names from which data is retrieved. For queries against agents, you must specify \$NODE\$ for the queries to work. Use the **Query Results Source** tab to control which managed systems you want to collect data from.

Query results source (continued)

By default, the query results source is defined by the Navigator item for which you accessed the Query editor in the view properties



You can include more query results by adding more managed systems to the **Query Results Source** pane
Select **Let user assign explicitly** to enable this feature

© Copyright IBM Corporation 2013

19

You can let the system determine which managed system to collect data from. Data is collected from any managed system that is assigned to the Navigator item to which the workspace belongs.

By selecting the **Let user assign explicitly** check box, you can include more data from other managed systems of the same type, even if the Navigator item is not assigned to them. If you create queries after accessing the Query editor from the toolbar, the **Query Results Source** tab is not visible.

Query results source (continued)

Command Line	Process ID	Process User CPU (Percent)	Process System CPU (Percent)
gnome-system-monitor	18369	0.51	0.22
/usr/X11R6/bin/X :0 -audit 0 -br -auth /var/lib/gdm/:0.Xauth...	6445	0.30	0.26
/usr/lib/vmware-tools/sbin32/vmware-guestd --background /...	3100	0.02	0.13
gnome-panel --sm-client-id default1	6829	0.01	0.31

Using the query created and running it as is against the OS Agent on VM03

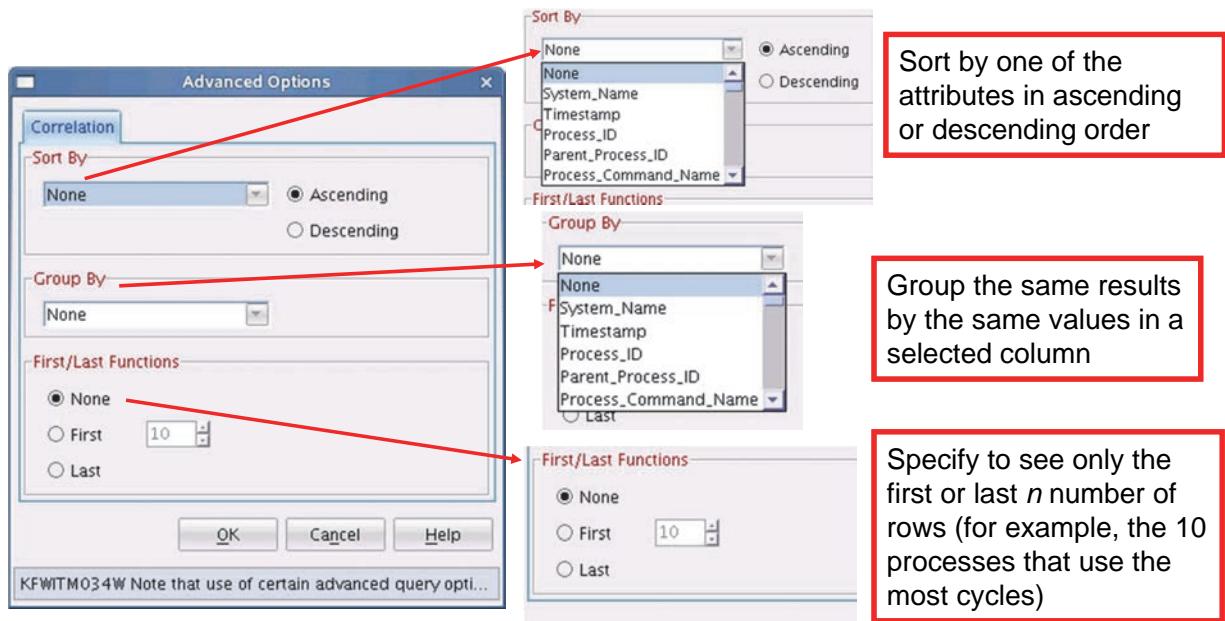
Using the query created and running it against the OS agent on VM01, VM02, and VM03 (notice that the filter on SystemName was removed in the query)

System Name	Command Line	Process ID	Process User CPU (Percent)	Process System CPU (Percent)
VM03:LZ	gnome-system-monitor	18369	0.51	0.22
VM01:LZ	/opt/IBM/ITM/jRE/ii6263/bin/java -mx96m -classpath.:./opt/I...	2028	0.31	1.24
VM03:LZ	/usr/X11R6/bin/X :0 -audit 0 -br -auth /var/lib/gdm/:0.Xauth...	6445	0.30	0.26
VM01:LZ	/usr/X11R6/bin/X :0 -audit 0 -br -auth /var/lib/gdm/:0.Xauth...	4613	0.16	0.13
VM01:LZ	/opt/IBM/ITM/jRE/ii6263/bin/java -Xms64m -Xmx256m -sho...	8615	0.12	0.90
VM01:LZ	/opt/IBM/ITM/ii6263/lw/java/bin/java -Declipse.security=Dosg...	17641	0.10	0.17
VM01:LZ	db2sys	31712	0.10	0.44
VM02:LZ	/opt/IBM/ITM/jRE/ii6263/bin/java -mx96m -classpath.:./opt/I...	12591	0.09	0.35
VM02:LZ	/usr/X11R6/bin/X :0 -audit 0 -br -auth /var/lib/gdm/:0.Xauth...	2988	0.08	0.06
VM01:LZ	/opt/IBM/ITM/ii6263/lz/bin/klzagent	2860	0.06	0.77
VM01:LZ	/opt/IBM/ITM/ii6263/lz/bin/klzagent	11525	0.05	0.73
VM01:LZ	/usr/lib/vmware-tools/sbin32/vmware-guestd --background /...	2667	0.02	0.17
VM01:LZ	/opt/IBM/ITM/ii6263/cq/bin/KfwServices -	17482	0.02	0.05
VM03:LZ	/usr/lib/vmware-tools/sbin32/vmware-guestd --background /...	3100	0.02	0.13
VM01:LZ	/opt/IBM/ITM/ii6263/sy/bin/ksy610 console	22334	0.01	0.02
VM02:LZ	zmd /usr/lib/zmd/zmd.exe --sleep 82199	2780	0.01	0.30
VM03:LZ	gnome-panel --sm-client-id default1	6829	0.01	0.30

Queries can be run against multiple agents with a single request. However, in a large environment, running queries this way affects performance.

Advanced options

Decide how the results are sorted and grouped, or how many are shown



© Copyright IBM Corporation 2013

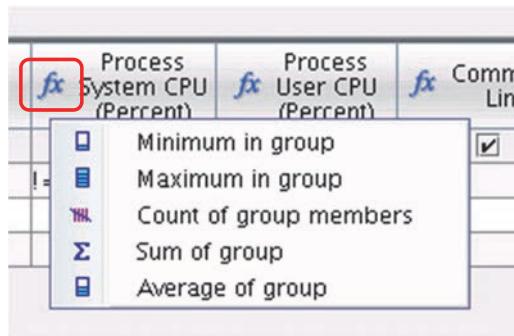
21

Advanced options

You have several advanced options to either sort query results, group them by a specific column, or show only the first or last entries of a query result.

Advanced options: Group By functions

- Using the **Group By** functions, you can return the sum, count, average, minimum, or maximum of all grouped data records
- Click the function symbol in the column header to access those options



© Copyright IBM Corporation 2013

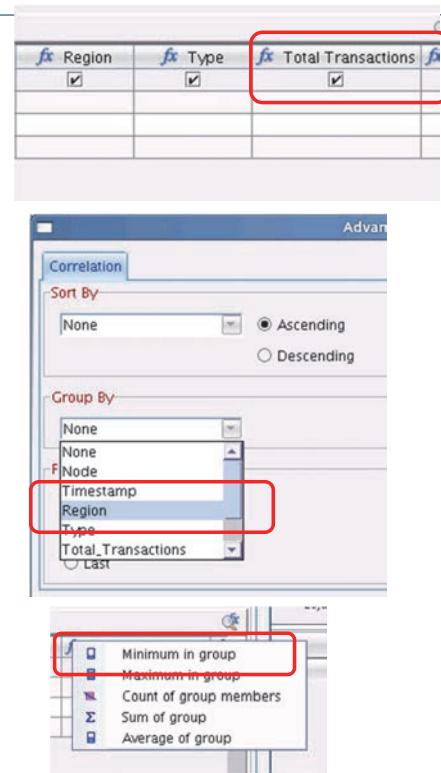
22

Advanced options: Group By functions

Use the **Group By** advanced option to specify additional functions to summarize or to count the occurrence of returned rows. You can also show the average, minimum, or maximum of values within the grouped rows of data.

Group By functions: Example

1. Create another query
2. Enter a name for the new query
3. Select the attribute you want to filter (Total Transactions)
4. Select the attribute to run the function against (Region)
5. Select the function (Minimum)
6. Save the query
7. Assign the query to the view



© Copyright IBM Corporation 2013

23

Group By functions: Example

This example illustrates the use of the **Group By** option with a column function.

Group By functions: Result

ABK Details - Region by Type - By Total Transaction

Region	Type	Total Transactions
North	ATM	6303
North	Online	6075
North	Branch	7558
North	CheckCard	12598
North	Phone	12305
South	ATM	9657
South	Online	9600
South	Branch	7170
South	CheckCard	14453
South	Phone	8741
West	ATM	13636
West	Online	10764
West	Branch	7828
West	CheckCard	6290
West	Phone	10426
East	ATM	6037
East	Online	9290
East	Branch	6096
East	CheckCard	12551
East	Phone	15389
MidWest	ATM	17160
MidWest	Online	12994
MidWest	Branch	14711

Minimum Transaction by Type in Each Region

Region	Type	Total Transactions
East	Branch	6037
MidWest	Online	5250
North	ATM	6075
South	ATM	7170
West	Branch	6290

Both queries collect the same data, but the query on the right filters it by worst performance

Group By functions: Result

The difference between these two tables is that the query used for the table on the right groups all data by **Region**. It then shows the minimum transaction **Type** in each region together with its **Total Transactions** value.

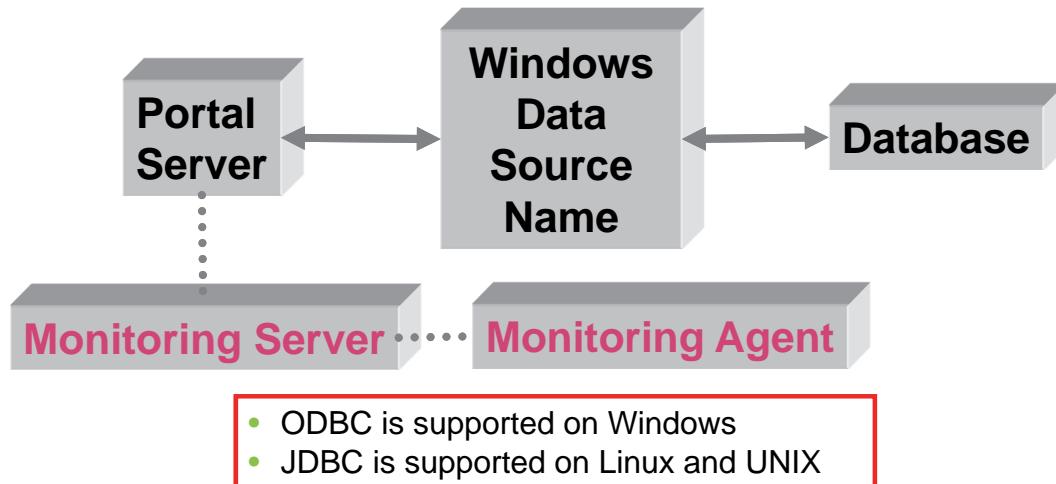


Note: Using the Advanced functions Sort, Group by, and First/Last makes the query ineligible for historical data collection.

Lesson 2. Creating a query against external databases

Lesson 2: Creating a query against external databases

- When you create a query against database data sources, the query is issued by the portal server where the data source is defined
- The database must be located on the computer that contains the portal server database



© Copyright IBM Corporation 2013

25

When you create a custom query against a database data source, the SQL statement is directly issued from the Tivoli Enterprise Portal Server to the database. Therefore, values are returned directly to Tivoli Enterprise Portal Server and are not available as attributes. You can populate views with that data, but you cannot build situations against it.

What this lesson is about

In this lesson, you learn how to create and manage queries to retrieve and manipulate data from external databases.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Retrieve data from ODBC data sources.
- Retrieve data from JDBC data sources.

Creating a query against ODBC data: Steps

1. Determine the data format
2. Create the data source
3. Configure the portal server
4. Create the custom query
5. Specify the SELECT statement
6. Test the new query

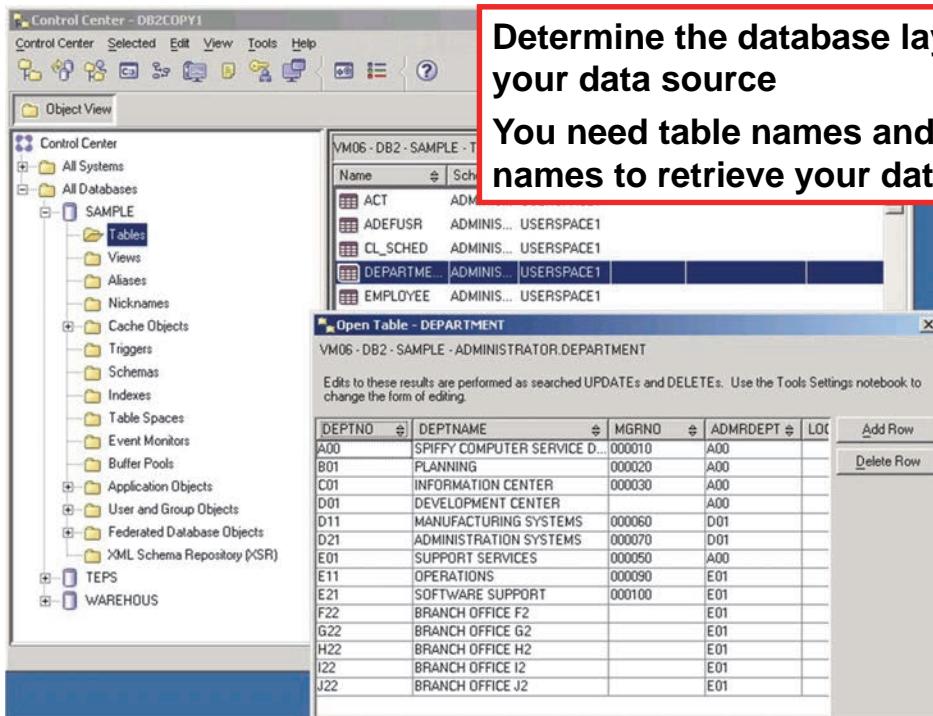
© Copyright IBM Corporation 2013

26

Creating a query against ODBC data: Steps

Several steps are required to create custom queries against ODBC data sources. You must perform some of these steps before defining the query itself.

Determining the data format (custom database)



Determine the database layout of your data source
You need table names and column names to retrieve your data

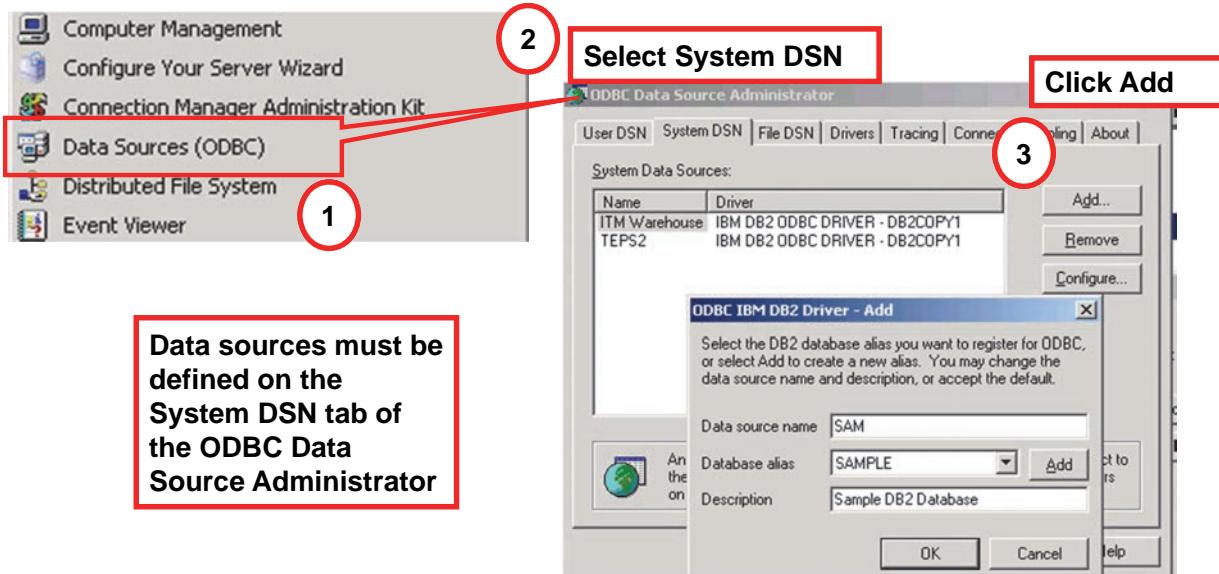
© Copyright IBM Corporation 2013

27

Determining the data format (custom database)

For any ODBC database, it is important to verify the database name and the table names. For more elaborate queries, you might also need the names of specific columns.

Creating the ODBC data source in Windows



© Copyright IBM Corporation 2013

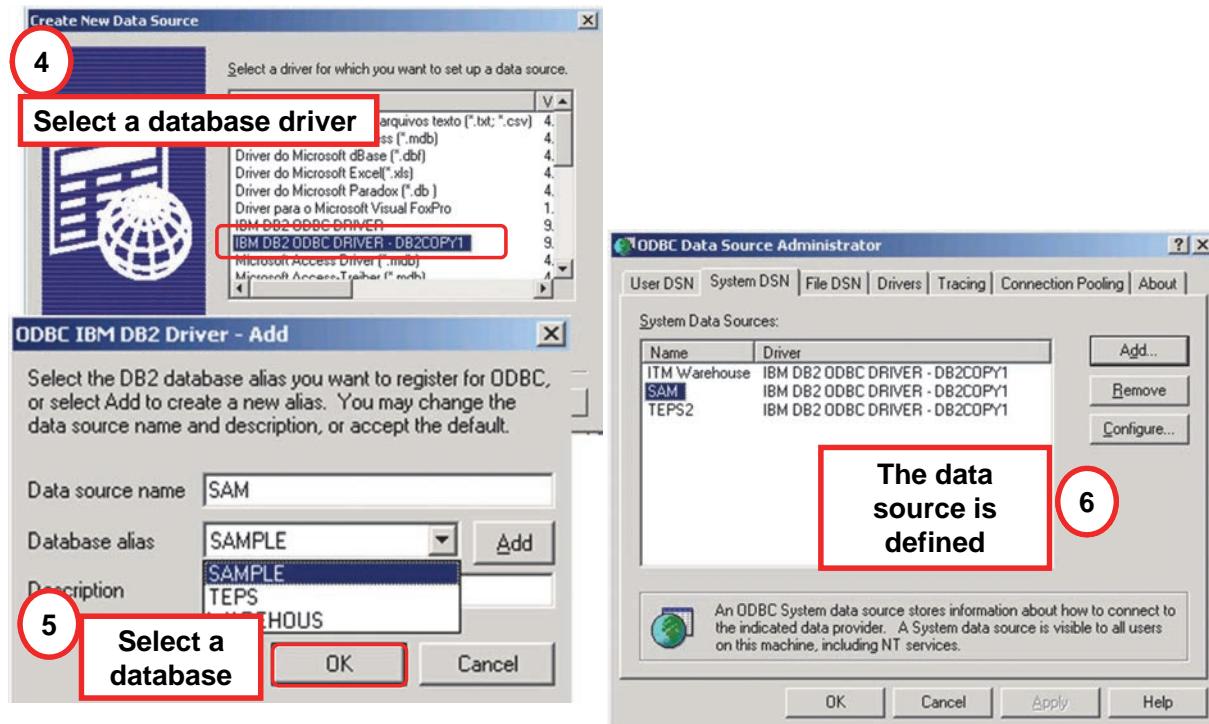
28

Creating the ODBC data source in Windows

You must register your custom data source with Windows as a System DSN. Access the ODBC Data Source Administrator under Windows through **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**.

The data source contains the driver to access the specific database. TEPS2 and ITM Warehouse are added automatically during the installation of Tivoli Monitoring. You must specify other data sources separately.

Creating the ODBC data source in Windows (continued)

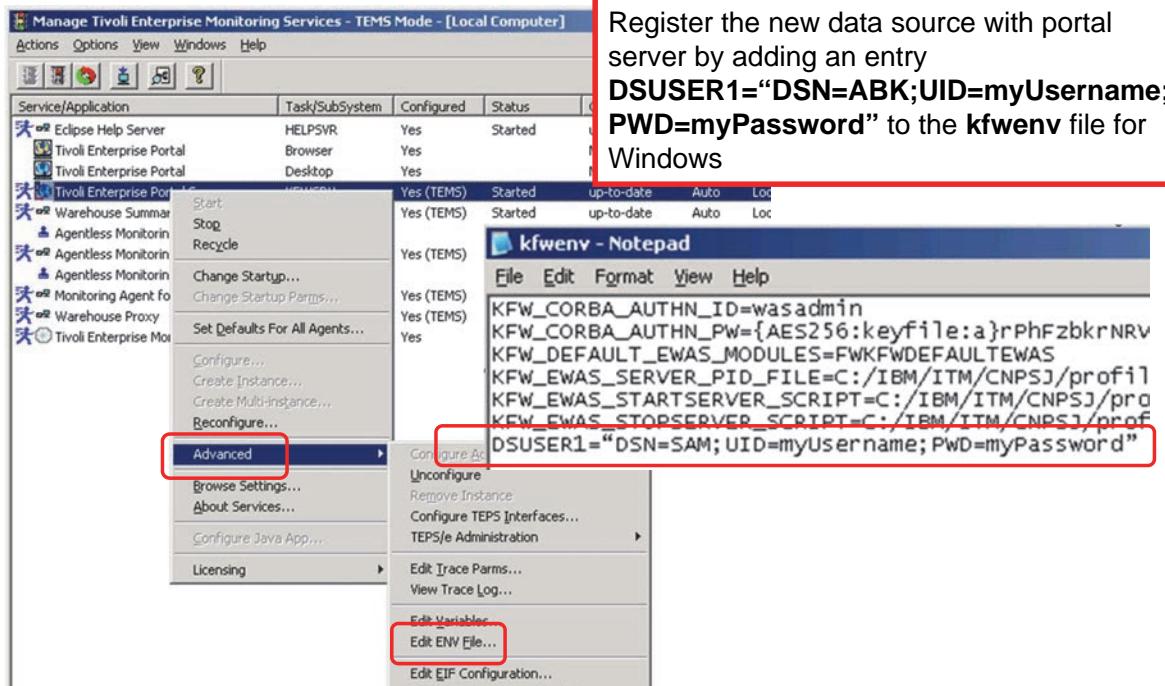


© Copyright IBM Corporation 2013

29

When the data source is added, you must assign a database to it.

Configuring the portal server



© Copyright IBM Corporation 2013

30

Configuring the portal server

The next step is to register the data source with the Tivoli Enterprise Portal Server. Access the environment file **kfwenv** from the **Tivoli Enterprise Portal Server** entry in Manage Tivoli Enterprise Monitoring Services.

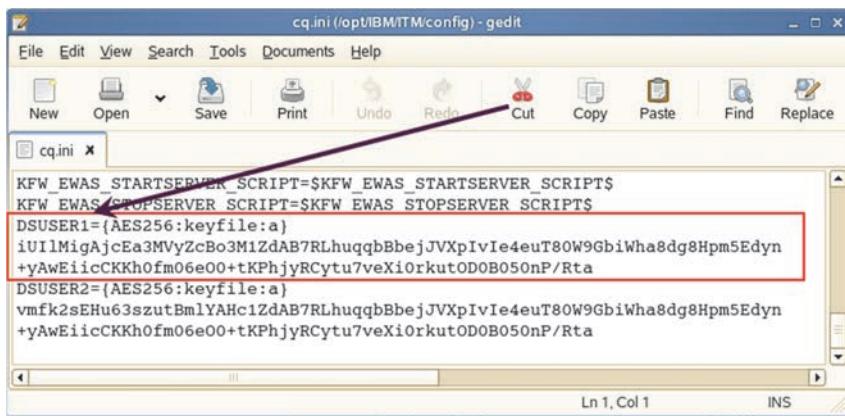
Configure the data source in the environment file by adding the entry
DSUSER1="DSN=ABK;UID=myUsername".

You can add more data sources in new columns by incrementing the DSUSER#. Adding data sources requires recycling the portal server. After you recycle the portal server, the data source is available in the Query editor.

Configuring the portal server using tacmd

The command line **tacmd configurePortalServer** can also be used to add data sources to the portal server

The advantage of this approach is that the data source, user ID, and password are encrypted in the **kfwenv** or **cq.ini** file



© Copyright IBM Corporation 2013

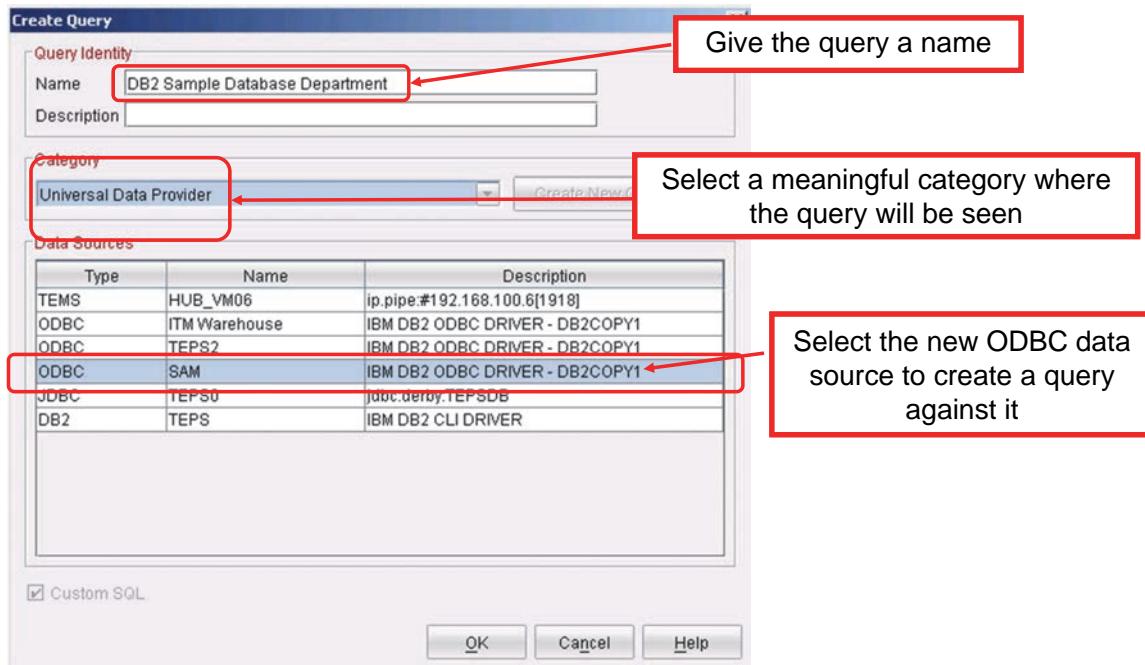
31

Configuring the portal server using tacmd

Use the **tacmd configurePortalServer** command to add security for your database access, because the user ID and password are encrypted.

Creating the custom query: Windows

The new data source is visible in the Query editor



© Copyright IBM Corporation 2013

32

Creating the custom query: Windows

After recycling the Tivoli Enterprise Portal Server, you can access the Tivoli Enterprise Portal and start creating a query. You can now see the new data source in the Query editor when creating a query.

You can see that the **Custom SQL** check box at the lower left corner of the editor is unavailable and preselected. You must create a custom SELECT statement.

Adding a data source on Linux and UNIX

- ODBC is not supported directly from the Tivoli Enterprise Portal Server in Linux and UNIX
- JDBC is supported in Linux and UNIX
- You can also add a DB2 data source by editing **cq.ini** and creating a **DSUSER1** entry:

```
DSUSER1=DSN=DatabaseName;UID=DatabaseInstanceOwner;PWD  
=password;CONNECTION_TYPE=DB2;CONNECTION_LIMIT=100
```

- For example:

The database name is not the same as the data source name
Be sure to specify the database name

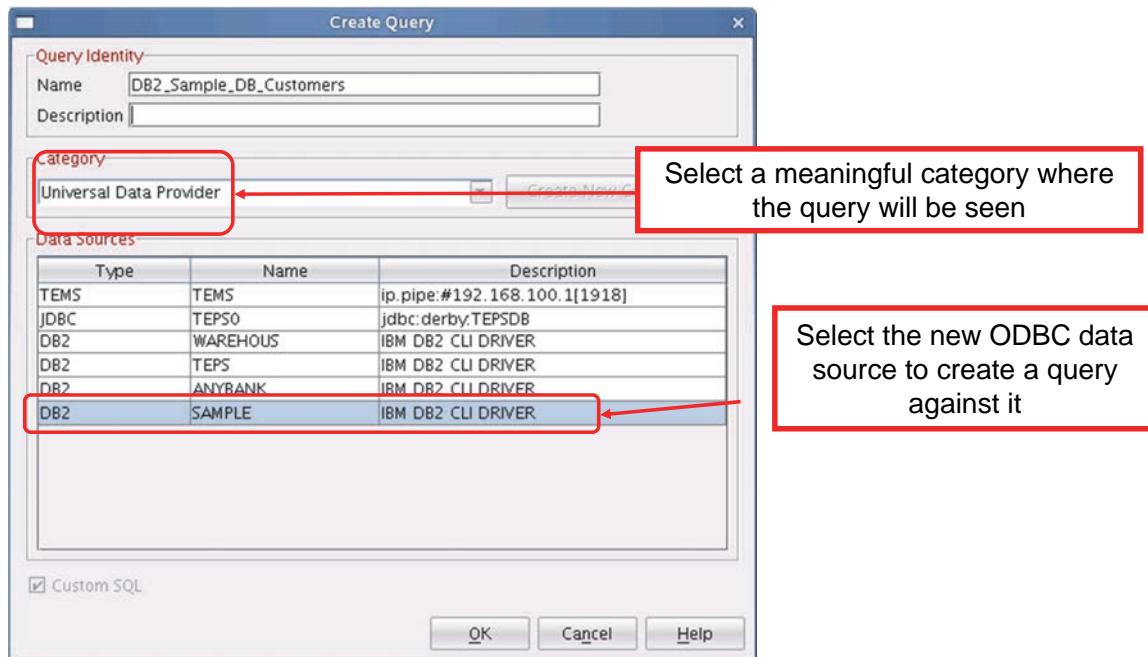
```
DSUSER1=DSN=SAMPLE;UID=db2inst1;PWD=object00;CONNECTION_TYPE=DB2;CONNECTION_LIMIT=100
```

Adding a data source on Linux and UNIX

Even though ODBC is only supported on Windows, you can access the database directly by updating the DSUSERx entries in a similar way.

Creating the custom query: Linux and UNIX

The new data source is visible in the Query editor



© Copyright IBM Corporation 2013

34

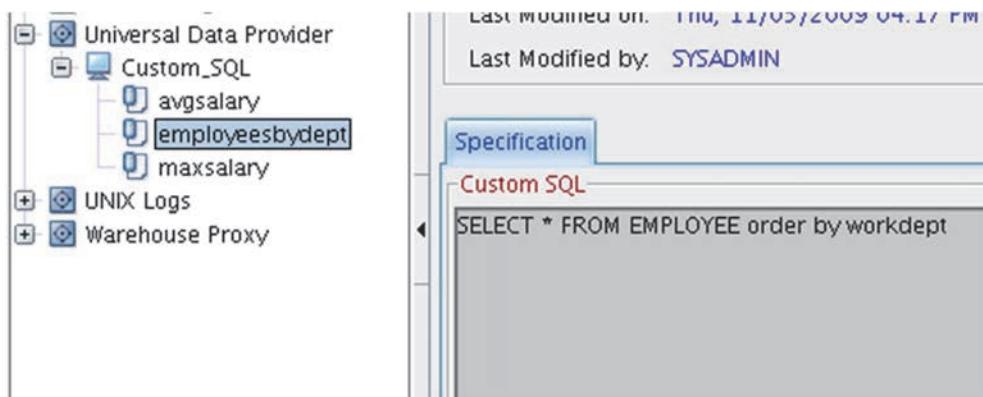
Creating the custom query: Linux and UNIX

After recycling the Tivoli Enterprise Portal Server, you can access the Tivoli Enterprise Portal and start creating a query. You can now see the new data source in the Query editor when you create a query.

You can see that the **Custom SQL** check box at the lower left corner of the editor is unavailable and preselected. You can create a custom SELECT statement.

Specifying the SELECT statement

Enter the query SQL statement



© Copyright IBM Corporation 2013

35

Specifying the SELECT statement

On the **Specification** tab, you can now enter your SQL statement. You can use any SELECT statement that the specific database can process. Only SELECT statements are supported, not UPDATE or INSERT.

Testing the new query

Test the query by building views

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	E
000010	CHRISTINE	I	HAAS	A00	3978	1995-01-01	PRES	
000110	VINCENZO	G	LUCCHESI	A00	3490	1988-05-16	SALESREP	
000120	SEAN		O'CONNELL	A00	2167	1993-12-05	CLERK	
200010	DIAN	J	HEMMINGER	A00	3978	1995-01-01	SALESREP	
200120	GREG		ORLANDO	A00	2167	2002-05-05	CLERK	
000020	MICHAEL	L	THOMPSON	B01	3476	2003-10-10	MANAGER	
000030	SALLY	A	KWAN	C01	4738	2005-04-05	MANAGER	
000130	DELORES	M	QUINTANA	C01	4578	2001-07-28	ANALYST	
000140	HEATHER	A	NICHOLLS	C01	1793	2006-12-15	ANALYST	
200140	KIM	N	NATZ	C01	1793	2006-12-15	ANALYST	
200220	REBA	K	JOHN	D11	0672	2005-08-29	DESIGNER	
200170	KIYOSHI		YAMAMOTO	D11	2890	2005-09-15	DESIGNER	
000220	JENNIFER	K	LUTZ	D11	0672	1998-08-29	DESIGNER	
000210	WILLIAM	T	JONES	D11	0942	1998-04-11	DESIGNER	
000200	DAVID		BROWN	D11	4501	2002-03-03	DESIGNER	

© Copyright IBM Corporation 2013

36

Testing the new query

You can now use the query as the basis for a workspace view. Ideally, you start with a table view to see the detailed data, and you make sure that what you see is what you intended to see. You can then use other view types to present the data in a different way.

Performance considerations

- Queries can have an impact on the performance of the Tivoli Enterprise Portal
- You can improve performance by limiting the amount of data that is collected
 - Select only columns needed for the view
 - Prefilter
- If one query can supply data for all views within a single workspace, that query is only issued once
- You can then filter the data for each view by using the **View properties > Filters** tab
 - Collects all columns
 - Provides postfiltering

Performance considerations

Queries can improve the performance of retrieving data in an effective way. By limiting the information requested from the agent, you can prefilter many rows to return only data necessary for views within a workspace.

Another rule is that if a workspace has multiple views using the same query, the query is issued only one time. You must decide whether you want to create one query for all views within a workspace, or one query for each view.

Student exercise environment

SUSE Linux ES R11

Host Name: **VM01**

Monitoring Infrastructure:

- Hub Monitoring Server
- Portal Server and database
- Portal desktop client
- Monitoring Agent for Linux OS
- Tivoli Log File agent
- IBM Tivoli Agent Builder agent
- Data Warehouse Proxy agent
- Summarization and Pruning agent

Other components:

- Apache web server
- AnyBank online application

SUSE Linux ES R11

Host Name: **VM02**

Monitoring Infrastructure:

- Remote Monitoring Server
- Monitoring Agent for Linux OS
- Tivoli Log File agent
- Java Web Start client

Other components:

- Apache web server

SUSE Linux ES R11

Host Name: **VM04**

Monitoring Infrastructure:

- Monitoring Agent for Linux OS

Other components:

- Apache web server

SUSE Linux ES R11

Host Name: **VM05**

© Copyright IBM Corporation 2013

38

Monitoring infrastructure

This slide presents the four systems you have available during your exercises. All the systems are SUSE Linux systems, and are running an Apache web server.

VM01 serves as the primary system, hosting the following items:

- Tivoli Monitoring Services infrastructure components
- An Apache web server
- An IBM DB2® database containing banking business data

A system named VM02 serves as a remote Tivoli Enterprise Monitoring Server, connected to the hub Tivoli Enterprise Monitoring Server on VM01. The two Tivoli Enterprise Monitoring Agents connects to the remote Tivoli Enterprise Monitoring Server on VM02.

VM04 contains a Netcool/OMNIbus installation, which can be used to receive SNMP traps and forwarded situation events from VM01. It also contains two Tivoli Enterprise Monitoring Agents that connect to the Tivoli Enterprise Monitoring Server on VM02.

VM05 contains no Tivoli code on it at all. It demonstrates the agentless monitoring topics covered in this course.

Student exercise



© Copyright IBM Corporation 2013

39

Student exercise

Open your Student Exercises book and perform the exercises for this unit.

Review questions

1. What are queries used for in the Tivoli Enterprise Portal?
2. Workspaces are only seen by the user who creates the workspace, unless the workspace is published. Is the same true for queries?
3. What is the difference between filtering the data in a view using the **Filters** tab instead of in the query itself?

Review answers

1. What are queries used for in the Tivoli Enterprise Portal?

The data for a table, chart, or relational table-based topology view is chosen by the query it uses. The query specifies the attributes to include in the view.

2. Workspaces are only seen by the user who creates the workspace, unless the workspace is published. Is the same true for queries?

No. Queries are global and are shared by various workspaces that reference them. Use care when changing a query because it affects all workspaces that refer to it.

3. What is the difference between filtering the data in a view using the Filters tab instead of in the query itself?

*When filtering is established through the **Filters** tab, the filtering occurs after the data is retrieved from the managed system. If many rows are returned, it might take some time to retrieve all of them and complete the filtering task. If you create a query that filters the data, and then assign that query to the view, the filtering is done at the managed system. The unwanted data is not processed by the portal server, which improves performance time.*

Summary

Now that you have completed this unit, you can perform the following tasks:

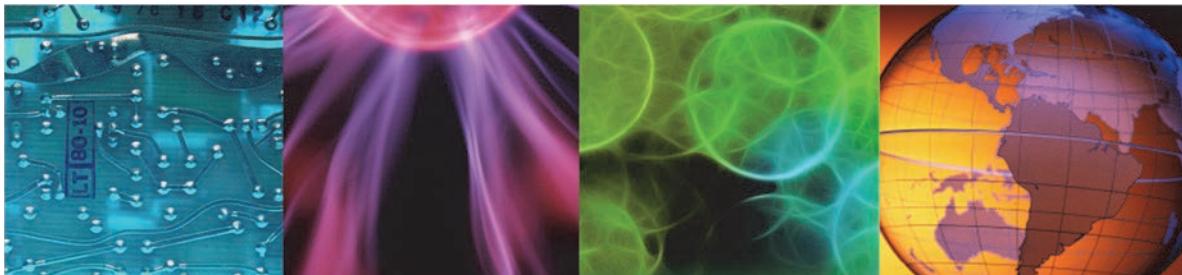
- Create and manage complex queries against Tivoli Enterprise Monitoring Agents
- Build custom queries against DB2 and ODBC data sources to include additional monitoring data in your solution



2 Advanced link topics



2 Advanced link topics



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit covers the creation of links to simplify navigating the Tivoli Enterprise Portal.

With links, you can navigate workspaces instead of using the Navigator. This feature can enhance usability for large enterprise installations with large Navigators, where locating specific workspaces can be difficult. You can also use links to access related or more detailed information about a particular entry in an overview workspace. By using advanced links, you can navigate to a workspace displaying information about just one data record.

How you check your progress

You can check your progress in the following ways:

- Review questions
 - Lab exercises
- 

Objectives

When you complete this unit, you can perform the following tasks:

- Explain the differences between simple and advanced links
- Explain the differences between the link types of absolute, relative, and dynamic
- Build relative links to provide navigation to a workspace that the user selects
- Build dynamic links to provide navigation to a workspace by substituting in workspace values automatically
- Create, manage, and issue an advanced link

Lesson 1. Understanding different types of links

Lesson 1: Understanding different types of links

Links: Overview

- By creating workspaces, views, and queries, you make sure that users can access all data that they need to monitor the enterprise in a meaningful way
- Because of the large number of Navigator items, it might be difficult to find specific workspaces of interest
- If you provide links as default navigation options, users can quickly navigate to related or more detailed information and can significantly improve usability
- Links are saved in a separate table in the portal server database

What this lesson is about

This lesson introduces the different types of links and covers some basic terminology.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe simple and advanced links.
- Explain the difference between dynamic, absolute, and relative links.

Links: Description

- Links are used to navigate between workspaces
- Link targets can be dynamic, based on certain criteria
- Links are defined using a Link wizard
- To define a link, you must specify the source of the link and a target workspace

© Copyright IBM Corporation 2013

4

Links: Description

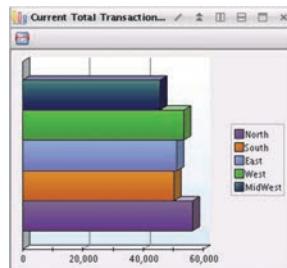
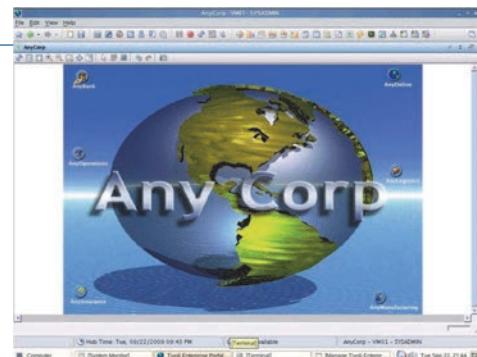
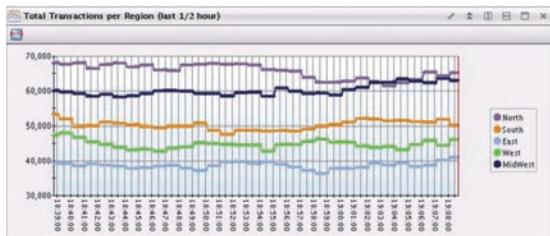
You can issue links from a right-click menu on a Navigator item, object in a graphic view, pie chart, bar chart, plot chart, or table row. You can also define default links that can be issued with a single mouse click.

For table rows and graphic view icons, you can include a link indicator to highlight the presence of a link. A link target is always another workspace.

Links: Sources

The link source can be one of the following items:

- Graphic view icon
- Row in a table
- Bar in a bar chart
- Pie in a pie chart
- Line in a plot chart
- Navigator item



File Size - Top Ten					
Path	File	Size (MB)	Owner	Group	Last Modified
/	sbin	0.008	root	root	08/25
/	tmp	0.006	root	root	11/02
/	dev	0.006	root	root	11/02
/	etc	0.006	root	root	11/02
/	lib	0.003	root	root	09/23
/	bin	0.002	root	root	09/23
/	root	0.001	root	root	11/02
/	usr	0.000	root	root	09/23
/	boot	0.000	root	root	09/23
/	media	0.000	root	root	11/02

© Copyright IBM Corporation 2013

5

Links: Sources

Links: Simple versus advanced

- A *simple* link is a shortcut from one workspace to another
 - A simple link can be used to navigate portal workspaces without using the Navigator
- An *advanced* link is used to manipulate the data that is displayed in the target workspace
 - You can use advanced links to pass values from the link source to the target workspace and dynamically filter or manipulate the data that is displayed
 - You can pass values by dynamically modifying queries to limit data that is retrieved from the agents

© Copyright IBM Corporation 2013

6

Links: Simple versus advanced

Tivoli Enterprise Portal provides various link types. The first differentiation is between a simple and an advanced link.

With a **simple** link, you can quickly navigate from one workspace to another. A simple link is like a shortcut on a web page to access another page or site. The main benefit of simple links is simplified navigation without having to locate the target in the Navigator. Using a simple link, you can also navigate between workspaces if the Navigator is hidden.

An **advanced** link builds on that capability, providing additional function that can pass values to the target workspace. The result is that users can limit or manipulate the data and context displayed in the target workspace views.

Using a simple link



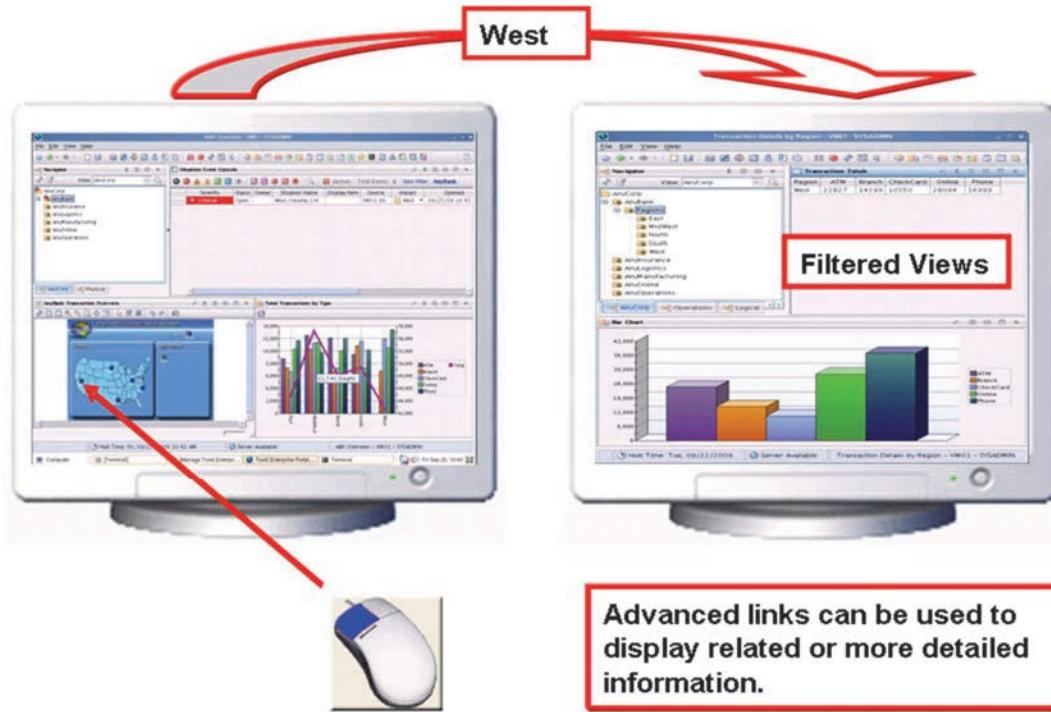
© Copyright IBM Corporation 2013

7

Using a simple link

The example illustrates the use of a simple link. It demonstrates how links can eliminate the need for displaying and using a Navigator to move between workspaces. When issuing a simple link, you navigate to the target workspace that is specified in the in the link definition.

Using an advanced link



© Copyright IBM Corporation 2013

8

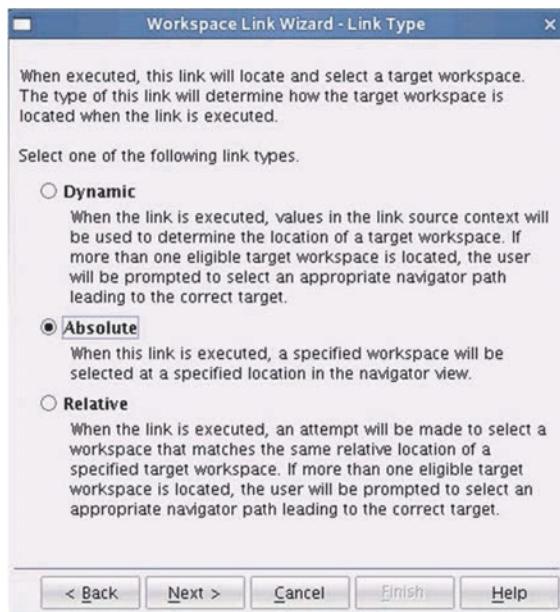
Using an advanced link

This example of an **advanced** link illustrates how data is sent from the source workspace to the target workspace. From there, you manipulate (in this case, filter) the data that is displayed in the target workspace views. When executing the link from different locations, such as different table rows, the data displayed in the target workspace is different.

In the example shown on the slide, the parameter that is passed on to the target workspace is the name of a region. This value is used by the target workspace view query to collect data only for that particular region and display it in the views.

Link types: Dynamic, absolute, and relative

- Dynamic, absolute, and relative links are options to define how to identify the target workspace during link execution
- They can be implemented as simple or advanced links



© Copyright IBM Corporation 2013

9

Link types: Dynamic, absolute, and relative

Three link types are available to define how a link target is identified during link execution. You can implement each of these link types as a **simple** or an **advanced** link. The most common type is **absolute**, which uniquely identifies the link target workspace. The workspace that is specified during link definition is always accessed during link execution. If the link target is not there or not available for a specific user, the link does not work.

Relative links can be used for environments where Navigator item names (such as host names) frequently change. Instead of expecting one particular workspace to be the link target, the target is identified by its substructure in the Navigator. Multiple workspaces can be available as possible link targets. During link execution, a user must select which workspace to access. This selection step can be a drawback for this implementation.

The **dynamic** link type option also includes navigating to a workspace that is not uniquely identified in the link definition. Different from relative links, the target is selected based on criteria from the link source during execution time. A variable is defined with the link, which is used to identify a target workspace.

An example for a dynamic link type implementation is a workspace that is product-provided with a specific agent. Every agent of the same type that is installed in the enterprise has the same workspace available. To identify an individual workspace as the link target, the link uses a value. This value can be a managed system name, or host name, to identify which of the possible target workspaces to access. Tivoli Enterprise Portal automatically performs the same step as a user selecting a target workspace when issuing a relative link.

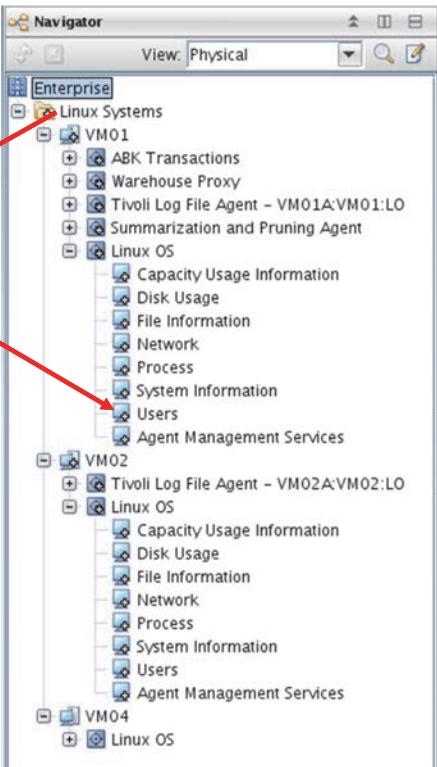
Link types: Absolute links

- Absolute links uniquely identify the link target workspace
- Most links are implemented as absolute links

Target uniquely identified in link definition

Link definition:
• Absolute link
• Link from Linux Systems to Users

Target selection during execution:
The link target is always the same
Users workspace under VM01



© Copyright IBM Corporation 2013

10

Link types: Absolute links

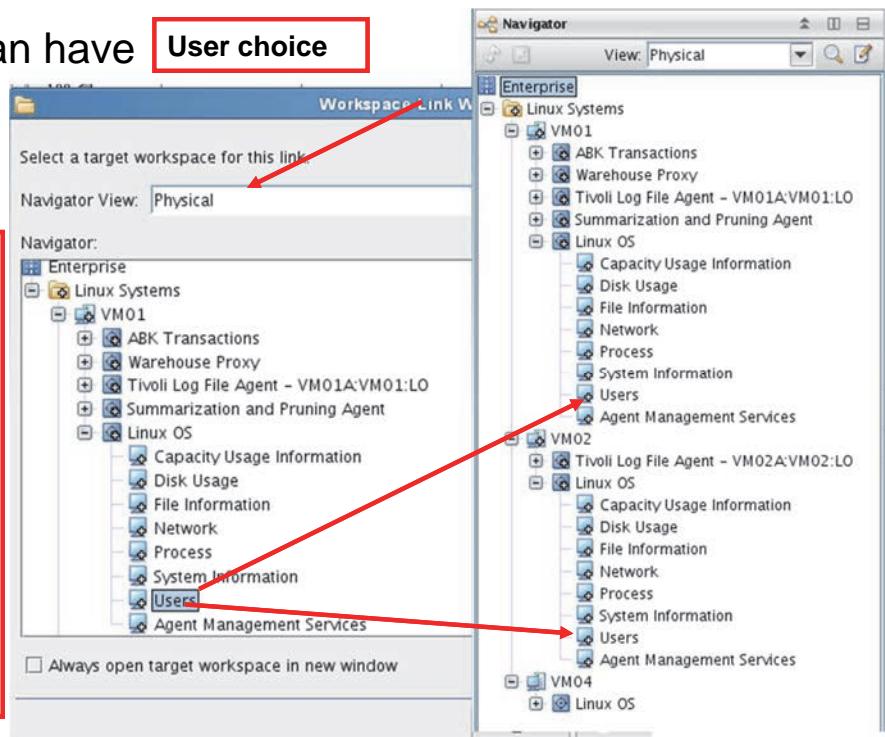
Absolute links always address one specific workspace in the enterprise as the target workspace. The link is defined with absolute target selection mode as a link from the Linux Systems Navigator item to the Users Navigator item. The target workspace is always the workspace on the specific Users Navigator item under VM01.

Link types: Relative links

- Relative links can have **User choice**
multiple target workspaces

Link definition:
 • **Relative link**
 • Link from **Linux Systems** to **Users**

Target selection during execution:
 A user must choose from multiple link targets that are displayed during execution



© Copyright IBM Corporation 2013

11

Link types: Relative links

The scenario shown on this slide uses the same link source and link target as the previous example describing the absolute link target selection type. This time, the link is implemented as a relative link.

Relative does not uniquely identify a single workspace as a potential target; instead, it identifies multiple workspaces. Therefore, the execution of this link results in an ambiguous situation. To successfully link to a specific workspace, the user is prompted to select a target from the list of possible targets.

Relative links compare the stored path with other paths available in the Navigator view. Next, there is a search for a Tivoli Enterprise Monitoring Server that is like the one stored in the target path. A common ancestor (in this case Linux Systems) defines the subtree to search for possible targets (in this case the Users navigator).

Link types: Dynamic links

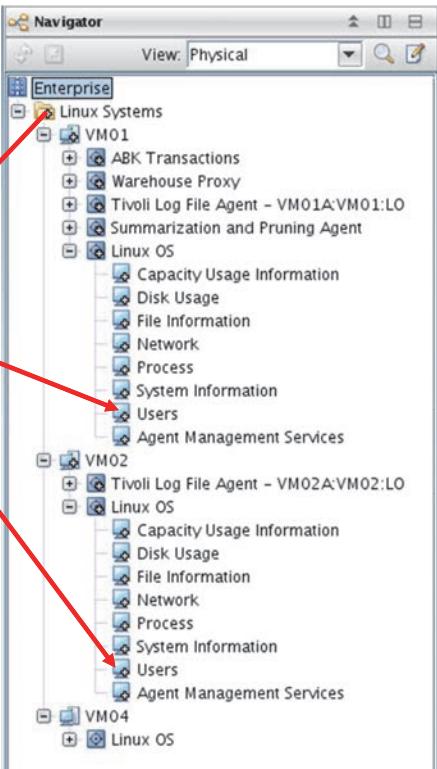
- Dynamic links work similarly to relative links
- The target selection is controlled through expressions

Link definition:
 • **Dynamic link**
 • Link from **Linux Systems** to **Users**

Target selection during execution:
 A target is selected based on parameters defined in the link definition

Automatic substitution

- Host name
- Managed system name
- IP address



© Copyright IBM Corporation 2013

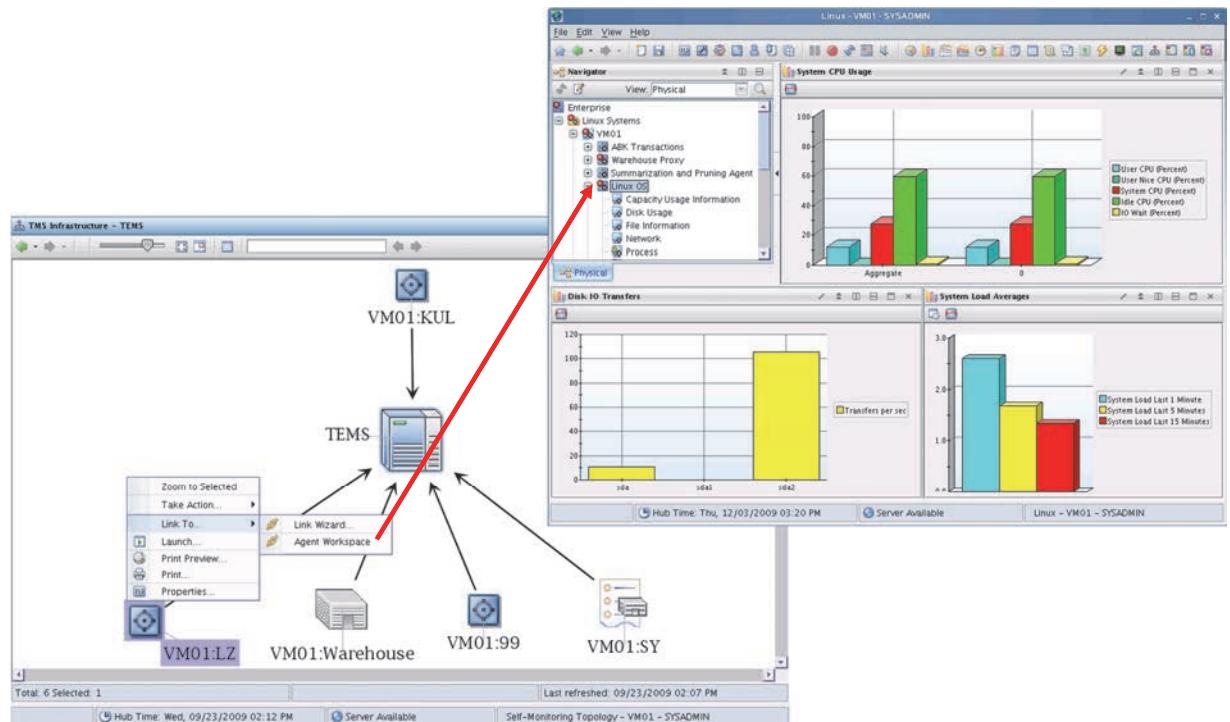
12

Link types: Dynamic links

This time the same link is implemented as a **dynamic** link, which also points to multiple target workspaces during link definition. Unlike relative links, dynamic links include defining a variable, which identifies the link target.

During execution, the current value of a link source accesses a specific workspace. This link type is useful for links *from* a view displaying entries from multiple managed systems *to* a view displaying managed system-specific information.

Dynamic links: Example



© Copyright IBM Corporation 2013

13

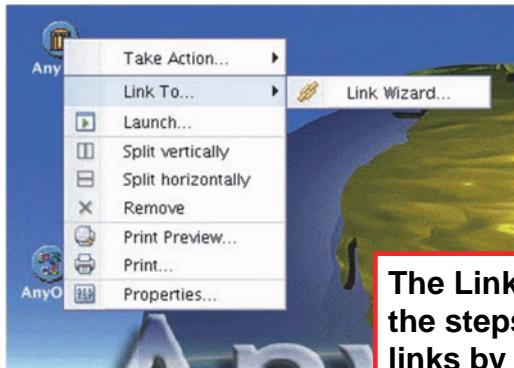
Dynamic links: Example

A product-provided implementation of a dynamic link is available in the Self-Monitoring Topology workspace on the Enterprise Navigator item in the Navigator Physical view. Right-click any icon in the TMS Infrastructure view and access the **Agent Workspace** link. You are directed to the overview workspace of the specific agent. The two values used to identify the target are the *managed system name* and the *product code*.

Lesson 2. Defining a simple, absolute link

Lesson 2: Defining a simple, absolute link

1. Build or select your target workspace
2. Access the source workspace
3. Position your mouse over the region of the source workspace where the link is needed
4. Right-click and start the Link wizard



The Link wizard can be accessed from a Navigator item, graphic view icon, or chart object

The Link wizard is a tool to guide you through the steps of generating simple and advanced links by specifying the required parameters

What this lesson is about

This lesson covers the creation a simple, absolute link. To start, you first define a simple, absolute link, which is the most common link type used. This slide describes some of the preparation that is required before you can successfully create links.

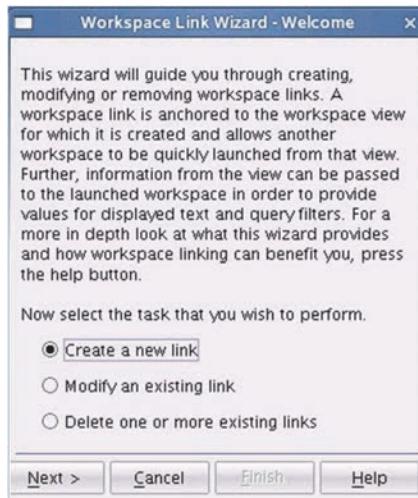
What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a simple, absolute link.
- Issue a simple, absolute link.

Using the Link wizard to define a new link

The first option is to define a new link or manage existing ones



© Copyright IBM Corporation 2013

15

Using the Link wizard to define a new link

After you launch the Link wizard, you must decide whether you want to create a link, modify an existing one, or delete one or multiple links. When modifying an existing link, you cannot modify the target of the link. If you must modify the target, create a new link.

Entering the link name and description

Enter a link name in the **Name** field and an optional description



A good way to name links is to include
the name of the target workspace

© Copyright IBM Corporation 2013

16

Entering the link name and description

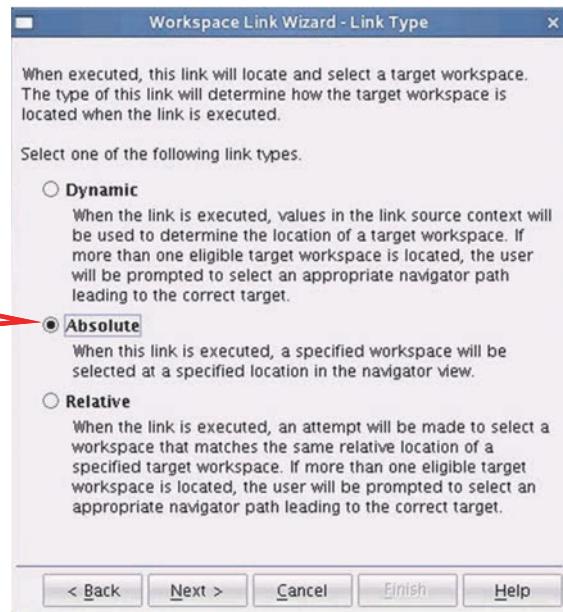
As a next step, you are prompted to type a link name and optional description. It is good practice to type the name of your target workspace. Users then know which workspace is accessed when they issue the link.

You can show the name for graphic view icons and table row link indicators. This display occurs when you hold the cursor over a link source or when right-clicking the source.

Defining the link target as absolute

Select a link type to define the link target

Select the most commonly used **Absolute** option, which uniquely identifies one specific workspace as the link target



© Copyright IBM Corporation 2013

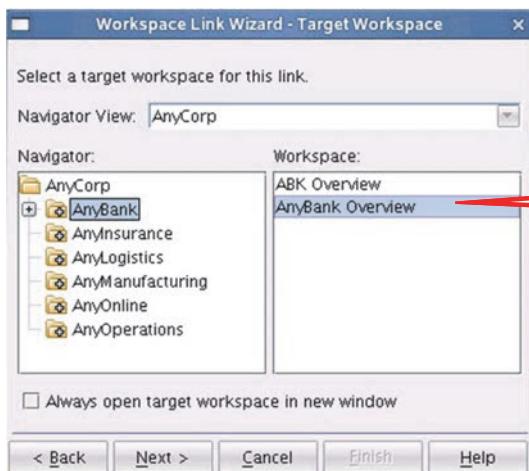
17

Defining the link target as absolute

The next menu requires you to indicate the target selection link type. In this first example, you see how to create an absolute link. For this type, you do not have to enter link target selection criteria other than selecting the link target workspace itself.

Selecting the target workspace

Select the target Navigator item that you want and any available workspace



Target workspaces can be in different Navigators

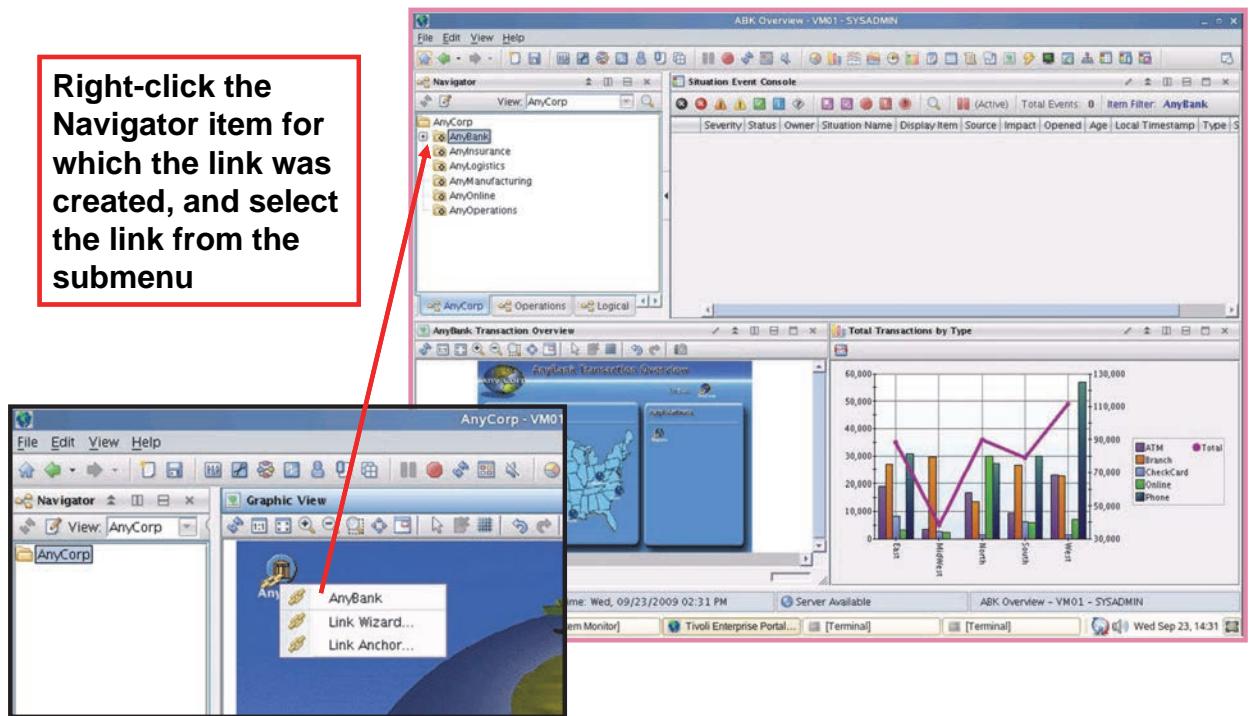
Because you are creating a simple link, you can finalize the link definition by clicking the **Next**, **Next**, and **Finish** buttons

Selecting the target workspace

To select your target workspace, follow these steps:

1. Select the Navigator you want to link to. You can navigate to a workspace that is in a different Navigator view.
2. Select a Navigator item within the Navigator.
3. Select an available workspace for this Navigator item. This view can include workspaces that are only available as link targets handed over by a link. Examples are workspaces that only function with filter parameters.
4. Because this link example is a simple link, no additional modifications must be made. Finalize the link by clicking **Next**, **Next**, and **Finish**.

Issuing the absolute link



© Copyright IBM Corporation 2013

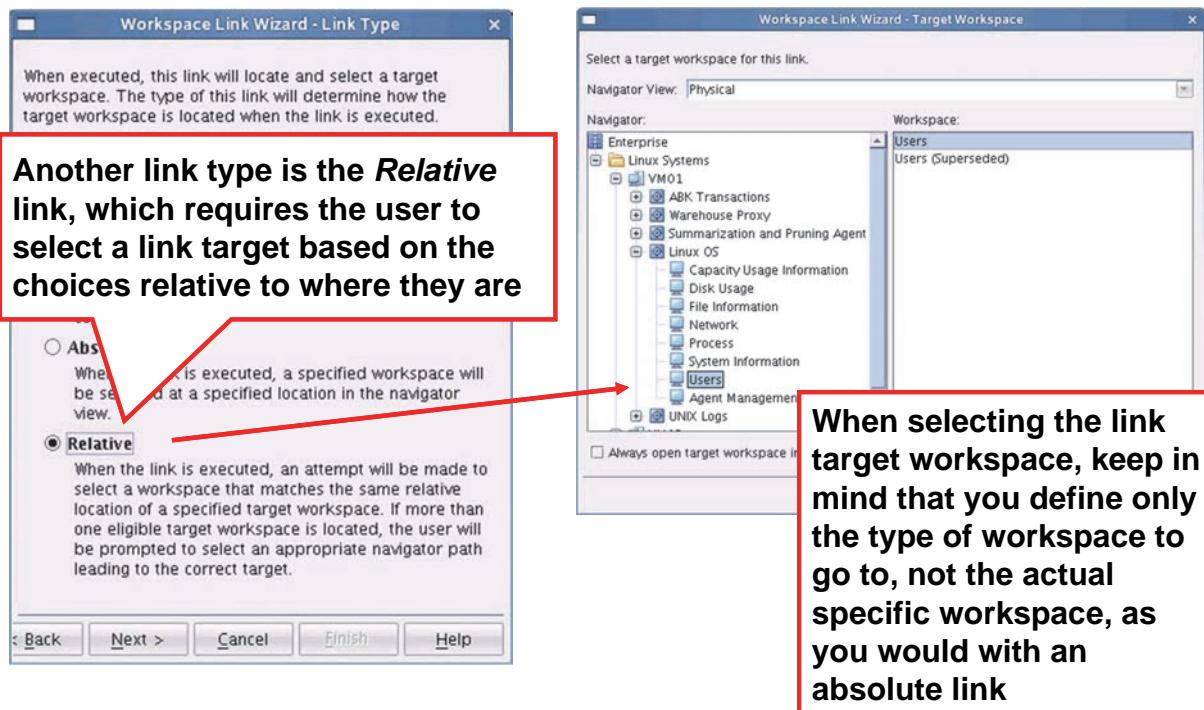
19

Issuing the absolute link

Now that the link is defined, you can issue it from the link source. This location is where the Link wizard was launched when you created the link. Links are saved automatically in a separate table in the Tivoli Enterprise Portal Server database.

Lesson 3. Defining a relative link

Lesson 3: Defining a relative link



© Copyright IBM Corporation 2013

20

What this lesson is about

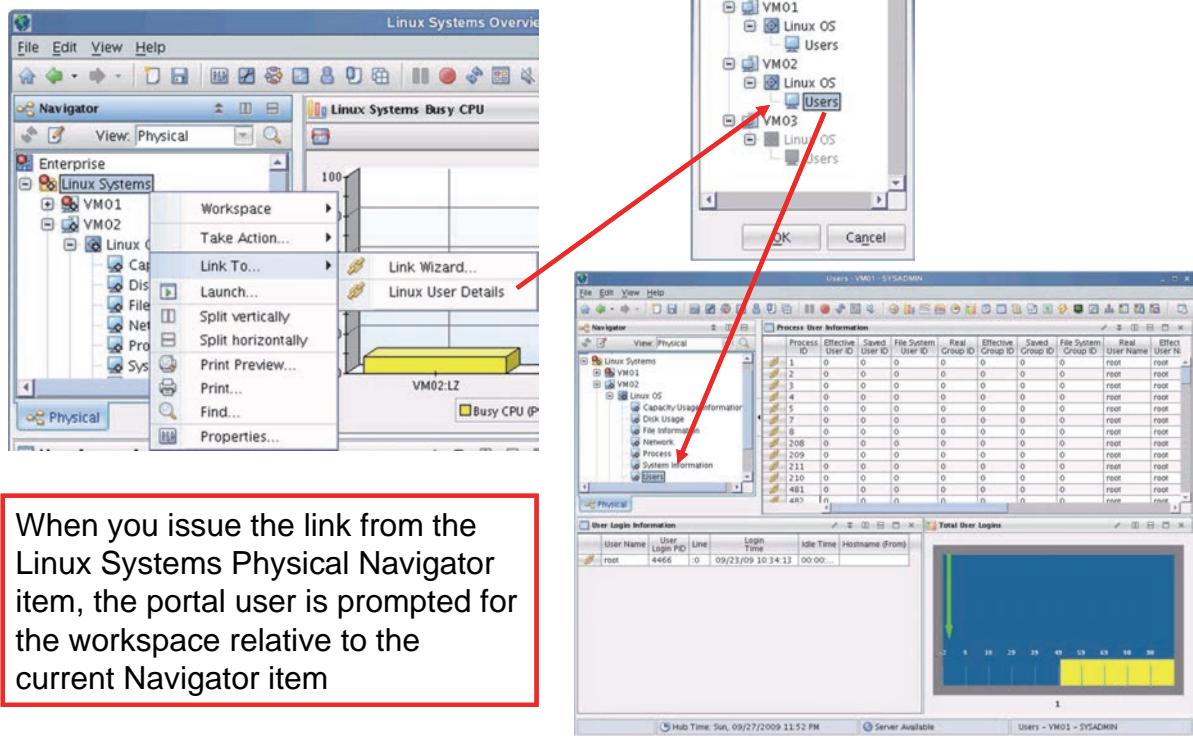
This lesson covers the creation of a relative link. Relative links work differently, prompting the user to identify a target workspace from a number of possible workspaces during execution.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a relative link.
- Issue a relative link.

Issuing the relative link



© Copyright IBM Corporation 2013

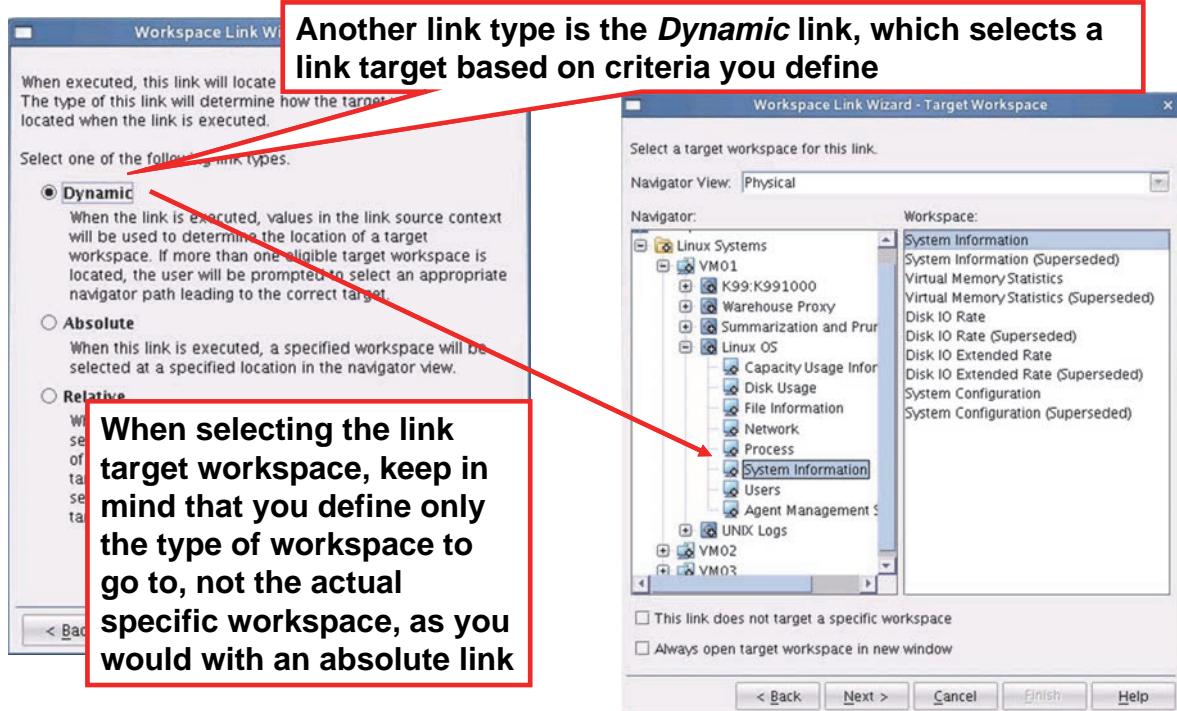
21

Issuing the relative link

When you issue a relative link, you are prompted to select the target workspace.

Lesson 4. Defining a dynamic link

Lesson 4: Defining a dynamic link



© Copyright IBM Corporation 2013

22

What this lesson is about

This lesson covers the creation of dynamic links, which identify a target workspace from a number of possible workspaces during execution. When you define a dynamic link, the target is chosen similarly to absolute links, but it is identified based on its type. It can be available multiple times in the Navigator; for example, multiple workspaces under each agent of the same type.

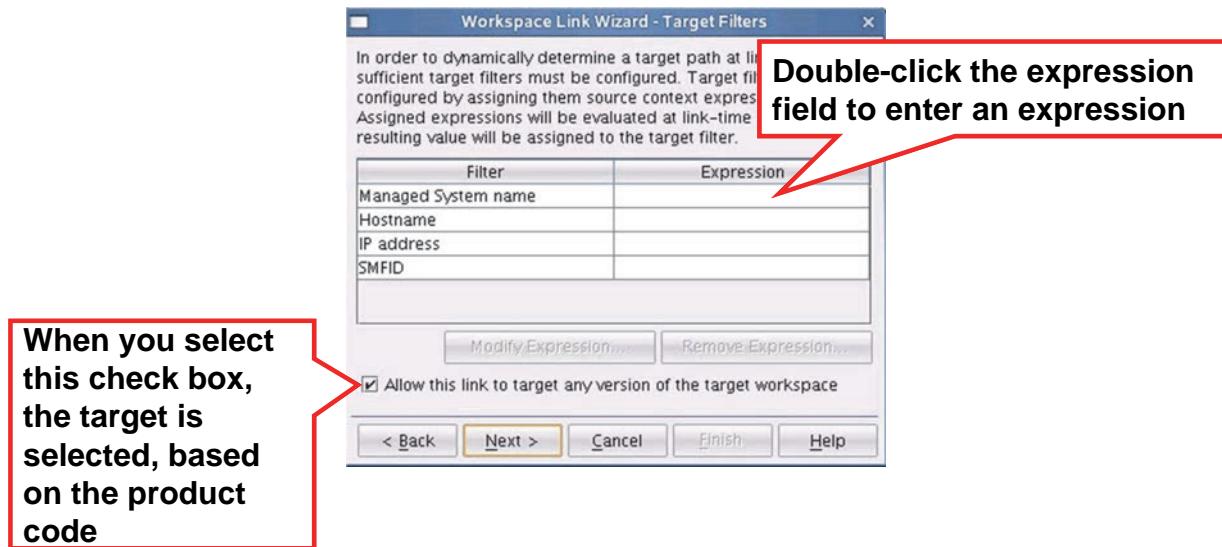
What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a dynamic link.
- Issue a dynamic link.

Defining the target selection criteria

You can substitute the managed system name, the host name or IP address for distributed systems, and the SMFID for z/OS systems to select the target workspace dynamically



© Copyright IBM Corporation 2013

23

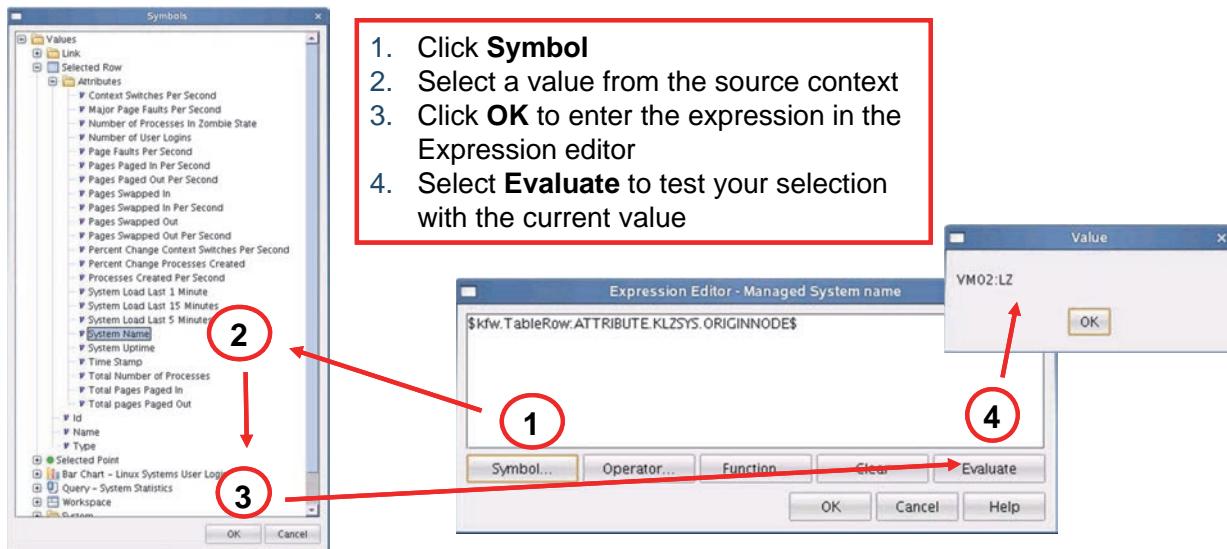
Defining the target selection criteria

Uniquely identify a target workspace during execution of the link. Enter an expression that identifies the managed system, host name, IP address, or SMFID to which the workspace belongs.

This expression is substituted when you issue the link with the parameter available for the expression at that time. To access available parameters, double-click an empty field in the Expression column, or highlight one of the filters and click **Modify Expression**.

Defining the target selection criteria (continued)

- After selecting the workspace, enter the criteria to be used during link execution to identify the link target
- Substitute any value that is part of the context of the source workspace



© Copyright IBM Corporation 2013

24

To build your expression, you can select from symbols and include operators and functions.

Typically, you substitute a symbol that represents the managed system name or host name where the agent is running.

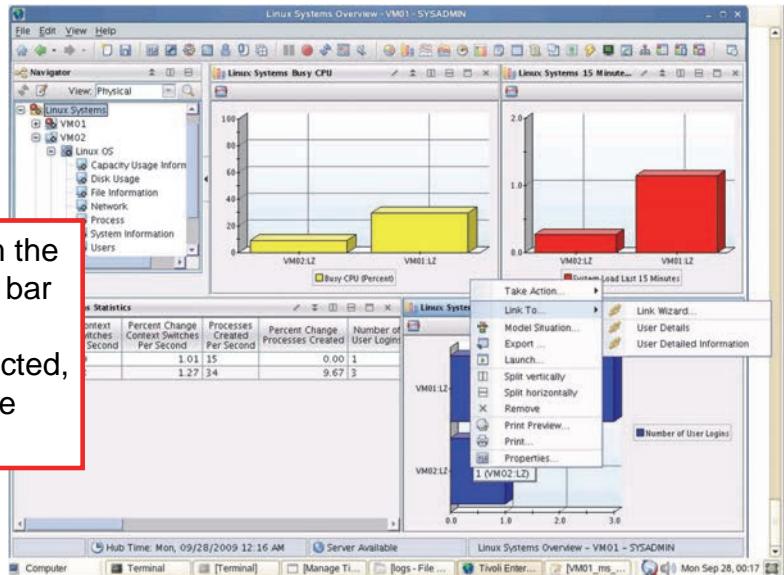
In the example, the **Selected Row > Attributes > Name** symbol identifies the managed system name of an agent. After you click **OK**, you can see the expression in the Expression editor.

Click **Evaluate** to validate the expression result. The result is based on the values available at the source from where the Link wizard was launched.

Issuing the dynamic link

When you issue the link, the target workspace is identified through the variable that you specified in the definition

When you issue the link from the **Linux System User Logins** bar chart in the Linux Systems workspace, the target is selected, depending on the value of the System Name



© Copyright IBM Corporation 2013

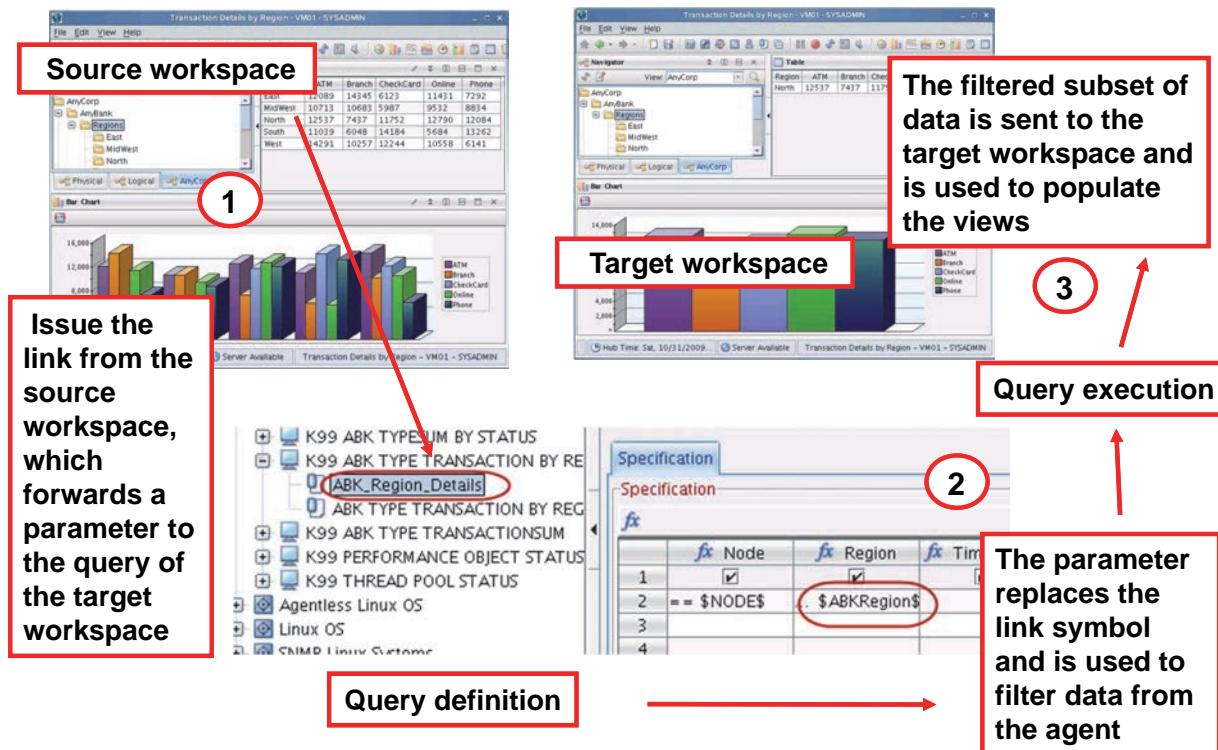
25

Issuing the dynamic link

When executing the dynamic link, a value available at the link source is used to identify the link target. In this case, the System Name attribute in the Linux System User Logins view contains the managed system names. This name is used to link to the System Information Navigator item under VM01 or VM02.

Lesson 5. Advanced links

Lesson 5: Advanced links



© Copyright IBM Corporation 2013

26

What this lesson is about

This lesson covers the creation of advanced links, which are the most flexible and complex links.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Explain the preparation involved in creating an advanced link.
- Build an advanced link.
- Customize the header and footer of a view in a workspace that is the target of an advanced link.
- Create a link anchor.

Before you start building an advanced link, you must understand how advanced links work. You also must learn some new terminology.

When executing an advanced link, information from the link source is used to build an expression that is passed on to the target workspace. This information is used in the target workspace queries or in the workspace views to manipulate the displayed data or information.

Data in the target workspace can be filtered. The queries in the target workspace views must contain variables to receive the filter criteria expression through the link. The query uses the filter to collect data from the agent that meets the filter criteria. The displayed content varies based on the source from where the link is issued.

New terminology: Link symbols

- When creating advanced links, you use variables to substitute values that are available as part of the source workspace context
- Those variables are called ***link symbols***
- They filter or manipulate data that is displayed in the target workspace views

© Copyright IBM Corporation 2013

27

New terminology: Link symbols

Advanced links use variables to manipulate data that is displayed in the target workspace views. Those variables are called ***link symbols***. You can specify them in various locations.

You can create any number of link symbols yourself or use existing ones that are provided in the **Parameters** menu of the Link wizard. Typically, customers add link symbols to the queries used in the target workspace.

New terminology: Link symbols (continued)

- Some link symbols are available in the **parameters** menu of the Link wizard
- You can add other link symbols in these ways:
 - Manually in the **Parameters** menu
 - Automatically in the **Parameters** menu by adding them to the queries, used in the target workspace views

Defining an advanced link: Preparation

1. Before you start the Link wizard to create an advanced link, create or modify your target workspace
2. Access the Query editor, and add a link symbol to your target workspace query

The screenshot shows the Query editor interface. On the left is a Navigator panel listing various workspace items, including 'K99 ABK TYPESUM BY STATUS', 'K99 ABK TYPE TRANSACTION BY RE', and 'ABK Region_Details'. The 'ABK Region_Details' item is selected and highlighted with a red oval. To the right is a 'Specification' dialog box with a table titled 'Specification'. The table has three columns: 'fx Node', 'fx Region', and 'fx Timestamp'. Row 1 has checked boxes in all three columns. Row 2 contains the expression '= = \$NODE\$.. \$ABKRegion\$'. A red oval highlights the '\$ABKRegion\$' part of the expression. The 'fx' icon is also circled in red.

Link symbols are specified inside of a pair of \$ and can have any unique name, for example **\$ABKRegion\$**

3. Assign the query to the workspace views, and adjust filters as needed, and save the workspace

© Copyright IBM Corporation 2013

29

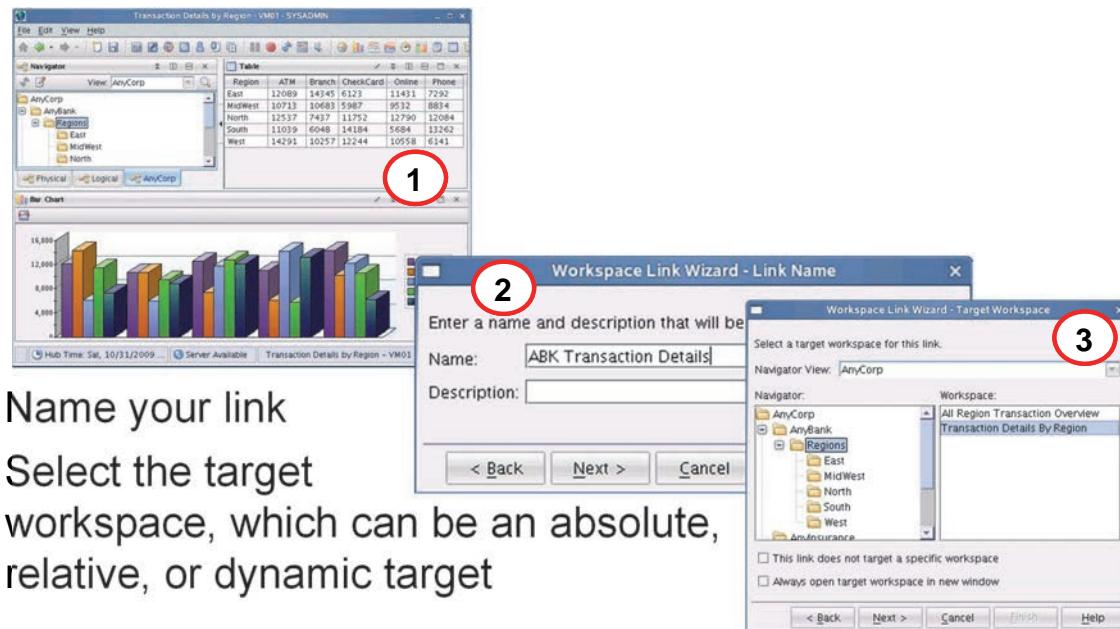
Defining an advanced link: Preparation

Before you create the advanced link in the Link wizard, add link symbols to the query that is used in the target workspace views. Assign the queries to the views in the target workspace to make the link symbols visible in the Link wizard. To add a link symbol, enter an arbitrary name and enclose it in \$ symbols, such as **\$ABKRegion\$**.

You can substitute required values during the execution of a link. You might not want to access this workspace directly from a Navigator item. If link symbol values cannot be determined, they are disregarded. In this case, it might be useful to make the target workspace accessible only through links, meaning that they cannot be accessed directly from the Navigator. This task is performed through the link anchor, which is described later in this unit in “[Link settings: Link anchor](#)” on page 84.

Launching the Link wizard

1. Access your source workspace and launch the Link wizard



© Copyright IBM Corporation 2013

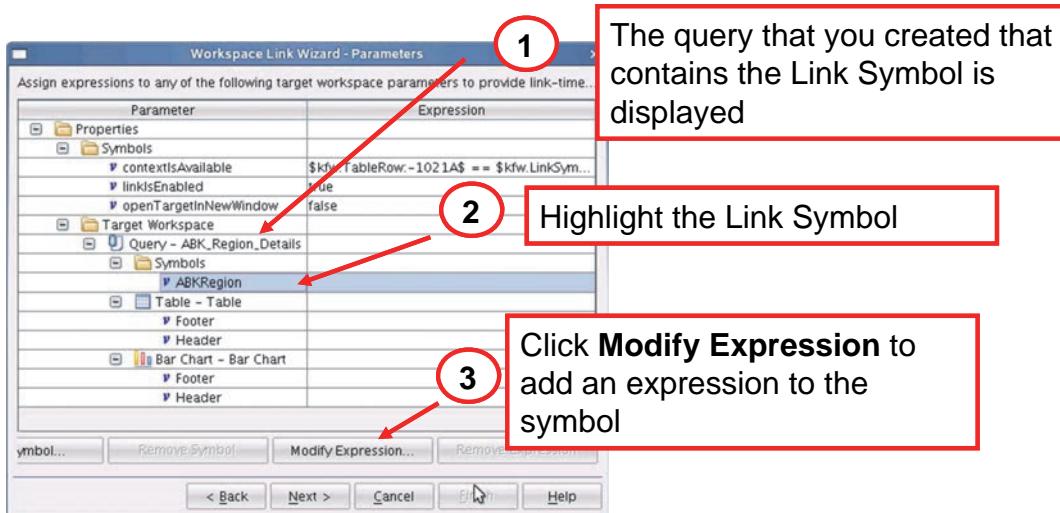
30

Launching the Link wizard

Creating advanced links works the same way as creating simple links. Launch the Link wizard and name the link. Select absolute, relative, or dynamic as the link target selector type, and select the workspace you want to link to. For dynamic links, you might have to specify your target selection criteria as described earlier in [Lesson 4, Defining a dynamic link](#), on page 66.

Assigning expressions to link symbols

A symbol that was added to a target workspace query is visible in the **Parameters** menu



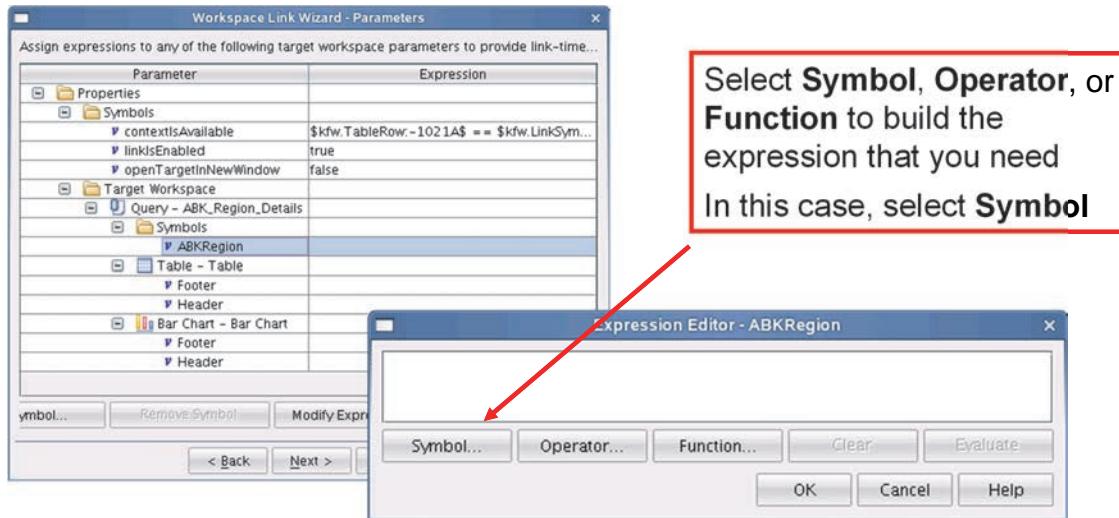
Assigning expressions to link symbols

After selecting your link target, click **Next** to access the **Parameters** menu of the Link wizard. This menu is only used for creating advanced links.

The link symbols are the means for passing values from the source to the target, and they are visible in the Parameters column. After creating new link symbols in your target workspace view queries and assigning the queries to the views, they appear in this list.

Some symbols are available by default to manipulate data in your target workspace views. This data can be the headers and footers. Links can be enabled and disabled, depending on certain criteria of your source. You can add symbols to manipulate other properties of the target workspace and its associated views.

Modifying expressions



© Copyright IBM Corporation 2013

32

Modifying expressions

You use expressions to assign a value to the link symbol during link execution. To change the expression, highlight the link and click **Modify Expression**. The Expression editor opens. In the editor, you can use symbols, operators, and functions to define your expressions.

Defining the link expression

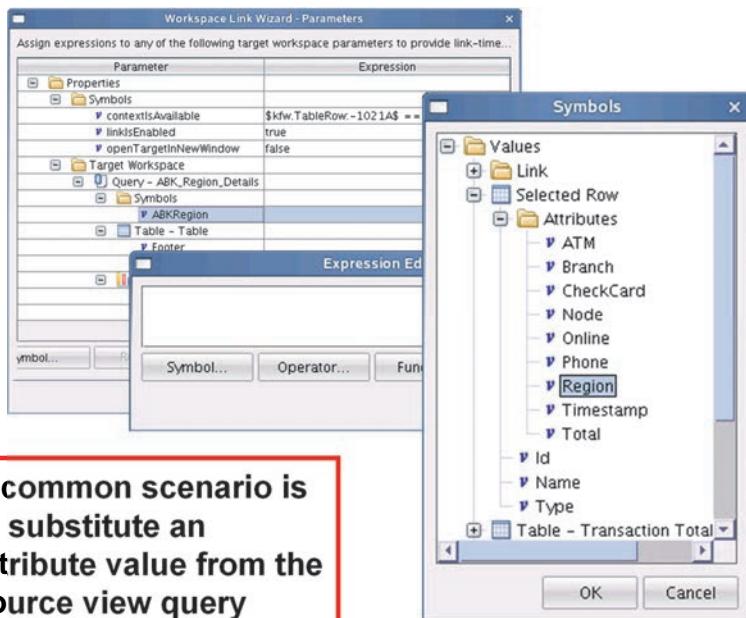
Specify an expression for every link symbol

You can select any value that is available as part of the link source workspace

Examples for values:

- Attributes
- Node name
- Managed system name
- User ID that is logged on
- Name of the source workspace
- Name of the Navigator
- Name of the Navigator item

A common scenario is to substitute an attribute value from the source view query



© Copyright IBM Corporation 2013

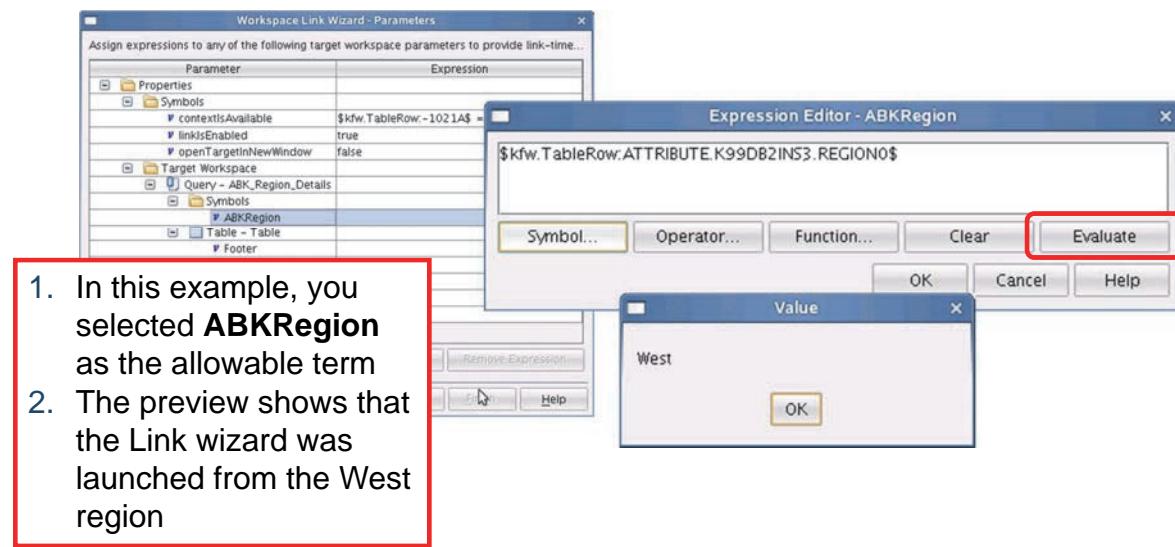
33

Defining the link expression

If the link is created from a view in the source workspace, the expression likely contains an attribute value. This first example uses the Region attribute to be substituted during link execution.

Validating the expression

- Expressions can be arbitrarily complex
- After you enter an expression, you can test it by clicking **Evaluate**



© Copyright IBM Corporation 2013

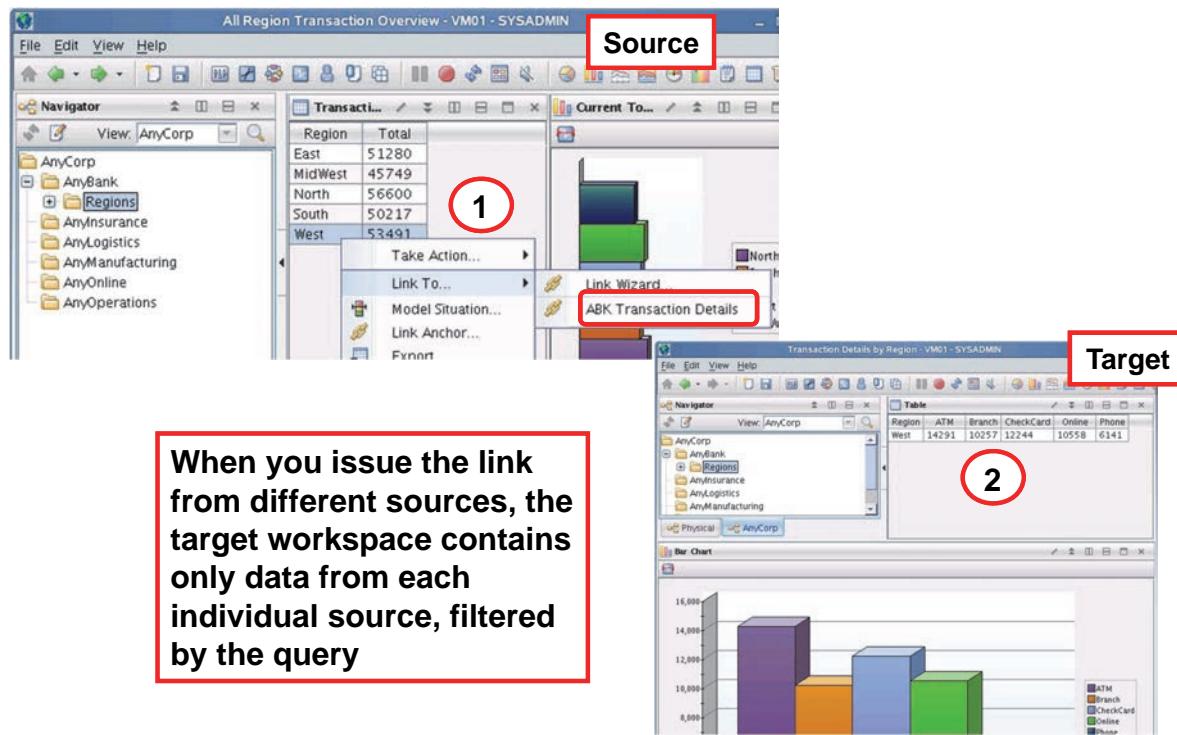
34

Validating the expression

After the value is selected, it is displayed in the Expression editor as a \$...\$ expression. You can use operators and functions to further refine and manipulate the expression.

Click **Evaluate** to test your expression. If the Value window does not show any value, the expression is not valid and does not work during link execution. Make sure that you receive a result before you finalize your link.

Issuing the advanced link



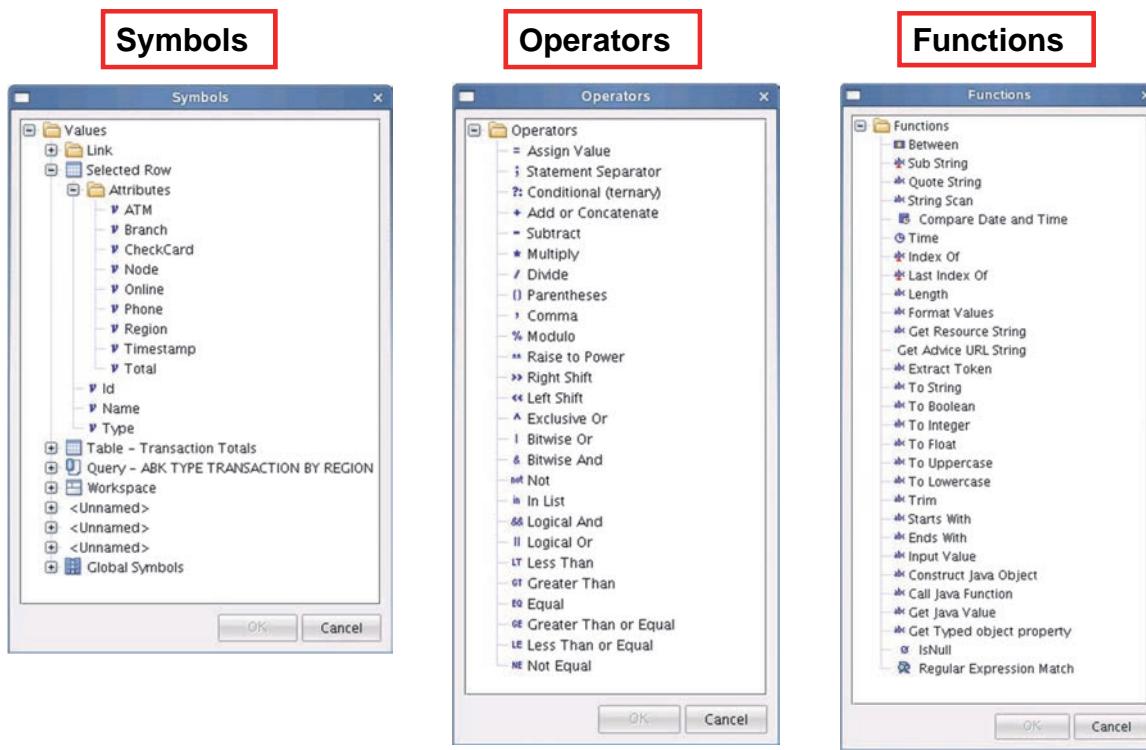
© Copyright IBM Corporation 2013

35

Issuing the advanced link

After the link is created, you can issue it from your source to test the result. You can create a link for a view from a table, bar chart, pie chart, and plot chart. The purpose is to show more detailed information about a particular view. You can issue the link from various sources to display different content in the target. The example uses a link from a bar chart view to show only data from one specific bar in the target workspace.

Expression editor: Components



© Copyright IBM Corporation 2013

36

Expression editor: Components

The **Symbols** menu shows all values that are available at the source from where you launch the Link wizard. In this case, the Link wizard was launched from a table view. Attributes from the table query are available to be substituted at execution time.

The **Symbols** menu shows different structures for different link sources. As an example, the structure for creating a link from a Navigator item does not contain attributes.

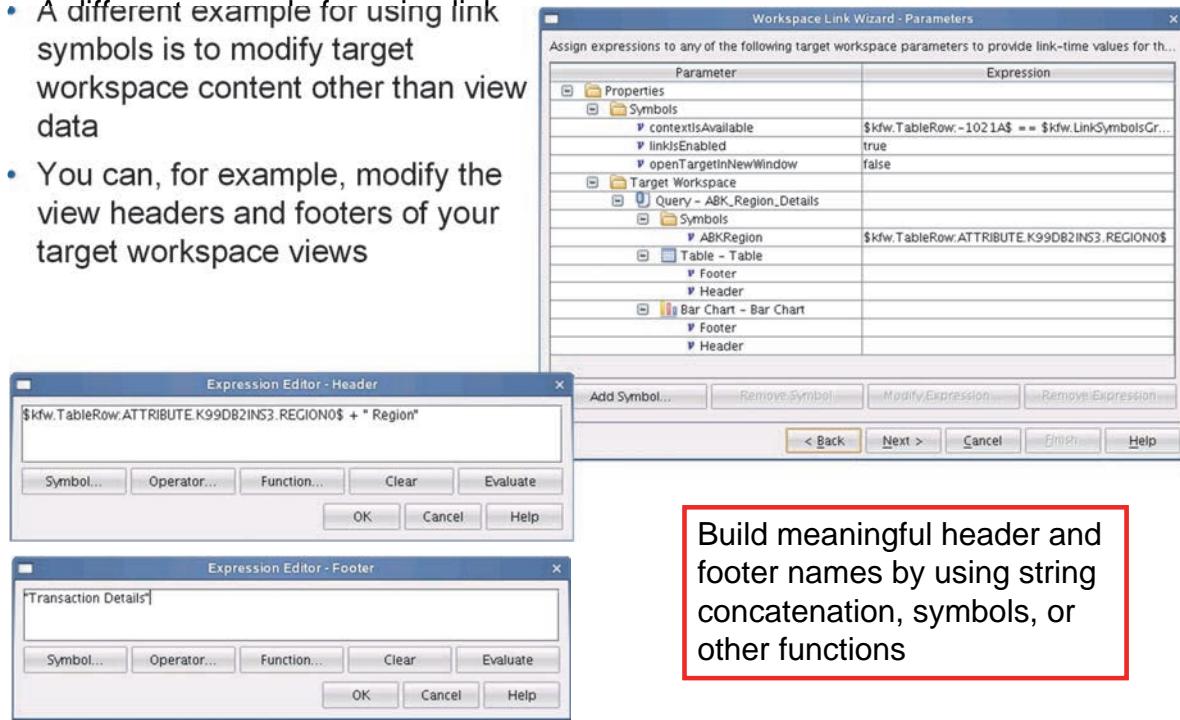
Values can be substituted from the source. These values are parameters for the Navigator, Navigator item, node, agent, workspace, views, queries, and attributes. Global values such as the ID of the logged-on user can also be used.

After selecting the values to be dynamically substituted during link execution, you can refine your expressions by using functions and operators.

Many of the functions and operators are described in detail in the Online help for Tivoli Enterprise Portal. Examples of how to use them are included also. Spend some time reading through the various options after you are familiar with the basics of generating advanced links.

Other link symbols: Header and footer

- A different example for using link symbols is to modify target workspace content other than view data
- You can, for example, modify the view headers and footers of your target workspace views



© Copyright IBM Corporation 2013

37

Other link symbols: Header and footer

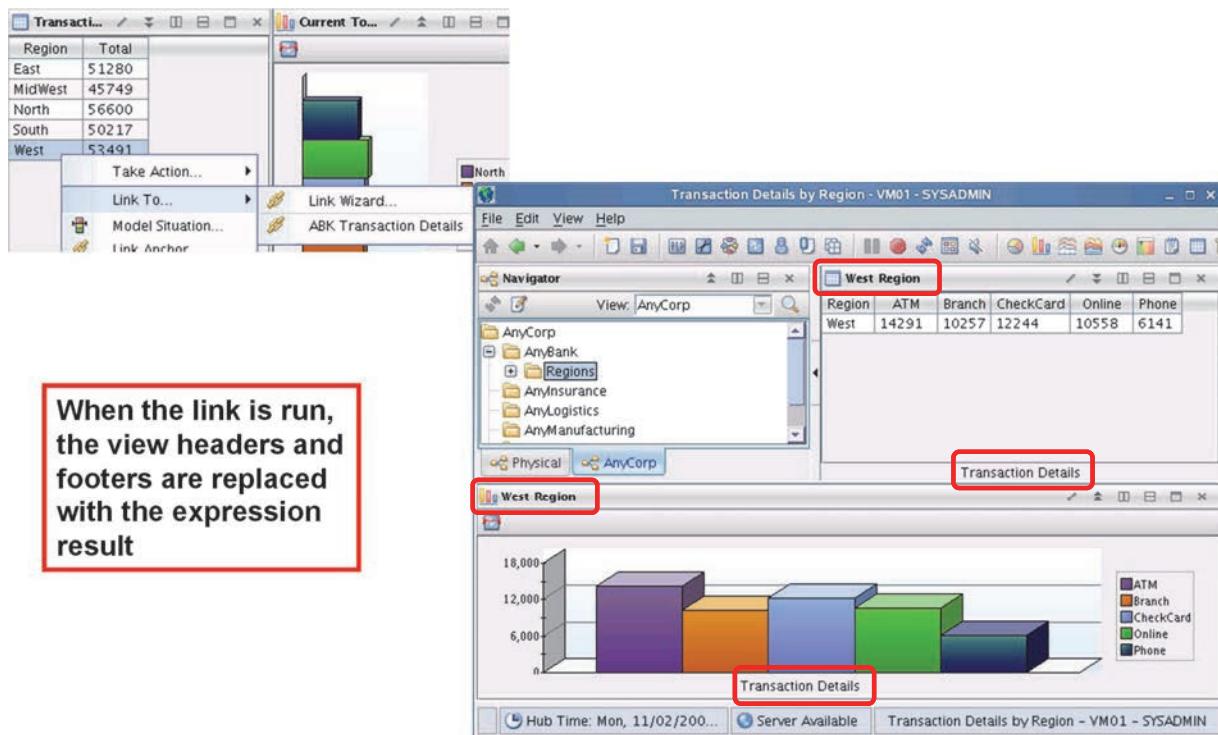
You can see on the previous slide that the content of your target workspace changes, displaying data for one region only.

This feature is useful, but there is not much information in the view to adequately describe the information that is displayed.

Header link symbols are available in the Link wizard **Parameters** menu. You can use them to substitute the headers of the views in the target workspace.

Creating the expressions works the same way as creating expressions for the link symbols you used previously. There are functions that can be used, such as TOUPPER, to convert strings to uppercase.

Issuing the link: Test the header and footer substitutions



© Copyright IBM Corporation 2013

38

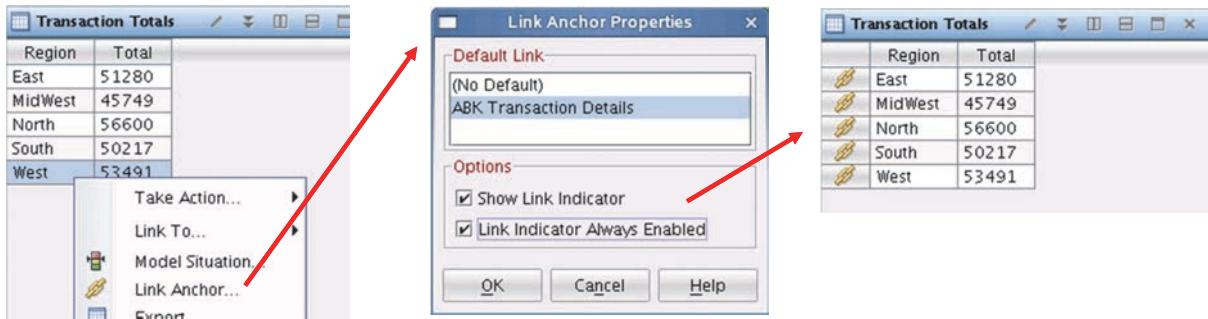
Issuing the link: Test the header and footer substitutions

Save the changes and select the link to review the result. The column headers in the target workspace are replaced with the Region name and the text that was added in quotation marks.

Link settings: Link anchor

The link anchor stores properties of links, such as these examples:

- The visibility of the link indicator
- The default action performed with a single click
- Whether the link is enabled, based on more criteria that can be specified in the Link wizard
- **Show Link Indicator** works with graphic views, but not bar chart views



© Copyright IBM Corporation 2013

39

Link settings: Link anchor

A **link anchor** controls the properties of link icons for graphic view icons and table views. You can define whether the indicator is visible, and whether a link is issued with a single mouse click. On charts, links are not visible. You right-click the entry to view the link.

The last option defines whether a test is performed to determine if the link is meaningful for each specific row or source. As a result of this test, the link indicator is made available or is displayed as unavailable. To make this feature work, you must specify additional settings in the Link wizard **Properties** menu.

An example is a table that shows a UNIX file system. You can enable links from table rows representing folders, and disable them for table rows representing files.

Student exercise



© Copyright IBM Corporation 2013

40

Student exercise

Perform the exercises for this unit.

Review questions

1. How many target workspaces does an absolute link have?
2. What tool do you use to create links?
3. Variables that are used to substitute values as part of the source workspace context are called _____.
4. What enables a default link action performed with a single click?

Review answers

1. How many target workspaces does an absolute link have?

One

2. What tool do you use to create links?

The Link wizard

3. Variables that are used to substitute values as part of the source workspace context are called

_____.

Link symbols

4. What enables a default link action performed with a single click?

A link anchor

Summary

Now that you have completed this unit, you can perform the following tasks:

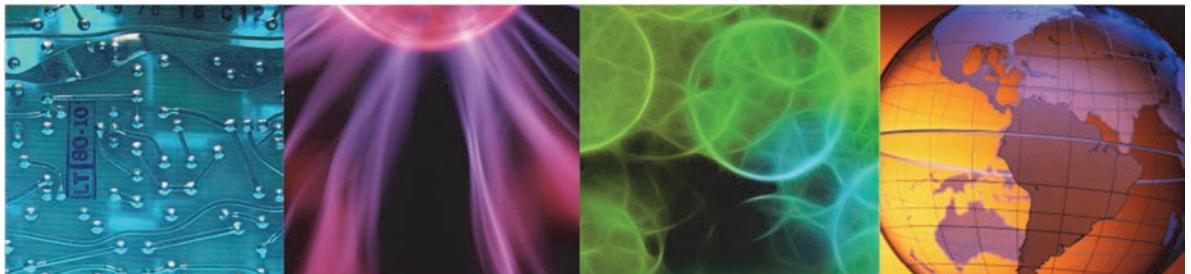
- Explain the differences between simple and advanced links
- Explain the differences between the link types of absolute, relative, and dynamic
- Build relative links to provide navigation to a workspace that the user selects
- Build dynamic links to provide navigation to a workspace by substituting in workspace values automatically
- Create, manage, and issue an advanced link



3 Monitoring your enterprise using advanced situation techniques



3 Monitoring your enterprise using advanced situation techniques



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit covers how to monitor the enterprise through expert advice, embedded situation events, correlated situation events, model situations, and dynamic thresholds.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises



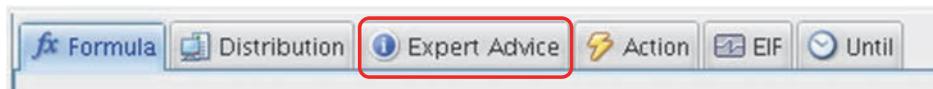
Objectives

When you complete this unit, you can perform the following tasks:

- Define expert advice
- Describe the use of embedded situations
- Build correlated situations that combine situations to create new situations
- Explain the differences between embedded and correlated situations
- Create situations based on current and historical data
- Construct thresholds that dynamically change for specific resources

Lesson 1. Expert advice

Lesson 1: Expert advice



- Using expert advice, you can enter a detailed description of the situation and describe how to solve the problem when the situation event occurs
- Expert advice can be provided in the following ways:
 - A simple text message
 - An HTML formatted text
 - A link to a website or HTML document
 - An expression that can perform these tasks:
 - Substitute situation event details to dynamically generate links to other documents or search strings
 - Contain various functions and operators

© Copyright IBM Corporation 2013

3

What this lesson is about

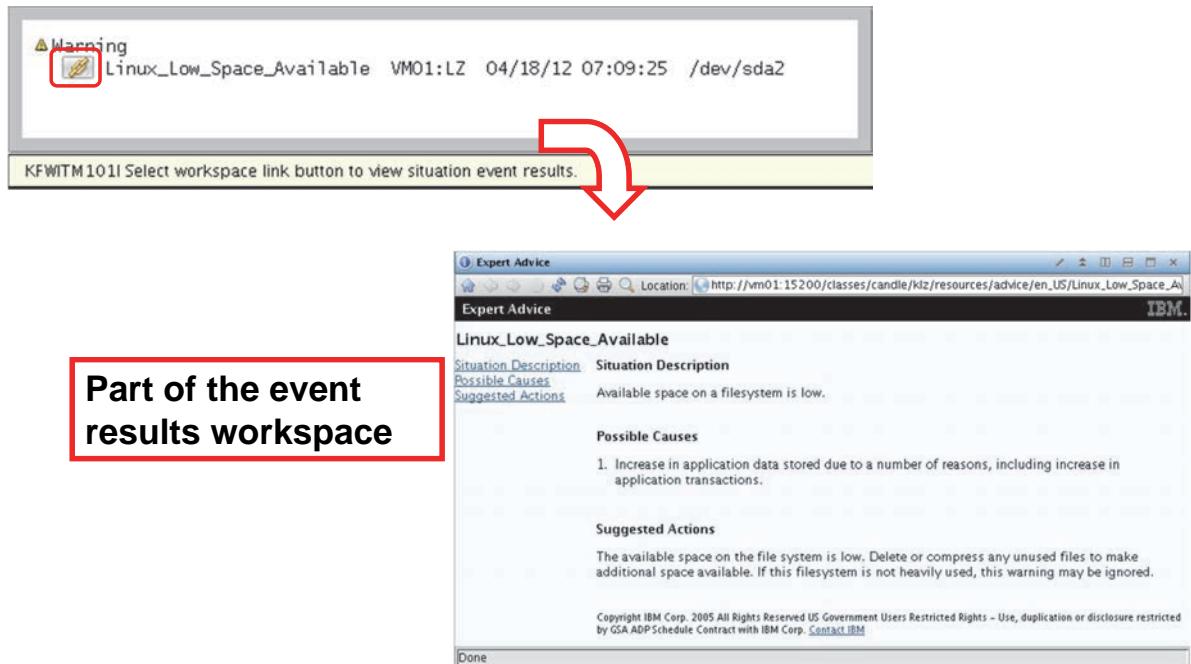
Expert advice is used to populate a view within the situation event results workspace. It provides guidance on how to solve the particular problem the situation monitors. Before creating expert advice, you must decide what type of message you want to display when the situation becomes true. This information can be a simple text message, or even an elaborate expression, linking to other documents and substituting situation event values.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Access expert advice for a situation event.
- Create customized expert advice for a situation.

Accessing expert advice



© Copyright IBM Corporation 2013

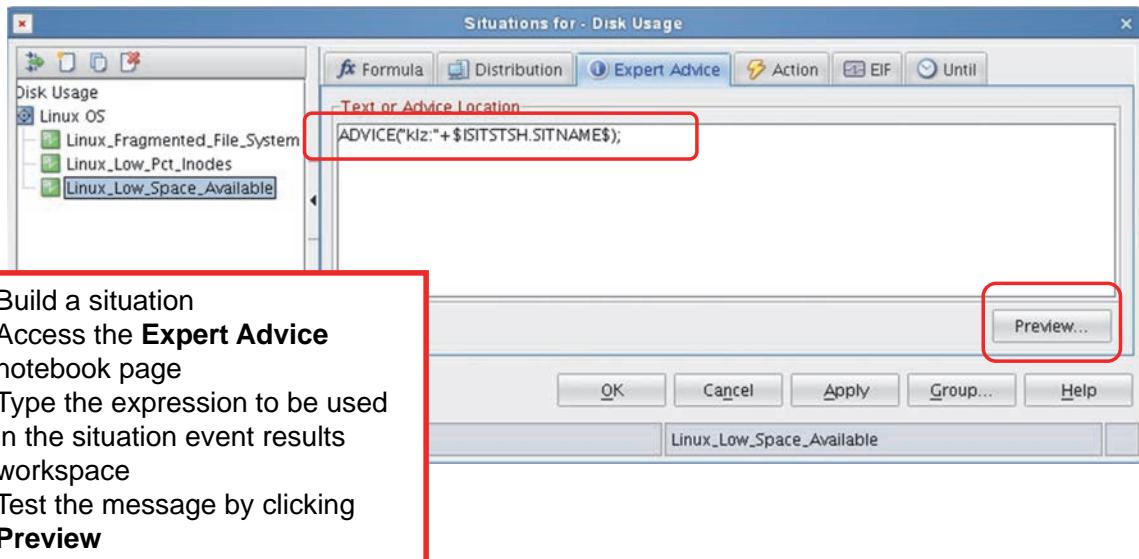
4

Accessing expert advice

You access expert advice as part of the situation event results workspace. The lower right viewing that workspace shows the message or document you want to display as part of the situation event. Presenting this useful information helps your operators to solve problems.

Specifying expert advice

Include comments or instructions to be read in the situation event results workspace when the situation becomes true



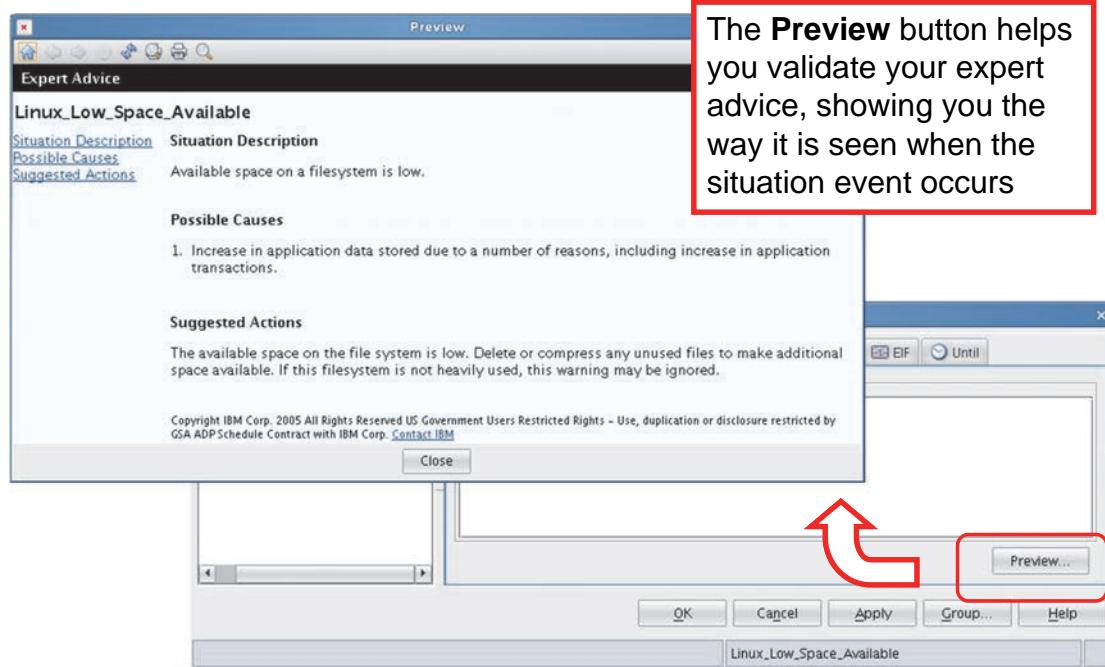
© Copyright IBM Corporation 2013

5

Specifying expert advice

When creating your expert advice, access the **Expert Advice** tab in the Situation editor and enter text, a URL, HTML, or an expression. Start simple and make sure that you understand how expert advice works. You can copy more elaborate expressions directly from the Link expression editor. That topic is covered later in this course.

Previewing expert advice



© Copyright IBM Corporation 2013

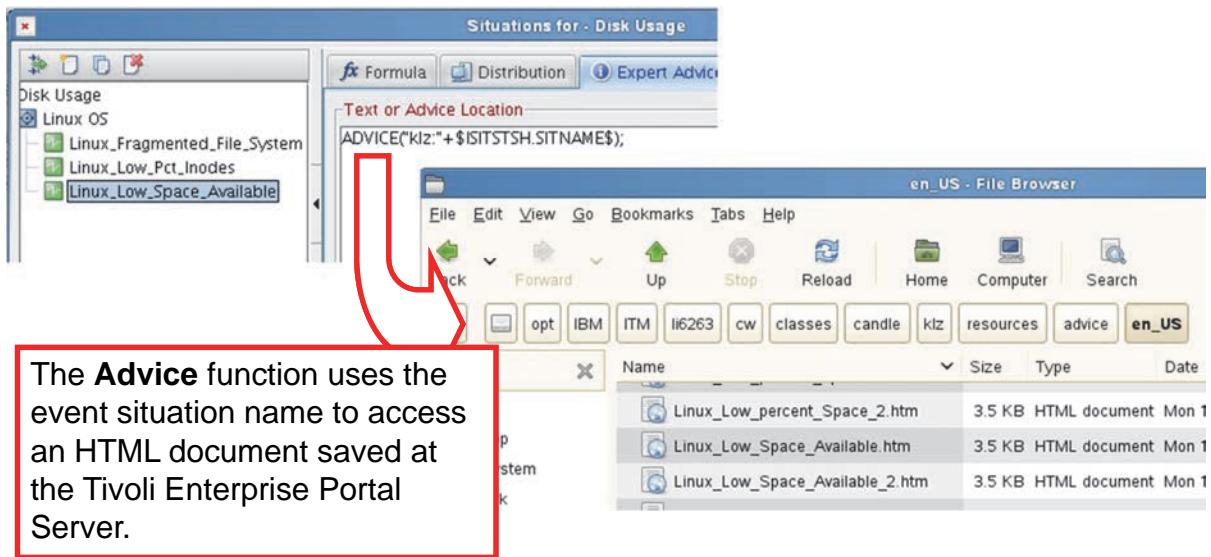
6

Previewing expert advice

Click **Preview** to validate your expression directly from the Situation editor, without having to generate a situation event and access the situation event results workspace. This function requires context to resolve certain variables. You might have to access the expert advice from the situation event results workspace to ensure that the variables resolve correctly.

Product-provided expert advice definition

Every product-provided situation comes with expert advice. It dynamically links to an HTML document, substituting the situation name to access the document.



© Copyright IBM Corporation 2013

7

Product-provided expert advice definition

All product-provided situations come with expert advice. This expert advice is saved as an HTML document at the Tivoli Enterprise Portal Server and is dynamically chosen based on the open situation event.

You can use product-provided expert advice as an example for creating your own expert advice, without the limitations of plain text. Using this method, maintenance is easier, and you do not have to memorize the syntax of more elaborate expressions.

Creating custom expert advice

Using a product-provided example, you can quickly create your own expert advice:

1. Access the storage location for product-provided expert advice documents
2. Save one of the documents under your new situation name
3. Modify the expert advice in the HTML document using a common text or HTML editor
4. Copy the advice expression from a product-provided situation of the same agent, preview your changes, and save the situation

The next four slides review these steps in detail

Creating custom expert advice

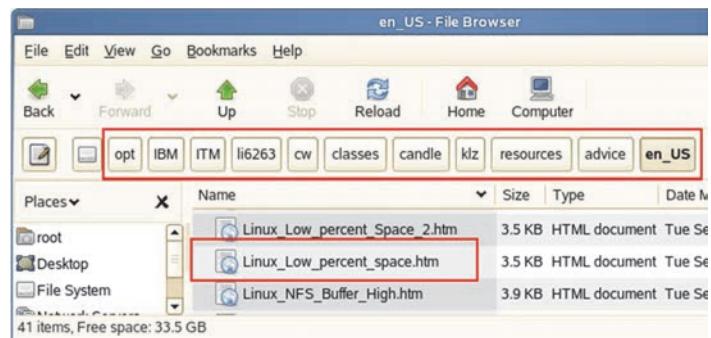
An efficient way of generating your own expert advice is to use the method provided with product-provided situations.

1. On Windows systems, access the directory **C:\IBM\ITM\CNB\classes\candle\kpc\resources\advice\en_us**, where **pc** stands for the specific product code of the agent you are working with.
On Linux or UNIX systems, access the directory **/opt/IBM/ITM/li6263/cw/classes/candle/kpc/resources/advice/en_us/**, where **pc** stands for the specific product code of the agent you are working with.
2. Copy and modify an existing expert advice HTML document. Make sure that you name it the same as the situation you want to use it for.
3. Use a text editor or HTML editor to open the document and enter the new expert advice.
4. Copy the entire expert advice expression from a product-provided situation expert advice of the same agent type into the expert advice expression field of your new situation. Preview your changes and save the situation.

Accessing expert advice HTML documents

- Each agent stores HTML advice documents in a separate directory at the portal server
- Access the directory:
 - **C:\IBM\ITM\CNB\classes\candle\kpc\resources\advice\language** (Windows)
 - **/opt/IBM/ITM/li6263/cw/classes/candle/kpc/resources/advice/language** (Linux or UNIX)

where **pc** stands for the specific product code of the agent with which you are working, and **language** is the language code



© Copyright IBM Corporation 2013

9

Accessing expert advice HTML documents

On Windows systems, the directory containing product-provided expert advice is located here:

C:\IBM\ITM\CNB\classes\candle\kpc\resources\advice\en_us

where **pc** stands for the specific product code of the agent you are working with.

On UNIX and Linux systems, the directory containing product-provided expert advice is located here:

/opt/IBM/ITM/li6263/cw/classes/candle/kpc/resources/advice/en_us/

where **pc** stands for the specific product code of the agent you are working with.

Some common product codes include these examples:

- **nt** for Windows
- **lx** for Linux
- **d2** for DB2
- **ux** for UNIX
- **m5** for OMEGAMON XE for z/OS

- **ip** for OMEGAMON XE for IMS
- **cp** for OMEGAMON XE for CICS
- **n3** for OMEGAMON XE for Mainframe Networks

Copying and saving a document

- Copy and paste an existing advice document
- Rename to the same name as the new situation
Make sure that the situation name is spelled correctly and that the document name has the .htm extension
- Edit and modify as needed

© Copyright IBM Corporation 2013

10

Copying and saving a document

The names of the advice documents, already in the specific directory, correspond to the product-provided situations for each agent. You can generate your own advice document by copying an existing one. Give it the same name as a situation you want to use it for, and make sure that the file extension is **.htm**.

Modifying the content

Access the document using an HTML or text editor

 Expert Advice

© Copyright IBM Corporation 2013

11

Modifying the content

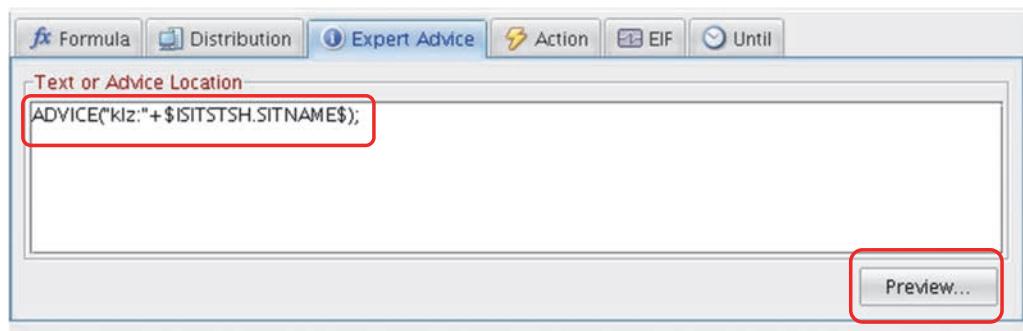
Use any text editor that you feel comfortable with to edit the advice document.

The default content includes the following elements:

- A description of the situation
 - The possible cause for the situation event
 - A description of how to handle and solve the particular problem

Copying advice expressions

- Using the Situation editor, copy the advice expression from a product-provided situation of the same agent, or enter it manually as shown



- Preview your new advice
- Save your changes

© Copyright IBM Corporation 2013

12

Copying advice expressions

In the Situation editor, copy the entire expert advice expression from a product-provided situation of the same agent type into the expert advice expression field of your new situation. Press Ctrl+C to copy and Ctrl+V to paste.

When you are done, verify that the expert advice works by clicking **Preview**. Save your situation.

More expert advice techniques

- Plain text
 - Comments or instructions to be read in the situation event results workspace when the situation triggers true
- Redirect to a website or HTML document
 - URL for a website or HTML document to show in the **Expert Advice** field when the situation event occurs
- Direct file access
 - Enter expert advice that directly accesses the HTML page
- Formatted HTML
 - You can use HTML in your expert advice to format the background and font
 - Variable substitution
 - Bookmark and HTML anchors

© Copyright IBM Corporation 2013

13

More expert advice techniques

Because every environment is different, you might have requirements for other options of using expert advice.

A simple method is to enter straight text as your expert advice. You can use HTML and even variable substitution to pass values to the expert advice when you access the situation event results workspace.

The following table summarizes the pros and cons of each of these expert advice techniques.

Table 1. Pros and cons of expert advice techniques

Technique	Pros	Cons
Plaintext	Easy to set up	<ul style="list-style-type: none"> Limited number of characters More difficult to maintain No text formatting
Redirect to a website or HTML document	<ul style="list-style-type: none"> Easy to set up More flexibility and number of characters Text formatting based on website or HTML document No character limitations in HTML document 	<ul style="list-style-type: none"> One URL for each situation, which must be maintained in the Situation editor No situation event details substituted
Formatted HTML using access to a single HTML document that uses anchors	Most flexible	<ul style="list-style-type: none"> Requires HTML knowledge Number of characters limited

Lesson 2. Embedded and correlated situations

Lesson 2: Embedded and correlated situations

- Includes other situations in a formula
- Easy setup
- Available situations depend on affinity
- Cannot combine multiple-result row attributes with other multiple-result row attributes

© Copyright IBM Corporation 2013

14

What this lesson is about

This lesson covers the use of embedded and correlated situations. With embedded and correlated situations, you can combine one or more situations with an already existing situation to make a new situation.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the difference between an embedded situation and a correlated situation.
- Create an embedded situation.
- Create a correlated situation.

Embedding situations into other situations can be compared to subroutines in programs.

The situations that you can include in other situations depend on the following elements:

- The managed system affinity
- The attribute group you chose when creating the situation

If you create a situation for multiple-result row attributes, only single-result row attribute situations can be combined with the existing formula.

Embedded situations

- With embedded situations, you can compare attribute conditions and situations within a given situation
- Embedded situations make it easy to take an existing situation and add a condition to it that is based on a situation
- Embedded situations have some restrictions:
 - You cannot use display items with an embedded situation
 - The properties of the embedding situation, such as monitoring interval, override the properties of the embedded situation

© Copyright IBM Corporation 2013

15

Embedded situations

Embedded situations make it easy to combine existing situations into new situations. Embedded situations contain at least one attribute and a situation result.

Embedded situation example: PrimeShift

Create a situation that will be evaluated only during prime shift (8 a.m. through 5 p.m.)

The screenshot shows a 'Formula' dialog with a table:

	Process Command Name	PrimeShift
1	... ('httpd2-prefork')	= = True
2		
3		

The screenshot shows a 'Select condition' dialog with the 'Condition Type' set to 'Situation Comparison'. The 'Situation Name' list includes:

- Linux_Process_stopped
- Linux_Process_Stopped_2
- Linux_RPC_Bad_Calls
- Linux_RPC_Bad_Calls_2
- Linux_System_Thrashing
- Linux_System_Thrashing_2
- MS_Offline
- NonPrimeShift
- PrimeShift** (highlighted with a red box)
- QMECAMON_ONLINE
- Weekday
- Weekend

© Copyright IBM Corporation 2013

16

Embedded situation example: PrimeShift

An example of embedding situations is the **prime shift** situation. In this situation, you want to monitor for this condition either during working hours or outside of working hours.

1. Create the situation that monitors prime shift.

This situation is not associated with any Navigator item. A **PrimeShift** situation is already included with IBM Tivoli Monitoring.

2. Embed this situation into another situation that monitors a condition you only during working hours.

If you are creating the situation outside of working hours, you can test the situation by exchanging **PrimeShift** with **NonPrimeShift**. **NonPrimeShift** hides the situation events during prime shift hours.

Correlated situations

- Correlated situations are made up entirely of other situations
- No attributes show in the formula notebook page, only other situations
- You can create new situations that are true when a combination of other situations becomes true
- For example, this situation is true when two of the three situations it is based on become true

Formula			
<i>fx</i>			
	Apache_Down_Linux @VM01:LZ	Apache_Down_Linux @VM02:LZ	Apache_Down_Linux @VM03:LZ
1	<code>== True</code>	<code>== True</code>	
2	<code>== True</code>		<code>== True</code>
3		<code>== True</code>	<code>== True</code>
4			

© Copyright IBM Corporation 2013

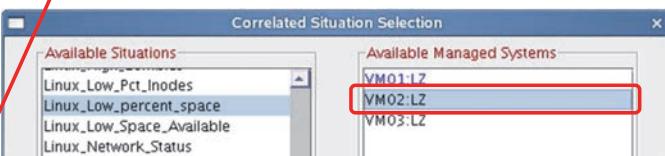
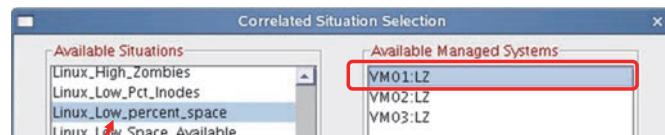
17

Correlated situations

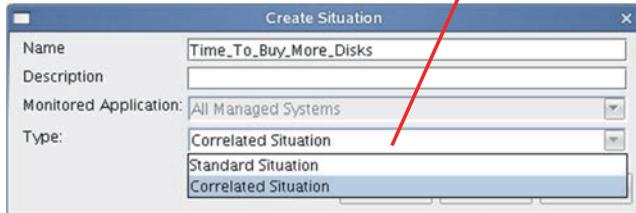
Correlated situations do not contain attributes, just other situations.

Correlated situations (continued)

- When you create situations combining attributes from different managed systems, select Correlated Situation from the menu
- Correlated situations run at the hub monitoring server



Correlated situation



Formula	
fx	Linux_Low_percent_space @VM01:LZ
1	= = True
2	= = True
3	

© Copyright IBM Corporation 2013

18

Correlated situations (continued)

Correlating situations enables comparing situation results from both of the following types of systems:

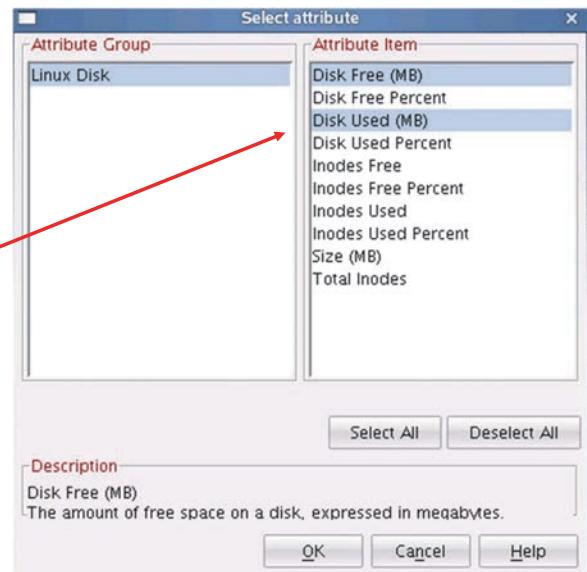
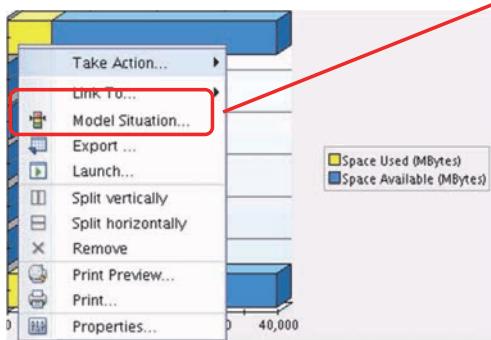
- Multiple types of managed systems
- The same type of managed system for attributes that otherwise cannot be combined in one formula

Correlated situations are evaluated at the hub monitoring server. This example assumes that you monitor disk space on two Linux systems. This example requires the use of **correlated** situations because the situations are running on different managed systems. This situation event is displayed only when both situations are true.

Lesson 3. Creating situations from a model

Lesson 3: Creating situations from a model

- Use monitored data to create situations
 - Include historical data
 - Use statistical functions
- Create the Model Situation from a data view
 - Chart view
 - Table view



© Copyright IBM Corporation 2013

19

What this lesson is about

This lesson covers the creation of situations from current or historical data. These situations are called ***model situations***.

What you should be able to do

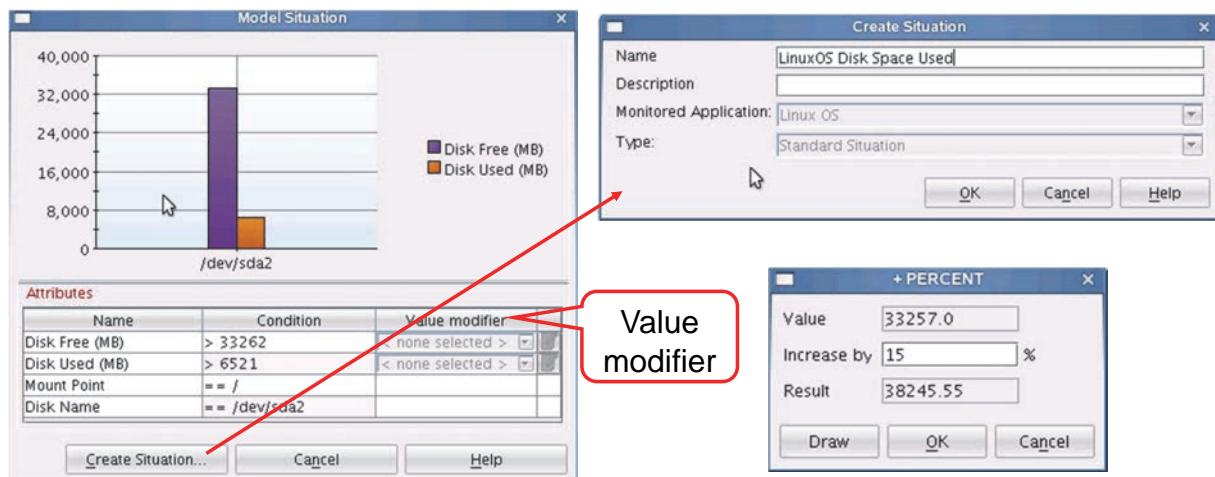
After completing this lesson, you should be able to perform the following tasks:

- Create a situation from current data.
- Create a situation from historical data.

You can create situations by modeling real-time data. Right-click an object in a chart view to pass the current attributes to the situation modeler, along with the current values.

Passing attributes from the view

- Adjust condition values as needed
- Add Value modifier +/-



Modify the value by percentage +/-

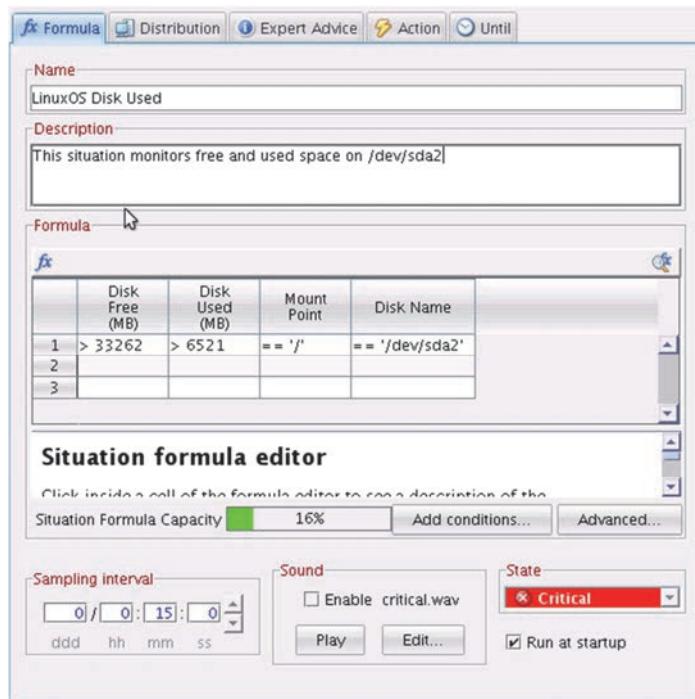
© Copyright IBM Corporation 2013

20

Passing attributes from the view

Any eligible key condition attribute is automatically selected and is not viewable in the list. You can adjust the values by a percentage plus or minus. The modified value passes to the Situation editor. When you have set the attributes to your satisfaction, click **Create Situation**. Type a situation name and optional description, and click **OK** to see the new situation in the Situation editor.

The new situation from current data



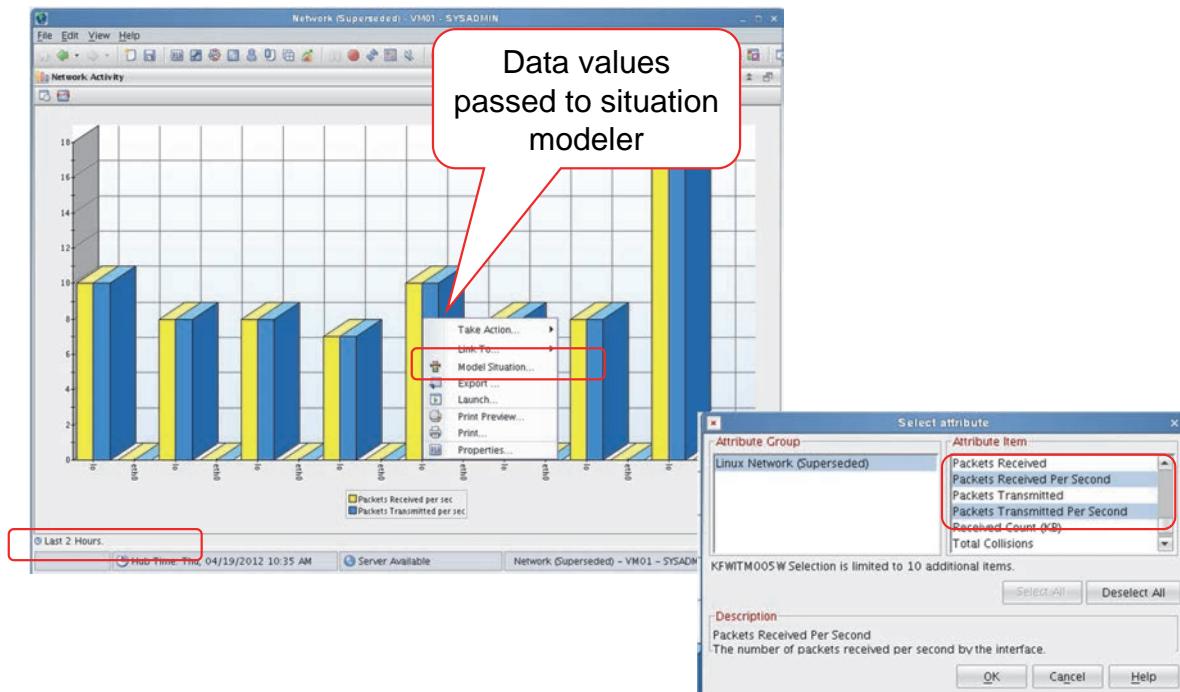
© Copyright IBM Corporation 2013

1-21

The new situation from current data

The Situation editor opens with the attributes and comparison values from the baseline section. Change the situation as needed, such as removing or adding attributes, and save the situation. The situation distributes to the managed system that hosted the view.

Using historical data to model situations



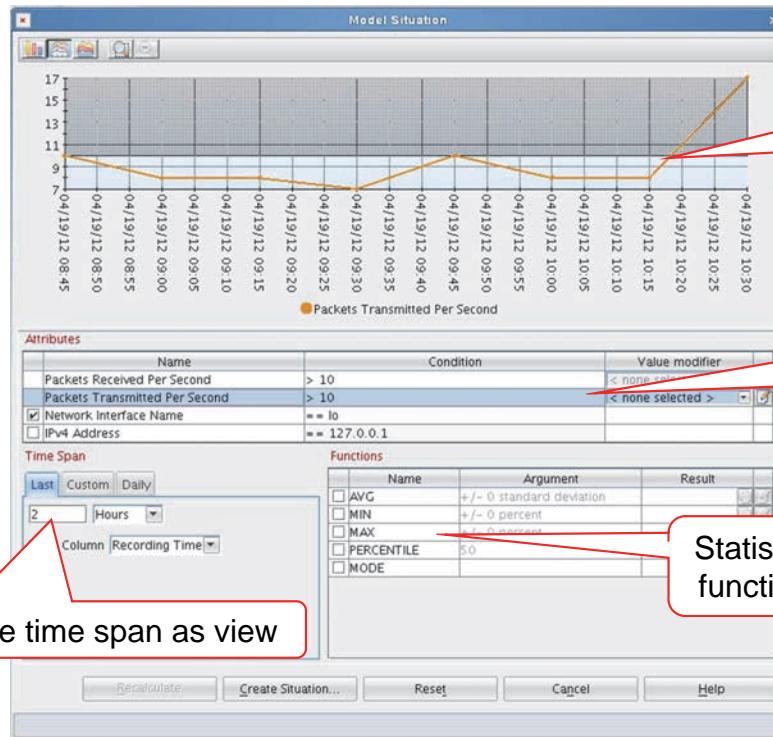
© Copyright IBM Corporation 2013

22

Using historical data to model situations

You can use historical data to model a situation. Use statistical functions that help you select optimal threshold values when creating a situation formula. The historical data passes to the situation modeler as a plot chart.

Situation model editor



© Copyright IBM Corporation 2013

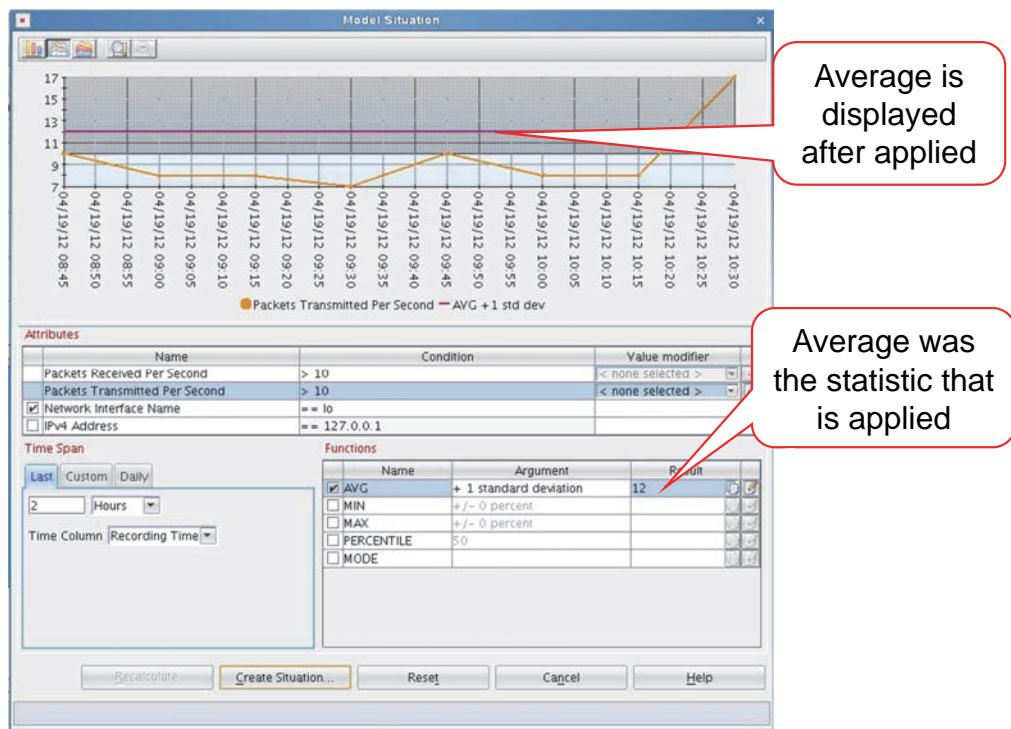
1-23

Situation model editor

The view passed the condition values. The values started from the point on the chart when the mouse invoked the menu. You can change the time span. Type a new value and click **Recalculate**.

When you click one of the numeric attributes, that attribute plots and the area above its value becomes shaded, as seen in this example. The shaded area represents the value that passed from the view for this attribute. You can click and drag the shaded area up or down to change the condition value.

Applying statistical functions



© Copyright IBM Corporation 2013

1-24

Applying statistical functions

Select a metric in the Functions table to calculate a statistical function and create a graph. This option assists in visualizing how the attribute behaves relative to its average, mode, minimum, and maximum.

Modifying statistical functions

After you modify the statistics, you can create the situation

Select Create Situation

Name	Condition	Value
Packets Received Per Second	> 10	< none
Packets Transmitted Per Second	> 12	< none
<input checked="" type="checkbox"/> Network Interface Name	= lo	
<input type="checkbox"/> IPv4 Address	= = 127.0.0.1	

Name	Argument	Result
<input checked="" type="checkbox"/> AVG	+ 1 standard deviation	12
<input type="checkbox"/> MIN	+/- 0 percent	
<input type="checkbox"/> MAX	+/- 0 percent	
<input type="checkbox"/> PERCENTILE	50	
<input type="checkbox"/> MODE		

Time Span: Last 2 Hours, Time Column: Recording Time

Functions: Recalculate, Create Situation...

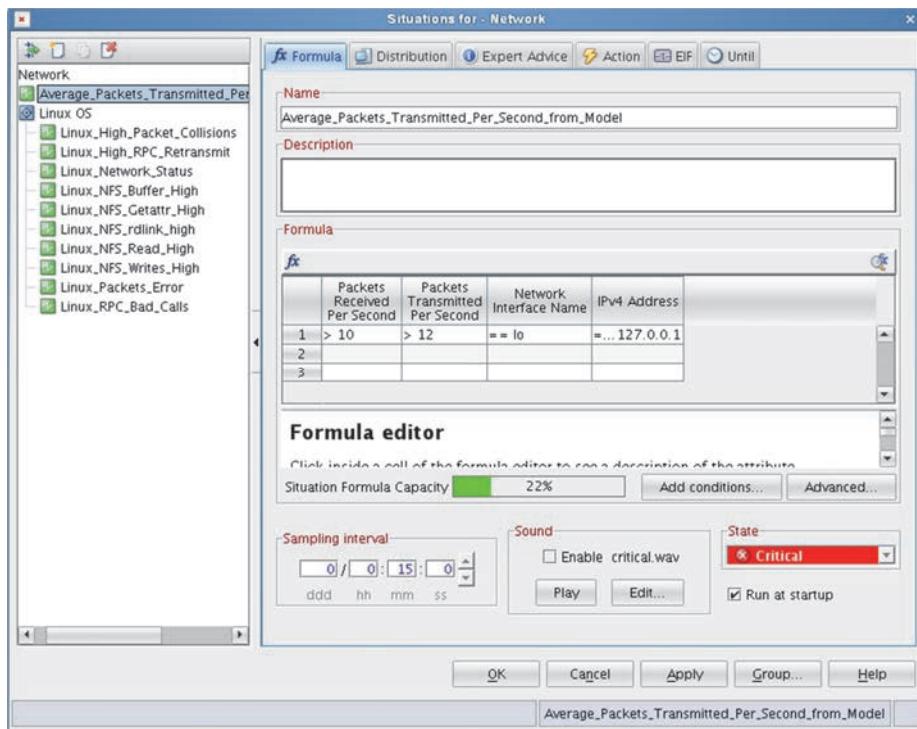
Create Situation Dialog:

- Name: Average_Packets_Transmitted_Per_Second_from_Model
- Description:
- Monitored Application: Linux OS
- Type: Standard Situation

Modifying statistical functions

You can apply modifiers to the condition values to further enhance the situation. Click the **Argument** field to enter standard deviations or plus or minus percentages. Clicking **Recalculate** applies any functions and calculates the result. Clicking **Copy** causes the modified condition value pass to the **Result** field.

The new situation from historical data



© Copyright IBM Corporation 2013

26

The new situation from historical data

The Situation editor opens, and the data that was calculated in the previous steps has been entered for you.

Lesson 4. Situation overrides for dynamic thresholds

Lesson 4: Situation overrides for dynamic thresholds

- Creating various threshold values for each agent with a single situation
- Overriding situation formula, based on the following factors:
 - Unique configuration or capacity of individual system
 - File server with much larger disk drives
 - Test environment where frequent thrashing is expected
 - Performance of a specific process or component (higher or lower processor utilization threshold for certain process or program)
 - Time variant thresholds
 - Night shift
 - Weekends or holidays
 - Comparison to historical values, or number of standard deviations from the mean

Note: Some situations are not eligible for dynamic thresholds

What this lesson is about

This lesson covers the creation of dynamic thresholds for a situation. In this process, a single situation can have multiple thresholds for individual managed systems.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a dynamic threshold for a situation.
- Describe how to adjust a dynamic threshold with calendar entries.
- Discuss the process to use historical data baselines to suggest a value to use as a dynamic threshold.

You can use dynamic thresholds to override situation settings on a single system, even if the situation distributes to multiple systems. You can base these overrides on the needs of the individual resource, service, line of business, or geographic location.

You can centrally manage the situation but have different comparison values on different managed systems. You can create time variant thresholds, which you can base on clock time or the calendar. You can also generate baselines for situation attributes based on historical data stored in the Tivoli Data Warehouse. Dynamic thresholds work with all distributed agents.

Some situations cannot use overrides. Situations not eligible for overriding include the following examples:

- Situations that are distributed only to the monitoring server
- Situations containing multiple attribute groups
- Situations containing expressions that involve the following aggregate column functions:
MIN, MAX, AVG, SUM, COUNT, CHANGE, PCTCHANGE
- Situations containing expressions involving these row functions:
STR, SCAN, MISSING, TIME, DATE
- Situations containing multiple expressions using the same attribute and the same relational operator. An example is `p_state != "running" AND p_state != "waiting"`.

Dynamic thresholds

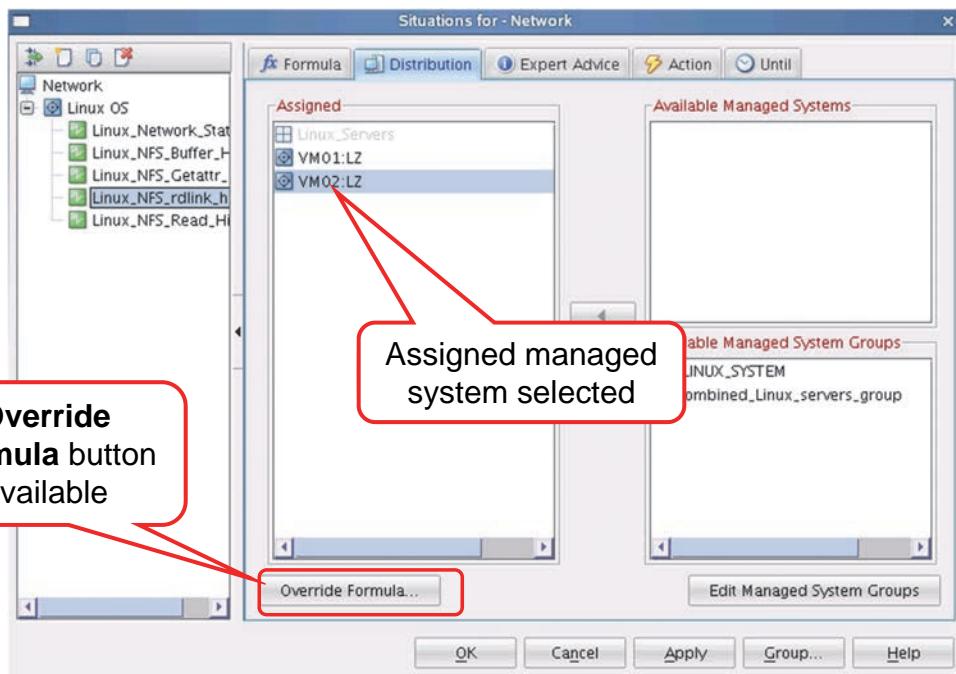
Capability for customizing monitoring by resource and time periods

- Company email server monitoring: central processor usage
 - Average is 60% in the morning
 - Average is 35% in the afternoon
 - Average is 5% off shift
- Linux: central processor usage
 - Average is 75%
 - Server A average is 60%
 - Server B average is 80% from 8:00 A.M. to 5:00 P.M.

Dynamic thresholds

A situation override is a set of expressions that substitute comparison values that are different from the values specified in the situation formula. You can apply these overrides to selected managed systems under specific conditions. Overriding a situation formula is useful if you need to adjust the situation comparison values. You can base such adjustments on local conditions or the current time.

Overriding a situation formula



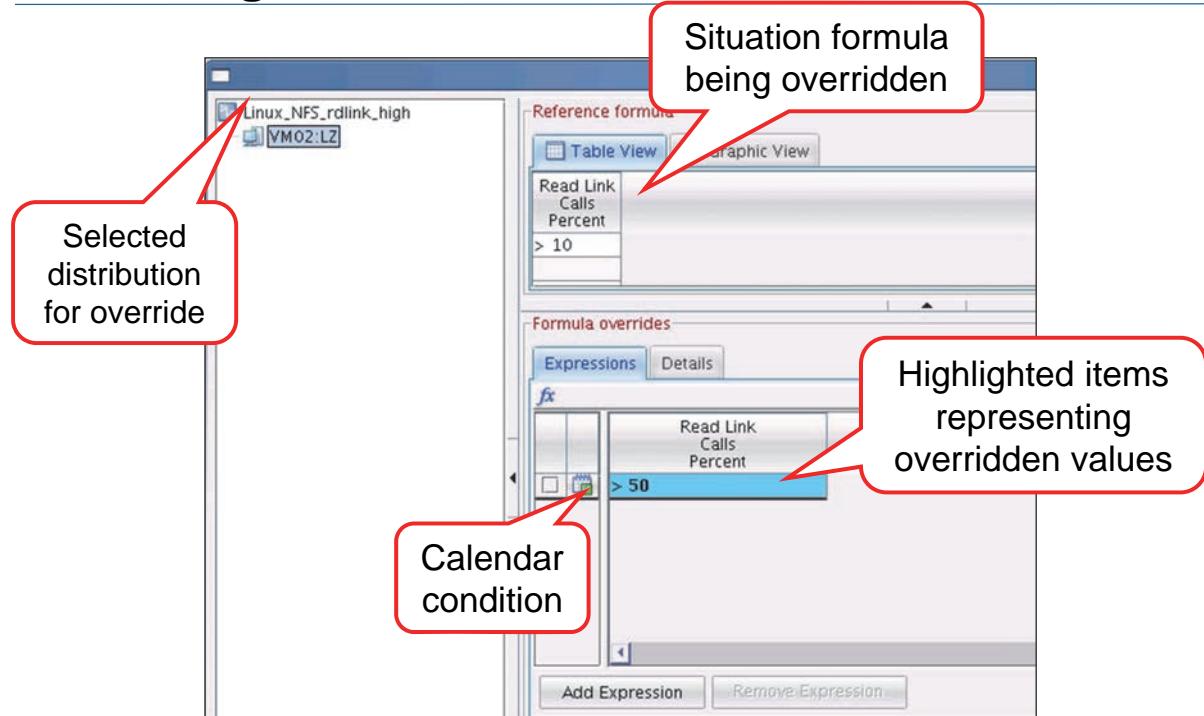
© Copyright IBM Corporation 2013

1-29

Overriding a situation formula

Apply situation overrides using the **Distribution** tab. You can set different formula overrides for distributing to multiple systems so that the base situation never needs changing.

Overriding a situation



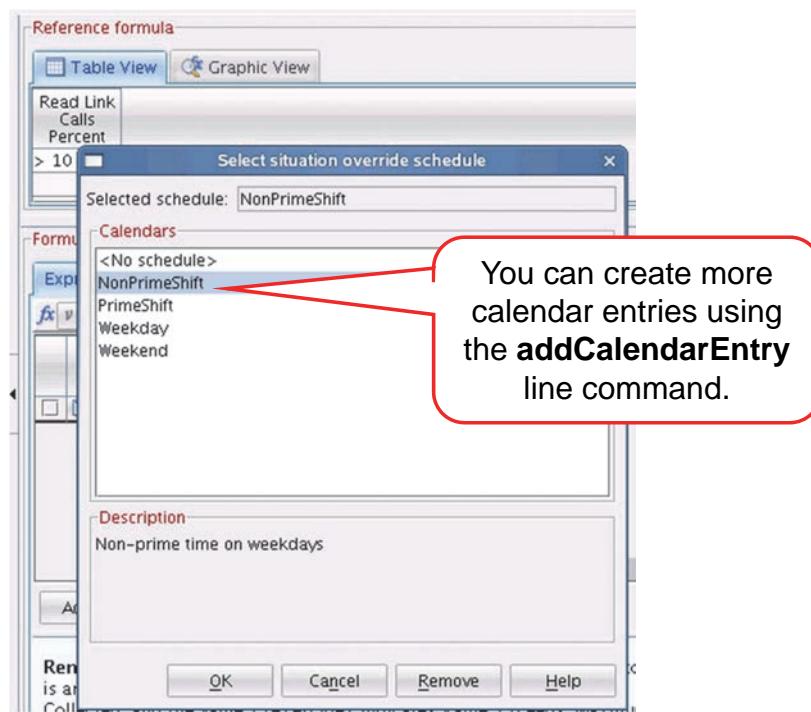
© Copyright IBM Corporation 2013

30

Overriding a situation

The values in the **Reference formula** area represent the original situation settings and cannot be changed. You can modify situation expressions for a specific managed system or managed system list in the **Formula overrides** area. Only situation attributes that are eligible for overriding are present in this area.

Applying a calendar schedule



© Copyright IBM Corporation 2013

1-31

Applying a calendar schedule

Tivoli Monitoring provides four calendar entries. You can use them so that the override applies only with a time, day, or date window. You can also set calendar overrides by using line commands.

Historical data baselines

- Calculating baseline (situation override) values
 - Overrides based on actual data, rather than arbitrary values (Based on historical data from the Tivoli Data Warehouse and agents)
 - Statistical functions for a situation attribute
- tacmd suggestBaseline
 - Calculates the baseline average
 - Start and end dates and times
 - Statistical functions, such as average, minimum, maximum, or standard deviations to the average
 - Capability to write output to a file (input to setOverride command)
- tacmd acceptBaseline
 - Has the same function as suggestBaseLine
 - Capability to calculate and commit baseline average
 - Single-command invocation

© Copyright IBM Corporation 2013

32

Historical data baselines

Use historical data to compute baselines. Before you generate a suggested baseline, determine which situation attributes are eligible for it. The **tacmd listSitAttributes** command shows you the attributes you can use for your baseline.

Historical baseline example

`./tacmd suggestBaseline`

```
-s Linux_Fragmented_File_System      <situation name
-u sysadmin                          <user ID
-m VM02                               <managed system name
-P "Space Used Percent LT 85"        <predicate (formula)
-f AVG+3                             <function Average, plus 3 standard
deviations
-d 1111005000000                      <start date: October 5, 2011, at midnight
-e 1111009120000                      <end date: October 9, 2011, at noon
-x /tmp/out.xml                       <write output to xml file
```

`./setOverride -x /tmp/out.xml -u sysadmin -w -h VM02`

Result: Replaces comparison value (85) with calculated baseline value

Historical baseline example

This example generates a baseline on the historical average over the range specified, plus three standard deviations from the mean. The application writes the results to a file, which you can read by using the **setOverride** command. You can also achieve the same results by running the **acceptBaseline** command using the same parameters.

Student exercise



© Copyright IBM Corporation 2013

34

Student exercise

Perform the exercises for this unit.

Review questions

1. How do you access expert advice for a situation event?
2. What are some of the attributes that you can substitute into expert advice?
3. What is the difference between embedded and correlated situations?

Review answers

1. How do you access expert advice for a situation event?

From the event results workspace. You can open the situation event results workspace by using either of the following methods:

- a. Clicking the link in the event flyover list that appears when you move the mouse over the event indicator
- b. Right-clicking the situation event in the situation event console and selecting **Situation Event Results**.

2. What are some of the attributes that you can substitute into expert advice?

You can substitute the following attributes:

- Situation name
- Monitoring server name
- Managed system name
- Display item (if set)
- Global time stamp
- Local time stamp
- Status

3. What is the difference between embedded and correlated situations?

Correlated situations are made up entirely of other situations. Embedded situations contain both attribute comparisons and other situations.

Summary

Now that you have completed this unit, you can perform the following tasks:

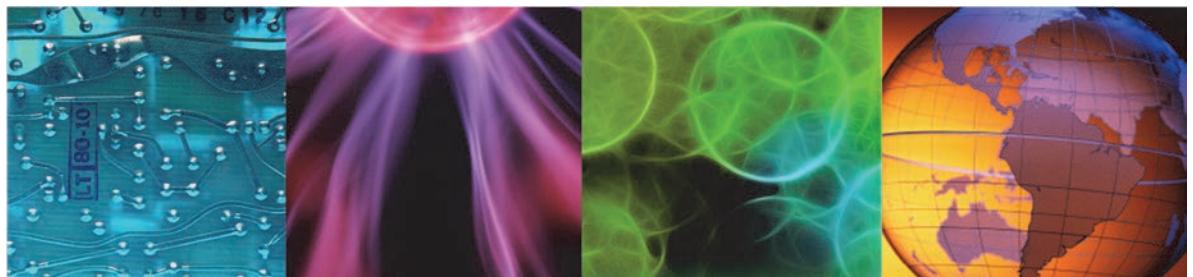
- Define expert advice
- Describe the use of embedded situations
- Build correlated situations that combine situations to create new situations
- Explain the differences between embedded and correlated situations
- Create situations based on current and historical data
- Construct thresholds that dynamically change for specific resources



4 Agent autonomy



4 Agent autonomy



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit covers how monitoring agents can function independently of the monitoring server.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises



Objectives

When you complete this unit, you can perform the following tasks:

- Describe the different levels of autonomy supported with agent autonomy
- Configure agents to run independently of a Tivoli Enterprise Monitoring Server
- Create private situations for monitoring agents
- Configure Netcool/OMNIbus to receive traps from monitoring agents
- Access and use the Agent Service Interface for monitoring agents

Lesson 1. Agent autonomy introduction

Lesson 1: Agent autonomy introduction

- Minimize gaps in event monitoring and data collection

The agents no longer must be dependent on the connection to the Tivoli Enterprise Monitoring Server
- Improve first failure reporting
- Events persistent over agent restart
- Improve monitoring availability and accuracy
- Increase Tivoli Monitoring product scalability
- Export events to independent event manager (OMNIbus)
- You can create private monitoring based on local criteria and needs

What this lesson is about

This lesson introduces the basics of agent autonomy.

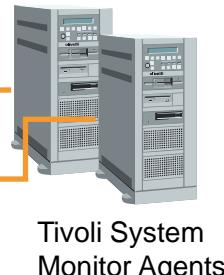
What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

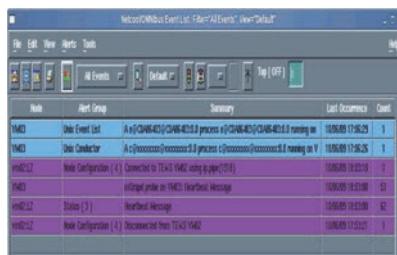
- Describe the benefits of agent autonomy
- Explain the various levels of agent autonomy

Agent autonomy overview

- Agents running in autonomous mode can perform these tasks:
- Run private situations
 - Send events as SNMP traps
 - Collect private historical data and warehouse private data



Tivoli System Monitor Agents



Netcool/OMNIbus

Show Table Information - MySQL Events						
SUSE Linux - Entertainment - News - Internet Search - References - Maps and Directions - Shopping - People and Companies - IBM						
Service interface						
Name	Definition	Type	Length	Minimum	Maximum	ENAMES
ORIGINNODE	KL2_CPM_System_Name	String	16	-	-	
TIMESTAMP	KL2_CPM_Timestamp	String	16	-	-	
CPUID	KL2_CPM_CPU_ID	String	4	-1	2147483647	4 - Aggregate
USRCPU	KL2_CPM_User_CPU	String	1	0	10000	
USRCPU	KL2_CPM_User_Non_CPU	String	1	0	10000	
SVCCPU	KL2_CPM_Service_CPU	String	1	0	10000	
ELSCPU	KL2_CPM_Exec_CPU	String	1	0	10000	
BLDCPU	KL2_CPM_Busy_CPU	String	1	0	10000	
WATCPU	KL2_CPM_Wait_O_CPU	String	1	0	10000	
USERBSCPU	KL2_CPM_User_Byte_Hd	String	1	-10000	10000	

Agent Service Interface

- Situations
- Historical data
- Queries



Tivoli Enterprise Monitoring Agents

© Copyright IBM Corporation 2013

4

Agent autonomy overview

Both Tivoli System Monitor Agents and Tivoli Enterprise Monitoring Agents can run in autonomous mode. All Tivoli System Monitor Agents are autonomous. They cannot be installed with any Tivoli Monitoring Services components. Tivoli Enterprise Monitoring Agents can run in autonomous mode, or they can run connected to a monitoring server.

Because an agent running in autonomous mode is not connected to a monitoring server, the statuses of the situations are not seen on the situation event console. They can send SNMP traps to SNMP event managers such as Netcool/OMNIbus. Another way to determine the status of the situations within an agent running in autonomous mode is to use the Agent Service Interface.

Agent autonomy features

- Traditional monitoring agents are connected to a monitoring server
- Monitoring agents can run unconnected to any monitoring server
- Complete agent features are available regardless of operation mode
- Agent begins monitoring immediately
 - Starts persistent enterprise situations and history data collection
 - Starts private situations and history data collection
 - Private situations are invisible to enterprise manager
- Private situation events emit SNMP traps (v1, v2, and v3)
- Enterprise situation events persist on local storage and upload to monitoring server upon reconnection
- Agent maintains basic situation operation statistics

Four levels of agent autonomy: Level 1

Base autonomous function

- No configuration is required
- If a monitoring agent loses connection to the monitoring server, the monitoring agent can continue running situations autonomously
When the agent reconnects, it uploads events that occurred while the agent was disconnected
- Some situations that use group functions cannot be evaluated completely on the monitoring agent alone, which means they cannot run
- You can view agent data using the service interface console, but not the portal client
- Base autonomous activities happen automatically on all agents that use the Tivoli Monitoring Services framework

© Copyright IBM Corporation 2013

6

Four levels of agent autonomy: Level 1

By default, agents that are included with IBM Tivoli Monitoring 6.3 have some level of autonomous agent capability.

Four levels of agent autonomy: Level 2

Monitoring agents can event *autonomously*

- Tivoli Monitoring events are typically routed through the hub monitoring server to the event receiver
Note: This method is not required
- The Tivoli Enterprise Monitoring Agent must connect to the monitoring server at least one time to receive enterprise situation definitions
- The user must place a trap configuration file in the agent installation and restart the agent to activate this function

Four levels of agent autonomy: Level 2

With Tivoli Monitoring 6.3, the agent can emit SNMP events directly to the event receiver. A monitoring server connection is no longer required for the agent to emit events.

Four levels of agent autonomy: Level 3

Private situations

- Agents can use locally defined situations to operate autonomously
- With Tivoli Monitoring 6.2.2 (and later), the agent can read locally defined, private situation definition files
- Private situations can also be used on an agent that connects to a Tivoli Enterprise Monitoring Server
However, private situations never report to the Tivoli Enterprise Monitoring Server
- With Tivoli Monitoring 6.2.3 Fix Pack 1 (and later), private situation events can be sent using SSL communication
- To use private situations, you must do three tasks:
 - Create a trap configuration file and provide it to the agent
 - Create a situation configuration file and provide it to the agent
 - Restart the agent

© Copyright IBM Corporation 2013

8

Four levels of agent autonomy: Level 3

This level of autonomy eliminates the need for the autonomous agent to ever connect to Tivoli Monitoring Infrastructure.

Four levels of agent autonomy: Level 4

Private history

- Using the private situation definitions, you can define locally which tables for which you want to collect historical data
- The definitions are in the `<pc>_situations.xml` file, which is in the `<install_dir>/localconfig/<pc>` directory.
- The syntax accepts the table name, the subnode names, how often to collect the data, and how many hours to keep the data
Data that is older than the retention setting is discarded
- The data defined using this method is not sent to the Tivoli Data Warehouse and cannot be seen in the Tivoli Enterprise Portal
- You can view the data by using the Agent Service Interface
- With IBM Tivoli Monitoring 6.2.3 Fix Pack 1(and later), you can store private history data

The agent must never connect to the monitoring server for this function to be enabled

Although private history is defined using the XML files that are used for private situations, no situations must be defined.

Lesson 2. Agent autonomy installation and configuration

Lesson 2: Agent autonomy installation and configuration

- Agent autonomy is an enhancement to the Tivoli Monitoring Agent framework
- Agent autonomy represents agent capability improvement for all agents, rather than a special agent
- Install the Tivoli Monitoring application agent as documented
- Autonomous operation is activated by default
- No default XML files are created by default
 - You must create them for traps to be sent

© Copyright IBM Corporation 2013

10

What this lesson is about

This lesson explains the installation and configuration details of agent autonomy.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the updates needed to an event manager to receive SNMP traps from an autonomous agent.
- Configure an autonomous agent to forward traps to an event manager.

You can use the following parameters in the agent configuration file to control agent autonomy:

- IRA_AUTONOMOUS_MODE

This parameter turns on the agent autonomous mode feature (Y) or off (N). The default is Y.

- IRA_AUTONOMOUS_LIMIT

This parameter defines the autonomous saved event limit. If the specified value is a number, then it is the maximum number of situation events to be saved by the agent, such as 500. If the specification is in disk space units, such as KB, MB, or GB, then it is the total amount of disk space to be used by the agent for saving situation event data, such as 5 MB. The default is 2 MB.

- IRA_EVENT_EXPORT_CHECKUSAGE_INTERVAL

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

- CTIRA_STANDALONE

This parameter is now obsolete and has no effect.

- IRA_EVENT_EXPORT_SIT_STATS

This parameter enables (Y) or disables (N) basic situation operation statistics data collection. The basic situation data includes situation first start time, situation first event time, situation last start time, situation last stop time, situation last true event time, situation last false event time, number of times situation recycles, and number of times situation enters autonomous operation. The default is Y (enabled).

- IRA_EVENT_EXPORT_SIT_STATS_DETAIL

This parameter enables (Y) or disables (N) detail situation operation statistics data collection. The detail data collection includes eight days of situation operation profile, such as hourly true sample count, hourly false sample count, hourly data row count, hourly true sample ratio, and hourly false event ratio. The default is detail data collection disabled (N).

- IRA_EVENT_EXPORT_SNMP_TRAP

This parameter enables (Y) or disables (N) SNMP trap emitter capability. When enabled, the SNMP trap configuration file is required and must exist for the agent to emit SNMP v1, v2 Traps, and v3 Informs to configured SNMP event managers. The default is Y (enabled).

- IRA_EVENT_EXPORT_SNMP_TRAP_CONFIG

This parameter specifies the fully qualified SNMP trap configuration file name. The default file name **pc_trapcnfg.xml** is in **\$CANDLE_HOME/localconfig/pc**, where **pc** is the agent product code. On z/OS, the default file name is a PDS Member name **pcTRAPS** of **DDNAME RKANDATV**.

SNMP MIB and trap types

- **candlebase and cansyssg**
MIBs (on IBM Tivoli Monitoring Agent DVD tool in the **mibs** folder)
include Agent Autonomy SNMP trap specifications
- These MIBs must be loaded into Netcool/OMNIbus so the traps can be displayed

```

File Edit View Terminal Tabs Help
VM01:/mnt/code/agents/mibs # pwd
/mnt/code/agents/mibs
VM01:/mnt/code/agents/mibs # more cansyssg.mib
-----
--          SNMP Trap Definitions
--          IBM Tivoli Monitoring System MIB
-----
-- Sep 20, 2000: Added candleEvent Trap definition
-- Feb 18, 2005: Added more specific trap severities
-- Jan 30, 2007: Updated trap description text
-- Nov 05, 2008: Changed description to reflect general use
-- Jan 08, 2009: Add autonomous agent stat table and traps
-- Apr 30, 2009: Changed autonomous agent stat entries
-- May 06, 2009: Added autoSip-Interval
-- May 18, 2009: Added agentSip-Source
-----
Candle-SYSTEM-MIB DEFINITIONS ::= BEGIN
    IMPORTS
        enterprises
        DisplayString
        TimeStamp
        OBJECT-TYPE
        TRAP-TYPE
        candle-Alert-MIB,
        candle-System-MIB,
        CandleTimeStamp
        FROM RFC1155-SMI
        FROM RFC1213-MIB
        FROM SNMPv2-TC
        FROM RFC-1212
        FROM RFC-1215;
    candle
        OBJECT IDENTIFIER ::= { enterprises 1667 }
    candleArchitecture
        OBJECT IDENTIFIER ::= { candle 1 }
    candleProduct
        OBJECT IDENTIFIER ::= { candle 2 }
    candle-Base-MIB
        OBJECT IDENTIFIER ::= { candleArchitecture 1 }
    candle-System-MIB
        OBJECT IDENTIFIER ::= { candleArchitecture 2 }
    candle-Alert-MIB
        OBJECT IDENTIFIER ::= { candleArchitecture 3 }

```

© Copyright IBM Corporation 2013

11

SNMP MIB and trap types

Whatever SNMP manager you decide to use, you must import the Management Information Bases (MIBs) necessary to understand the traps that the autonomous agent sends. In this case, it is the **cansyssg** MIB.

SNMP trap types

There are three types of traps:

- AgentStatusEvent
- AgentSitSampledEvent
- AgentSitPureEvent

The details for what is sent with each type of trap are in the following table.

Table 2. Information that is sent with SNMP trap types

Trap name	Trap type	Information sent
AgentStatusEvent	20	agent life cycle status situation name origin node time stamp category severity status text interval (heartbeat status only)
AgentSitSampledEvent	21	application name table name situation name origin node time stamp situation context sample interval source category severity predicates attribute list
AgentSitPureEvent	22	application name table name situation name origin node time stamp situation context source category severity predicates attribute list

Simple SNMP trap configuration XML example



The screenshot shows a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", and "Help". The main area contains the following XML code:

```
<SNMP>
<TrapDest name="VM04" Address="192.168.100.104" BindAddress="192.168.100.102" />
<Situation name="*" target="VM04" />
<StatTrap name="EE_HEARTBEAT" sev="1" interval="1" cat="3" />
</SNMP>
```

When the autonomous mode is entered, this trap configuration file sends all situation events to an SNMP manager at IP address 192.168.100.104.

It also sends a heartbeat message named EE_HEARTBEAT every minute.

© Copyright IBM Corporation 2013

13

Simple SNMP trap configuration XML example

Trap configuration files are not created by default, by using a command-line interface, or by using a graphical user interface. You must create them manually by using XML. However, they can be simple XML files, as shown on the slide.

Note: This simple example sends SNMP traps for situations that are true even when the monitoring agent is connected to the monitoring server.

Another SNMP trap configuration file

```
<SNMP>
  <TrapDest name="US-East" Address="elsun01" />
  <TrapDest name="US-West" Address="elaix05" Version="v2" IP="6" />
  <TrapDest name="US-HQ"   Address="9.52.106.95"
    Community="{AES256:keyfile:a}P0hUrmUhCgfFwimS+Q6w+w=="
    STAT="Y" />

  <TrapAttrGroup Table="NT_Paging_File" TrapAttrList="Server_Name,%_Usage" />

  <situation name="*" target="US-HQ" />
  <situation name="Free_DiskSpace_Low" target="US-West" SEV="4" />
  <situation name="Is_KFC_Running" target="US-East" CAT="5" mode="HY" />
  <Situation name="NT_Paging_File_Critical" target="US-East" sev="5" cat="0"
    mode="HY" />

  <StatTrap name="EE_HEARTBEAT" Interval="5" />
</SNMP>
```

This trap configuration file shows the format of an SNMPv2 trap

Use trap configuration files to control where the traps are sent by situation name.

Agent autonomy messages

Netcool/OMNibus Event List : Filter="All Events", View="Default"		
Node	Event Type	Message
VM02:LZ	Threshold (0)	Linux_AMS_Alert_Critical ()
VM02:LZ	Threshold (0)	Linux_Process_High_Cpu
VM02:LZ	Threshold (0)	Linux_System_Thrashing
VM02:LZ	ITM_StatEvent	Entering Autonomous Operation Mode
VM02:LZ	ITM_StatEvent	Monitoring situation Linux_High_Zombies stopped
VM02:LZ	ITM_StatEvent	Exiting Autonomous Operation Mode
VM02:LZ	ITM_StatEvent	Monitoring situation Linux_High_Zombies stopped
VM02:LZ	Node Configuration (4)	Connected to TEMS VM02_RTEMS using ip.pipe(1918)
VM02:LZ	Node Configuration (4)	Disconnected from TEMS VM02_RTEMS
VM02:LZ	Status (3)	Entering Autonomous Operation Mode

These messages were received when the Linux OS agent was disconnected from the remote monitoring server and was running autonomously

This message was received when the Linux OS agent was reconnected to the remote monitoring server and is no longer running autonomously

Agent autonomy messages

Agents can submit traps to inform you when they are running autonomously and when they are not.

Agent autonomy messages (continued)

The screenshot shows the Netcool/OMNIbus Event List window with the following details:

Node	Alert Group	Summary	Last Occurrence	Count
VM02:LZ	Threshold (0)	Calculator_Down	1/19/2012 01:33:38 P	1
VM03	Unix Event List	A e@CDA86467@CDA86467:0.0 process e@CDA86467@CDA86467:0.0 running on V	1/19/2012 12:54:10 P	1
VM03	Unix_Conductor	A c@xxxxxxxx@xxxxxxxx:0.0 process c@xxxxxxxx@xxxxxxxx:0.0 running on V	1/19/2012 12:54:08 P	1
VM02 RTEMS	ITM_ManagedSystem	MS Offline	1/19/2012 01:33:13 P	1
VM02:LZ	Node Configuration (4)	Disconnected from TEMS RTEMS	1/19/2012 01:15:11 P	1
VM02:LZ	Status (3)	Monitoring situation Linux_High_Zombies stopped	1/19/2012 01:15:11 P	1
RTEMS	ITM_ManagedSystem	MS_Offline[(Status-*OFFLINE AND Reason<>"FA") ON RTEMS (Status-*OFFL)]	1/19/2012 01:09:30 P	1
VM03		mttrapd probe on VM03: Heartbeat Message	1/19/2012 01:08:04 P	10
VM02:LZ	Node Configuration (4)	Connected to TEMS RTEMS using ip.pipe(1918)	1/19/2012 01:05:10 P	1
VM02:LZ	Node Configuration (4)	Disconnected from TEMS RTEMS	1/19/2012 12:59:03 P	1

After the remote monitoring server is shut down again, the Linux OS agent is again disconnected from the hub monitoring server and a message is issued. The situation Calculator_Down is able to send a trap to Netcool, even though it is not connected to a monitoring server.

Lesson 3. Private situations

Lesson 3: Private situations

Private situation features

- You can create agent monitoring criteria that is pertinent to local environment and operation needs
- A local administrator creates human-readable XML that persists across agent restart
- Private situations
 - Are invisible to Tivoli Enterprise Portal centrally managed infrastructure existence, monitoring data, or events
 - Are limited to simple predicate expression
 - Run concurrently with enterprise situations, but do not interfere, supersede, override, or reconfigure enterprise situations
 - Use all available agent features and services, such as pure and sample situation processing, reflex automation, operation log, history data collection, and traces
 - Have events delivered immediately through SNMP traps

© Copyright IBM Corporation 2013

17

What this lesson is about

This lesson discusses private situations, which can submit traps to a local event management solution.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a private situation from an existing enterprise situation.
- Build a private situation manually.
- Create definitions to collect private history.

Two ways to create private situations

- The tacmd commands can export enterprise situation definition XML, which can be input to an agent
 - tacmd operands to use are:
 - bulkexportsit (for multiple situations)
 - viewsit (for a single situation)
 - This method is useful in migrating existing enterprise situations and duplicating them into private situations
 - Beware of dual notification if you use this method
- Text editor manually
 - This method is useful when creating new situations that are not existing enterprise situations

You can have a private situation and an enterprise situation with the same name

There are two syntaxes for private situations, depending on how you create the private situation.

Private situation configuration specification

- Implements basic predicate logic
 - Expandable (by default up to 10) attribute triplets
 - A triplet contains these three items:
 - Attribute
 - Operator
 - Threshold
 - Operators consist of EQ, NE, GT, LT, GE, LE
 - Triplets are connected by all AND or OR logic connectors
 - *MISSING function supported
 - Attribute scale and enumeration supported
 - Support reflex automation command; you can specify:
 - Command to run
 - Run for each item
 - Run every time it is true, or only the first time
 - Support history data collection

© Copyright IBM Corporation 2013

19

Private situation configuration specification

Private situations provide most of the same capabilities that situations provide.

XML elements for private situations

<PRIVATECONFIGURATION>

Root element

<PRIVATESET>

<SITUATION>

NAME=

INTERVAL=

CRITERIA=

<CMD>

<AUTOSOPT>

WHEN=

FREQUENCY=

WHERE=

<DISTRIBUTION>

<HISTORY>

<WAREHOUSE>

AUTOSOPT is required if an action <CMD> is specified

DISTRIBUTION is required for products with subnodes (subagents)

Optional

Optional

© Copyright IBM Corporation 2013

20

XML elements for private situations

This slide shows the format of private situations that are created manually.

Simple private situation example

```
<Situation Name="Check_Free_DiskSpace" Interval="001500"  
          Criteria="*VALUE NT_Logical_Disk.Disk_Name *EQ  
          C: *AND  
          *VALUE NT_Logical_Disk.%_Free *LE 5" >  
</Situation>
```

Or

```
<Situation>Check_Free_DiskSpace</Situation>  
<Interval>001500</Interval>  
<Criteria>  
  <![CDATA[=*VALUE NT_Logical_Disk.Disk_Name  
  *EQ C: *AND *VALUE NT_Logical_Disk.%_Free *LE 5" >  
</Criteria>
```

This XML shows two ways to create a situation named Check_Free_DiskSpace that checks every 15 minutes for the C: drive having less than 5% free space

Simple private situation example

This slide shows the two syntaxes and how they can both perform equivalent functions.

Private situation example with reflex automation

```
<privateconfiguration>
<pripvatesit>
    <situation name="Check_Process_CPU_Usage" interval="000100"
        criteria="![CDATA[
            *VALUE NT_Process_64.%_Processor_Time *GE 5
            *AND *VALUE NT_Process_64.Priority_Base *NE 0
            *AND *VALUE NT_Process_64.Process_Name *NE _Total ]]>
    </situation>
    <cmd>
        <![CDATA[echo "&{NT_Process_64.Process_Name}" >>
        "c:\temp\CPU_Process.out"]]>
    </cmd>
    <autosopt when="N" frequency="N" ></autosopt>
</pripvatesit>
</privateconfiguration>
```

This XML shows a situation named Check_Process_CPU_Usage that runs a command when the conditions are met

© Copyright IBM Corporation 2013

22

Private situation example with reflex automation

Private situations support simple reflex automation.

Private situation configuration *MISSING support

- Identical to enterprise situation format
- Exceptions:
 - Wildcards are not supported
 - Numeric values are not supported
 - 10 maximum values in missing list
 - 63 maximum characters per value

*Private situation configuration *MISSING support*

The MISSING function is so vital to monitoring for an attribute that its absence indicates a problem, for example, the name of a process that you expect to be running.

Correct examples of *MISSING support

- NT_Missing_Process:
 *IF *MISSING NT_Process.Process_Name *EQ ('System')
- Linux_Missing_Antivirus:
 *IF *MISSING Linux_Process.Process_Command_Name *EQ ('rtvscand', 'symcfgd')
- Linux_Antivirus_Not_Running:
 *IF *VALUE Linux_Process.State *NE Trace *AND *MISSING
 Linux_Process.Process_Command_Name *EQ ('rtvscand', 'symcfgd')

© Copyright IBM Corporation 2013

24

*Correct examples of *MISSING support*

You can use the *MISSING function to look for more than one missing attribute.

Incorrect examples of *MISSING support

- INIT_Missing:

- *IF *MISSING Linux_Process.Process_ID *EQ ('1')
or
▪ *IF *MISSING Linux_Process.Process_ID *EQ (1)

Numeric values are not supported

- BAD_Linux_Antivirus_Not_Running:

*IF *MISSING Linux_Process.Process_Command_Name
*EQ ('rtvscand', 'symcfgd') *AND *VALUE Linux_Process.State *NE Trace

MISSING must be the last entry in a logical AND sequence

- BAD_Linux_Virus_Problem:

*IF *VALUE Linux_Process.Command_Name *EQ 'BigBadVirus' *OR
*MISSING Linux_Process.Command_Name *EQ ('rtvscand',
'symcfgd')

OR logic is not supported

© Copyright IBM Corporation 2013

25

Incorrect examples of *MISSING support

You cannot use the *MISSING function for these situations:

- Search for numeric values
- In anything but the first stage of an *AND sequence
- With OR logic

Private history

- Uses existing short-term history capability
 - Output local binary history and header files
 - File trim frequency, size, and retention period
- Private history file name: PVTHIST.tablename
- Run concurrently with enterprise defined history data collection of the same application table
- Private history data can be warehoused into the Tivoli Data Warehouse for reporting using the <WAREHOUSE> element
- Read private history data using the Service Interface request

© Copyright IBM Corporation 2013

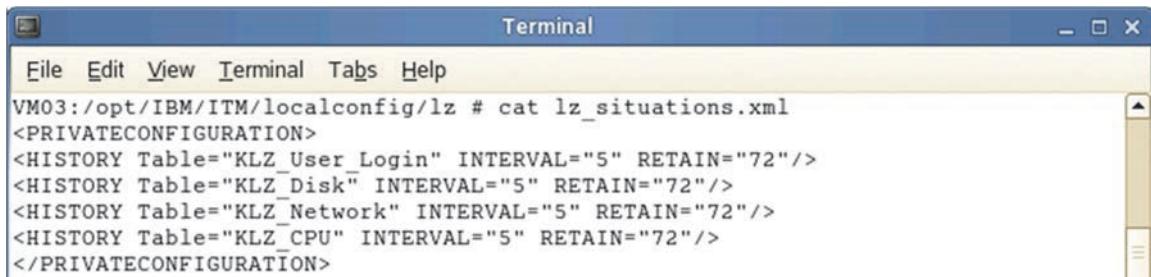
26

Private history

You can use private history to examine private situations data over a period of time.

Simple private configuration file for historical data collection

- This simple example collects four Linux OS agent attribute groups
- You can get the names of the attribute groups from the ATTR file on Linux and UNIX located at:
`/opt/IBM/ITM/li6263/lz/tables/ATTRLIB/klz.atr`



The screenshot shows a terminal window titled "Terminal". The command `cat lz_situations.xml` is run, displaying the following XML code:

```
VM03:/opt/IBM/ITM/localconfig/lz # cat lz_situations.xml
<PRIVATECONFIGURATION>
<HISTORY Table="KLZ_User_Login" INTERVAL="5" RETAIN="72"/>
<HISTORY Table="KLZ_Disk" INTERVAL="5" RETAIN="72"/>
<HISTORY Table="KLZ_Network" INTERVAL="5" RETAIN="72"/>
<HISTORY Table="KLZ_CPU" INTERVAL="5" RETAIN="72"/>
</PRIVATECONFIGURATION>
```

© Copyright IBM Corporation 2013

27

Simple private configuration file for historical data collection

Do not modify the attribute files. However, you can examine them to determine the names of the attribute groups you can use to collect historical data.

The syntax of the History statement is listed here:

<HISTORY> Optional. Use the history element to specify each attribute group that you want to collect historical data for. The agent does not support multiple <HISTORY> specifications for the same TABLE. The XML parser processes duplicated <HISTORY> specifications as update scenarios. The final updated attribute value is the value in effect and is always output to the agent's Operation Log.

TABLE= This parameter specifies the application attribute group name.

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

INTERVAL= Optional. This parameter specifies the historical data collection interval in minutes. The minimum collection interval is 1 minute and the maximum is 1440 (24 hours). Valid intervals are values that divide evenly into 60 or are divisible by 60: an interval below 60 could be 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, and 30; an interval greater than 60 could be 120, 180, 240, and so on, up

to 1440. If you enter an invalid value, no history will be collected for the specified attribute group. The default is 15.

RETAIN= Optional. This parameter defines the short-term history data retention period in hours. The default is 24 hours, and the minimum retention period is 1 hour. There is no limit other than that imposed by storage space on the computer. After the retention limit has been reached, the oldest data samples are deleted as new samples arrive. The default is **24**.

Lesson 4. Netcool/OMNIbus integration

Lesson 4: Netcool/OMNIbus integration

- Requires Netcool/OMNIbus V7.2.1 FP10 or later installed; Netcool/OMNIbus ObjectServer
- Requires IBM Tivoli Monitoring Agent installation DVD; MIB and rule file

© Copyright IBM Corporation 2013

28

What this lesson is about

This lesson describes the configuration necessary so that Netcool/OMNIbus can receive private situations.

What you should be able to do

After completing this lesson, you should be able to configure Netcool/OMNIbus to receive private situations.

Netcool/OMNibus integration steps

- Copy rule files from agent installation DVD

ibm-TIVOLI-CANSYSSG-MIB.include.snmtrap.lookup
ibm-TIVOLI-CANSYSSG-MIB.include.snmtrap.readme
ibm-TIVOLI-CANSYSSG-MIB.include.snmtrap.rules

to Netcool/OMNibus Probe location \$OMNIHOME/probes/arch/.

For example:

- C:\Program Files\IBM\tivoli\Netcool\omnibus\probe\win32
- Examine probe properties file \$OMNIHOME/probes/arch/mttrapd.props:
 - Ensure that UDP or ALL protocol property is set
 - Check RulesFile property specification; typically mttrapd.rules
 - Check MIBDirs property specification

If you do not install the multithreaded TRAPD probe, Netcool/OMNibus cannot receive the situation events forwarded from the Tivoli Enterprise Monitoring Server.

However, you do not have to install and configure the TRAPD probe to receive SNMP traps sent from agents running in autonomous mode.

Netcool/OMNIbus integration steps (continued)

- Copy **CANSYSSG.MIB** from agent the installation DVD to Netcool/OMNIbus Probe location (MIBDirs specification)
\$OMNIHOME/common/mibs/

CANSYSSG.mib requires these common SNMP MIB files:

- IANAifType-MIB.mib, RFC1155-SMI[rfc1155].mib, RFC-1212.mib, RFC1213-MIB.mib, RFC1213-MIB[rfc1213].mib, RFC1215-TRAP.mib, SNMPv2-SMI.mib, SNMPv2-C[rfc2579].mib

They can be downloaded from the Internet.

- Edit default rule file (RuleFile specification)
\$OMNIHOME/probes/arch/
- Add lookup table reference as the first definition
 - Add rules reference in proper processing order

© Copyright IBM Corporation 2013

30

Netcool/OMNIbus integration steps

The rules allow for Netcool/OMNIbus to provide duplicate detection and problem resolution when situation events are closed.

The following text is the **mttrapd.props** file used for this course.

```
#####
#
# Licensed Materials - Property of IBM
#
#
#
# (C) Copyright IBM Corp. 2002, 2006, 2011. All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication
# or disclosure restricted by GSA ADP Schedule Contract
# with IBM Corp.
#
```

```
# =====
# Module Information:
#
# DESCRIPTION:
# mttrapd.props
#
#
# Ident: $Id: mkprops 1.21 2003/08/20 15:29:09 csmith Development $
#
#####
#####
#
# Property NameDefault
#
# Generic Properties
#
# AuthPassword      : ''
# AuthUserName      : ''
# AutoSAF          : 0
# Buffering         : 0
# BufferSize        : 10
# Help              : 0
# LookupTableMode   : 3
# Manager           : 'mttrapd'
# MaxLogFileSize    : 1048576
# MaxRawFileSize     : -1
# MaxSAFFFileSize   : 1048576
# MessageLevel      : 'warn'
# MessageLog        : '$OMNIHOME/log/mttrapd.log'
# MsgDailyLog       : 0
# MsgTimeLog        : '0000'
# Name              : 'mttrapd'
# NetworkTimeout    : 0
# PollServer         : 0
# Props.CheckNames   : TRUE
# PropsFile          : '$OMNIHOME/probes/<arch>/mttrapd.props'
# RawCapture         : 0
# RawCaptureFile     : '$OMNIHOME/var/mttrapd.cap'
# RawCaptureFileAppend: 0
# RawCaptureFileBackup: 0
# RetryConnectionCount: 15
```

```

# RetryConnectionTimeOut: 30
# RulesFile           : '$OMNIHOME/probes/<arch>/mttrapd.rules'
# SAFFileName         : ''
# Server              : 'NCOMS'
# ServerBackup        : ''
# StoreAndForward     : 1
# Version             : 0
#
# Specific Properties
#
# BindAddress         : ''
# ConfPath            :
'$OMNIHOME/probes/<arch>:/export/build/tmp/${USERNAME} or
${LOGNAME}/solaris/Omnibus36/var'
# FlushBufferInterval : 0
# Heartbeat           : 60
# LogStatisticsInterval: 0
# MIBDirs             : '$OMNIHOME/common/mibs'
# MIBFile             : '$OMNIHOME/probes/<arch>/mib.txt'
# MIBs                : 'ALL'
# NoNameResolution    : 0
# NoNetbiosLookups   : 0
# NonPrintableAsHex  : 0
# PersistentDir       : '$OMNIHOME/var'
# Port                : 162
# Protocol            : 'UDP'
# QuietOutput         : 1
# SleepTime           : 1
# SocketSize          : 8192
# TrapQueueMax       : 20000
# snmpv3ONLY          : 0
# snmpv3MinSecurityLevel : 1
# DSALog              : 0
# DSAPeriod           : 30
#
#####
#
# Add your settings here
#
#####

```

```
Protocol      : 'UDP'  
RulesFile    : '$OMNIHOME/probes/linux2x86/mttrapd.rules'  
MIBDirs     : '$OMNIHOME/common/mibs'
```

Lesson 5. Agent Service Interface

Lesson 5: Agent Service Interface

- Agent Service Interface performs these main functions:
 - Accepts and validates client requests
 - Dispatches requests to appropriate agent components for processing
 - Gathers and formats reply data
 - Sends response data and status to request originator
- Uses the IBM Tivoli Monitoring Basic Services, sharing the same default 1920 port for all agents on a system
 - Each agent registers as an Internet server application
- Uses XML over HTTP/HTTPS for communication
- Product provides basic client program as a starting point

What this lesson is about

This lesson explains how to retrieve data from an agent running autonomously.

What you should be able to do

After completing this lesson, you should be able to retrieve data from an autonomous agent from the Agent Service Interface.

Reference

IBM Tivoli Monitoring Administrator's Guide

Use the Agent Service Interface to retrieve information from an installed agent, whether it is a Tivoli Enterprise Monitoring Agent or Tivoli System Monitor Agent. Log in to the local operating system to get a report of agent information, private situations, private histories, queries, and attribute descriptions and current values.

Agent Service Interface features

- Between client program and Tivoli Monitoring agent:
 - How to access the agent components and monitoring data
 - How to initiate agent service request, request format, parameters, and procedures (HTTP POST method)
 - How to interpret agent response, data format, and status
- Between agent components:
 - How to publish component services and make them available
 - How to receive service requests, request details, and processing requirements

Agent Service Interface requests

- Agent Property Information
- Private Situation Configuration
- Application Attribute File List
- Application Attribute File
- Application Subnode List
- Application Table Report
- Application Situation List
- Application Situation Statistics
- Application Situation Summary
- Private Situation Control
- Read Situation History
- Logon Authentication Data

© Copyright IBM Corporation 2013

33

Agent Service Interface requests

For more information about the Agent Service Interface requests, refer to the *IBM Tivoli Monitoring Administrator's Guide*.

Agent Service Interface example

Point your browser to the 1920 port of the agent you want to inspect

Click here

You must provide credentials

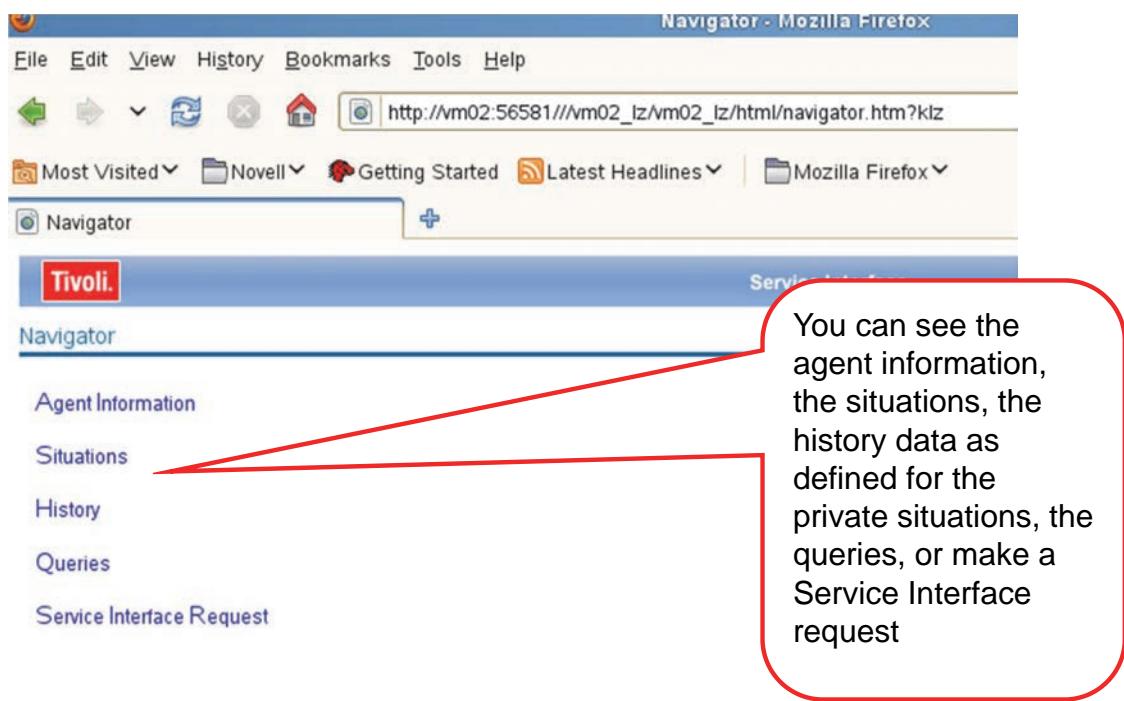
© Copyright IBM Corporation 2013

34

Agent Service Interface example

You can use Microsoft Internet Explorer or Mozilla Firefox to access the Agent Service Interface.

Agent Service Interface example (continued)



© Copyright IBM Corporation 2013

35

From the navigator screen, you can select what you want to see using the Agent Service Interface.

Agent Service Interface: Agent information

Attribute	Value
HOSTNAME	VM02
NODENAME	VM02_LZ
SUBSYSID	
NODEINFO	Linux<
PRODUCT	LZ
VERSION	06.23.00
LEVEL	A=00 Ii6263 C=06 23.00.00 II6263 G=06 23.00.00 II6263
PATCHLEVEL	A=00 Ii6263 C=06 23.00.00 II6263,G=06 23.00.00 II6263;
AFFINITY	%IBM STATIC 134 000000000000J0u0a4
BOOTTIME	Thu Apr 19 13:56:45 2012

```

Running /opt/IBM/TM/Ii6263/lib/kzagent>> /opt/IBM/TM/logs/VM02_LZ_1334872600.log 2>&1
• Thu Apr 19 13:56:40 GMT+8 2012
• _=/28605*/usr/bin/env
• ALSA_CONFIG_PATH=/etc/alsa-pulse.conf
ATTRLIB=/opt/IBM/TM/Ii6263/lib/tables/ATTRLIB
CANDLEHOME=/opt/IBM/TM
CLASSPATH=.:/opt/IBM/TM/RE/Ii6263/lib:/opt/IBM/TM/classes/InstallRAS.jar:/opt/IBM/TM/classes/IiTMclasses/cnp_vbjorball.jar:/opt/IBM/TM/classes/cnp.jar:/opt/IBM/TM/classes/chart.jar:/opt/IBM/TM/classes/install_common.jar:/opt/IBM/TM/classes/lcu4_3_4.jar:/opt/IBM/TM/classes/lp_de.jar:/opt/IBM/TM/classes/lp_es.jar:/opt/IBM/TM/classes/lp_ko.jar:/opt/IBM/TM/classes/lp_fr.jar:/opt/IBM/TM/classes/lp_ja.jar:/opt/IBM/TM/classes/lp_zh_CN.jar:/opt/IBM/TM/classes/lp_zh_TW.jar:/opt/IBM/TM/classes/lp_ru.jar:/opt/IBM/TM/classes/lp_zh.jar:/opt/IBM/TM/classes/lp_zh_hist
• COLORTERM=gnome-terminal
• COMMAND=kzagent
• CPU=i686
• CSHEDIT=emacs
• CTIRA_HIST_DIR=/opt/IBM/TM/Ii6263/z/hist

```

You can see the details about the agent here

© Copyright IBM Corporation 2013

36

Agent Service Interface: Agent information

The agent information can show the product code of the agent, the software level, when the agent was started, and the environment variable settings.

Agent Service Interface: Situations

Situations		+
Tivoli.		Service Interface
Apache_Down_Linux (Not running)		
<hr/>		
Attribute	Value	
TYPE	Sampled	
INTERVAL	30	
ROWSIZE	1540	
FIRSTSTARTTIME	Tue Apr 17 05:48:51 2012	
LASTSTARTTIME	Tue Apr 17 05:48:51 2012	
LASTSTOPTIME	Tue Apr 17 05:58:53 2012	
FIRSTEVENTTIME	NA	
LASTTRUETIME	NA	
LASTFALSETIME	Tue Apr 17 05:58:49 2012	
TIMESRECYCLED	1	
TIMESAUTONOMOUS	2	
<hr/>		
DAY	DATE	Thu Apr 19 00:00:00 2012
	TRUESAMPLES	0
	FALSESAMPLES	0
	TRUERATIO	0.00%
	FALSERATIO	0.00%
	HOURROWS	00000000000000000000000000000000
	HOURTRUE	00000000000000000000000000000000
	HOURFALSE	00000000000000000000000000000000
<hr/>		
Linux_AMS_Alert_Critical		
<hr/>		
Attribute	Value	

These details are about all the situations defined on the agent

© Copyright IBM Corporation 2013

37

Agent Service Interface: Situations

You can see both private and enterprise situations with this tool.

Agent Service Interface: History

History Display - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://vm02.50187//vm02_lz/vm02_lz/html/historicaldatadisplay.html?klz

Most Visited Novell Getting Started Latest Headlines Mozilla Firefox

History Display

Tivoli.

Service Interface

Choose a history table

- KLZCPU
- KLZDISK
- KLZLOGIN
- KLZNET

Select time range

Start date: 4/1/2012

Start time: 11:00 AM

End date: 4/21/2012

End time: 11:00 AM

Click "Report" after making table, column, and time selections.

Report

Select all columns Deselect all columns

Selected	Column name	Column description
<input type="checkbox"/>	ORIGINNODE	System Name
<input type="checkbox"/>	TIME	
<input type="checkbox"/>	CPUTIME	
<input type="checkbox"/>	SYSFCPU	System CPU
<input type="checkbox"/>	IDLECPU	Idle CPU
<input type="checkbox"/>	BUSYCPU	Busy CPU
<input type="checkbox"/>	WAITCPU	Wait IO CPU
<input type="checkbox"/>	USRSYSFCPU	User Sys Pct
<input type="checkbox"/>	STEALCPU	Steal CPU Percent

© Copyright IBM Corporation 2013

38

Agent Service Interface: History

You can see only private history with this tool.

Agent Service Interface: History (continued)

The screenshot shows the Tivoli History Display interface. On the left, there's a sidebar with 'Tivoli' branding and a 'Service Interface' header. Below the header, it says 'Choose a history table' with options: KLZCPU, KLZDISK, KLZLOGIN, and KLZNET. It also has a 'Select time range' section with date and time inputs. A note says 'Click "Report" after making table, column, and time selections.' To the right is a 'Select columns for KLZCPU' dialog box. It lists various CPU states with checkboxes: ORIGINNODE, TIMESTAMP, CPUID, USRCPU, USRNCPU, SVSCPU, IDLE, BUSYCPU, WAITCPU, USRSYSCPU, and STEALCPU. Below this are 'Select all columns' and 'Deselect all columns' buttons. A red arrow points from the 'Report' button to a callout box that says 'Then click Report'. Another red arrow points from the 'Select all columns' button to a callout box that says 'Click the columns you want data collected from'. At the bottom, there's a table titled 'History' with columns: System Name, Timestamp, CPU ID, User CPU, User Nice, System CPU, Idle CPU, Busy CPU, Wait IO CPU, User Sys Pct, and Steal CPU Percent. The table contains several rows of data.

System Name	Timestamp	CPU ID	User CPU	User Nice	System CPU	Idle CPU	Busy CPU	Wait IO CPU	User Sys Pct	Steal CPU Percent
VM02.LZ	1120419134559000	-1	26	0	6	9908	32	0	433	-2
VM02.LZ	1120419134559000	0	26	0	6	9908	32	0	433	-2
VM02.LZ	1120419135059000	-1	30	0	9	9938	62	0	363	-2
VM02.LZ	1120419135059000	0	30	0	9	9938	62	0	363	-2
VM02.LZ	1120419135059000	-1	23	0	6	9948	52	0	330	-2
VM02.LZ	1120419135559000	0	20	0	6	9948	52	23	363	-2
VM02.LZ	1120419140045000	-1	10	0	10	9957	43	0	330	-2
VM02.LZ	1120419140045000	0	10	0	10	9957	43	0	330	-2

© Copyright IBM Corporation 2013

39

Agent Service Interface: Queries

Select the table that you want to inspect

Name	Display	Type	Length	Minimum	Maximum	ENUMS
ORIGINNODE	KLZ_CPU.System_Name	2	64	-1	2147483647	-1 Aggregate
TIMESTAMP	KLZ_CPU.Timestamp	2	16			
CPUID	KLZ_CPU.CPU_ID	4		-1	10000	
USRCPU	KLZ_CPU.User_CPU	1		0	10000	
USRNCPU	KLZ_CPU.User.Nice_CPU	1		0	10000	
SYSCPU	KLZ_CPU.System_CPU	1		0	10000	
IDLECPU	KLZ_CPU.Idle_CPU	1		0	10000	
BUSYCPU	KLZ_CPU.Busy_CPU	1		0	10000	
WAITCPU	KLZ_CPU.Wait_IO_CPU	1		0	10000	
USRSYSCPU	KLZ_CPU.User.Sys_Pct	1		-10000	10000	
STEALCPU	KLZ_CPU.Steal_CPU_Percent	4		-2	10000	-1 Not Available -2 Not Collected

ORIGINNODE	TIMESTAMP	CPUID	USRCPU	USRNCPU	SYSCPU	BUSYCPU	WAITCPU	USRSYSCPU	STEALCPU
VM02.LZ	1120419140321000	-1	86	0	26	112	0	330	-2
VM01.LZ	1120419140321000	0	86	0	26	112	0	330	-2

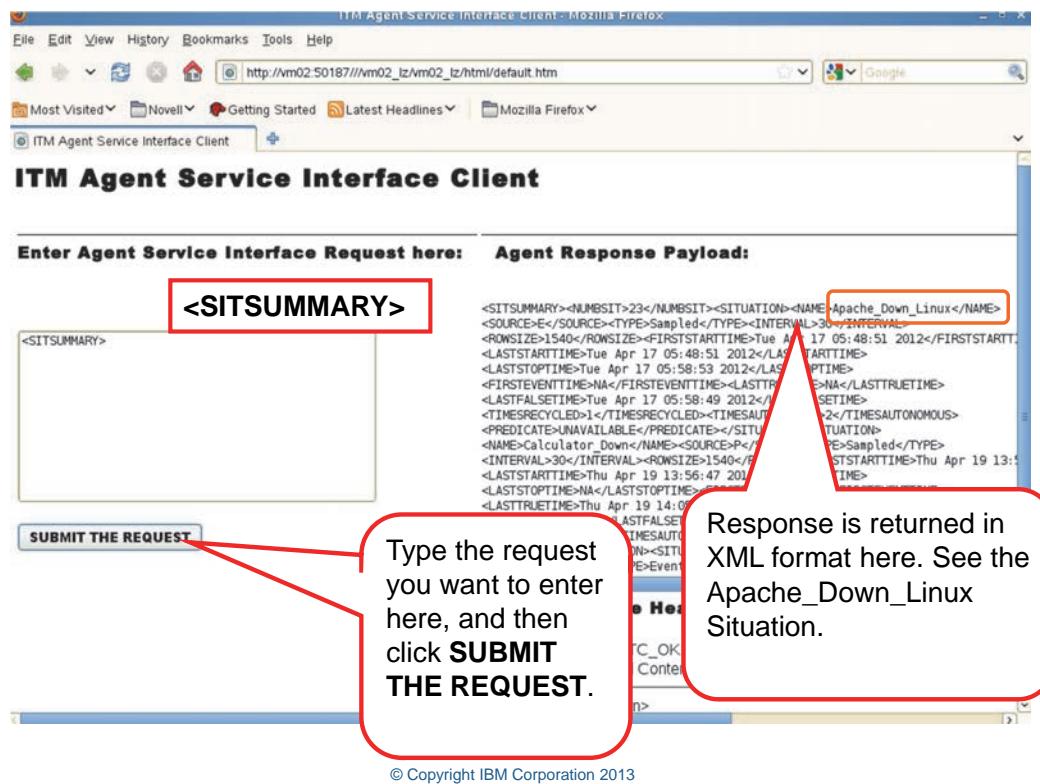
Data from the table you selected is displayed here

© Copyright IBM Corporation 2013

40

Agent Service Interface: Queries

Agent Service Interface: Service Interface request

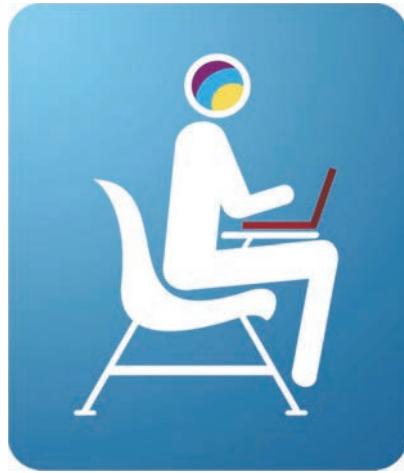


41

Agent Service Interface: Service Interface request

You can copy and paste the payload into a temporary file to view with a browser, which can format the XML to enhance the data's readability.

Student exercise



© Copyright IBM Corporation 2013

42

Perform the exercises for this unit.

Review questions

1. Do private situations show in the situation event console?
2. True or false: By default, the autonomous agent mode feature is enabled.
3. When a private situation sends a trap to Netcool/OMNIbus, can the trap be closed automatically when the private situation is closed?

Review answers

1. Do private situations show in the situation event console?

No, private situations do not show in the situation event console.

2. True or false: By default, the autonomous agent mode feature is enabled.

True. This feature is enabled by default.

3. When a private situation sends a trap to Netcool/OMNibus, can the trap be closed automatically when the private situation is closed?

*If the rule file provided with the agent installation media, **ibm-TIVOLI-CANSYSSG-MIB.include.snmptrap.rules**, is installed, then closing the private situation closes the previously sent trap.*

Summary

Now that you have completed this unit, you can perform the following tasks:

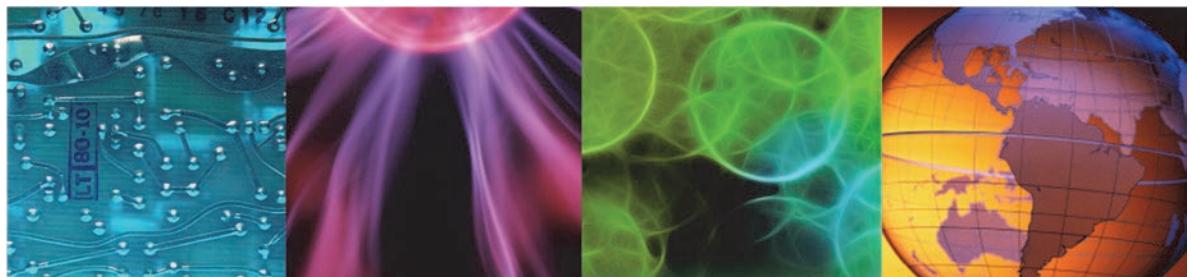
- Describe the different levels of autonomy supported with agent autonomy
- Configure agents to run independently of a Tivoli Enterprise Monitoring Server
- Create private situations for monitoring agents
- Configure Netcool/OMNIbus to receive traps from monitoring agents
- Access and use the Agent Service Interface for monitoring agents



5 Working with policies



5 Working with policies



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit covers how to automate problem resolution through actions and policies.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

Objectives

When you complete this unit, you can perform the following tasks:

- Use the Workflow Editor
- Create and validate a policy
- Issue commands from a policy
- Activate a policy
- Automatically solve a problem and notify users of the result

Lesson 1. The Workflow Editor

Lesson 1: The Workflow Editor

- You can implement policies to fully automate workflow strategies
- You create policies using a workflow editor
- Policies have the following characteristics:
 - Include a number of predefined activities
 - Use connector lines to control which activities run when and in what sequence
 - Run at the monitoring server

© Copyright IBM Corporation 2013

3

What this lesson is about

This lesson covers the Workflow Editor.

What you should be able to do

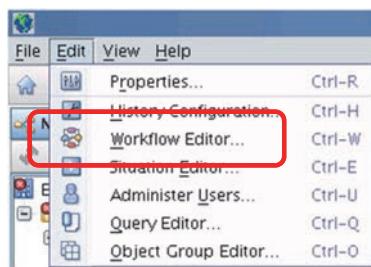
After completing this lesson, you should be able to perform the following tasks:

- Open the Workflow Editor.
- Describe the Workflow Editor user interface.

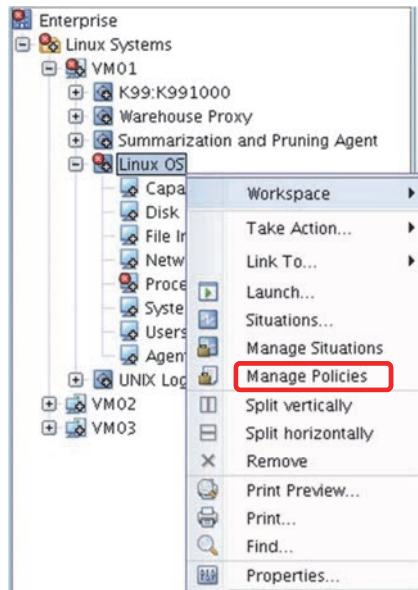
The next logical step in managing your enterprise is to provide the ability for the systems to repair themselves and to react to failed actions. Although these abilities might not be possible in all scenarios, Tivoli Monitoring provides an option to include workflow automation or policies. You can create elaborate solutions for solving a problem automatically by simulating steps that otherwise must be performed manually. With this option, you can have the system automate many standard operator tasks.

Accessing the Workflow Editor

- From the toolbar
- From the **Edit** menu



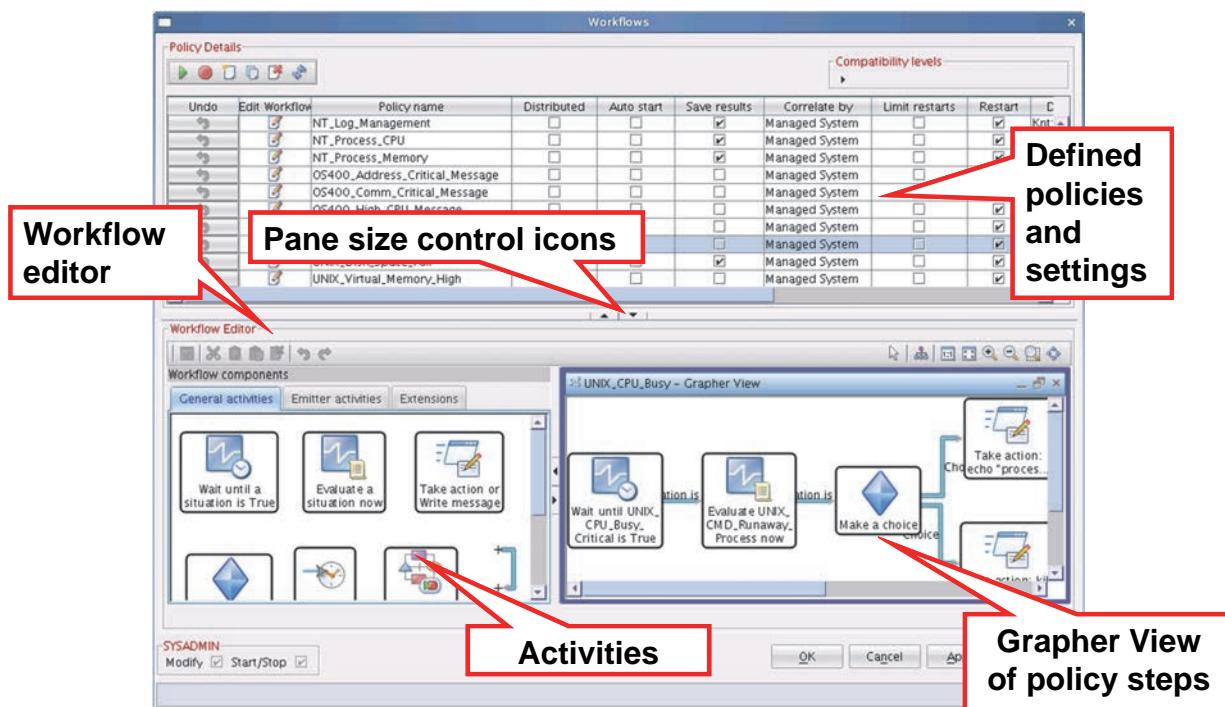
- By pressing **Ctrl+W**
- From the Manage Policies entry on a managed system, or System Navigator item



Accessing the Workflow Editor

You can access the Workflow Editor in these four ways.

The Workflow Editor: Content



© Copyright IBM Corporation 2013

5

The Workflow Editor: Content

The Workflow Editor consists of two panes:

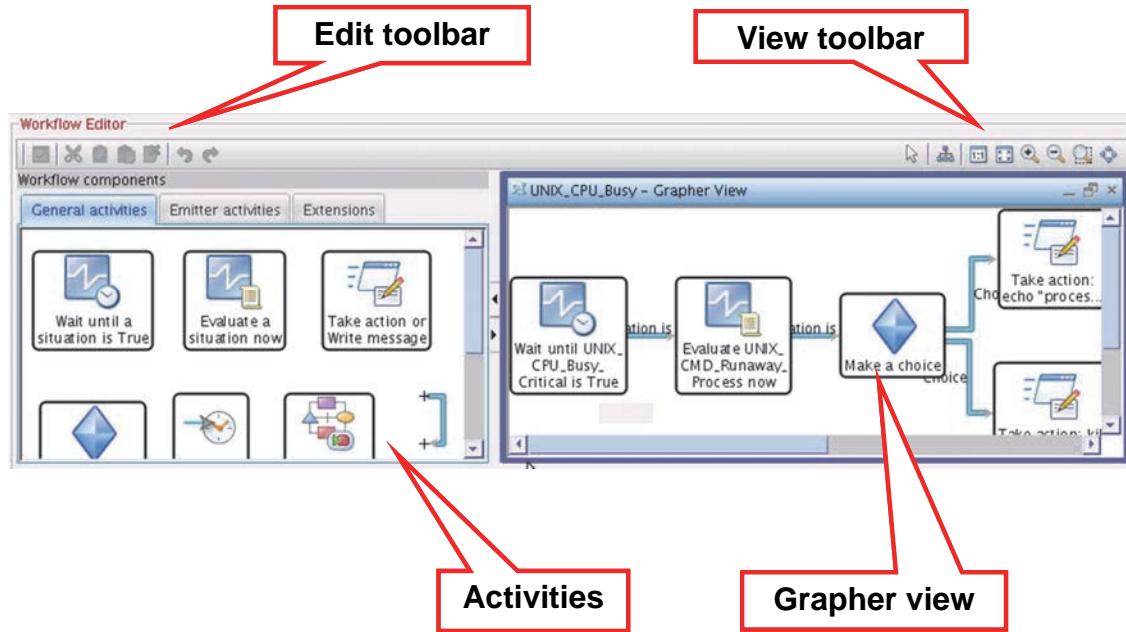
- The Policy Details pane lists all policies that are stored at the hub monitoring server. You can modify settings from here directly.
- In the Workflow Editor pane, you program your policy. This pane has three tabs on the left and one editor field on the right. It also contains an editing toolbar and a viewing toolbar.

The activities palette groups activities under a **General activities** tab, an **Emitter activities** tab, and an **Extensions** tab.

You can resize these two panes in the following ways:

- Dragging the border between the two panes
- Selecting the upward triangle on the border
- Selecting the downward triangle on the border

The Workflow Editor pane

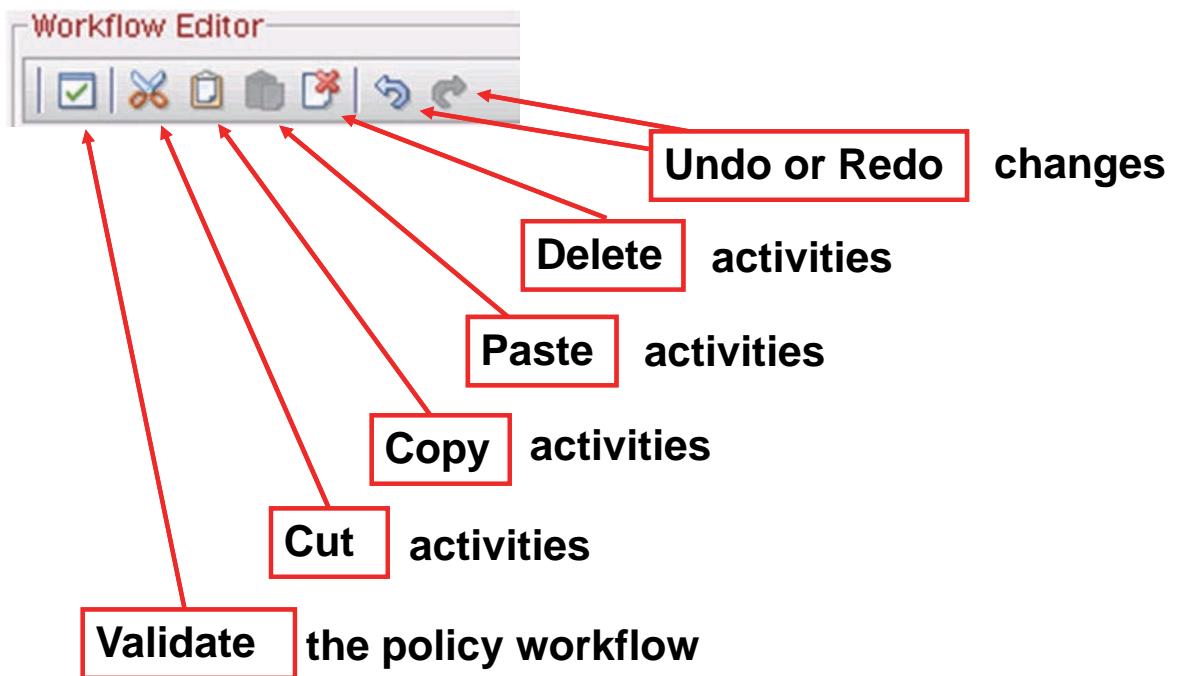


© Copyright IBM Corporation 2013

6

The Workflow Editor pane

Edit toolbar



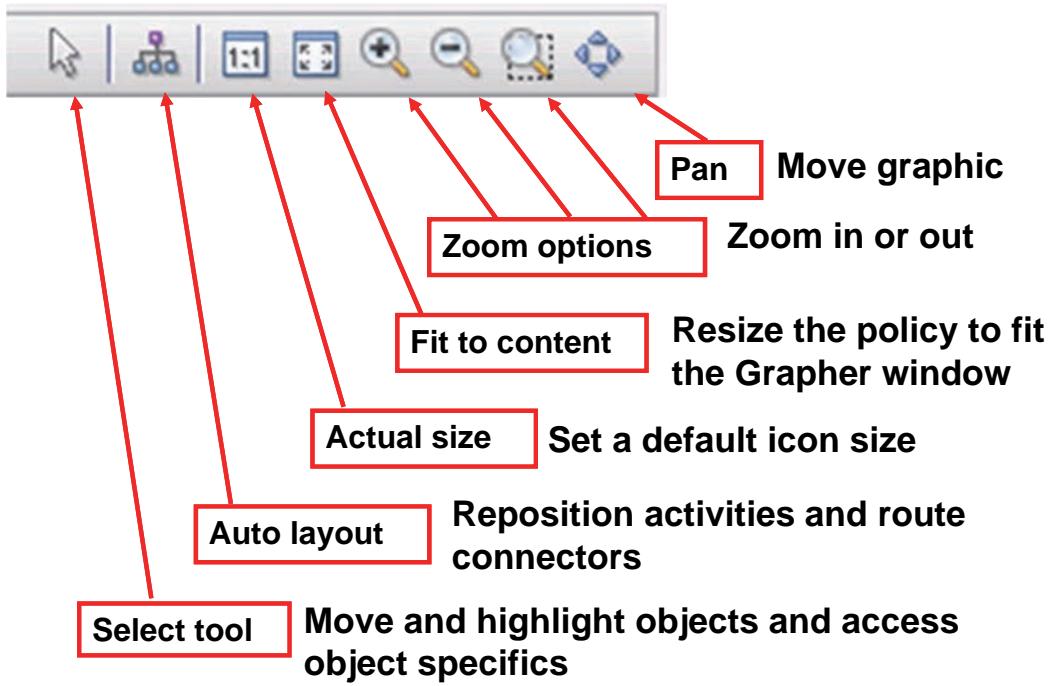
© Copyright IBM Corporation 2013

7

Edit toolbar

You can cut and paste groups of activities from one workflow into another.

View toolbar



© Copyright IBM Corporation 2013

8

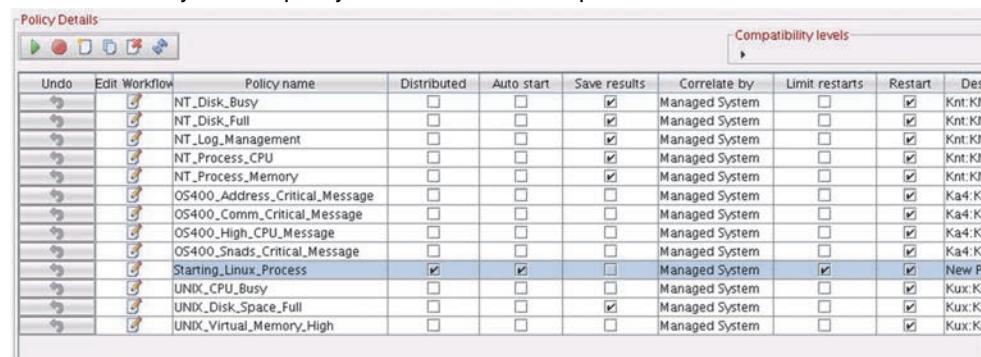
View toolbar

You can toggle the **Select** and **Pan** tools.

Policy details

You can view and modify policy settings in the Policy Details view by selecting or clearing check boxes.

- Policy name is a unique alphanumeric name up to 32 characters
You cannot modify it after the policy is applied
- Distributed distributes the policy to managed systems
- Auto start defines whether a policy is started automatically when the monitoring server starts
- Save results defines whether activity results are stored
- Correlate by defines the operation mode of the policy
- Limit restarts limits the restart of a policy to five times within 24 hours
- Restart recycles the policy after a thread is completed



The screenshot shows the 'Policy Details' window with a toolbar at the top containing icons for Undo, Redo, Edit Workflow, Save, Print, and Help. Below the toolbar is a table titled 'Compatibility levels' with columns for Undo, Edit Workflow, Policy name, Distributed, Auto start, Save results, Correlate by, Limit restarts, Restart, and Description. The table lists several policies such as NT_Disk_Busy, NT_Disk_Full, NT_Log_Management, NT_Process_CPU, NT_Process_Memory, OS400_Address_Critical_Message, OS400_Comm_Critical_Message, OS400_High_CPU_Message, OS400_Snads_Critical_Message, Starting_Linux_Process, UNIX_CPU_Busy, UNIX_Disk_Space_Full, and UNIX_Virtual_Memory_High. The 'Save results' checkbox is checked for most policies, while 'Distributed' is mostly unchecked. The 'Correlate by' column shows Managed System for most policies, except for Starting_Linux_Process which is set to New Pd. The 'Restart' column shows Knt:KN for most, except for Starting_Linux_Process which is set to New Pd. The 'Description' column shows Knt:KN for most, except for Starting_Linux_Process which is set to New Pd.

Undo	Edit Workflow	Policy name	Distributed	Auto start	Save results	Correlate by	Limit restarts	Restart	Description
↶	↶	NT_Disk_Busy	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Knt:KN
↶	↶	NT_Disk_Full	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Knt:KN
↶	↶	NT_Log_Management	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Knt:KN
↶	↶	NT_Process_CPU	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Knt:KN
↶	↶	NT_Process_Memory	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Knt:KN
↶	↶	OS400_Address_Critical_Message	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ka4:K4
↶	↶	OS400_Comm_Critical_Message	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ka4:K4
↶	↶	OS400_High_CPU_Message	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ka4:K4
↶	↶	OS400_Snads_Critical_Message	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ka4:K4
↶	↶	Starting_Linux_Process	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed System	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	New Pd
↶	↶	UNIX_CPU_Busy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Kux:KL
↶	↶	UNIX_Disk_Space_Full	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Kux:KL
↶	↶	UNIX_Virtual_Memory_High	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Managed System	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Kux:KL

© Copyright IBM Corporation 2013

9

Policy details

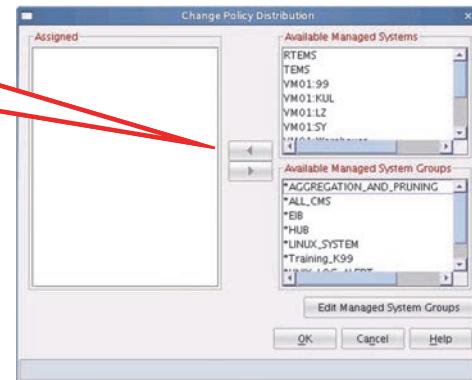
The *policy name* must be unique and describe the purpose of the policy.

Policy distribution

Policy distribution determines where situations are evaluated that run within a policy

Set the distribution by selecting the Distributed field in the policy settings.

In the menu, highlight all systems that you want to include and add them using the left arrow.



© Copyright IBM Corporation 2013

10

Policy distribution

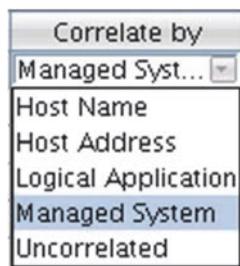
The distribution of a policy does not affect the distribution of situations that are used by the policy. The distribution of the situation and distribution of a policy should be consistent, although it is not a requirement.

You can distribute a policy to fewer agents than are included in the distribution of a situation. In this case, the policy does not respond to events from agents that are absent from the distribution of the policy. Do not distribute a policy to agents that are not contained in the distribution of the situation.

Policies run at the monitoring server to which the managed systems are connected.

Correlation modes

Correlation is used to define the path of execution that a policy uses for running activities on different managed systems.



- By **Host Name** or by **Host Address** correlates activities of monitoring agents running on the same physical machine.
- By **logical application** group correlates activities of monitoring agents that belong to an application group or other logical group.

Correlation modes

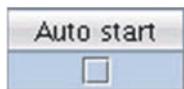
The **correlation mode** defines where activities are issued and which managed systems are involved during the execution of a policy.

Some examples are as follows:

- Correlating by **Host Name** or **Host Address** issues activities on the same node where situations were evaluated. These activities might include other agents running on the same host.
- When you correlate by **Logical Application Group**, you can run activities on managed systems that are part of the same managed system list.
- You can select **Uncorrelated** when it does not matter which system or set of systems has triggered the policy. For example, you might want a policy that pages an operator. It can wait until the situation that triggered the policy is no longer true, for any host that is running. You do not care which server has encountered the issue, only that a page is sent when any server first encounters a problem. The page that is sent can include the details of which server triggered the workflow. No further pages are sent until the situation is false on all servers.

Additional options

Other settings include these examples:



- Whether the policy runs at system startup



- Whether the policy is recycled after it finishes
The policy recycles, beginning at the triggering activity after all paths through the workflow have terminated



- Whether you want to limit restarting the policy to five times within 24 hours

Additional options

When a policy is set to **Auto start**, it starts together with the monitoring server.

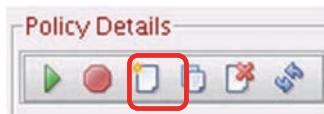
The **Restart** option defines that a policy recycles and starts again after the trigger situation is true.

Limit restarts is a troubleshooting feature that you can use to recycle the policy only four times within 24 hours, which prevents it from restarting continuously.

Lesson 2. Creating a policy

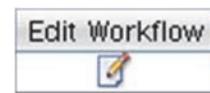
Lesson 2: Creating a policy

- Create a new policy

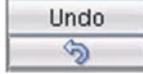


- Change the name and other properties

- Create the policy workflow



- Select activities and connect them in the order you want them to be run

- The Undo  icon is enabled for each policy if there are unsaved changes
- Save changes

© Copyright IBM Corporation 2013

13

What this lesson is about

This lesson introduces the step to create a workflow, or policy, with the Workflow Editor.

What you should be able to do

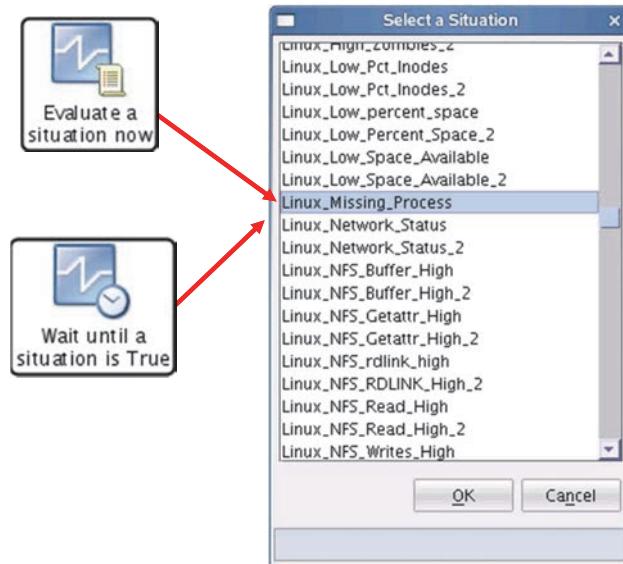
After completing this lesson, you should be able to perform the following tasks:

- Create a workflow.
- Describe various situation activities, emitter activities, and extensions.

When creating a policy, always begin by entering a name. After you apply the policy to the monitoring server using the **Apply** or **OK** button, you can no longer modify the policy name.

Situation activities

- **Evaluate a situation now** immediately evaluates a situation and continues with true or false
- **Wait until a situation is True** waits for a situation interval and suspends the workflow indefinitely until the situation becomes true



© Copyright IBM Corporation 2013

14

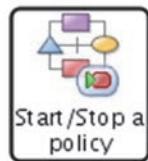
Situation activities

When you use activities, double-clicking palette objects makes them **sticky**, which means that they are available with every mouse click. This feature is useful when connecting multiple activities in a row.

Starting and stopping activities



- **Start/Stop a situation**
activates or deactivates a situation



- **Start/Stop a policy**
activates or deactivates another policy

© Copyright IBM Corporation 2013

15

Starting and stopping activities

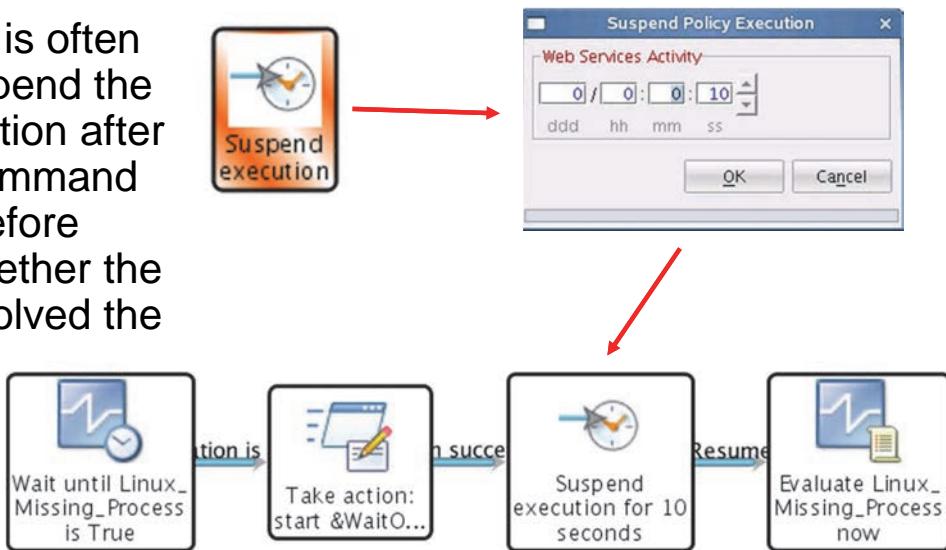
As part of a policy, you can start and stop other situations or policies.

Suspending a policy execution

- **Suspend execution** suspends a policy operation for a specified amount of time

- **Usage:**

This activity is often used to suspend the policy operation after a system command is issued, before verifying whether the command solved the problem



© Copyright IBM Corporation 2013

16

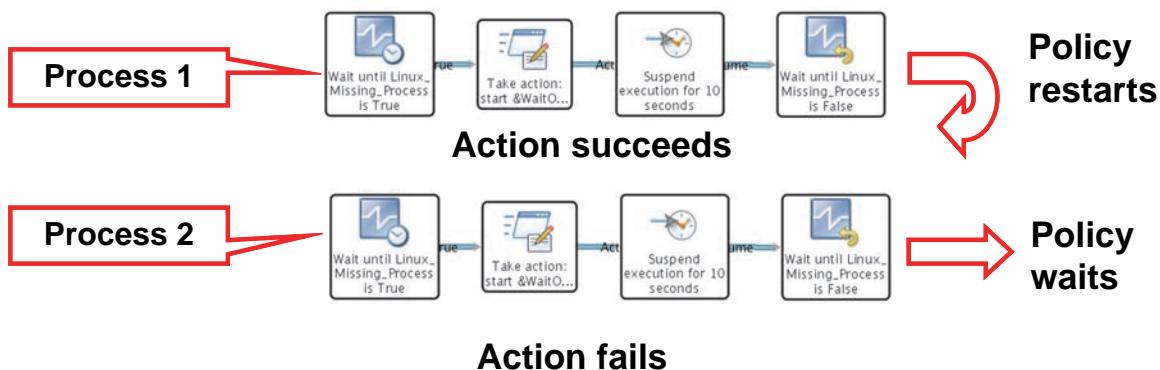
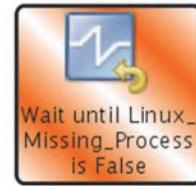
Suspending a policy execution

One way of suspending a policy is by including a **Suspend execution** activity in your workflow. Use that option every time you must wait for something to happen before the workflow continues, such as a system command being issued.

Waiting until a situation is false

- **Wait until a situation is False** suspends a policy operation until a specific situation returns to false
- Example:

When an atomized situation returns separate situation events for each item (in this case, each process that is not running), a different policy thread is created for each item



© Copyright IBM Corporation 2013

17

Waiting until a situation is false

The policy runs a different thread for each situation event item when both of these conditions occur:

- You select the **Wait until a situation is False** activity.
- You use an atomized situation as the policy trigger situation, meaning that you specify a display item.

Take action or write a message activity

- Actions in policies can send commands to the following locations:
 - The monitoring server
 - The agent that returned the situation data
 - Any arbitrary agent in the enterprise, even ones that are connected to another monitoring server
- They work similar to actions in situations



© Copyright IBM Corporation 2013

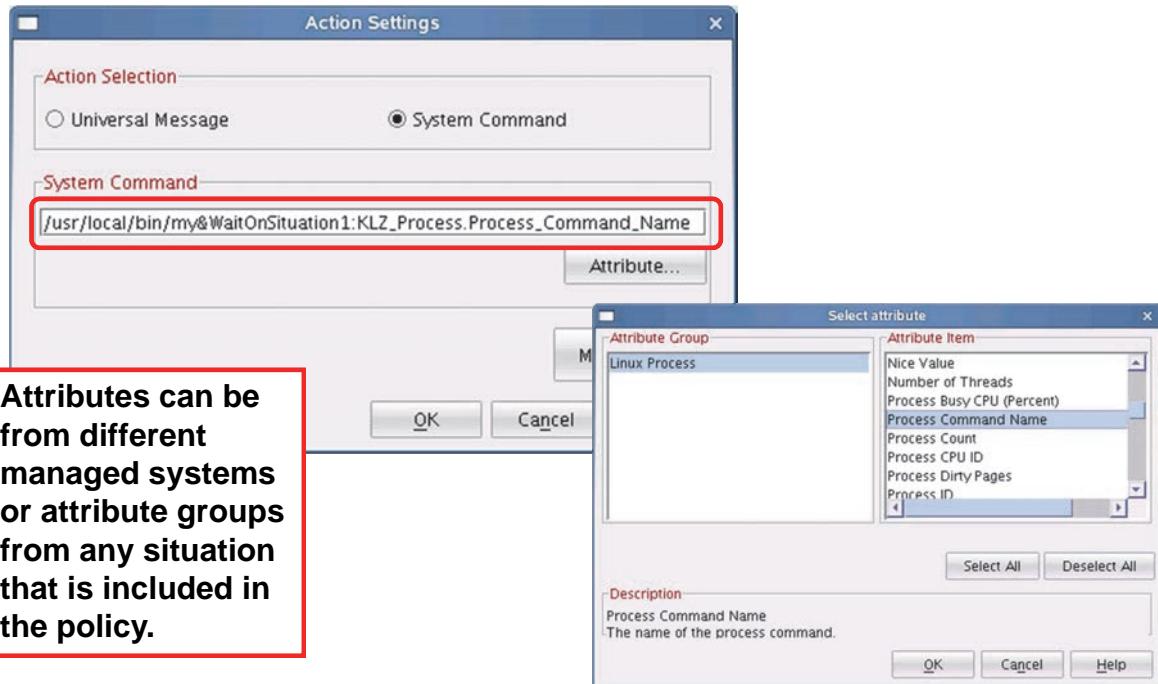
18

Take action or write a message activity

This activity works like the action you can specify in a situation. You can issue a system command or write a universal message. There are some differences, which are covered in the rest of this unit.

Take action activity: Attribute substitution

You can substitute attributes at execution time



© Copyright IBM Corporation 2013

19

Take action activity: Attribute substitution

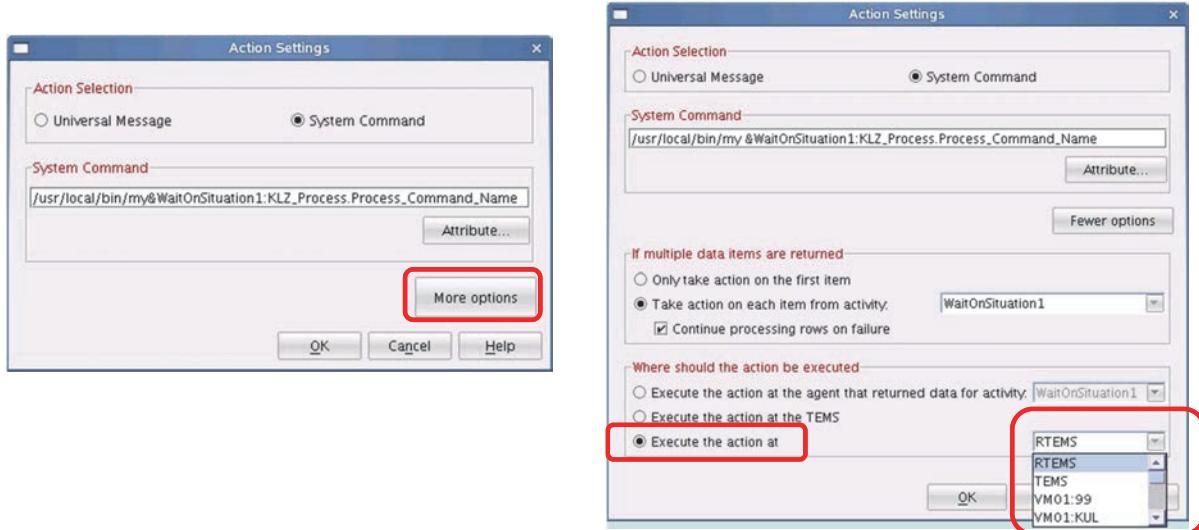
As with actions in situations, you can substitute attribute values in your policy system command. Those attribute values are replaced with actual values when the policy runs, and are based on the situation results within the policy.

You can substitute only values that were collected before the action taking place in the workflow. This restriction ensures that the attribute values are available when the command is issued.

If no situations are embedded, you cannot select any attributes.

Take action activity: More options

- Policies do not run on an interval basis
- You can issue actions at any system in the enterprise



© Copyright IBM Corporation 2013

20

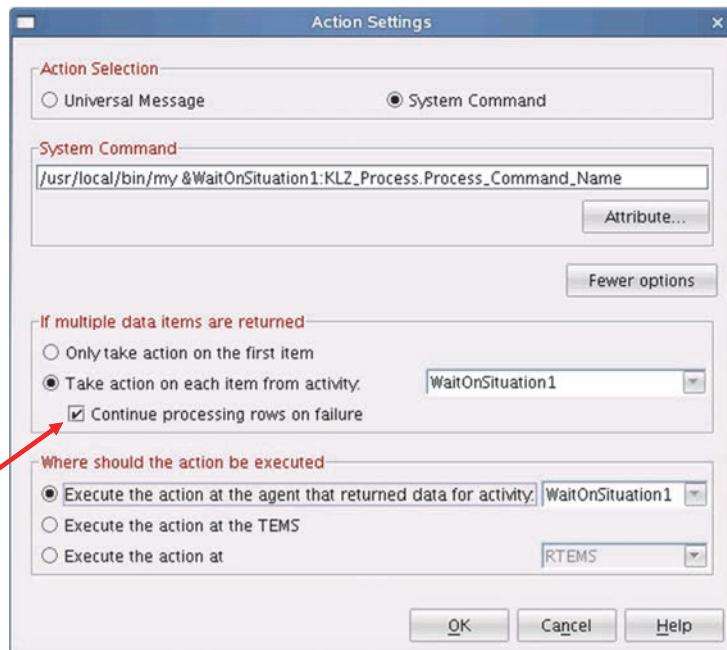
Take action activity: More options

The main differences between actions in situations and actions in policies are some of the optional settings:

- Policies do not run on an interval basis. Therefore, there is no selection to issue a command on a specific interval.
- Policy actions can run on any managed system in the enterprise.
- You can select **Continue processing rows on failure** to prevent policy suspension for other situation result rows that trigger the policy.

Take action activity: Continue on failure

- A situation event can have multiple result rows, for example, multiple processes that are missing on a specific system
- You can elect to issue a command for each row
- Select the option to continue processing rows on failure to issue commands for any result row, even if one of the commands fails



© Copyright IBM Corporation 2013

21

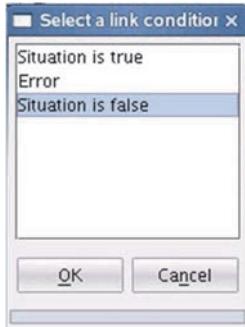
Take action activity: Continue on failure

During normal operation of an action activity, multiple situation event result rows are only issued if all commands return a success value of 0. Continuing to process rows on failure lets you process all rows, even if one or more commands fail.

You can react on failure or success with a different connector to the next activity in the policy. Different threads can be created for different situation event result rows.

Connector

- You use the connector tool  to define the policy logic
- Connecting two activities using the connector prompts you with link conditions, which define what rule is applied for starting the next activity



- Double-clicking the connector tool makes it sticky, and you can connect multiple activities without having to reselect the icon

© Copyright IBM Corporation 2013

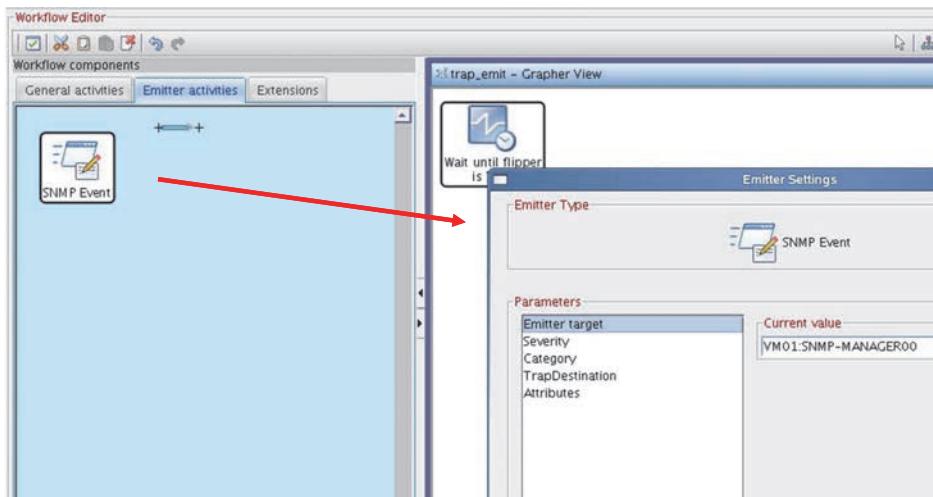
22

Connector

Which link conditions are available depends on the preceding activity from where the connector originates. Different activities have different link condition options.

Emitter activities

- When an emitter is installed, the equivalent activity is displayed in the policy editor under the **Emitter activities** tab
- The SNMP Event emitter activity is available after you install the IBM Tivoli Universal Agent
- There are other emitter activities that require separately purchased non-IBM Tivoli Monitoring products



© Copyright IBM Corporation 2013

23

Emitter activities

SNMP emitter targets can be any event management system that accepts SNMP traps. For the severity, select **cleared**, **ineterminate**, **warning**, **minor**, **critical**, or **major**. The many categories include **network topology events**, **status events**, **application alert events**, and so on.

Trap destinations are the hosts that are designated to receive SNMP traps. Your trap destination is any SNMP manager you define, and you can substitute attribute values when you issue the SNMP trap.

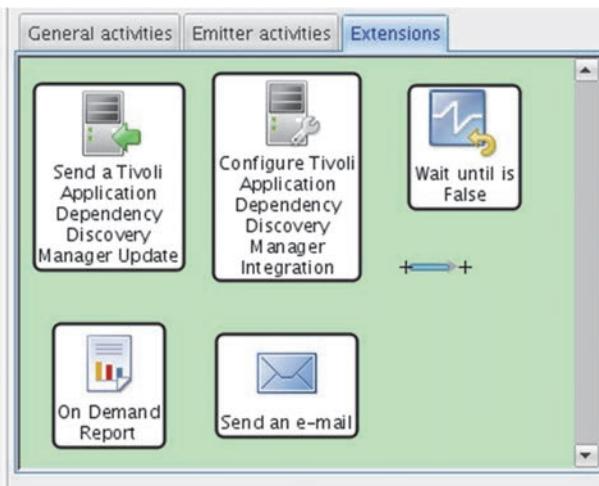
Several emitters are available:

- SNMP Event:** This activity is for the SNMP Gateway on Windows, available with the Universal Agent. Only this emitter comes with Tivoli Monitoring.
- NetView Event:** IBM Tivoli Alert Adapter for TME 10 Tivoli NetView.
- TEC Event:** IBM Tivoli Alert Manager for Tivoli Enterprise Console; IBM Tivoli Alert Adapter for Tivoli Enterprise Console.

Extensions

The **Extensions** tab shows:

- Send a Tivoli Application Dependency Discovery Manager Update activity
- Configure Tivoli Application Dependency Discovery Manager Integration activity
- Wait until is False activity
- On Demand Report activity
- Send an e-mail activity



© Copyright IBM Corporation 2013

24

Extensions

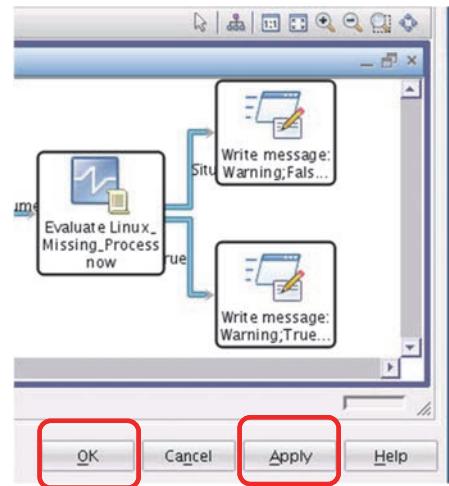
Tivoli Application Dependency Discovery Manager delivers automated discovery and configuration tracking capabilities to build application maps that provide real-time visibility into application complexity. With the first two extensions, you can write policies in IBM Tivoli Monitoring that integrate with Tivoli Application Dependency Discovery Manager.

On Demand Report sends an on-demand request to retrieve attribute data for a specified monitoring agent report. You can specify filters to be used in the request, limiting data specific to the situation that triggered the workflow.

Send an e-mail sends an email to a single recipient in response to a triggered situation event or activity in your workflow.

Saving changes

- After finalizing the policy, commit any details and workflow changes to the monitoring server
- The workflow is automatically validated, and you receive error messages for invalid entries
- Changes to multiple policies can be saved at the same time
- The Undo icon is disabled after you commit changes



© Copyright IBM Corporation 2013

25

Saving changes

Lesson 3. Workflow validation

Lesson 3: Workflow validation

- All changes to the workflow must be validated before you can save them
- The Validate button is enabled when you make changes to a workflow
- Workflow validation is performed at these times:
 - When you select the Validate button
 - When you click **Apply** or **OK**
- A notification opens
- Any error messages are displayed once



© Copyright IBM Corporation 2013

26

What this lesson is about

This lesson teaches the process to validate, activate, and troubleshoot a workflow.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Validate a workflow.
- Activate a workflow
- Start and stop a workflow
- Troubleshoot a workflow.

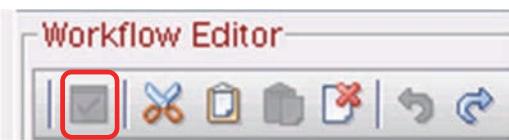
The **Validate** option verifies whether a policy is valid. You can validate manually by clicking **Validate**, or automatically by saving the policy. If the policy has logical errors, you cannot save the policy. You are prompted to fix the errors first.

Workflow validation: Success

Before validation



After successful validation



If validation is successful,
the icon becomes gray

© Copyright IBM Corporation 2013

27

Workflow validation: Success

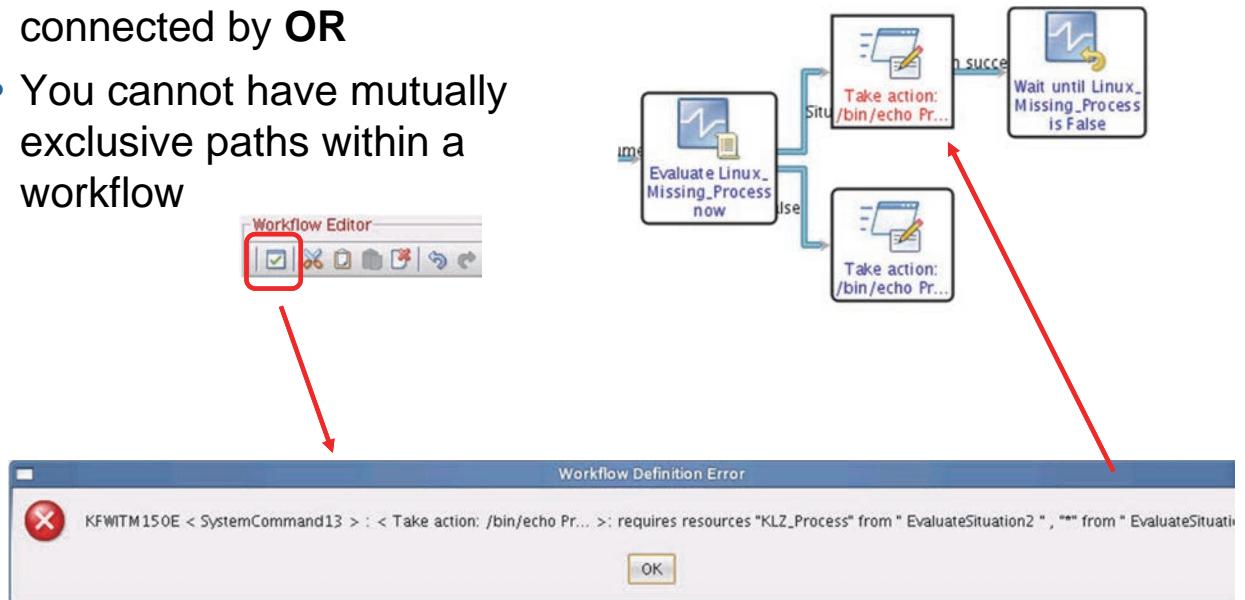
Workflow validation: Failure

- If the validation of a policy fails, the display indicates that failure in the following ways:
 - Activity borders changes from rounded edges to square edges
 - Activity text changes from black to red
- You also receive an error message because you did not follow certain rules, such as these examples:
 - The policy must not be empty
 - Correlated policies must start with a Wait until a situation is True activity
 - System commands and emitters can only use attributes from situation activities that preceded them with an appropriate connector condition
 - All AND connectors must be possible; there can be no mutually exclusive paths



Workflow validation: Policy rules

- Activities cannot be connected by **OR**
- You cannot have mutually exclusive paths within a workflow



© Copyright IBM Corporation 2013

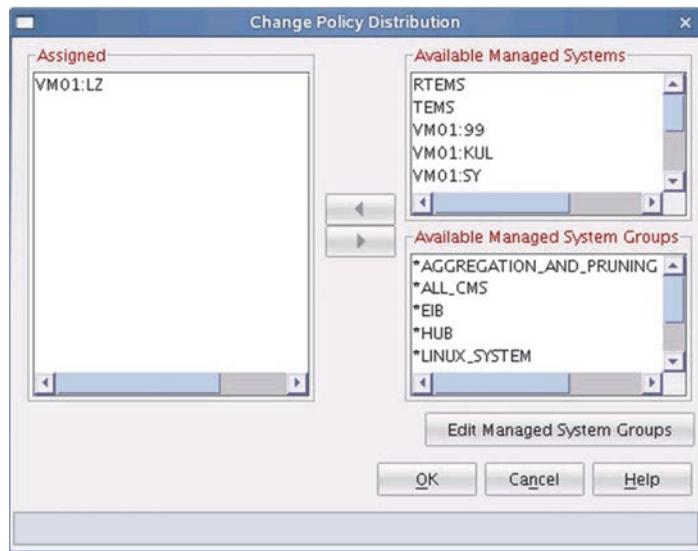
29

Workflow validation: Policy rules

An important rule when creating workflows is that all paths entering an activity must be true before the activity starts. Therefore, no OR conditions can be in a policy. You cannot start an activity if one or another entering path is true. For example, a **Take action** activity never runs if both entering paths can never be true. The exiting paths of the first **Take action** are mutually exclusive. When you validate the policy, this error is caught and you cannot save this policy.

Activating the policy

- Distribute the policy to a managed system or managed system group
- Distributing a policy does not affect the distribution of situations used by the policy



© Copyright IBM Corporation 2013

30

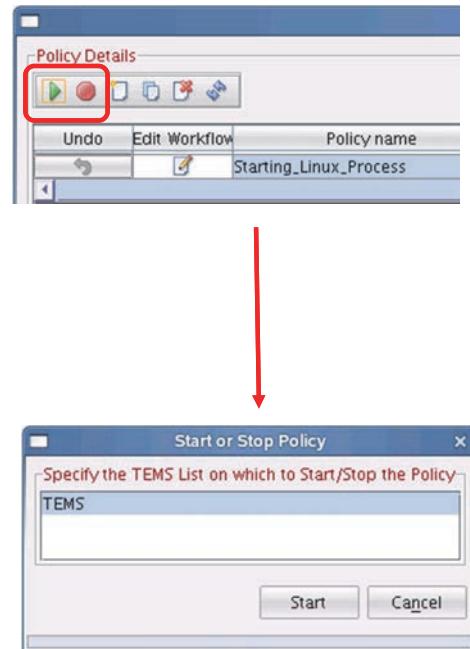
Activating the policy

When you distribute a policy to a managed system or managed system group, embedded situations are not distributed with the policy. They must be distributed to the target systems using another mechanism. For example, you can use the Situation Editor.

With the advent of situation groups, the logic to modify a distribution of a situation becomes more complex. You can distribute a situation to an agent through one or more managed system groups. You can also distribute the situation through one or more groups.

Policy functions: Start and stop

- Multiple policies can be started or stopped at the same time
- An automatic prompt reminds you to commit any unsaved changes of selected policies
- Policies can run on hub and remote monitoring servers
- A list of monitoring servers is displayed on which the policy can be started



© Copyright IBM Corporation 2013

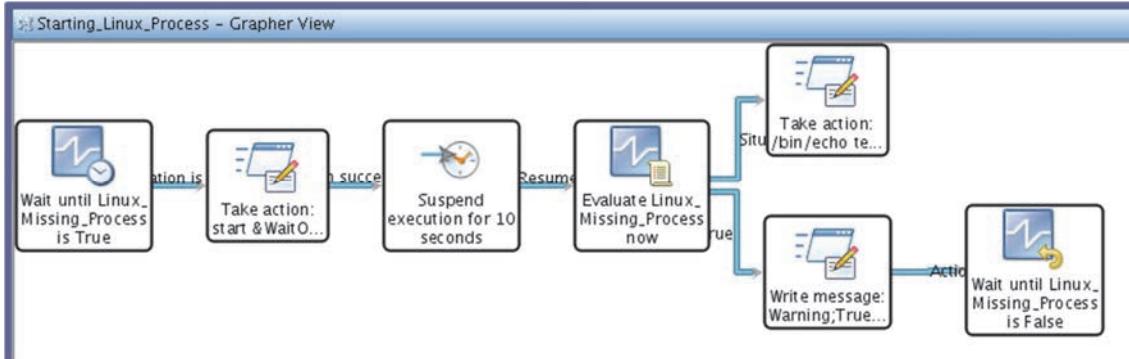
31

Policy functions: Start and stop

When starting a policy manually, you are prompted with a list of available monitoring servers where the policy can run. If more than one server is displayed, you can select multiple servers on which to start the policy.

Policy usage scenarios: Policy workflow

Undo	Edit Workflow	Policy name	Distributed
		Starting_Linux_Process	<input checked="" type="checkbox"/>



© Copyright IBM Corporation 2013

32

Policy usage scenarios: Policy workflow

This policy example is based on one or multiple processes not running. After that situation becomes true, the policy workflow starts.

1. Try to restart the missing process. To give the command enough time to finish before the next activity, a 10-second wait activity is included in the workflow.
2. Re-evaluate the trigger situation to see whether the problem persists.
3. Send a message, depending on the outcome of that activity. Different messages are sent for true and false.

Troubleshooting a policy workflow

You can troubleshoot the policy workflow using the **Universal Message Console**

The screenshot shows two windows. The top window is a 'Filters' dialog box with a table. Row 2 has a condition `MessageText == 'Starting_Linux_Process'` highlighted with a red box. A callout box to the right says: 'Specify a filter in the view properties to include messages from only one policy'. The bottom window is a 'Policy Log - VM01 - SYSADMIN' window showing a list of messages. One message in the log is also highlighted with a red box.

	Message Text	Timestamp	Category	Originnode
1				
2	MessageText == 'Starting_Linux_Process'			
3				
4				

	Message Text	Timestamp	Category	Originnode
1	PolicyStarting_Linux_Process started.	09/28/09 09:09:36	K048010	TEMS
2	PolicyStarting_Linux_Process, activity Embed Situation: Linux_Missing_Process: <*> has started.	09/28/09 09:09:36	K048073	TEMS
3	PolicyStarting_Linux_Process deactivated by external request.	09/28/09 09:09:39	K048077	TEMS
4	PolicyStarting_Linux_Process activated.	09/28/09 09:09:39	K048071	TEMS
5	PolicyStarting_Linux_Process started.	09/28/09 09:09:39	K048010	TEMS
6	PolicyStarting_Linux_Process, activity Embed Situation: Linux_Missing_Process: <*> has started.	09/28/09 09:09:39	K048073	TEMS
7	PolicyStarting_Linux_Process, activity Embed Situation: Linux_Missing_Process: <VM01:L2> has started.	09/28/09 09:15:17	K048073	TEMS
8	PolicyStarting_Linux_Process: <VM01:L2> ended.	09/28/09 09:15:17	K048009	TEMS

© Copyright IBM Corporation 2013

33

Troubleshooting a policy workflow

While a policy is running, messages are written to the universal message log. To see messages about only one particular policy or one type of activity, you can filter the Universal Message Console to display only those messages. This feature can be used as an effective method of troubleshooting policies.

Student exercise



© Copyright IBM Corporation 2013

34

Student exercises

Perform the exercises for this unit.

Review questions

1. What is the Workflow Editor pane called where the policy steps are arranged and connected?
2. Are there Undo and Redo change icons in the Workflow Editor?
3. How are attributes substituted into commands issued by policies?

Review answers

1. What is the Workflow editor pane called where the policy steps are arranged and connected?

The Grapher view

2. Are there Undo and Redo change icons in the Workflow editor?

Yes

3. How are attributes substituted into commands issued by policies?

*In the Action Settings window, click **Attribute**. In the Select Attribute window, select an attribute group and an attribute.*

Summary

Now that you have completed this unit, you can perform the following tasks:

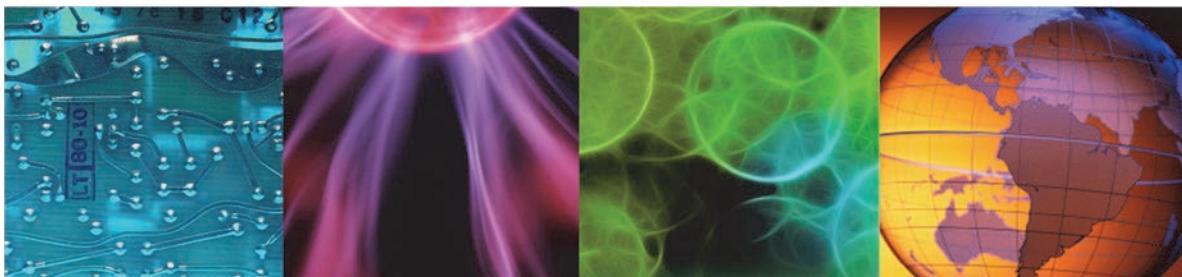
- Use the Workflow Editor
- Create and validate a policy
- Issue commands from a policy
- Activate a policy
- Automatically solve a problem and notify users of the result



6 Agentless monitoring



Agentless monitoring



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit covers how to use agentless monitoring in IBM Tivoli Monitoring.

How you check your progress

You can check your progress in the following ways:

- Review questions
 - Lab exercises
- 

Objectives

When you complete this unit, you can perform the following tasks:

- Explain the agentless monitoring features and function
- Plan and configure an agentless monitoring environment
- Troubleshoot an agentless monitoring problem

Lesson 1. Agentless monitoring overview

Lesson 1: Agentless monitoring overview

- Agentless monitors supervise the health of your enterprise from a small set of centralized servers
- They provide multiple solutions to provide flexibility for customer environments
 - Traditional deep agents for core business servers
- **IBM Tivoli Monitoring 6.1 and 6.2 provided**
 - Agent Builder and Universal Agent-Based Custom Solutions
 - Multiple mechanisms supported: SNMP, JMX, WMI, HTTP
 - Expand remote data providers in Agent Builder
 - IBM Tivoli Monitoring Solutions supporting both local and remote deployment
- **IBM Tivoli Monitoring 6.2.1 provided**
 - OS Agentless Monitoring for popular operating systems
 - AIX, HP, Linux, Windows, and Solaris (Solaris Operating Environment)

What this lesson is about

This lesson provides an overview of the agentless monitoring capabilities in IBM Tivoli Monitoring. Agentless monitoring has been part of Tivoli Monitoring since version 6.1. Updates have made this function easier to use.

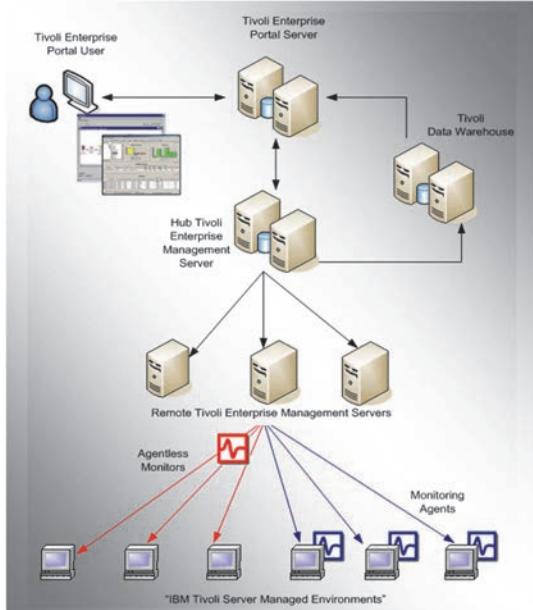
What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Explain how agentless monitoring differs from agent based monitoring
- Describe some of the metrics that can be collected with agentless monitoring.

Agent or agentless

- ☒ **Agent-based technology** runs directly on a managed server and collects data based on policy set locally or by the management server
- ☒ **Agentless technology** can run on a management server or a dedicated server and gets its data using a remote application programming interface (API)
 - *Agentless* does not mean that nothing is present or running
Some basic operating system function or base application function is running to provide the information as requested over the network
 - Resources are still being used and services must be running on the server



© Copyright IBM Corporation 2013

4

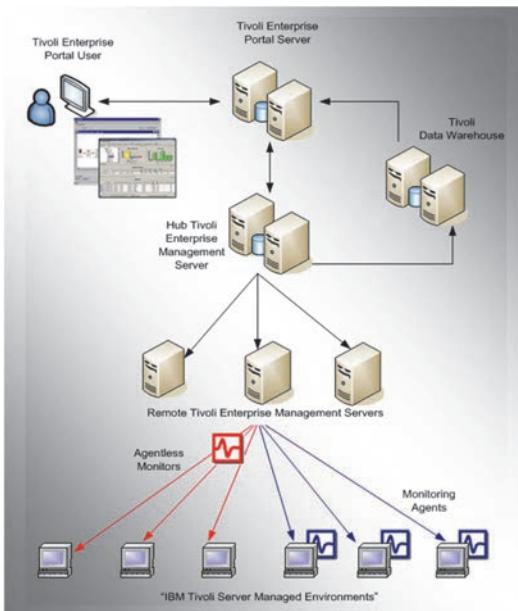
Agent or agentless

Agentless technologies that Tivoli Monitoring supports include these examples:

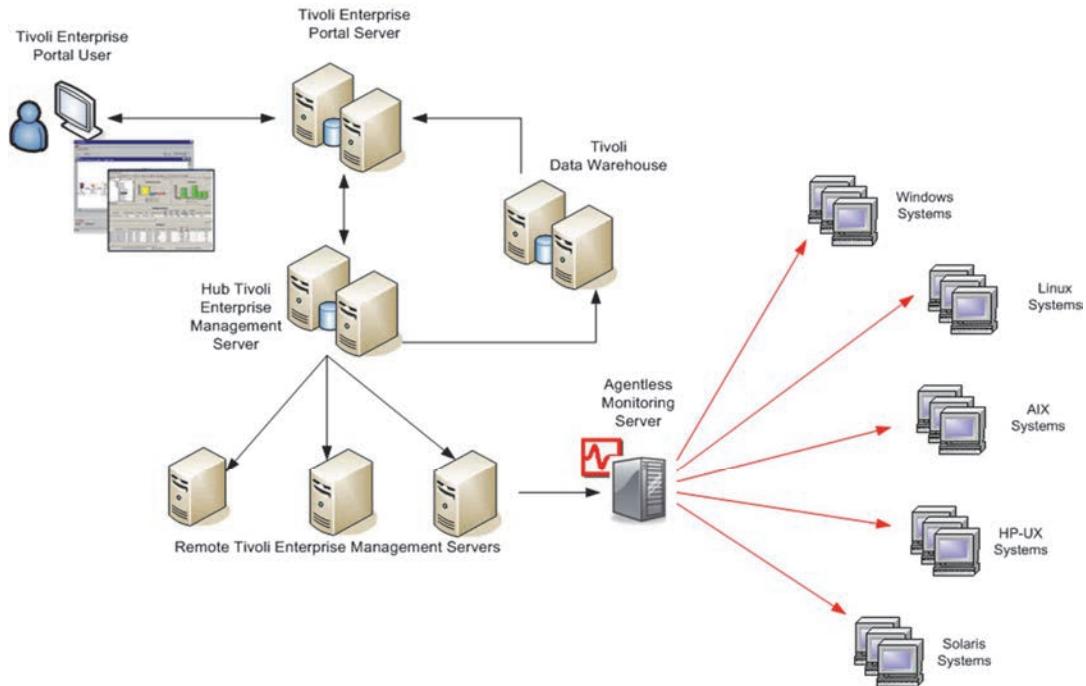
- Simple Network Management Protocol (SNMP)
- Common Information Model (CIM)
- Windows Management Infrastructure (WMI)

Understanding agent and agentless technology

- Tivoli has both agent and agentless technology
- Agent technology
 - Many of the IBM Tivoli Monitoring for technologies
 - Database
 - Operating systems
- Agentless technology
 - Custom Agent from the Agent Builder
 - IBM Tivoli Monitoring for Virtual Servers
 - IBM Tivoli Monitoring for Applications
 - Operating systems



Agentless monitoring architecture



© Copyright IBM Corporation 2013

6

Agentless monitoring architecture

This slide illustrates having a dedicated server for the Agentless Monitoring Server, which spreads the load away from the monitoring server. The result is greater throughput and improved performance.

Comparing agent and agentless technologies

	Agent 	Agentless 
Service provider No ability to put agents in a customer's environment	Unsuitable	<input checked="" type="checkbox"/> Suitable
Speed of implementation	Low	<input checked="" type="checkbox"/> High
Agent maintenance Deployment and update distributions of agent	High	<input checked="" type="checkbox"/> Low
Impact to testing Locked-down server environments	High	<input checked="" type="checkbox"/> Low
IT resource impacts Server, network	<input checked="" type="checkbox"/> Primarily server Agent message suppression	Primarily network Remote polling on server
Resilience and availability Unique inherent single points of failure	<input checked="" type="checkbox"/> Higher, store and forward	Lower. Susceptible to network problems
Command and control capabilities Take actions easily	<input checked="" type="checkbox"/> High	Low
Granularity and coverage of monitoring metrics	<input checked="" type="checkbox"/> Greater access	Dependent on standards
Data availability Real-time responsiveness	<input checked="" type="checkbox"/> High	Polling lag, network delay
Security	<input checked="" type="checkbox"/> Highly secure communication	Standards dependent

Comparing agent and agentless technologies

This table shows you the typical impacts of agent and agentless technology on your resources. There are some exceptions. For instance, you might have an agentless target machine that is being monitored using WMI. In this case, the performance of the target machine might be affected if too many resources are polled too often.

Agentless monitoring for operating systems

- Lightweight OS monitoring targeting key metrics and basic situations to satisfy simple monitoring needs
- Uses standard protocols to gather metrics
- Uses the Agent Builder runtime for data collection and remote node capabilities

KR2	Agentless Monitoring for Windows Operating Systems
KR3	Agentless Monitoring for AIX Operating Systems
KR4	Agentless Monitoring for Linux Operating Systems
KR5	Agentless Monitoring for HP-UX Operating Systems
KR6	Agentless Monitoring for Solaris Operating Systems

© Copyright IBM Corporation 2013

8

Agentless monitoring for operating systems

Agentless Monitoring Servers exist for Windows, AIX, Linux, HP-UX, and Solaris systems.



Note: Some agents cannot run on certain platforms. For instance, you cannot install and run the Agentless Monitoring for Windows Operating Systems and configure it to use WMI on a non-Windows platform.

Windows OS Monitoring: Windows APIs, or SNMP v1-based, v2c-based, v3-based monitoring

Category	Attributes
Memory	Memory Type (Virtual or Physical), Used Memory, Available Memory, Description, Total Memory, % Used Memory, % Available Memory, Memory Allocation Errors
Disk and File System Capacity	Disk Type, Total Disk Space, Available Disk Space, Description, % Used Disk Space, % Available Disk Space, Disk Allocation Errors
Processor	CPU Number, % CPU Utilization
Network	Interface Type, Physical Address, MTU, Speed (bps) Status, Bytes In, Discarded Inbound Packets, Inbound Packet Errors, Inbound Protocol Errors, Bytes Out, Outbound Discarded Packets, Outbound Packet Errors
Process and Service Availability	Process Name, Process ID, Process Path, Process Parameters, Process Type, Status, Cumulative CPU Time(s), Memory Utilization (KB)
System	System Name, System Uptime, System Date, Current Time, Current User Logins, Total Running Processes, Maximum Allowed Processes, % Allowed Processes In Use, Description, System Contact, System Location
User Account Information (Windows APIs only)	Name, Description, Account Type, Is Disabled, Domain, Full Name, Is Local Account, Lockout, Is Password Changeable, Does Password Expire, Is Password Required, Status
Terminal Services (Windows APIs only)	Active Sessions, Inactive Sessions, Total Sessions, Active Sessions Percentage of, Inactive Sessions Percentage of Session Name, Percentage of Privileged Time, Percentage of Processor Time, Percentage of User Time, Private Size, Virtual Size, Working Set
Windows Event Log (Windows APIs only)	Security, Application, System logs (informational, warning, critical)

AIX OS Monitoring: SNMP v1-based, v2c-based, v3-based monitoring

Category	Attributes
Memory	Memory Type (Virtual, Physical, Swap), Used Memory, Available Memory, Description, Total Memory, % Used Memory, % Available Memory, Memory Allocation Errors
Disk and File System Capacity	Disk Type, Total Disk Space, Available Disk Space, Description, % Used Disk Space, % Available Disk Space, Disk Allocation Errors, Mount Point
Processor	CPU Number, % CPU Utilization
Network	Interface Type, Physical Address, MTU, Speed (bps) Status, Bytes In, Discarded Inbound Packets, Inbound Packet Errors, Inbound Protocol Errors, Bytes Out, Outbound Discarded Packets, Outbound Packet Errors
Process and Service Availability	Process Name, Process ID, Process Path, Process Parameters, Process Type, Status, Cumulative CPU Time(s), Memory Utilization (KB)
System	System Name, System Uptime, System Date, Current Time, Current User Logins, Total Running Processes, Maximum Allowed Processes, % Allowed Processes In Use, Description, System Contact, System Location, System Load (1 min, 5 min, 15 min)
Volume Group, Logical Volume, Physical Volume, Page System	Disk Type, Total Space, Available Space, % Used Space, % Available Space, Mount Point
User Account Volume	Name, Description, Account Type, Is Disabled, Domain, Full Name, Is Local Account, Lockout, Is Password Changeable, Does Password Expire, Is Password Required, Status

HP-UX OS Monitoring: SNMP v1, v2c, v3 based monitoring

Category	Attributes
Memory	Memory Type (Virtual, Physical, Swap), Used Memory, Available Memory, Description, Total Memory, % Used Memory, % Available Memory, Memory Allocation Errors
Disk and File System Capacity	Disk Type, Total Disk Space, Available Disk Space, Description, % Used Disk Space, % Available Disk Space, Disk Allocation Errors, Mount Point
Processor	CPU Number, % CPU Utilization
Network	Interface Type, Physical Address, MTU, Speed (bps) Status, Bytes In, Discarded Inbound Packets, Inbound Packet Errors, Inbound Protocol Errors, Bytes Out, Outbound Discarded Packets, Outbound Packet Errors
Process and Service Availability	Process Name, Process ID, Process Path, Process Parameters, Process Type, Status, Cumulative CPU Time(s), Memory Utilization (KB)
System	System Name, System Uptime, System Date, Current Time, Current User Logins, Total Running Processes, Maximum Allowed Processes, % Allowed Processes In Use, Description, System Contact, System Location, System Load (1 min, 5 min, 15 min)

Linux OS Monitoring: SNMP v1-based, v2c-based, v3-based monitoring

Category	Attributes
Memory	Memory Type (Virtual, Physical, Swap), Used Memory, Available Memory, Description, Total Memory, % Used Memory, % Available Memory, Memory Allocation Errors
Disk and File System Capacity	Disk Type, Total Disk Space, Available Disk Space, Description, % Used Disk Space, % Available Disk Space, Disk Allocation Errors, Mount Point
Processor	CPU Number, % CPU Utilization
Network	Interface Type, Physical Address, MTU, Speed (bps) Status, Bytes In, Discarded Inbound Packets, Inbound Packet Errors, Inbound Protocol Errors, Bytes Out, Outbound Discarded Packets, Outbound Packet Errors
Process and Service Availability	Process Name, Process ID, Process Path, Process Parameters, Process Type, Status, Cumulative CPU Time(s), Memory Utilization (KB)
System	System Name, System Uptime, System Date, Current Time, Current User Logins, Total Running Processes, Maximum Allowed Processes, % Allowed Processes In Use, Description, System Contact, System Location, System Load (1 min, 5 min, 15 min)

Solaris OS Monitoring: CIM-XML or SNMP v1-based, v2c-based, v3-based monitoring

Category	Attributes
Memory	Memory Type (Virtual, Physical, Swap), Used Memory, Available Memory, Description, Total Memory, % Used Memory, % Available Memory, Memory Allocation Errors
Disk and File System Capacity	Disk Type, Total Disk Space, Available Disk Space, Description, % Used Disk Space, % Available Disk Space, Disk Allocation Errors, Mount Point
Processor	CPU Number, % CPU Utilization
Network	Interface Type, Physical Address, MTU, Speed (bps) Status, Bytes In, Discarded Inbound Packets, Inbound Packet Errors, Inbound Protocol Errors, Bytes Out, Outbound Discarded Packets, Outbound Packet Errors
Process and Service Availability	Process Name, Process ID, Process Path, Process Parameters, Process Type, Status, Cumulative CPU Time (s), Memory Utilization (KB)
System	System Name, System Uptime, System Date, Current Time, Current User Logins, Total Running Processes, Maximum Allowed Processes, % Allowed Processes In Use, Description, System Contact, System Location, System Load (1 min, 5 min, 15 min)

Agentless monitoring scalability

- Agentless monitors are multiple-instance agents
- Support for up to 10 active instances on a single system, for example:
 - 2 AIX, 2 HP-UX, 2 Linux, 2 Solaris, 2 Windows
 - 4 Windows, 3 AIX, 3 Linux
 - 5 Windows, 5 Linux
 - 10 HP-UX
- Each instance supports communication with 100 remote nodes
 - 10 instances x 100 remote nodes = 1000 monitored systems
- Agent Builder runtime is multithreaded for parallel data collections, with a thread pool shared across all attribute groups in all remote nodes
- When starting the agents, give them a few minutes to reach their steady state
 - Initial collections of data can be slow to process because of the initial data volume load

© Copyright IBM Corporation 2013

9

Agentless monitoring scalability

Performance tuning environment variables

Variable name	Default	Description
CDP_DP_CACHE_TTL	60	Time in seconds before a query will trigger a new data collection.
CDP_DP_THREAD_POOL_SIZE	60	The number of threads created to perform background data collections. The Thread Pool is shared among all attribute groups in all remote nodes in an agent.
CDP_DP_REFRESH_INTERVAL	60	The interval in seconds at which each attribute group cache is updated in the background.
CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT	2	The number of seconds to wait for a data collection to happen before timing out and returning cached data.
CDP_SNMP_RESPONSE_TIMEOUT	2	The number of seconds to wait for each request to time out. Each row in an attribute group is a separate request
CDP_SNMP_MAX_RETRIES	2	The number of times to retry sending the SNMP request after a response timeout
CDP_NT_EVENT_LOG_GET_ALL_ENTRIES_FIRST_TIME	NO	Configures whether the Windows Event Log data provider should report old log entries on startup, or only new ones.
CDP_NT_EVENT_LOG_CACHE_TIMEOUT	3600	Cache lifetime in seconds of an event from the Windows Event Log.
CDP_PURE_EVENT_CACHE_SIZE	100	Number of pure events held in cache at any one time. When a query is made, reports all events in the cache at that time. When cache is full, oldest events are removed to make room for new ones.

© Copyright IBM Corporation 2013

10

Performance tuning environment variables

These environment variables are in the environment variable file for the instance of the agentless monitor you are running. For the Agentless Monitoring for Linux Operating Systems, that file is **r4.ini**. On a Linux operating system, it is in the **/opt/IBM/ITM/config** directory.

If you are running multiple instances of the agent, the name of the instance is reflected in the file name of the environment variable file (**r4_instanceName.ini**).

Agentless management

- Each remote monitor has self-monitoring attribute tables that you can use to monitor the collection process
- Performance Object Status attributes:
 - Last collection errors encountered
 - Last collection start and finish times
 - Last or average collection duration
 - Refresh interval
 - Number of collections
 - Cache hit, miss, hit percent
 - Intervals skipped (most useful)
- Thread Pool attributes
- You can create situations for these attribute groups to notify of collection failures
- Deploy an operating system agent to the Agentless Monitoring Server to watch processor utilization, memory, and network utilization of the monitors

Data Collection Status				
Query Name	Error Code	Last Collection Start	Last Collection Finished	
Network	NO ERROR	07/30/08 13:35:26	07/30/08 13:35:26	
Processes	NO ERROR	07/30/08 13:35:26	07/30/08 13:35:26	
Processor	NO ERROR	07/30/08 13:35:26	07/30/08 13:35:26	
System	NO ERROR	07/30/08 13:35:26	07/30/08 13:35:26	
hrStorageTable	NO ERROR	07/30/08 13:35:26	07/30/08 13:35:26	
Disk	NO ERROR	07/30/08 13:35:23	07/30/08 13:35:23	
Memory	NO ERROR	07/30/08 13:35:27	07/30/08 13:35:27	

© Copyright IBM Corporation 2013

11

Agentless management

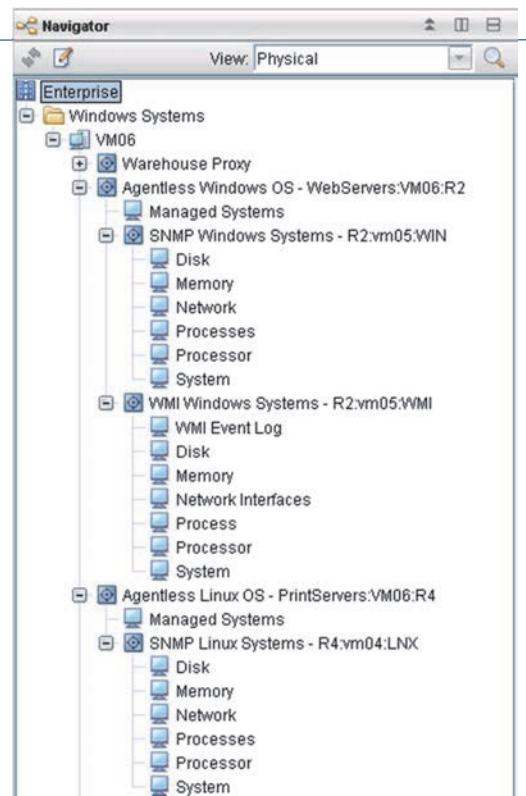
Agentless OS monitoring metric overview

- Key operating system metrics returned:
 - Logical and Physical Disk Utilization
 - Network Utilization
 - Virtual and Physical Memory
 - System Level Information
 - Aggregate Processor Utilization
 - Process Availability
- Default situations for:
 - Disk Utilization
 - Memory Utilization
 - Processor Utilization
 - Network Utilization

By default, situations are started to monitor the agentless monitoring targets.

Remote node capabilities

- One agent can represent more than one monitored entity
 - Multiple remote systems in one agent
 - Each remote node has a unique managed system name so that it can be in different managed system groups for situations
- One agent can represent different types of entities
 - Windows and Solaris agents can monitor different sets of data on different systems
- Multiple instances of an agent can reside on the same agent server



© Copyright IBM Corporation 2013

13

Remote node capabilities

Agentless monitoring data collection and platforms

- Agentless monitors can run from most Tivoli Monitoring supported platforms:
 - Windows (x86 and x64, not IA64)
 - x/p/z Linux
 - Solaris
 - AIX
 - HP-UX
- Agentless monitors can remotely monitor older versions of listed operating systems and other Linux distributions, depending on capabilities
- If you want to use the Windows API data collectors, the agentless monitor must run on a Windows platform

Customers can configure different data providers for the agentless monitors:

Agentless Monitoring for AIX OS

SNMP v1, v2c, v3

Agentless Monitoring for HP-UX OS

SNMP v1, v2c, v3

Agentless Monitoring for Linux OS

SNMP v1, v2c, v3

Agentless Monitoring for Solaris OS

CIM-XML

SNMP v1, v2c, v3

Agentless Monitoring for Windows OS

- Windows APIs
 - Windows Management Instrumentation (WMI)
 - Performance Monitor (Perfmon)
 - Event Log

- SNMP v1, v2c, v3

Lesson 2. Agentless monitoring planning and configuration

Lesson 2: Agentless monitoring planning and configuration

- You can configure all of the agentless OS monitors either from the Managed Tivoli Enterprise Monitoring Services panel or from the command-line interface
- All agentless OS monitors can use the SNMP protocol to obtain data
More protocols are available on Solaris (CIM) and Windows (Windows API)
- More configuration might be required on the remote system, depending on the remote operating system to be monitored
- Impact of configuration decisions
Capacity, speeds, response time, hardware requirements

© Copyright IBM Corporation 2013

15

What this lesson is about

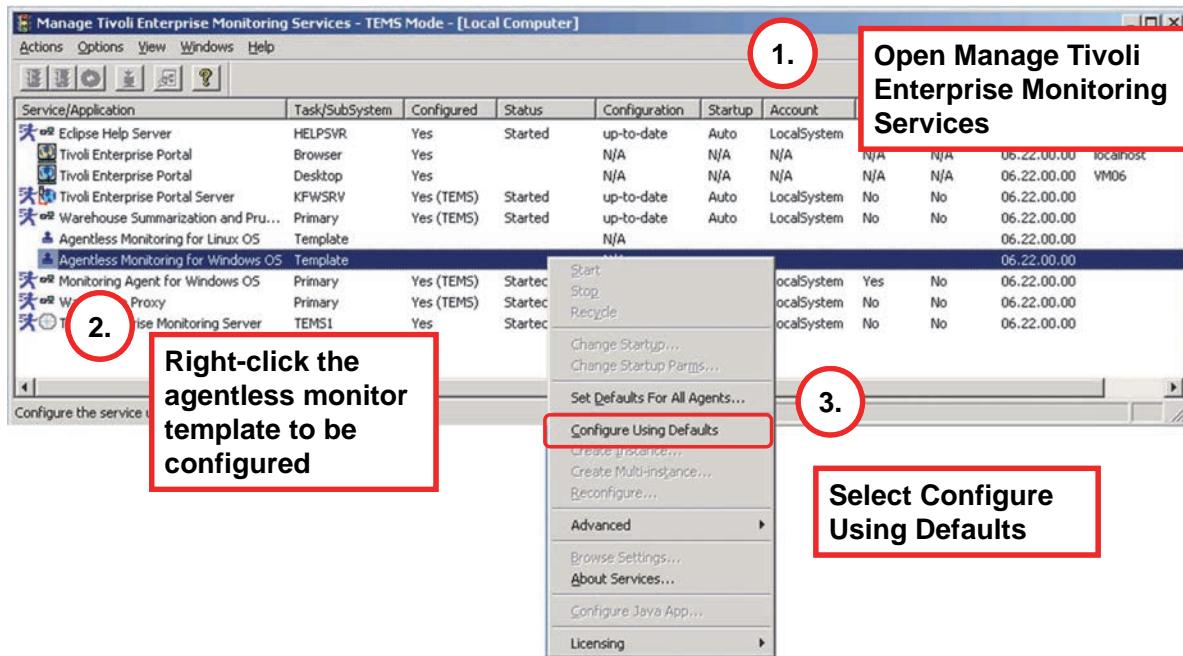
This lesson covers the planning and configuration of an agentless monitoring solution in IBM Tivoli Monitoring. Common Information Model is supported only when a Solaris target is specified.

What you should be able to do

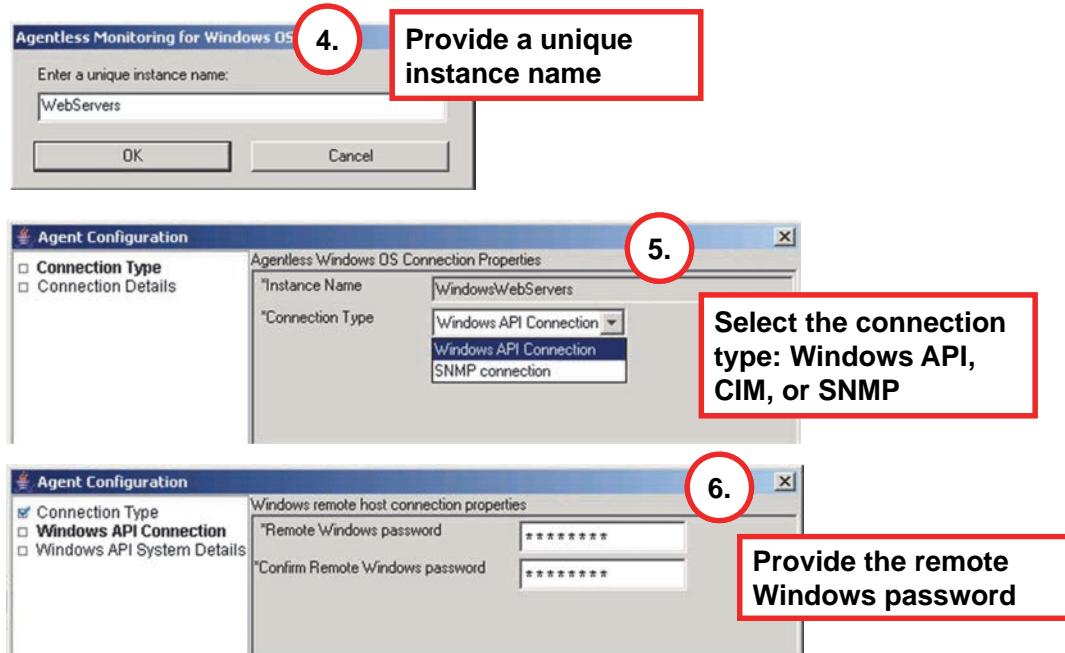
After completing this lesson, you should be able to perform the following tasks:

- Explain the steps that are involved in planning agentless monitoring solution.
- Configure an agentless monitoring solution on Windows, Solaris, Linux, and UNIX, and confirm it is functional.

Windows configuration

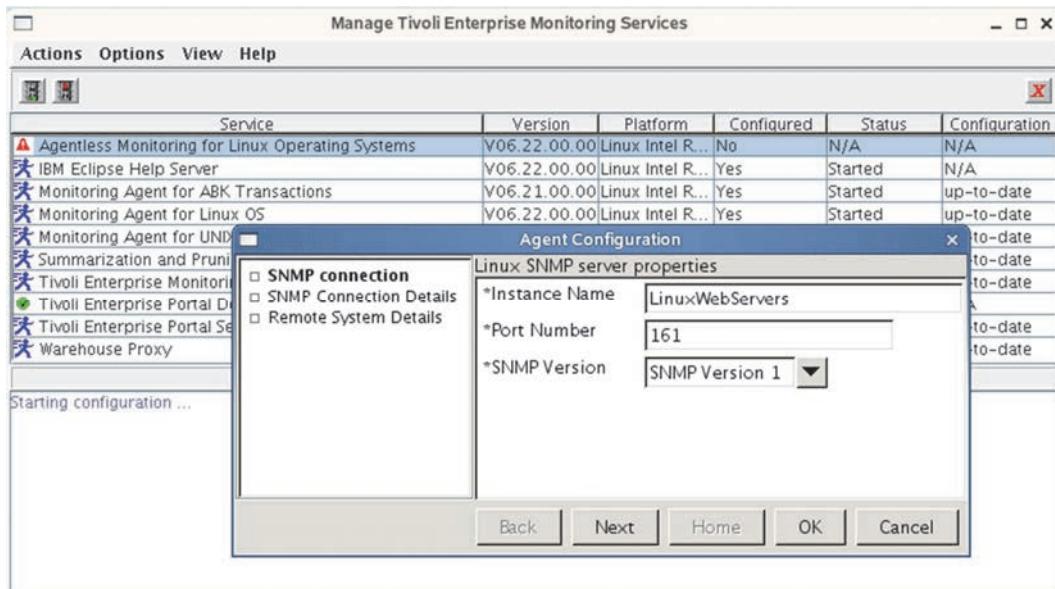


Windows configuration (continued)



SNMP configuration

If you use SNMP, select the port number and SNMP version to use



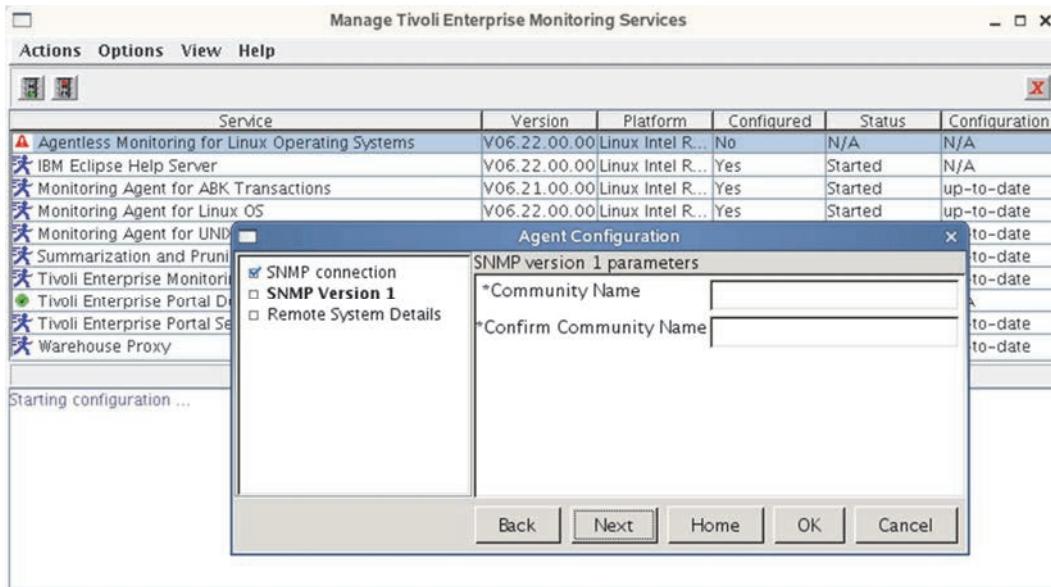
© Copyright IBM Corporation 2013

18

SNMP configuration

SNMP community name configuration

If you use SNMP v1, select the community name to use with the SNMP protocol on the remote system



© Copyright IBM Corporation 2013

19

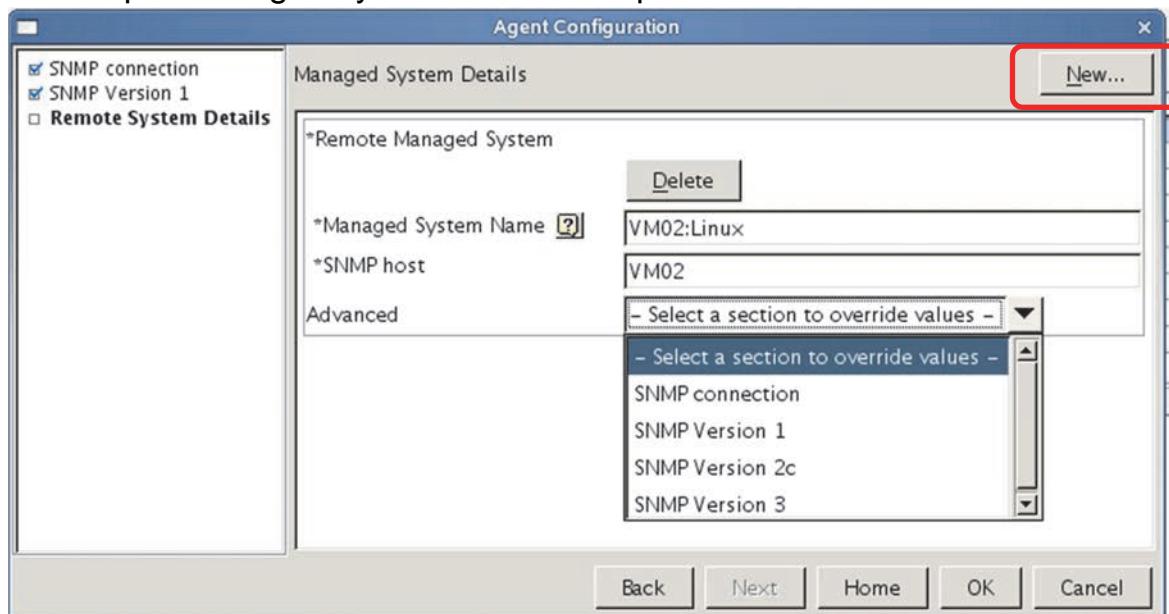
SNMP community name configuration

SNMP community names are case-sensitive. If the community name does not match exactly, then you cannot see any data from the SNMP daemon returned.

SNMP managed system name configuration

If you use SNMP, click **New** to add the system to be monitored

A unique managed system name is required



© Copyright IBM Corporation 2013

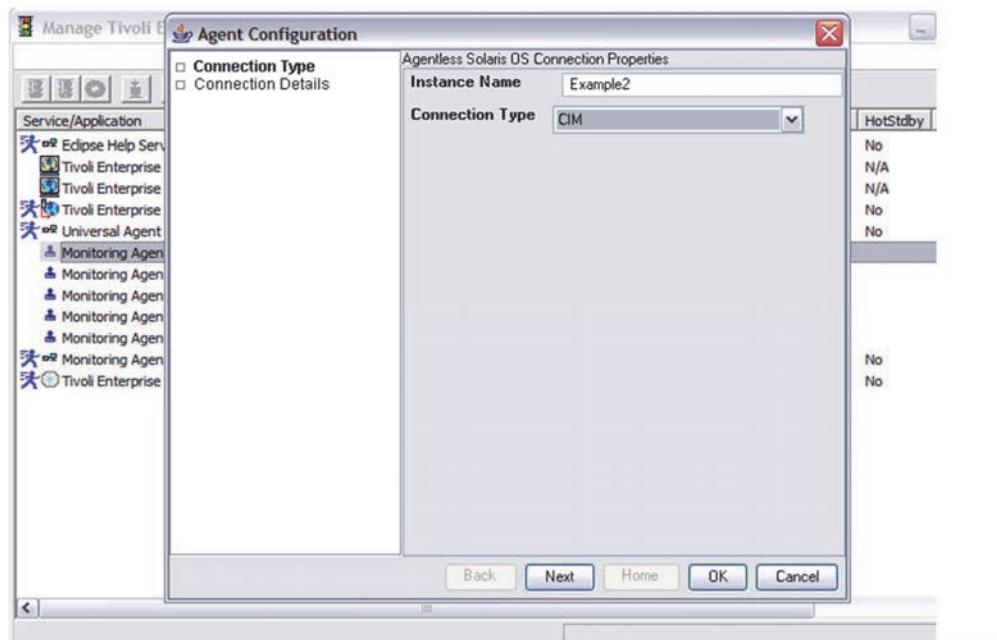
20

SNMP managed system name configuration

Managed system names must be unique within the associated framework.

CIM configuration (1 of 3)

If you use CIM, select the connection type to use



© Copyright IBM Corporation 2013

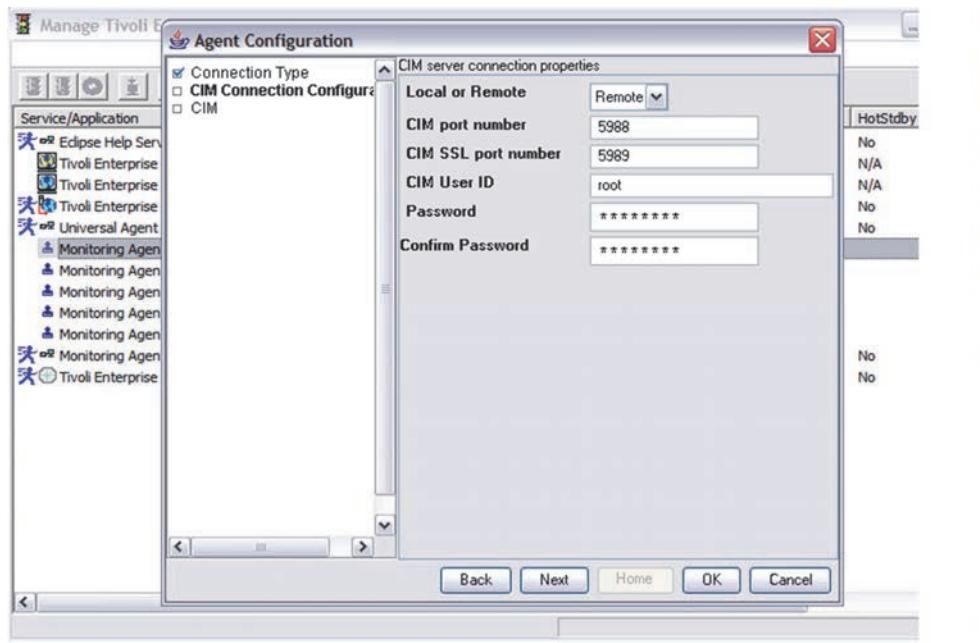
21

CIM configuration (1 of 3)

CIM is supported only for Solaris targets.

CIM configuration (2 of 3)

If you use CIM, select the CIM port number, CIM SSL port number, CIM user ID, and password

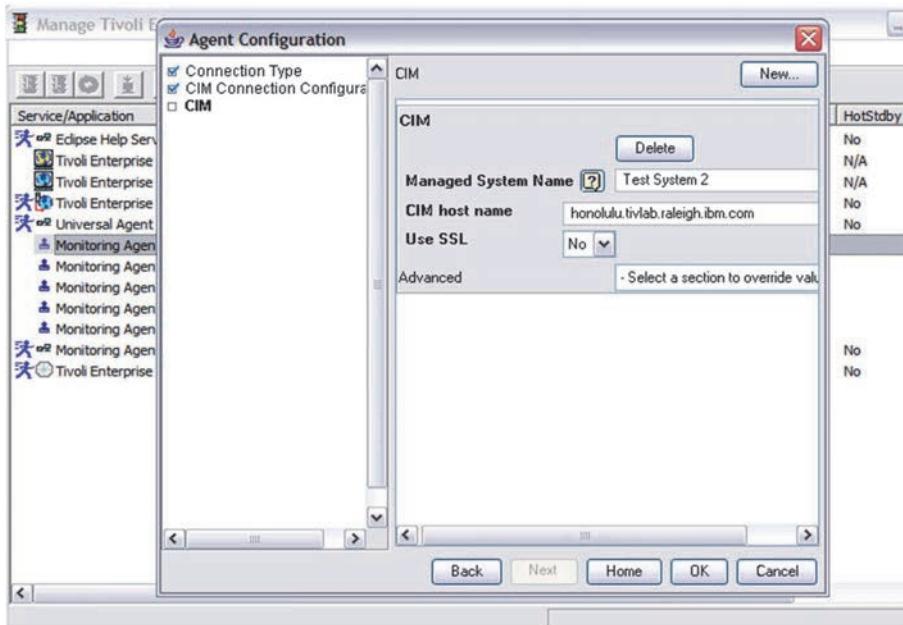


© Copyright IBM Corporation 2013

22

CIM configuration (3 of 3)

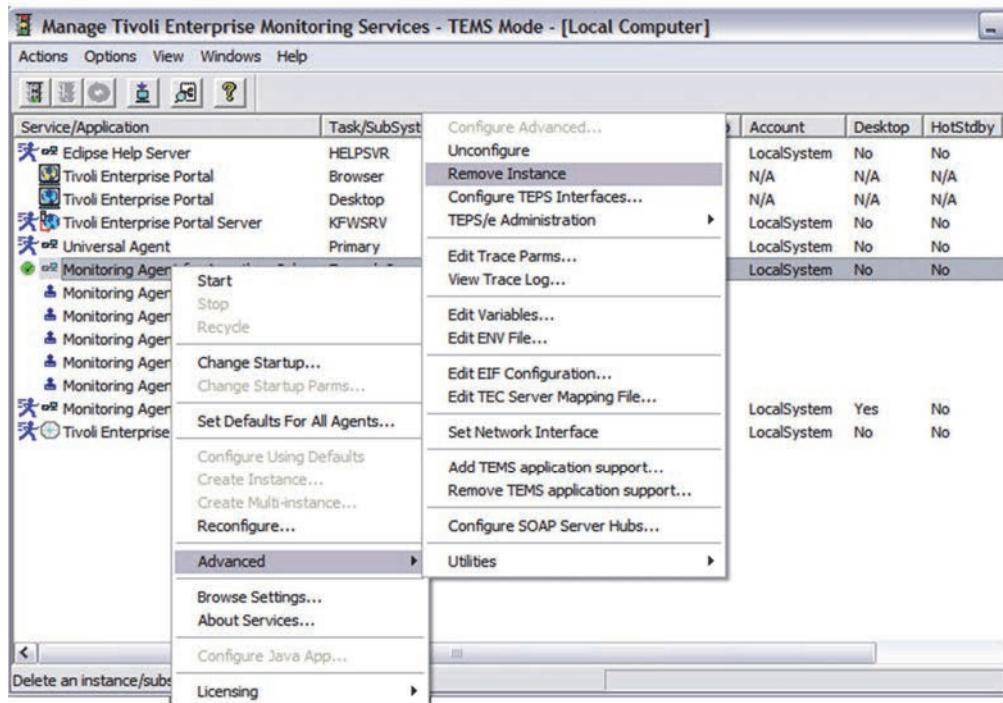
If you use CIM, click **New** to add the system to be monitored
A unique managed system name is required



© Copyright IBM Corporation 2013

23

Removing the instance on Windows



© Copyright IBM Corporation 2013

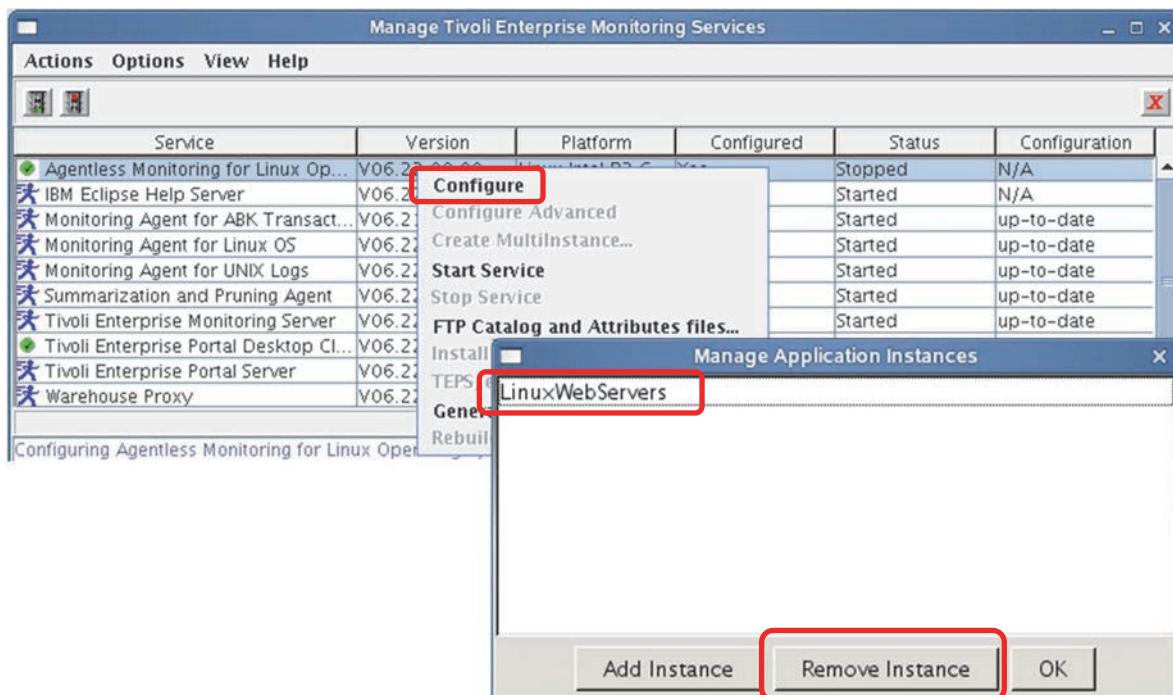
24

Removing the instance on Windows

If any abnormalities occur, follow these steps to remove the configured instance:

1. Right-click the instance in the Manage Tivoli Enterprise Monitoring Services application.
2. Select **Advanced**.
3. Select **Remove Instance**.

Removing the instance on Linux and UNIX



© Copyright IBM Corporation 2013

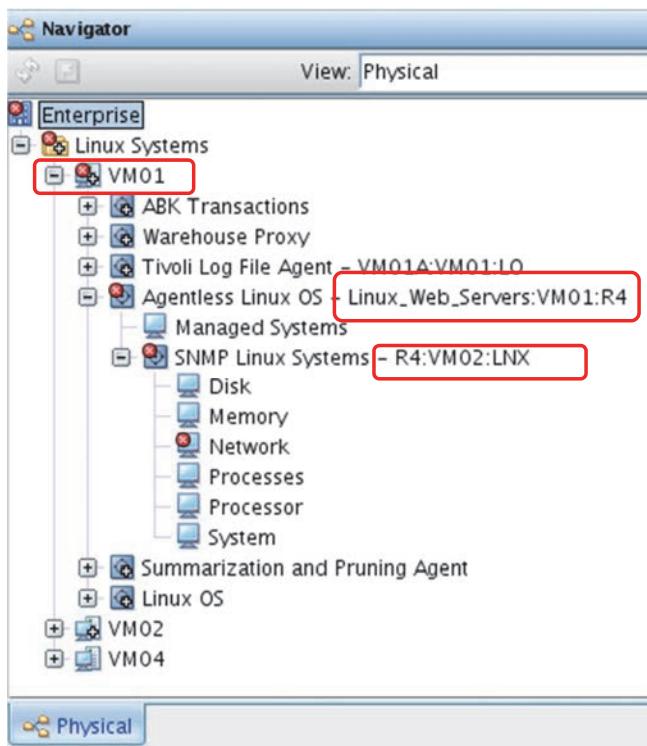
25

Removing the instance on Linux and UNIX

If any abnormalities occur, follow these steps to remove the configured instance:

1. Right-click the **Agentless Monitor** template in the Manage Tivoli Enterprise Monitoring Services application.
2. Select **Configure**.
3. Select the instance you want to remove.
4. Click **Remove Instance**.

Configuration verification



Observe the hierarchy of the agentless system that was the destination

The workspaces contain data about the destination system

© Copyright IBM Corporation 2013

26

Configuration verification

To verify a successful configuration, open the Manage Tivoli Enterprise Monitoring application and start the instance. Look in the portal client for the new workspaces.

Lesson 3. Troubleshooting agentless monitoring

Lesson 3: Troubleshooting agentless monitoring

- Typical target issues:
 - Connectivity
 - SNMP or CIM daemon not running
 - Incorrect version of SNMPD or CIM
 - SNMPD not configured correctly
- Typical Tivoli Monitoring environment issues:
 - Incorrect workspaces
 - Situations not firing

What this lesson is about

This lesson covers the basic troubleshooting of an agentless monitoring solution in IBM Tivoli Monitoring.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Explain the typical issues that are encountered when deploying an agentless monitor.
- List a few troubleshooting tips to help you solve an agentless monitoring issue.

Troubleshooting connectivity

- Firewall

- SNMP needs ports 161 and 162 open
- CIM needs ports 5988 and 5989 open

- TCP stack

Verify TCP connectivity to the remote system using ping and nslookup

- Domain name server

Use nslookup and route to verify that the remote system is known to your domain

Troubleshooting agent log files

- Default location:

%CANDLE_HOME%\TMAITM6\logs_<pc>_k<pc>agent_<instance>_<timestamp>-01.log (**Windows**)

\$CANDLE_HOME/logs/<hostname>_<pc>_<instance>_<timestamp>-01.log (**UNIX or Linux**)

- You can turn up the trace level of different to debug specific problems

Problem area	KBB_RAS1 setting
General Startup/Initialization	ERROR (UNIT:query ALL) <i>running on Windows</i> ERROR (UNIT:ct_main ALL) <i>running on UNIX or Linux</i>
WMI Data Provider	ERROR (UNIT:WMI ALL)
Perfmon Data Provider	ERROR (UNIT: QueryClass ALL)
SNMP Data Provider	ERROR (UNIT:SNMP ALL)
Windows Event Log Data Provider	ERROR (UNIT:EventLog ALL) (UNIT:WinLog ALL)
CIM-XML Data Provider	ERROR (UNIT:CIM ALL)

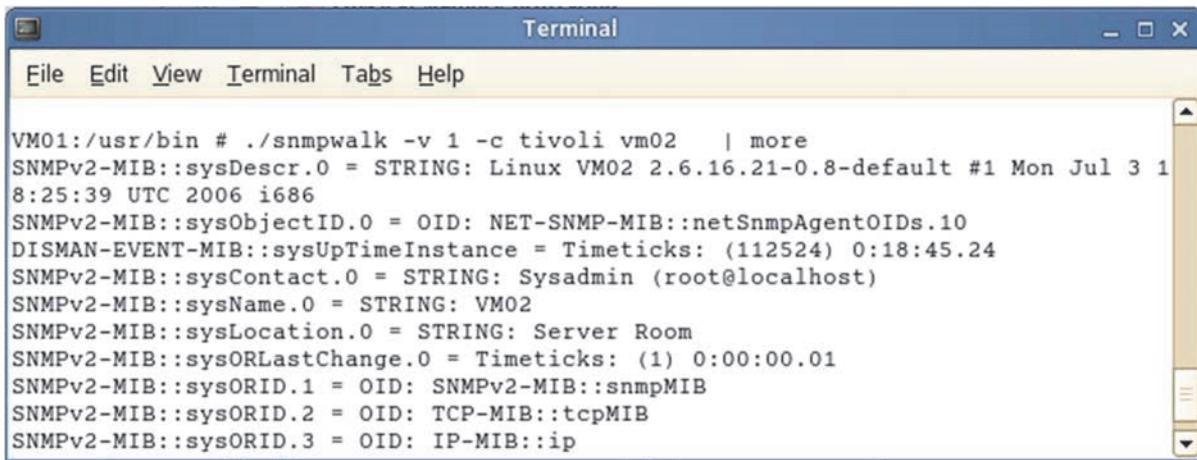
Troubleshooting tips

With agentless monitoring, you must devote some preparation time to verifying the destination system configurations

- Verify that SNMP daemons are installed, configured, and started
- Show MIB branches in SNMP configuration files
- Verify Windows passwords and user account rights for Windows API collection
- Verify patch levels for endpoint systems
 - AIX and SLES 9 systems are most likely to require SNMP patches
 - Solaris CIM-XML systems are most likely to require CIM patches
 - Red Hat, HP-UX, and Windows include patch levels

Troubleshooting tips (continued)

If possible, use tools like **snmpwalk**, **WMIExplorer**, and **perfmon** to verify that the metrics are displayed before you try them in the Tivoli Enterprise Portal



The screenshot shows a terminal window with the title "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main pane displays the output of the command "VM01:/usr/bin # ./snmpwalk -v 1 -c tivoli vm02 | more". The output lists various SNMP MIB variables and their values:

```
VM01:/usr/bin # ./snmpwalk -v 1 -c tivoli vm02 | more
SNMPv2-MIB::sysDescr.0 = STRING: Linux VM02 2.6.16.21-0.8-default #1 Mon Jul 3 1
8:25:39 UTC 2006 i686
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (112524) 0:18:45.24
SNMPv2-MIB::sysContact.0 = STRING: Sysadmin (root@localhost)
SNMPv2-MIB::sysName.0 = STRING: VM02
SNMPv2-MIB::sysLocation.0 = STRING: Server Room
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (1) 0:00:00.01
SNMPv2-MIB::sysORID.1 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.2 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.3 = OID: IP-MIB::ip
```

Student exercise



© Copyright IBM Corporation 2013

32

Student exercise

Perform the exercises for this unit.

Review questions

1. What are the three protocols used by agentless monitors?
2. When using agentless monitoring, is there any required software or configuration on the computer to be monitored?

Review answers

1. What are the three protocols used by agentless monitors?

Simple Network Management Protocol (SNMP), Common Information Model (CIM), and Windows Management Instrumentation (WMI)

2. When using agentless monitoring, is there any required software or configuration on the computer to be monitored?

Yes. Whether you are using SNMP, CIM, or WMI, you must allow access for the agentless monitor to access the target machine to retrieve the data.

Summary

Now that you have completed this unit, you can perform the following tasks:

- Explain the agentless monitoring features and function
- Plan and configure an agentless monitoring environment
- Troubleshoot an agentless monitoring problem



7 Agent Management Services



7 Agent Management Services



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

Agent Management Services are used to monitor the availability of agents and respond automatically if an agent becomes unhealthy or exits unexpectedly.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

Objectives

When you complete this unit, you can perform the following tasks:

- Manage agents with Agent Management Services commands
- Configure the Agent Management Services components
- Issue Agent Management Services Take Action commands
- Navigate and use Agent Management Services workspaces and views

Lesson 1. Monitoring the agents

Lesson 1: Monitoring the agents

- Proxy Agent Services are renamed in IBM Tivoli Monitoring 6.2.2 to Agent Management Services to better describe their function
- More base agents are supported
- There are new capabilities to report and manage agent instances

© Copyright IBM Corporation 2013

3

What this lesson is about

Proxy Agent Services was introduced in Tivoli Monitoring 6.2.1, and has been renamed Agent Management Services. This lesson introduces Agent Management Services, which ensures that monitoring agents stay active and continue to provide monitoring information.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the purpose of Agent Management Services
- Explain the basic function of Agent Management Services.

Basic functions

- Monitor the availability of agents and respond automatically if an agent operates abnormally or ends unexpectedly
- Start, stop, restart, manage, and unmanage an agent
- Stop noncritical processes running on servers
- Enable the deployment of new agents to the operating system

© Copyright IBM Corporation 2013

4

Basic functions

If an agent has stopped, you can have it restart automatically by using Agent Management Services.

Results of using Agent Management Services

- Operators spend less time on agent management
- No longer need to configure situations to be notified of agent failures
- Improves agent availability
- Increases the ability to meet customer-defined service level agreements during peak utilization times

© Copyright IBM Corporation 2013

5

Results of using Agent Management Services

Before Agent Management Services, the customer had to ensure that agents were continuing to function. The ability to monitor the monitoring agents is now a feature of the IBM Tivoli Monitoring product.

Installation

Services installed automatically with these agents:

- Linux OS agent
- UNIX OS agent
- Windows OS agent

Portal user interface contents:

- Agent Management Services workspace
 - Agents' Management Status view
 - Agents' Alerts view
 - Agents' Runtime Status view
 - Agents' Management Definition view
- Agents' Management Log workspace
 - Agents' Management Log view

© Copyright IBM Corporation 2013

6

Installation

Agent Management Services is installed as part of the basic installation. Facilities are provided to use and manage these services.

Managing the agents

- Functions for watchdog and auto-restart are embedded into OS monitoring agents
- A separate process is provided, called the physical watchdog, that monitors the operating system agent itself
- Take Action commands are available that stop, start, enable, and disable monitoring for each agent
- An event-driven situation is provided for critical availability status problems
- Availability policies in the form of XML files are provided by default for the OS agent, physical watchdog, Tivoli Data Warehouse Proxy Agent, Summarization and Pruning Agent, and Agent Builder Agents

© Copyright IBM Corporation 2013

7

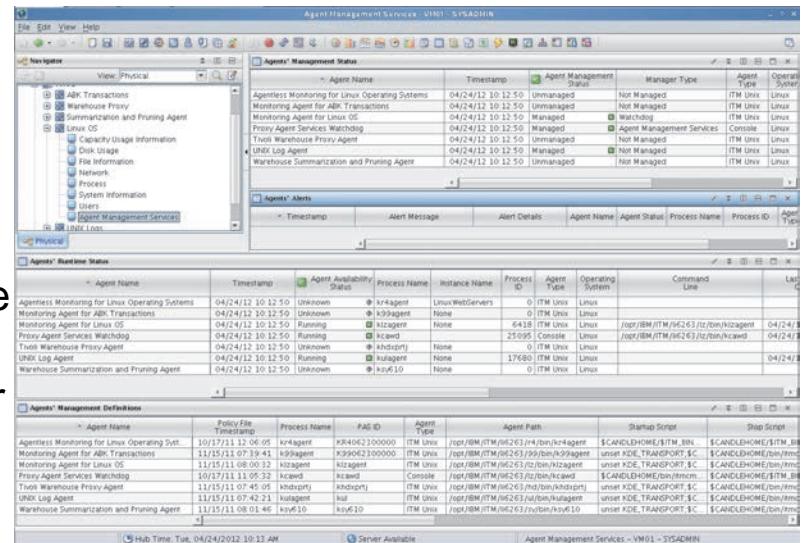
Managing the agents

There are two parts to the Agent Management Services:

- The operating system agents ensure that the separate watchdog program is running.
- The watchdog program ensures that the agents it monitors are running.

Controlling with the portal user interface

- You can start, stop, and restart an agent manually
- Operator can decide whether an Agent Management Service instance takes control of a particular agent



© Copyright IBM Corporation 2013

8

Controlling with the portal user interface

As with other agent monitoring functions, the portal is the user interface that provides the functions for managing the Agent Management Services. Workspaces are provided for Agent Management Services and are available through the physical Navigator.

Lesson 2. Components and configuration

Lesson 2: Components and configuration

- Agent management workspace views provide information for managing for these base agents:
 - Linux OS
 - Windows OS
 - UNIX OS
 - Log file agent
 - Warehouse Proxy
 - Warehouse Summarization and Pruning
 - Agent Builder Agents
 - Agentless Monitoring Agents
- Agent Management Services is not supported on the following OSs:
 - i5/OS
 - z/OS

Agent Name	Timestamp	Agent Management Status	Manager Type
Agentless Monitoring for Linux Operating Systems	04/24/12 10:12:50	Unmanaged	Not Managed
Monitoring Agent for ABK Transactions	04/24/12 10:12:50	Unmanaged	Not Managed
Monitoring Agent for Linux OS	04/24/12 10:12:50	Managed	<input checked="" type="checkbox"/> Watchdog
Proxy Agent Services Watchdog	04/24/12 10:12:50	Managed	<input checked="" type="checkbox"/> Agent Management Services
Tivoli Warehouse Proxy Agent	04/24/12 10:12:50	Unmanaged	Not Managed
UNIX Log Agent	04/24/12 10:12:50	Managed	<input checked="" type="checkbox"/> Not Managed
Warehouse Summarization and Pruning Agent	04/24/12 10:12:50	Unmanaged	Not Managed

© Copyright IBM Corporation 2013

9

What this lesson is about

This lesson covers the components and configuration of Agent Management Services. Tivoli Monitoring provides workspaces for managing agents with Agent Management Services. Individual agents are listed, and the current management status information is displayed in tables.

What you should be able to do

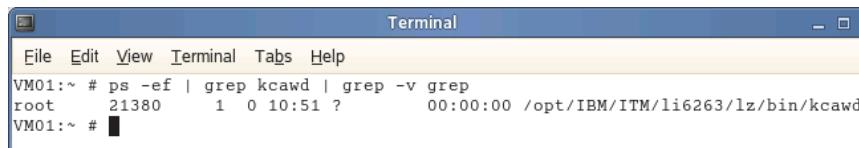
After completing this lesson, you should be able to perform the following tasks:

- Describe the components of Agent Management Services

Agent Management Services watchdog component

- Is used only to watch the OS agent
- Is a stand-alone executable file with the OS agents
- Runs as the process kcawd.exe on Windows
- Runs as the process kcawd on Linux and UNIX systems

Example:



```
VM01:~ # ps -ef | grep kcawd | grep -v grep
root    21380      1  0 10:51 ?          00:00:00 /opt/IBM/ITM/li6263/lz/bin/kcawd
VM01:~ #
```

- Is listed in the Agents' Runtime Status and Agents' Management Definitions views

Agent Name	Timestamp	Agent Availability Status	Process Name	Instance Name
Agentless Monitoring for Linux Operating Systems	04/24/12 10:12:50	Unknown	kr4agent	Linux WebServers
Monitoring Agent for ABK Transactions	04/24/12 10:12:50	Unknown	k99agent	None
Monitoring Agent for Linux OS	04/24/12 10:12:50	Running	livelogs	None
Proxy Agent Services Watchdog	04/24/12 10:12:50	Running	kcawd	None
Tivoli Warehouse Proxy Agent	04/24/12 10:12:50	Unknown	twhelp	None
UNIX Log Agent	04/24/12 10:12:50	Running	kulagent	None
Warehouse Summarization and Pruning Agent	04/24/12 10:12:50	Unknown	ksy610	None

© Copyright IBM Corporation 2013

10

Agent Management Services watchdog component

The **kcawd** process is the Agent Management Services watchdog component. The Agent Management Services watchdog is used only to watch the OS agent. The watchdog is a stand-alone program file with the OS agents. If this process is not running, then the agent watchdog is not monitoring the OS agents to ensure that they are active.

The agent watchdog component restarts the **kcawd** process.

Agent watchdog component

- Runs inside the OS agent process as a logical component
- Watches any monitoring agent that has an XML file in the common agent package (CAP) directory of the OS agent installation
- Performs specific availability monitoring actions against an agent, based on the policy in the agent's CAP file

© Copyright IBM Corporation 2013

11

Agent watchdog component

The agent watchdog ensures that any monitoring agents that have been configured to be monitored are active. The physical watchdog component can ensure that the OS agent is active, ensuring that the agent watchdog functions remain active.

CAP directory definition environment variable



```

File Edit View Terminal Tabs Help
VM01:/opt/IBM/ITM/config # cat lz.ini
# =====
# Generic Settings
# =====
# @preserve CTIRA_HOSTNAME CTIRA_SYSTEM_NAME
KGL_KGLMSGBASE=$CANDLEHOMES/$ARCHITECTURE$/bin/KGLMSENU
ATTRLIB=$CANDLEHOMES$/BINARCH$/$PRODUCTCODE$/tables/ATTRLIB
CT_CMSLIST=$NETWORKPROTOCOLS:$HOSTNAMES:$NETWORKPROTOCOL$MIRRORS
CTIRA_HIST_DIR=$CANDLEHOMES$/BINARCH$/$PRODUCTCODE$/hist
CTIRA_LOG_PATH=$CANDLEHOMES/logs
KBB_ACF1_PATH=$CANDLEHOMES/$ARCHITECTURE$/bin
KBB_RAS1=ERROR
ITM_BINARCH=$BINARCH$
KBB_VARPREFIX=%
KBB_RAS1_LOG=%(CTIRA_LOG_PATH)${RUNNINGHOSTNAME}_$PRODUCTCODE%_$(systask)_$(sysutctime)-.log INVENTORY=%(CTIRA_LOG_PATH)${RUNNINGHOSTNAME}_$PRODUCTCODE%_$(systask).inv COUNT=03 LIMIT=5 PRESERVE=1 MAXFILES=9
KDC_FAMILIES=$NETWORKPROTOCOLS
LOGSHOME=$CANDLEHOMES/logs
LD_LIBRARY_PATH=$ICCRTE_DIRS:$SGSKLIB:$CANDLEHOMES/$BINARCH$/$PRODUCTCODE$/lib:$CANDLEHOMES/$ARCHITECTURE$/lib
LIBPATH=$ICCRTE_DIRS:$SGSKLIB:$CANDLEHOMES/$BINARCH$/$PRODUCTCODE$/lib:$CANDLEHOMES/$ARCHITECTURE$/lib
PATH=/bin:/usr/bin:$CANDLEHOMES/$BINARCH$/$PRODUCTCODE$/bin:$CANDLEHOMES/$ARCHITECTURE$/bin
SHLIB_PATH=$ICCRTE_DIRS:$SGSKLIB:$CANDLEHOMES/$BINARCH$/$PRODUCTCODE$/lib:$CANDLEHOMES/$ARCHITECTURE$/lib:/lib:/usr/lib
SQLLIB=$CANDLEHOMES/tables/cicatrsql/SOLLIB
KDCBO_HOSTNAME=$HOSTNAMES
KDC_PARTITION=$KDC_PARTITIONNAME$ SCOMMENT$TIRA STANDALONE=Y
ITMDEPLOY TIMEOUT=300
ITMDEPLOY AGENTDEPOT=$CANDLEHOMES/tmaitm6/agentdepot
RUNNINGHOSTNAME=$RUNNINGHOSTNAME
KCA_CAP_DIR=$CANDLEHOMES/$BINARCH$/$PRODUCTCODE$/bin/CAP:$CANDLEHOMES/config/CAP:/opt/IBM/CAP
KCA_IP_DIR=$CANDLEHOMES/$BINARCH$/$PRODUCTCODE$/bin/pasipc
KCA_CMD_TIMEOUT=30
KCA_DISCOVERY_INTERVAL=30

```

Example of a Linux OS agent specification of KCA_CAP_DIR

© Copyright IBM Corporation 2013

12

CAP directory definition environment variable

This example shows how the common agent package files are accessed. In the environment file for an OS agent, the KCA_CAP_DIR variable specifies the directories and the order to search them to locate the CAP files.

CAP file

- Default Linux OS file directory is /opt/IBM/ITM/config/CAP
- Files are named k**_default.xml, where ** represents a product or component code

The screenshot shows two windows side-by-side. On the left is a GIMP window titled 'klz_default.xml (/opt/IBM/ITM/config/CAP) - gedit' displaying the XML content of the CAP file. On the right is a Terminal window titled 'Terminal' showing the command 'ls -l' being run in the directory '/opt/IBM/ITM/config/CAP'. The terminal output lists several XML files, with the last few highlighted by a red box: 'k99_default.xml', 'kca_default.xml', 'khd_default.xml', 'klz_default.xml', 'kr4_default.xml', 'ksy_default.xml', 'kul_default.xml', and 'kwgcap.xsd'. The XML code in the GIMP window includes sections for agentName, procName, productCode, agentType, agentPath, agentVersionScript, agentBuildNumberScript, managerType, maxRestarts, startScript, and returnCodeList.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Copyright IBM Corp. 2009 All Rights Reserved -->
<subagents xmlns="http://www.tivoli.ibm.com/itm/kwgcaps" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tivoli.ibm.com/itm/kwgcaps kwgcaps.xsd">
  <!-- DO NOT EDIT THIS PORTION OF THE CAP FILE -->
  <subagent id="klzagent">
    <agentName>Monitoring Agent for Linux OS</agentName>
    <procName>klzagent</procName>
    <productCode>lz</productCode>
    <agentType>ITM_UNIX</agentType>
    <agentPath>$CANDLEHOME/$ITM_BINARCH/lz/bin/itmver.sh lz</agentPath>
    <agentVersionScript>
      <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/itmver.sh lz</command>
    </agentVersionScript>
    <agentBuildNumberScript>
      <command>$CANDLEHOME/$ITM_BINARCH/lz/bin/itmbuildnum.sh lz</command>
    </agentBuildNumberScript>
    <platform xsi:type="PlatformLinux"></platform>
  <!-- END -->
  <checkFrequency>120</checkFrequency>
  <managerType>WatchDog</managerType>
  <maxRestarts>4</maxRestarts>
  <startScript>
    <command>unset KDE_TRANSPORT;$CANDLEHOME/bin/itmcmd agent start lz</command>
  </startScript>
  <returnCodeList>
    <returnCode type="OK">0</returnCode>
    <returnCode type="OK">2</returnCode>
  </returnCodeList>
</subagent>
</subagents>
```

```
VM01:/opt/IBM/ITM/config/CAP # ls -l
total 56
-rwxrwxrwx 1 root root 5618 2011-11-15 07:39 k99_default.xml
-rwxrwxrwx 1 root root 5036 2011-10-17 11:05 kca_default.xml
-rwxrwxrwx 1 root root 4072 2011-11-15 07:45 khd_default.xml
-rwxrwxrwx 1 root root 1793 2011-11-15 08:00 klz_default.xml
-rwxrwxrwx 1 root root 4943 2011-10-17 12:06 kr4_default.xml
-rwxrwxrwx 1 root root 4088 2011-11-15 08:00 ksy_default.xml
-rwxrwxrwx 1 root root 3284 2011-11-15 07:42 kul_default.xml
-rwxrwxrwx 1 root root 9929 2011-10-17 11:05 kwgcap.xsd
-rwxrwxrwx 1 root root 1005 2012-04-18 13:08 pas.dat
VM01:/opt/IBM/ITM/config/CAP #
```

© Copyright IBM Corporation 2013

13

CAP file

This slide shows the contents of the Linux OS CAP file. Settings in the CAP files control the agent watchdog monitoring functions.

CAP file parameters

- **cpuThreshold**
 - Maximum average percent CPU time that the agent process might consume over the checkFrequency seconds before considered unhealthy and then restarted
 - A positive integer from 0 to 100
- **memoryThreshold**
 - Maximum average amount of working set memory that the agent process can consume over the checkFrequency seconds before being considered unhealthy and restarted
 - A string of integers between 0 and 9
 - The threshold must include a unit of either kilobytes, megabytes, gigabytes, or terabytes

© Copyright IBM Corporation 2013

14

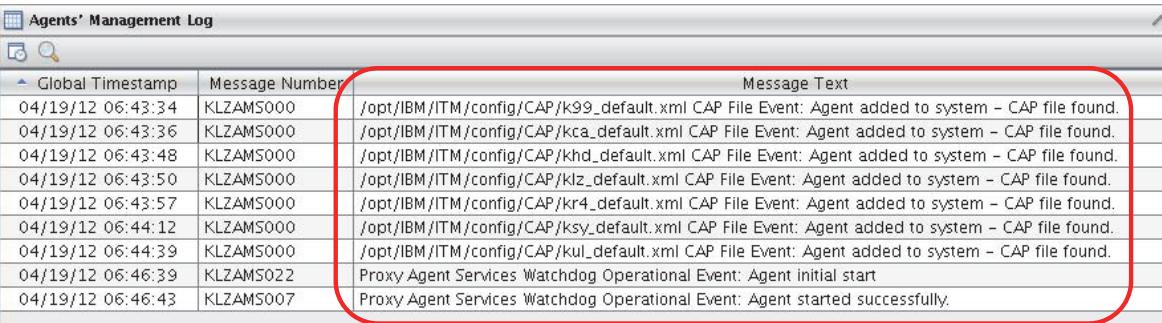
CAP file parameters

The agent watchdog component by default checks whether agents are active every 30 seconds. If an agent is inactive, the agent watchdog can restart the agent up to the number of times specified in the **maxRestarts** parameter.

CPU and memory threshold parameters are established for determining whether an agent is functioning as expected. The agent is evaluated during the interval of time specified in the **checkFrequency** setting. You can then restart it if these thresholds are exceeded during this interval.

Agents' Management Log

- Agents' Management Log workspace and view
- Message text shows the CAP files used for agents added to Agent Management Services



Global Timestamp	Message Number	Message Text
04/19/12 06:43:34	KLZAMS000	/opt/IBM/ITM/config/CAP/k99_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:43:36	KLZAMS000	/opt/IBM/ITM/config/CAP/kca_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:43:48	KLZAMS000	/opt/IBM/ITM/config/CAP/khd_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:43:50	KLZAMS000	/opt/IBM/ITM/config/CAP/klz_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:43:57	KLZAMS000	/opt/IBM/ITM/config/CAP/kr4_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:44:12	KLZAMS000	/opt/IBM/ITM/config/CAP/ksy_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:44:39	KLZAMS000	/opt/IBM/ITM/config/CAP/kul_default.xml CAP File Event: Agent added to system - CAP file found.
04/19/12 06:46:39	KLZAMS022	Proxy Agent Services Watchdog Operational Event: Agent initial start
04/19/12 06:46:43	KLZAMS007	Proxy Agent Services Watchdog Operational Event: Agent started successfully.

© Copyright IBM Corporation 2013

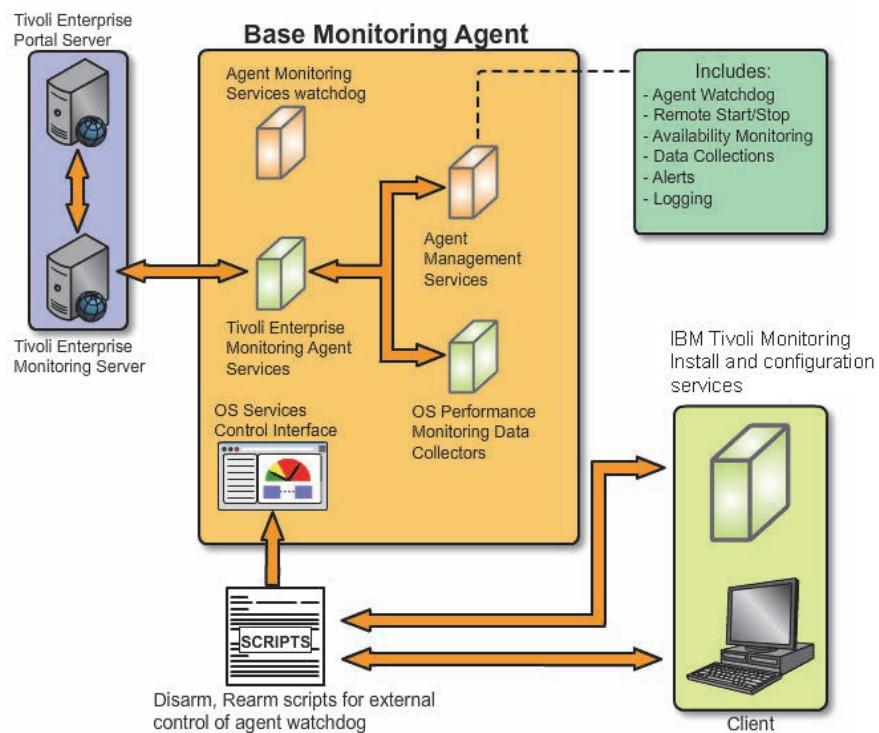
15

Agents' Management Log

To access the Agents' Management Log workspace, perform the following steps:

1. Right-click the Agent Management Services navigator item.
2. Click **Workspace** and select the Agents' Management Log workspace.

Interaction with Tivoli Monitoring components



© Copyright IBM Corporation 2013

16

Interaction with Tivoli Monitoring components

Managing Agent Management Services

- System monitor agents are managed by Agent Management Services by default
- Includes agents created with Tivoli Monitoring Agent Builder running on the same system, though not managed by default
- Use the **disarmWatchdog** command to suspend management
- Use the **rearmWatchdog** command to enable the watchdog

© Copyright IBM Corporation 2013

17

Managing Agent Management Services

In addition to the product-provided agents, Agent Management Services can also monitor agents that the Agent Builder builds. This function extends the Agent Management Services capabilities. It works only with those agents built by Agent Builder that are running on the same system.

Watchdog process control

- Shell script files, as used by a particular Linux OS Agent, are located here:

/opt/IBM/ITM/lz/bin/

- Start with these scripts:

/opt/IBM/ITM/bin/itmcmd execute lz stopWatchdog.sh

/opt/IBM/ITM/bin/itmcmd execute lz disarmWatchdog.sh

/opt/IBM/ITM/bin/itmcmd execute lz statusWatchdog.sh

/opt/IBM/ITM/bin/itmcmd execute lz startWatchdog.sh

/opt/IBM/ITM/bin/itmcmd execute lz rearmWatchdog.sh

Watchdog process control

Shell script files control the watchdog process. Scripts are available for stopping, disarming, displaying the status, rearming, and starting the watchdog process.

For example, you suspend management by using the **disarmWatchdog** command. This command disables the Agent Management Services watchdog for the operating system agent. It also disables Agent Management Services for any agents created with the Tivoli Monitoring Agent Builder on the same system.

Lesson 3. Take Action commands

Lesson 3: Take Action commands

- AMS Start Agent
 - Start an agent that is under Agent Management Services management. For a multiple-instance agent, use AMS Start Agent Instance
- AMS Stop Agent
 - Stop an agent that is under Agent Management Services management
- AMS Start Agent Instance
 - Start a instance of the monitoring agent
- AMS Start Management
 - Put an agent under Agent Management Services management
 - This task is useful when the agent is taken offline intentionally because you are ready to resume running the agent, and you have the agent managed
- AMS Stop Management
 - Remove an agent from Agent Management Services management
 - This task is useful when you want to take an agent offline and not have it restart automatically
- AMS Recycle Agent Instance
 - Stop and start any agent with a single request
- AMS Reset Agent Restart Count
 - Reset the daily restart count of an agent to 0.

© Copyright IBM Corporation 2013

19

What this lesson is about

This lesson introduces the Take Action commands that are part of Agent Management Services.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the steps to run a Take Action command that is part of Agent Management Services
- Explain how to verify that the Take Action command was successful.

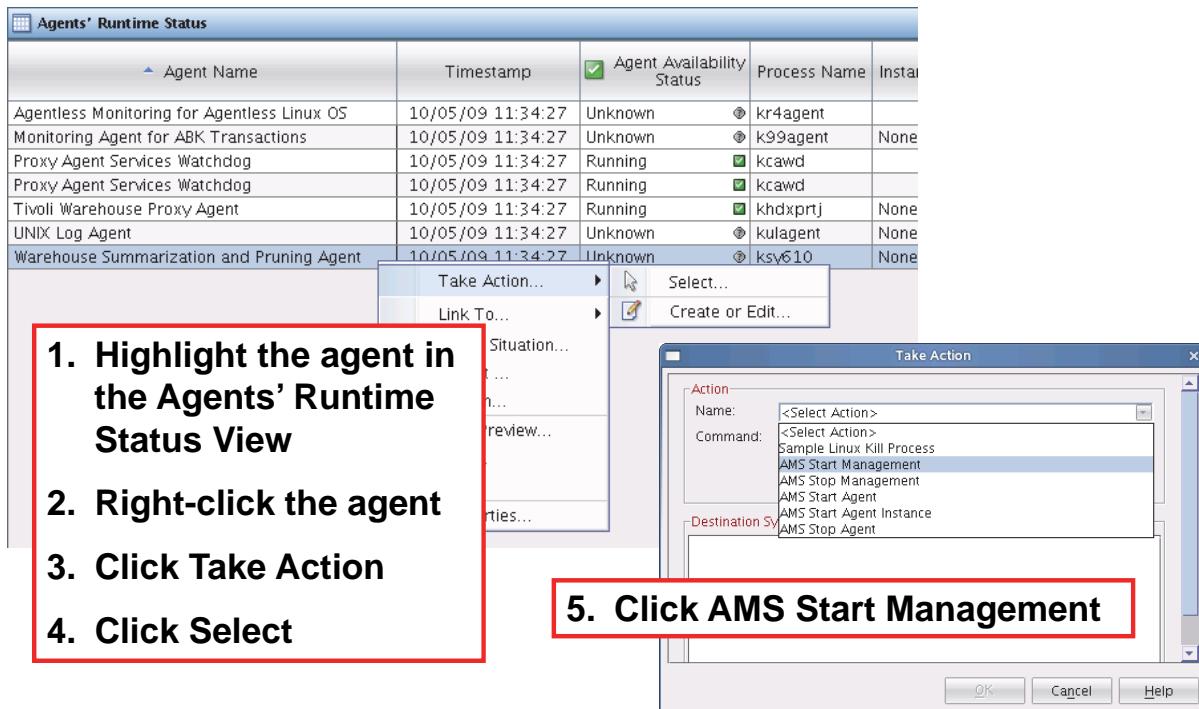
With the portal interface functions, you can issue Take Action commands. For the Agent Management Services, there are five Take Action functions provided by the product, as shown on

this slide. You might be prompted to provide the name of the agent when running some of these commands.



Note: The ability to recycle an operating system agent using the AMS Recycle Agent Instance is a useful feature of Agent Management Services.

Take Action command example



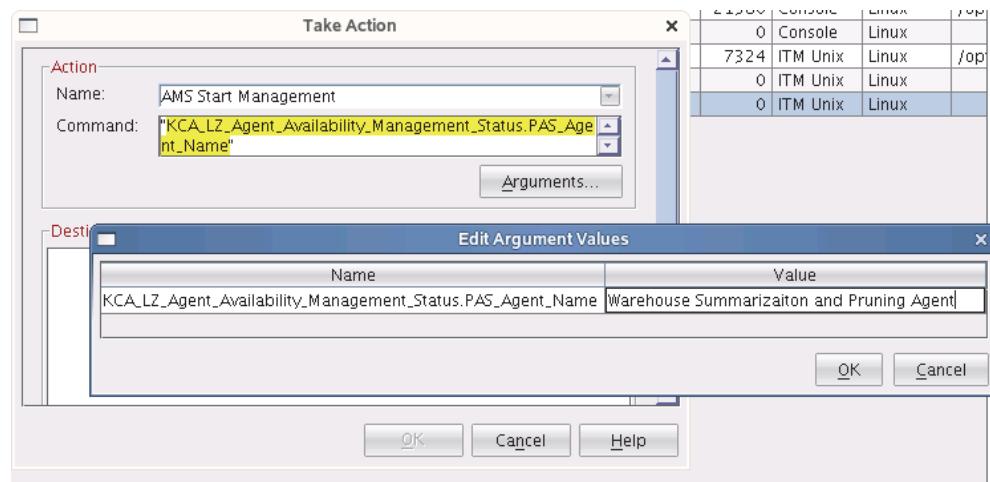
© Copyright IBM Corporation 2013

20

Take Action command example

The next few slides demonstrate the steps involved to use a Take Action command. In this example, you select the Warehouse Summarization and Pruning agent as the agent that is being managed. You highlight the agent and right-click to view the drop-down menu choices. From these choices, you select **Take Action**. When the action choices are displayed, you select **AMS Start Management**.

Take Action command example (continued)



6. Type the name of the agent

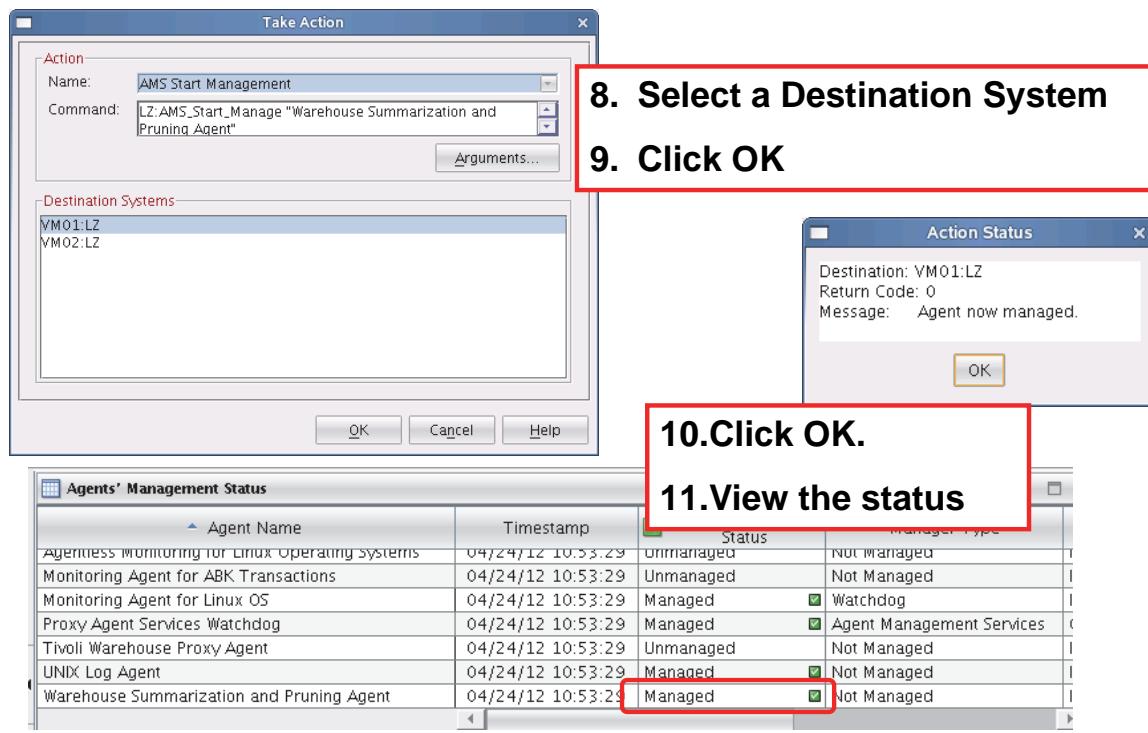
7. Click OK

© Copyright IBM Corporation 2013

21

In the **Value** field of the Edit Argument Values pane, you provide the name of the agent. You enter it as it is shown in the Agents' Runtime Status Agent Name column. Then, you click **OK**.

Take Action command example (continued)



© Copyright IBM Corporation 2013

22

Specify the destination where this command is issued. From the destination choices, select one or more managed systems. Click **OK** on the Take Action window to issue the command.

After the command is issued, an Action Status window opens. This window contains the notification that the action has been issued and indicates the return code. A return code of **0** indicates that the action is successful. Click **OK** to clear this window. Return to the Agents' Management Status view to view the Agent Management Status column. The status is Managed.

Automated procedure

- Agent Management Services queries the OS to determine whether an agent is stopped
 - If the process is not running, Agent Management Services attempts to restart it
- If a managed agent is configured but not started, it is automatically started within 10 minutes of being configured
- Managed agents whose configured instances are started by the user:
 - Are discovered immediately
 - Show in the Agents' Runtime Status view

Agent Name	Timestamp	Agent Availability Status	Process Name	Instance Name	Process ID	Agent Type	Operating System	
Agentless Monitoring for Linux Operating Systems	04/24/12 10:53:29	Unknown	kr4agent	LinuxWebServers	0	ITM Unix	Linux	
Monitoring Agent for ABK Transactions	04/24/12 10:53:29	Unknown	k99agent	None	0	ITM Unix	Linux	
Monitoring Agent for Linux OS	04/24/12 10:53:29	Running	klzagent	None	6418	ITM Unix	Linux	/opt/IE
Proxy Agent Services Watchdog	04/24/12 10:53:29	Running	kcawd		25095	Console	Linux	/opt/IE
Tivoli Warehouse Proxy Agent	04/24/12 10:53:29	Unknown	khdxprtj	None	0	ITM Unix	Linux	
UNIX Log Agent	04/24/12 10:53:29	Running	kulagent	None	17680	ITM Unix	Linux	
Warehouse Summarization and Pruning Agent	04/24/12 10:53:29	Running	ksy610	None	21488	ITM Unix	Linux	/opt/IE

© Copyright IBM Corporation 2013

23

Automated procedure

The Agents' Management Status view from the Agent Management Services workspace provides current information about the agent.

Automated procedure (continued)

- If the number of connection attempts to the monitoring server exceeds **CTIRA_MAX_RECONNECT_TRIES**:
 - The agent attempts to shut down
 - The default is 0
- If the Agent Management Services Watchdog (shown as Proxy Agent Services Watchdog) is running, it immediately restarts the agent
- If you want the agent to shut down when **CTIRA_MAX_RECONNECT_TRIES** is exceeded, disable this watchdog process
- Use the AMS Stop Management action to disable this watchdog process

The **CTIRA_MAX_RECONNECT_TRIES** environment variable is used with the Agent Management Services.

Agent Management Services configuration

- Putting agents under management by watchdog remotely using the portal interface:
 - Easy method of enabling monitoring of an agent versus editing the agent's policy file
 - Select an agent row from the Agents' Management Status view
 - Select AMS Start Manage action to enabling watchdogging and the reporting of availability status
- Manual editing of availability policy files
 - Use only to change a default policy setting, not to enable monitoring, though it can be used for both
 - Do not update identifying information for the agent unless you are certain about what you are doing
 - The watchdog uses the identifying information to locate the agent on the file system and identify it in the OS process table

© Copyright IBM Corporation 2013

25

Agent Management Services configuration

On this slide are two methods for configuring agent management. One method uses the GUI; the other method involves manual editing.

Agent Management Services verification

Verifying a successful configuration

- The Agents' Management Definitions view shows current policy file information
- Agent Availability Status column in the Agent's Runtime Status view reflects the current runtime status of agent process
- Agents' Management Status view shows a value of Managed for Agent Management Status

© Copyright IBM Corporation 2013

26

Agent Management Services verification

After updating a configuration, you must verify it.

The following views are displayed in the Agent Management Services workspace and can help with the verification:

- Agents' Management Definitions
- Agents' Runtime Status
- Agents' Management Status

Lesson 4. Agent Management Services workspace

Lesson 4: Agent Management Services workspace

The screenshot shows the 'Agent Management Services - VM01 - SYSADMIN' interface. The left pane is a Navigator tree with 'Enterprise' selected, showing 'VM01' and its components: ABK Transactions, Warehouse Proxy, Summarization and Pruning Agent, and Linux OS (with sub-options like Capacity Usage Information, Disk Usage, File Information, Network, Process, System Information, Users, and Agent Management Services). The right pane is divided into three sections:

- Agents' Management Status:** A table with columns 'Agent Name', 'Timestamp', 'Agent Management Status', and 'Managed'. It lists several agents: 'Agentless Monitoring for Linux Operating Systems' (Unmanaged), 'Monitoring Agent for ABK Transactions' (Unmanaged), 'Monitoring Agent for Linux OS' (Managed), 'Proxy Services Watchdog' (Managed), and 'Tivoli Warehouses Prov Agent' (Unmanaged).
- Agents' Alerts:** An empty table with columns 'Timestamp', 'Alert Message', 'Alert Details', 'Agent Name', and 'Agent Status'.
- Agents' Runtime Status:** A table with columns 'Agent Name', 'Timestamp', 'Agent Availability Status', 'Process Name', 'Instance Name', 'Process ID', 'Agent Type', 'Operating System', and 'Status'. It shows the status of the same five agents from the first table.

At the bottom, there are status indicators: 'Hub Time: Tue, 04/24/2012 10:57 AM', 'Server Available', and 'Agent Management Services - VM01 - SYSADMIN'.

© Copyright IBM Corporation 2013

27

What this lesson is about

This lesson introduces the workspaces and views that are part of Agent Management Services.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the various views and workspaces that are part of Agent Management Services

This slide shows the Agent Management Services workspace. It contains information in views with these titles:

- Agents' Management Status
- Agents' Alerts
- Agents' Runtime Status
- Agents' Management Definitions

Agents' Management Definitions view

Agents' Management Definitions						
Agent Name	Policy File Timestamp	Process Name	PAS ID	Agent Type	Agent Path	
Agentless Monitoring for Linux Operating Syst...	10/17/11 12:06:05	kr4agent	KR4062300000	ITM Unix	/opt/IBM/ITM/ii6263/r4/bin/kr4agent	
Monitoring Agent for ABK Transactions	11/15/11 07:39:41	k99agent	K99062300000	ITM Unix	/opt/IBM/ITM/ii6263/99/bin/k99agent	
Monitoring Agent for Linux OS	11/15/11 08:00:32	klzagent	klzagent	ITM Unix	/opt/IBM/ITM/ii6263/lz/bin/klzagent	
Proxy Agent Services Watchdog	10/17/11 11:05:32	kcawd	kcawd	Console	/opt/IBM/ITM/ii6263/lz/bin/kcawd	
Tivoli Warehouse Proxy Agent	11/15/11 07:45:05	khdxprtj	khdxprtj	ITM Unix	/opt/IBM/ITM/ii6263/hd/bin/khdxprtj	
UNIX Log Agent	11/15/11 07:42:21	kulagent	kul	ITM Unix	/opt/IBM/ITM/ii6263/ui/bin/kulagent	
Warehouse Summarization and Pruning Agent	11/15/11 08:01:46	ksy610	ksy610	ITM Unix	/opt/IBM/ITM/ii6263/sy/bin/ksy610	

Startup Script	Stop Script	Status Script
unset KDE_TRANSPORT;\$CANDLEHOME/bin/... \$CANDLEHOME/bin/itmcmd agent stop 99	\$CANDLEHOME/\$ITM_BINARCH/lz/bi	
unset KDE_TRANSPORT;\$CANDLEHOME/bin/... \$CANDLEHOME/bin/itmcmd agent stop hd	\$CANDLEHOME/\$ITM_BINARCH/lz/bi	
unset KDE_TRANSPORT;\$CANDLEHOME/bin/... \$CANDLEHOME/bin/itmcmd agent stop lz	\$CANDLEHOME/\$ITM_BINARCH/lz/bi	
\$CANDLEHOME/\$ITM_BINARCH/lz/bin/age...	\$CANDLEHOME/\$ITM_BINARCH/lz/bin/a...	\$CANDLEHOME/\$ITM_BINARCH/lz/bi
unset KDE_TRANSPORT;\$CANDLEHOME/bin/... \$CANDLEHOME/bin/itmcmd agent stop sy	\$CANDLEHOME/\$ITM_BINARCH/lz/bi	
unset KDE_TRANSPORT;\$CANDLEHOME/bin/... \$CANDLEHOME/bin/itmcmd agent stop ul	\$CANDLEHOME/\$ITM_BINARCH/lz/bi	
\$CANDLEHOME/bin/itmcmd execute -c lz ...	\$CANDLEHOME/\$ITM_BINARCH/lz/bin/s...	\$CANDLEHOME/\$ITM_BINARCH/lz/bi

Configuration Script	Operating System	Operating System Version	Operating System Name	Kernel Release	Check Frequency	Service Name	Dependencies
	Linux				180		
	Linux				120		
	Linux				120		
	Linux				180		
	Linux				120		
	Linux				120		
	Linux				120		

© Copyright IBM Corporation 2013

28

Agents' Management Definitions view

The Agents' Management Definitions view contains the following information about agents:

- Agent Name:** The subagent name.
- Policy File Timestamp:** The date and time of the CAP file.
- Process Name:** The process name of the managed agent.
- PAS ID (PAS is now AMS):** The PAS subagent ID.
- Agent Type:** The watched agent type. Valid values are Unknown (0), ITM_Unix (1), Console (2), Windows_Service (3), Discover_ITM (4), Discover_Bin (5), Linux_Service (6), and ITM_Windows (7).
- Agent Path:** The fully qualified path to the agent.
- Startup Script:** The agent startup script.
- Status Script:** The agent status script.
- Stop Script:** The agent stop script.
- Check Frequency:** The frequency to check status, in seconds.

- **Maximum Daily Restarts:** The maximum number of restarts allowed. The clock begins at midnight.
- **Memory Threshold:** The maximum memory allowed.
- **Memory Unit:** The maximum memory allowed units. Valid values include Bytes (0), KB (1), MB (2), and GB (3).
- **% CPU Threshold:** The maximum CPU allowed.
- **Server Name:** The origin node of the collecting agent.
- **Timestamp:** The date and time when the Tivoli Enterprise Monitoring Server samples the data. It uses the standard 16-character date and time format (`CYYMMDDHHMMSSmmm`).

Agents' Runtime Status view

Agents' Runtime Status								
Agent Name	Timestamp	Agent Availability Status	Process Name	Instance Name	Process ID	Agent Type	Operating System	
Agentless Monitoring for Linux Operating Systems	04/24/12 11:09:09	Unknown	kr4agent	LinuxWebServers	0	ITM Unix	Linux	
Monitoring Agent for ABK Transactions	04/24/12 11:09:09	Unknown	k99agent	None	0	ITM Unix	Linux	
Monitoring Agent for Linux OS	04/24/12 11:09:09	Running	klzagent	None	6418	ITM Unix	Linux	
Proxy Agent Services Watchdog	04/24/12 11:09:09	Running	kcawd		25095	Console	Linux	
Tivoli Warehouse Proxy Agent	04/24/12 11:09:09	Unknown	khdxprjt	None	0	ITM Unix	Linux	
UNIX Log Agent	04/24/12 11:09:09	Running	kulagent	None	17680	ITM Unix	Linux	
Warehouse Summarization and Pruning Agent	04/24/12 11:09:09	Running	ksy610	None	22431	ITM Unix	Linux	

Command Line	Last Health Check	Check Frequency	Daily Restart Count	Total Size (Pages)
	N/A	180	0	0
	N/A	180	0	0
/opt/IBM/ITM/ii6263/lz/bin/klzagent	04/24/12 11:07:14	120	0	34,734
/opt/IBM/ITM/ii6263/lz/bin/kcawd	04/24/12 11:09:01	120	0	10,478
	N/A	120	0	0
	04/24/12 11:09:01	120	0	30,163
/opt/IBM/ITM/ii6263/sy/bin/ksy610 console	04/24/12 11:09:01	120	1	164,023

Resident Size	Process System CPU (Percent)	Process User CPU (Percent)	Number of Threads	Page Faults Per Second	Agent Host Name	IP Address	Parent Process ID	User Name	Server Name
0	0	0	0	0	VM01	192.168.100.101	0	Not Available	VM01:LZ
0	0	0	0	0	VM01	192.168.100.101	0	Not Available	VM01:LZ
3,837	36	41	62	15	VM01	192.168.100.101	1	root	VM01:LZ
1,187	0	0	7	0	VM01	192.168.100.101	1	root	VM01:LZ
0	0	0	0	0	VM01	192.168.100.101	0	Not Available	VM01:LZ
1,912	1	0	55	0	VM01	192.168.100.101	1	Not Available	VM01:LZ
20,210	778	644	54	208	VM01	192.168.100.101	1	root	VM01:LZ

© Copyright IBM Corporation 2013

29

Agents' Runtime Status view

The Agents' Runtime Status view contains this information:

- Agent Name:** The watched agent name.
- Timestamp:** The date and time when the Tivoli Enterprise Monitoring Server samples the data. It uses the standard 16-character date and time format (CYYMMDDHHMMSSmmm).
- Agent Availability Status:** The watched agent availability status. Valid values include Unknown (0), Not_found (1), Stopped (2), Start_Pending (3), Running (4), Manually_Stopped (5), Stop_Pending (6), and Not_Configured (7). For agents with an Availability Status of Running, use the attribute group to see run time properties of the agent, such as its Process ID and Thread Count.
- Process Name:** The process name.
- Instance Name:** The instance name of the running Tivoli Monitoring agent.
- Process ID:** The process ID.
- Agent Type:** The watched agent type. Valid values are Unknown (0), ITM_Unix (1), Console (2), Win_Service (3), Discover_ITM (4), Discover_Bin (5), Linux_Service (6), and ITM_Windows (7).

- **Operating System:** The operating system identification. Valid values are Unknown (0), Windows (1), and Linux (2).
- **Command Line:** The command line.
- **Last Health Check:** The last health check timestamp.
- **Check Frequency:** The frequency to check status, in seconds.
- **Daily Restart Count:** The restarts within a period of a day.
- **Total Size (Pages):** The total memory size in pages.
- **Resident Size:** The process resident size.
- **Process System CPU (Percent):** The system CPU.
- **Process User CPU (Percent):** The user CPU time.
- **Number of Threads:** The thread count.
- **Page Faults Per Second:** The total page faults.
- **Agent Host Name:** The host name of the agent.
- **IP Address:** The IP address of the agent.
- **Parent Process ID:** The parent process ID.
- **User Name:** The user name of the running managed agent.
- **Server Name:** The origin node of the collecting agent.

Agents' Management Status view

Agents' Management Status					
Agent Name	Timestamp	Agent Management Status	Manager Type	Agent Type	Operating System
Agentless Monitoring for Linux Operating Systems	04/24/12 11:40:04	Unmanaged	Not Managed	ITM Unix	Linux
Monitoring Agent for ABK Transactions	04/24/12 11:40:04	Unmanaged	Not Managed	ITM Unix	Linux
Monitoring Agent for Linux OS	04/24/12 11:40:04	Managed	Watchdog	ITM Unix	Linux
Proxy Agent Services Watchdog	04/24/12 11:40:04	Managed	Agent Management Services	Console	Linux
Tivoli Warehouse Proxy Agent	04/24/12 11:40:04	Unmanaged	Not Managed	ITM Unix	Linux
UNIX Log Agent	04/24/12 11:40:04	Managed	Not Managed	ITM Unix	Linux
Warehouse Summarization and Pruning Agent	04/24/12 11:40:04	Managed	Not Managed	ITM Unix	Linux

Service Name	Agent Version	Build Number	Server Name
06230000	201107141418	VM01:LZ	
06230000	201110070939	VM01:LZ	
06230000	12141	VM01:LZ	
06230000	12141	VM01:LZ	
06230000	d1228a	VM01:LZ	
06230000	12141	VM01:LZ	
06230000	d1177a	VM01:LZ	

© Copyright IBM Corporation 2013

30

Agents' Management Status view

The Agents' Management Status view contains this information:

- **Agent Name:** The watched agent name.
- **Timestamp:** The date and time when the Tivoli Enterprise Monitoring Server samples the data. The standard 16-character date and time format is used (CYYMMDDHHMMSSmmm).
- **Agent Management Status:** The watched agent management status. Valid values include Unmanaged (0), Managed (1), and Watchdog (2). A value of Managed means that the agent is under the management of Agent Management Services. A value of Unmanaged means that it is known, but not under the management of Agent Management Services.
- **Manager Type:** The enumerated value defining the manager type. Valid values include Unknown (0), Not_Managed (1), Agent_Management_Services (2), Watchdog (3), and External (4). A value of Agent Management Services means that Agent Management Services is responsible. A value of NotManaged means that the agent is not under availability monitoring by any application. A value of Externally means that some other application besides Agent Management Services is responsible for availability monitoring of the agent, such as Tivoli System Automation.

- **Agent Type:** The watched agent type. Valid values are Unknown (0), ITM_Uinx (1), Console (2), Windows_Service (3), Discover_ITM (4), Discover_Bin (5), Linux_Service (6), and ITM_Windows (7).
- **Operating System:** The operating system identification. Valid values are Win2000 (0), Win2003 (1), Win2008 (2), AIX (3), Linux (4), UNKNOWN (5), and NA (-1).
- **Service Name:** The service name.
- **Agent Version:** The VRM information for the agent.
- **Build Number:** The build number information for the agent.
- **Server Name:** The origin node of the collecting agent.

Agents' Alerts view

Agents' Alerts			
Timestamp	Alert Message	Alert Details	Agent Name
04/24/12 11:07:02	Agent abnormally stopped	N/A	Warehouse Summarization and Pruning Agent

Agent Status	Process Name	Process ID	Agent Type	Operating System	Server Name
Stopped	ksy610	0	ITM Unix	Linux	VM01:LZ

© Copyright IBM Corporation 2013

31

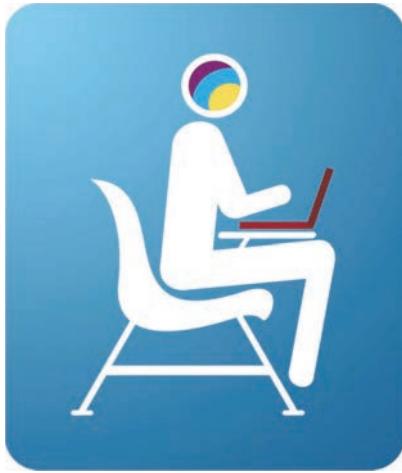
Agents' Alerts view

The Agents' Alerts view contains Agent Management Services error information that includes the following components:

- **Agent Name:** The watched agent name.
- **Agent Status:** The agent status. Valid values include Unknown (0), Not_found (1), Stopped (2), Start_Pending (3), Running (4), Manually_Stopped (5), Stop_Pending (6), and Not_Configured (7).
- **Agent Type:** The watched agent type. Valid values are Unknown (0), ITM_Unix (1), Console (2), Windows_Service (3), Discover_ITM (4), Discover_Bin (5), Linux_Service (6), and ITM_Windows (7).
- **Alert Details:** The alert message details.
- **Alert Message:** The alert message. Valid values include Availability_policy_removed (1), Managed_agent_removed_from_system (2), Unmanaged_agent_removed_from_system (3), Agent_abnormally_stopped (4), Agent_exceeded_restart_count (5), Agent_restart_failed (6), Agent_overutilizing_memory (7), Agent_overutilizing_CPU (8), and Agent_manual_stop_failed (9).

- **Operating System:** The operating system identification. Valid values are Unknown (0), Windows (1), and Linux (2).
- **Process ID:** The process ID.
- **Process Name:** The process name.
- **Server Name:** The origin node of the collecting agent.
- **Timestamp:** The date and time when the Tivoli Enterprise Monitoring Server samples the data. The standard 16-character date and time format is used (CYYMMDDHHMMSSmmm).

Student exercise



© Copyright IBM Corporation 2013

32

Student exercises

Perform the exercises for this unit.

Review questions

1. What is used in the portal interface to control the agents with Agent Management Services?
2. What are the XML files called that are used to manage the agents with Agent Management Services?
3. With which agents is Agent Management Services installed automatically?
4. What is the purpose of the physical watchdog, the process named **kcawd**?

Review answers

1. What is used in the portal interface to control the agents with Agent Management Services?

Take Action commands from the Agent Management Services workspace views

2. What are the XML files called that are used to manage the agents with Agent Management Services?

CAP or common agent package files

3. With which agents is Agent Management Services installed automatically??

The operating system agent for Linux, UNIX, and Windows

4. What is the purpose of the physical watchdog, the process named **kcawd**?

To monitor the OS agent and ensure that it is running

Summary

Now that you have completed this unit, you can perform the following tasks:

- Manage agents with Agent Management Services commands
- Configure the Agent Management Services components
- Issue Agent Management Services Take Action commands
- Navigate and use Agent Management Services workspaces and views



8 Managing situation event integration



8 Managing situation event integration



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

This unit examines the integration of Netcool/OMNibus, Tivoli Enterprise Console, and IBM Tivoli Monitoring.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

Objectives

When you complete this unit, you can perform the following tasks:

- Describe the ways in which IBM Tivoli Monitoring integrates with Netcool/OMNibus and the Tivoli Enterprise Console
- List the steps involved in forwarding situation events to Netcool/OMNibus
- Forward situation events to Netcool/OMNibus and the Tivoli Enterprise Console
- Customize slot values for situations sent to Netcool/OMNibus and the Tivoli Enterprise Console

Lesson 1. Integration points to other event repositories

Lesson 1: Integration points to other event repositories

- You can forward situation events to these locations:
 - Netcool/OMNIbus
 - Tivoli Enterprise Console
- You can show events in workspaces with the following view types:
 - Common Event Console
 - Tivoli Event Console
- You can send SNMP events to SNMP managers from the following sources:
 - The Universal Agent
 - Monitoring agents running in autonomous mode

© Copyright IBM Corporation 2013

3

What this lesson is about

This lesson explores the ways that IBM Tivoli Monitoring integrates with other event repositories.

What you should be able to do

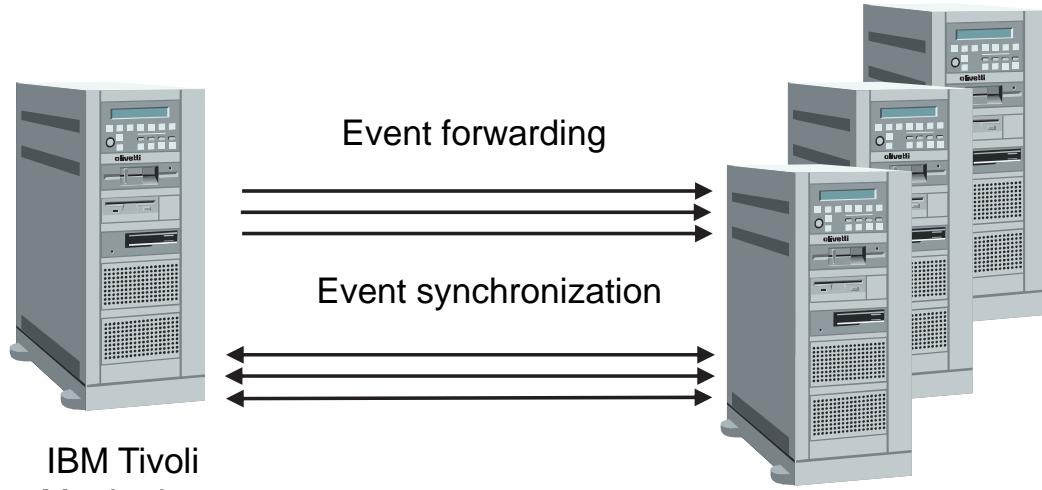
After completing this lesson, you should be able to perform the following tasks:

- Explain how Tivoli Monitoring can forward situation events to other event repositories.
- Describe how Tivoli Monitoring can display events from other event repositories in the portal client.

The Tivoli Enterprise Portal can integrate with other event repositories in three ways:

- You can forward situation events to other event repositories.
- You can show event repositories in the portal server.
- You can send SNMP traps to other SNMP managers.

Situation event integration



- Netcool/OMNIBus
- Tivoli Enterprise Console
- Event Synchronization Component
- Event Integration Facility

© Copyright IBM Corporation 2013

4

Situation event integration

Many installations manage events from multiple event emitters with a central console, such as Tivoli Enterprise Console or Netcool/OMNIBus. Tivoli Monitoring can forward situation events to external event managers, and you can synchronize these events as their status changes.

Situation event integration using workspaces

- Situation event console view
 - IBM Tivoli Monitoring situation events
 - Events for specific Navigator item
 - Sort, filter, acknowledge, or close (pure event)
- Common event console view
 - Normalized view of events from multiple event managers
 - Sort, filter, acknowledge, or close (pure event)
- Tivoli Enterprise Console view
 - Events sent to Tivoli Enterprise Console
 - Same function as Java console
 - Filter by severity, status, or source

Situation event status can be seen and managed in these types of views. The views offer various levels of function and scope.

Common Event Console view

The screenshot shows a table titled "Common Event Console" with columns: Severity, Status, Repository, Name, Display Item, Source, Time Created, and Type. The table lists 12 events. Several rows are highlighted with red boxes around the "Name" and "Display Item" columns, specifically for entries from ITM, OMNIBus, and TEC repositories.

Severity	Status	Repository	Name	Display Item	Source	Time Created	Type
Critical	Open	ITM	Invalid_ssh_attempt		VM01:KUL	10/02/09 11:12:17	Pure
Warning	Open	ITM	Apache_Down_Linux		VM01:LZ	10/02/09 10:54:45	Sampled
Critical	Open	OMNIBus	ITM_Log_Entries		VM01:KUL	10/02/09 11:12:16	Pure
Warning	Open	OMNIBus	ITM_KLZ_Process		VM01:LZ	10/02/09 10:54:45	Sampled
Warning	Open	OMNIBus	Unix_Event_List			10/02/09 10:34:01	Pure
Warning	Open	OMNIBus	Unix_Event_List		VM03	10/02/09 09:35:17	Pure
Warning	Open	OMNIBus	Unix_Conductor		VM03	10/02/09 09:35:04	Pure
Unknown	Open	OMNIBus					Pure
Critical	Open	TEC	ITM_Log_Entries				Pure
Warning	Open	TEC	ITM_KLZ_Process				Sampled
Harmless	Open	TEC	TEC_Start				Pure
Minor	Open	TEC	ITM_Generic				Pure

You can forward one situation event to more than one management system

You can consolidate events from multiple event management systems

© Copyright IBM Corporation 2013

6

Common Event Console view

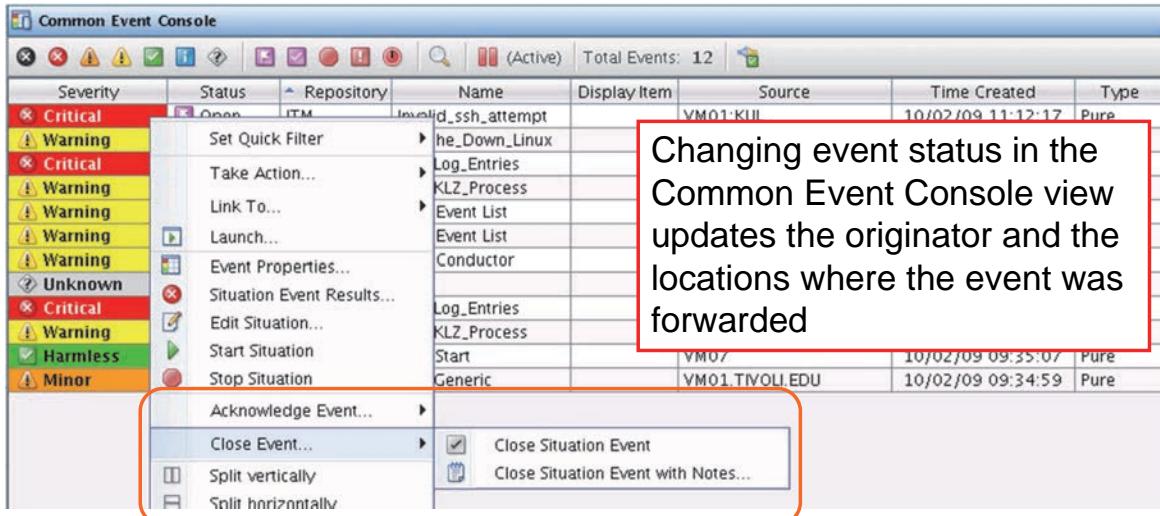
You can see situation events in a A view, which is limited to events generated by Tivoli Monitoring itself. The Common Event Console view displays a normalized look at events from multiple event managers. You can sort and filter the events in this view to make event management easier.



Note: You cannot close sampled events in IBM Tivoli Monitoring, but you can close them in Tivoli Enterprise Console and Netcool/OMNIBus. When that happens, the sampled situation event is not closed in IBM Tivoli Monitoring. It is acknowledged with a specified expiration time for that acknowledgment.

If the acknowledgment of the situation expires and the situation is still true, then a new situation event is opened in Tivoli Enterprise Console and Netcool/OMNIBus. If the situation becomes false at any time, then it resets itself in Tivoli Monitoring, closing the situation event. The default acknowledgment expiration time is 59 minutes.

Common Event Console: Close or acknowledge events



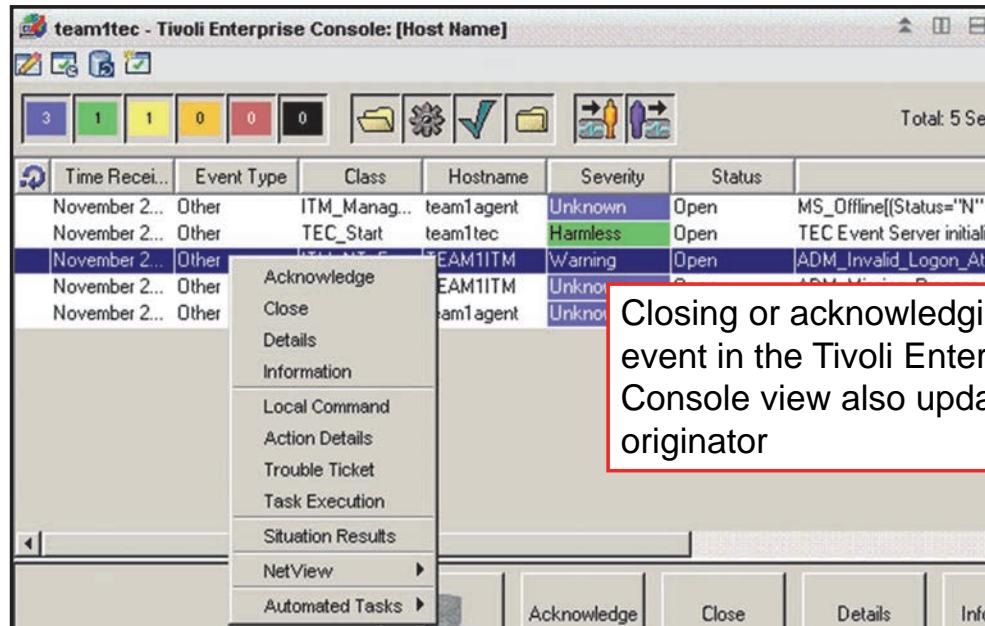
© Copyright IBM Corporation 2013

7

Common Event Console: Close or acknowledge events

You can centrally manage events from this view and make changes to the status of events flow to the originating sources. You can also see status changes made in either Tivoli Enterprise Console or Netcool/OMNIbus in this view.

Tivoli Enterprise Console view: Close or acknowledge events



© Copyright IBM Corporation 2013

8

Tivoli Event Console view: Close or acknowledge events

The Tivoli Enterprise Console view provides the same functions as Tivoli Enterprise Console itself.

Agent autonomy forwards events to Netcool/OMNIbus

Netcool/OMNIbus Event List : Filter="All Events", View="Default"					
File Edit View Alerts Tools		Default		Help	
Node	Alert Group	Summary			Last Occurrence
vm02:LZ	Threshold (0)	Apache Down Linux (Process_Command_Name = httpd2-prefork)			10/06/09 18:15:04
VM03	Unix Event List	A e@CA8C403@CA8C403:0.0 process e@CA8C403@CA8C403:0.0 running on			10/06/09 17:06:29
VM03	Unix Conductor	A c@xxxxxxxxx@xxxxxxxx:0.0 process c@xxxxxxxx@xxxxxxxx:0.0 running on V			10/06/09 17:06:26
vm02:LZ	Status (3)	Heartbeat Message			10/06/09 18:15:01
vm02:LZ	Node Configuration (4)	Disconnected from TEMS VM02			10/06/09 18:14:34
vm02:LZ	Status (3)	Monitoring situation Linux_High_Zombies stopped			10/06/09 18:14:34
VM03		mttrapd probe on VM03: Heartbeat Message			10/06/09 18:12:01
vm02:LZ	Node Configuration (4)	Connected to TEMS VM02 using ip.pipe(1918)			10/06/09 18:03:18

© Copyright IBM Corporation 2013

9

Agent autonomy forwards events to Netcool/OMNIbus

Another way situation events integrate with Netcool/OMNIbus is by having agents running in autonomous mode sending events.

Lesson 2. Situation event integration implementation

Lesson 2: Situation event integration implementation

- Plan your environment
 - Single hub monitoring server and single event server
 - Single hub monitoring server and multiple event servers
 - Multiple hub monitoring servers and single event server
 - Multiple hub monitoring servers and multiple event servers
- For Netcool/OMNIbus, install and configure the Tivoli EIF (Event Integration Facility) probe
 - Copy **tivoli_eif.rules** to ObjectServer
 - Recycle the Tivoli EIF OMNIbus probe
 - Install the Netcool/OMNIbus fix pack 10 (or later):
7.2.1.4-TIV-NCOMNIbus-linux2x86-FP0010.jar

© Copyright IBM Corporation 2013

10

What this lesson is about

This lesson covers the implementation steps to integrate Tivoli Monitoring with other event repositories.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Explain the steps to integrate Tivoli Monitoring to other event repositories.

This slide summarizes the implementation tasks.

Installing the event synchronization component

- All event servers
 - Tivoli Enterprise Console
 - Netcool/OMNIbus
- Install using the wizard or command line
- When running on the Tivoli Enterprise Console, copy the baroc files to the target Tivoli Enterprise Console Server, import the CLASSES, compile and import the updated rule base
- Requires event server restart

© Copyright IBM Corporation 2013

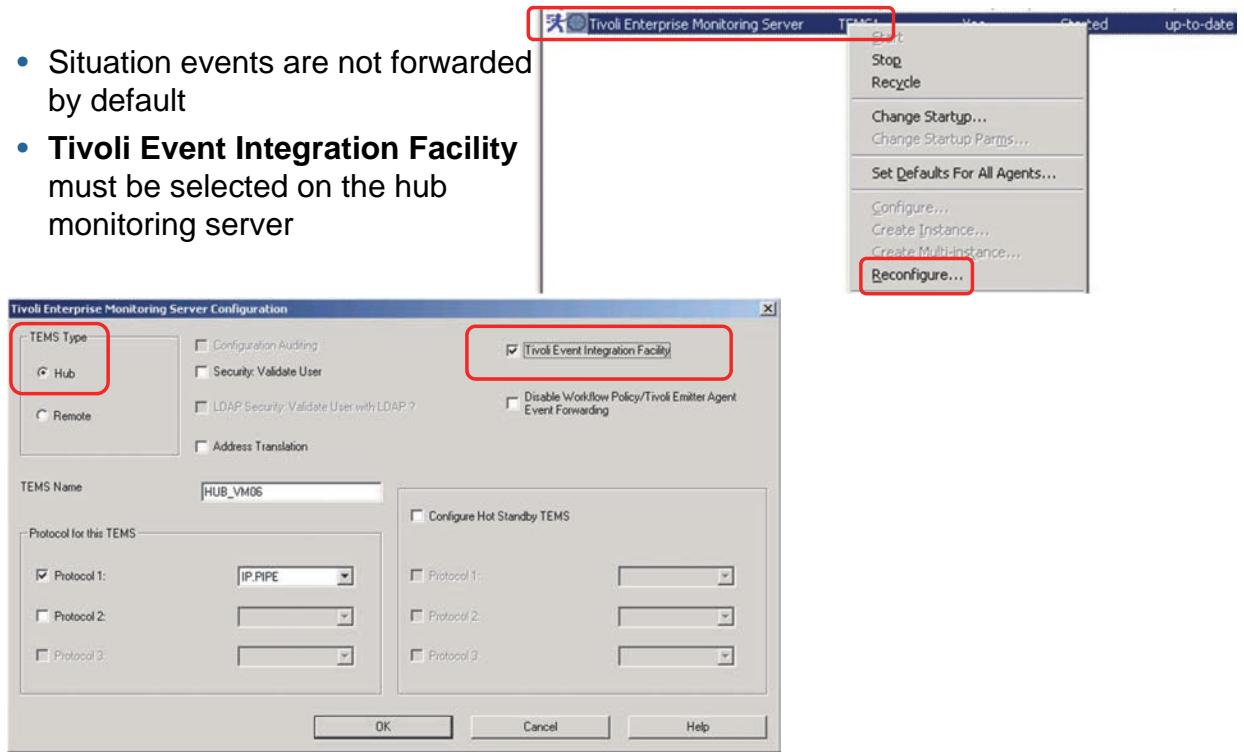
11

Installing the event synchronization component

Event synchronization probes are installed on each event server that participates in central event management. These probes use application programming interfaces to send event notification and status changes.

Configuring the monitoring server to forward events (Windows)

- Situation events are not forwarded by default
- **Tivoli Event Integration Facility** must be selected on the hub monitoring server



© Copyright IBM Corporation 2013

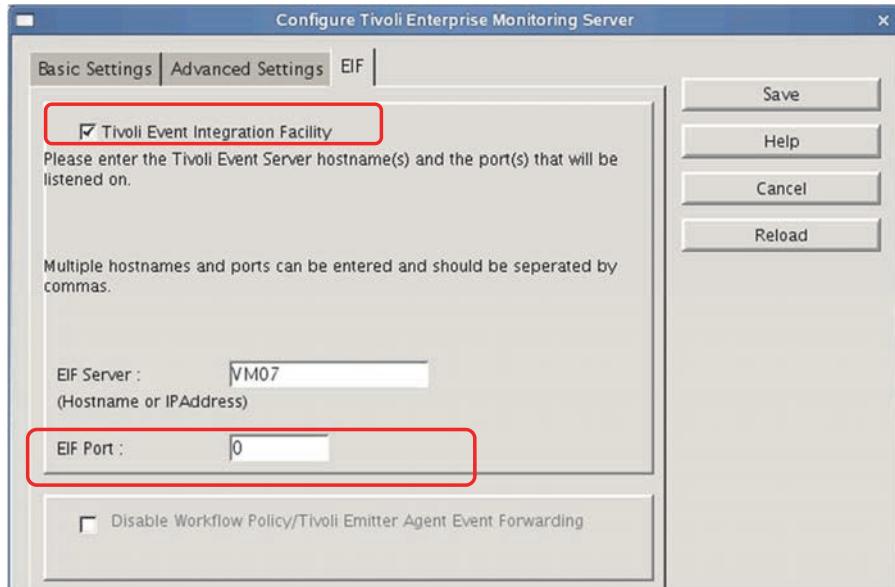
12

Configuring the monitoring server to forward events (Windows)

On Windows systems, you must configure the monitoring server to support event forwarding by selecting the **Tivoli Event Integration Facility** check box. You also must specify the EIF server and the port number. If you change the EIF server, the port number, or whether or not you want situation events forwarded to an EIF server, you must recycle the monitoring server.

Configuring the monitoring server to forward events (Linux)

- Situation events are not forwarded by default
- The **Tivoli Event Integration Facility** must be selected on the hub monitoring server



© Copyright IBM Corporation 2013

13

Configuring the monitoring server to forward events (Linux)

On Linux systems, you must configure the hub monitoring server to support event forwarding by selecting the **Tivoli Event Integration Facility** check box. You also must specify the EIF server and the port number. If you change the EIF server, the port number, or whether or not you want situation events forwarded to an EIF server, you must recycle the monitoring server.

Event forwarding on z/OS

- Event forwarding on z/OS has almost identical functions as distributed platforms, but with minor differences
- Uses the ported version of the EIF library to deliver events to event listeners (kef component)
- Uses in-core event cache; cannot check generated events in an external cache file
- Updated attribute files are not dynamically refreshed
 - These files include dynamically uploaded Universal Agent attribute files
 - They must be manually refreshed using the refreshTecInfo CLI
- Event destination (EVNTSEVER) and event map tables (EVNTMAP) are used

Event forwarding on z/OS (continued)

- DDNAMES in the proc

```
//QA1DEVMP      DD DISP=SHR,DSN=&RVHILEV..&SYS..RKDSEVMP  
//QA1DEVSR      DD DISP=SHR,DSN=&RVHILEV..&SYS..RKDSEVSR  
//DSEVT        DD DISP=SHR,DSN=&RHILEV..&SYS..RKDSEVT
```

- Member KMSOMTEC in RKANPARU

EIF configuration file for the default event destination

- Members in DSEVT

Dynamically generated EIF configuration files for alternate event destinations (EVTDS nnn , where nnn is the destination ID)

- Environment variables in KDSENV

```
KDS_RUN=KRANDREG.KRANDREG;KFAOMTEC.KFAOTMGR;KSMOMS.KSMOMS;KSH  
CMS.WEBSQL  
KMS_OMTEC_INTEGRATION=YES  
KMS_DISABLE_TEC_EMITTER=YES
```

- Event mapping and resource bundle files are stored in RKANDATV (KppMAP and KppRSI)

Configuring monitoring server to forward events (z/OS)

```
KDS62PP3 ----- SPECIFY ADVANCED CONFIGURATION VALUES -----
Enable Web Services SOAP Server => Y (Y, N)
Enable Tivoli Event Integration Facility (EIF) => N (Y, N)
Enable startup console messages => I (I, N)
Enable communications trace => N (Y, N, D, M, A)
Reconnect after TCP/IP recycle => N (Y, N)
Enable storage detail logging => Y (Y, N)
  Storage detail logging: Hours => 0 (0-24) Minutes => 60 (0-60)
  Flush VSAM buffers: Hours => 0 (0-24) Minutes => 30 (0-60)
Virtual IP Address (VIPA) type => N (S=Static, D=Dynamic, N=None)
Minimum extended storage => 768000 K
Maximum storage request size => 16 (Primary) => 23 (Extended)
Language Locale => 1           (Press F1=Help for a list of codes)

Persistent datastore parameters:
Maintenance procedure prefix => KPOPROC
Datastore file high-level prefix => rhilev.rte
Volume => Storclas =>
Unit => Mgmtclas =>

Enter=Next F1=Help F3=Back F5=Advanced F10=CMS List
```

From the Configure the TEMS ICAT panel

- Select Specify configuration values. Press F5 (Advanced)
- Set Enable Tivoli Event Integration Facility to Y (default is N)

© Copyright IBM Corporation 2013

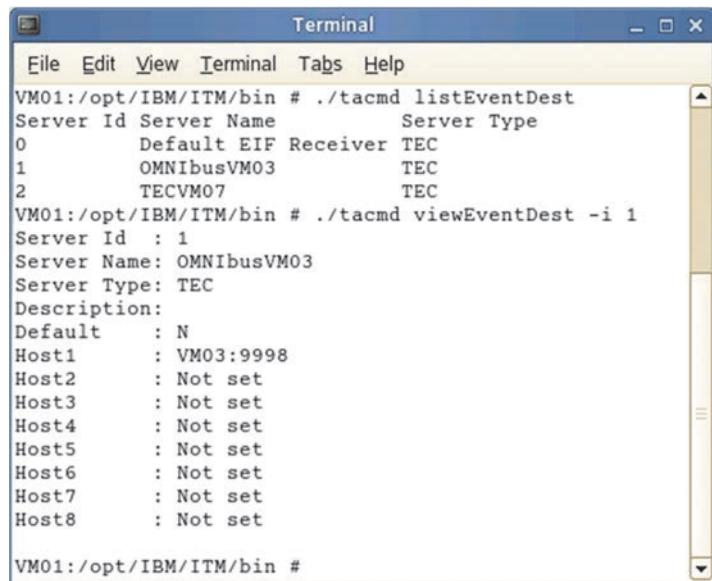
16

Configuring monitoring server to forward events (z/OS)

Event destination creation

Create the event destinations using the tacmd commands:

- tacmd createEventDest
- tacmd listEventDest
- tacmd viewEventDest
- tacmd deleteEventDest
- tacmd refreshTECinfo



The screenshot shows a terminal window titled "Terminal". The window contains the following command-line session:

```
VM01:/opt/IBM/ITM/bin # ./tacmd listEventDest
Server Id Server Name           Server Type
0      Default EIF Receiver    TEC
1      OMNIBusVM03             TEC
2      TECVM07                 TEC
VM01:/opt/IBM/ITM/bin # ./tacmd viewEventDest -i 1
Server Id : 1
Server Name: OMNIBusVM03
Server Type: TEC
Description:
Default   : N
Host1     : VM03:9998
Host2     : Not set
Host3     : Not set
Host4     : Not set
Host5     : Not set
Host6     : Not set
Host7     : Not set
Host8     : Not set
VM01:/opt/IBM/ITM/bin #
```

© Copyright IBM Corporation 2013

17

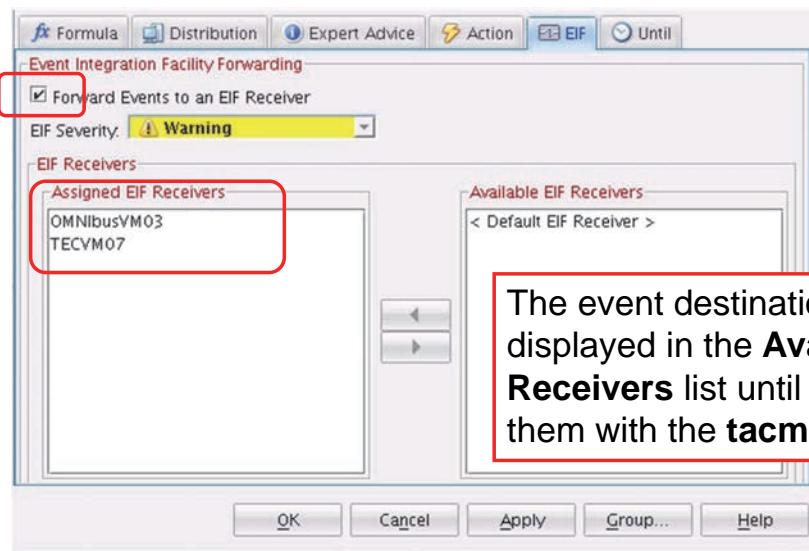
Event destination creation

You must create and manage event destinations from the command-line interface.

Use the **tacmd refreshTECinfo** command to trigger the Event Forwarder to reprocess any of these updated items without recycling the hub monitoring server:

- Event destinations
- EIF configurations
- Custom event mapping files

Editing the situation using the EIF notebook page



© Copyright IBM Corporation 2013

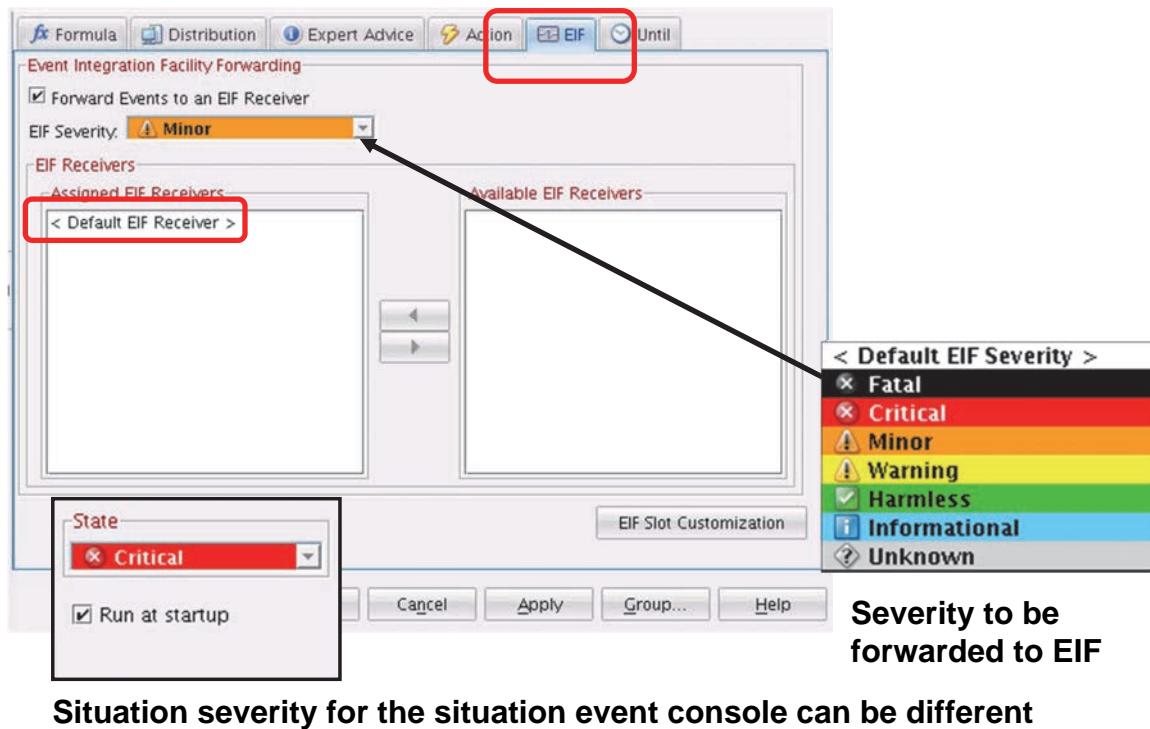
18

Editing the situation using the EIF notebook page

The event destinations created using the **tacmd createEventDest** command are displayed on the **Available EIF Receivers** tab.

Lesson 3. Situation event forwarding customization

Lesson 3: Situation event forwarding customization



© Copyright IBM Corporation 2013

19

What this lesson is about

This lesson demonstrates how to customize an event that is forwarded from Tivoli Monitoring to another event repositories.

What you should be able to do

After completing this lesson, you should be able to explain the steps to customize an event forwarded to another event repositories from Tivoli Monitoring.

Administration tasks include specifying which situations events are forwarded. You perform this action with the **EIF** tab in the Situation editor. You can send a different severity than what the situation has. If you have multiple event receivers, you can assign the situation to one or more of them.

Events forwarded to Netcool/OMNibus ObjectServer

The same event with a different status based on the EIF severity setting

Netcool/OMNibus Event List : Filter="All Events", View="Default"

Node	Alert Group	Summary	Last Occurrence	Count	Type
VM01:KUL	ITM_Log_Entries	Invalid_ssh_attempt{[STRSCAN(Description_U, N'Authentication failure for root	10/02/09 16:04:15	3	ITM Problem
VM03	Unix Event List	Netcool/OMNIbusEventConnector process running on has connected as username	10/02/09 15:41:13	1	Problem
VM03	Unix Conductor	A e@CDA86-403:0.0 process e@CDA86-403@CDA86-403:0.0 running on	10/02/09 09:35:17	1	Problem
VM03		mitrapd probe on VM03: Heartbeating to CDA86-403:0.0			
VM01:LZ	ITM_ManagedSystem	Managed_System_OFFLINE([Status='N']) CDA86-403:0.0			

Enterprise Status - VM01 - SYSLOG

Situation Event Console

Severity	Status	Owner	Situation Name	Display Item	Source	Impact
Critical	Open		Invalid_ssh_attempt		VM01:KUL	Monitored

© Copyright IBM Corporation 2013

20

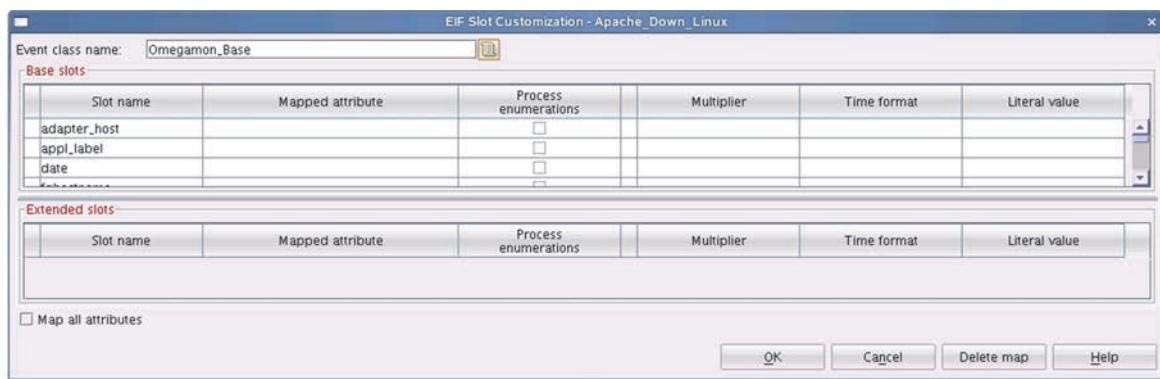
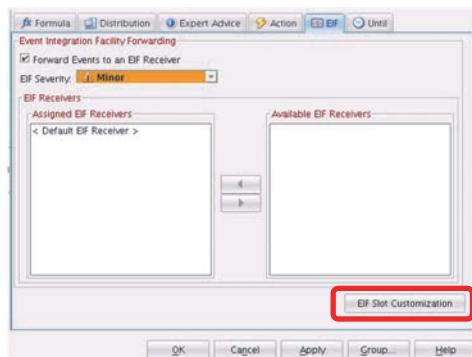
Events forwarded to Netcool/OMNibus ObjectServer

This example shows a situation event forwarded to the Netcool/OMNibus ObjectServer. The Netcool/OMNibus operator can manage these events in the same way as usual.

Notice that the event severity is Warning. This example illustrates that the administrator can send a different severity than what is defined on the situation itself (Critical).

EIF Slot Customization editor

- Use the EIF Slot Customization editor to customize how situation events are mapped to forwarded EIF events
- You map each EIF slot you want forwarded



© Copyright IBM Corporation 2013

21

EIF Slot Customization editor

You can modify the contents of the forwarded event from the EIF Slot Customization editor.

Event slot customization features

- Users can customize the generated event using the Event Map editor in the portal console
- Users can specify an optional user-defined event map per situation for these purposes:
 - Select the list of attributes to be included in the generated Tivoli Enterprise Console event
 - Customize the msg slot
 - When the Base Slot name is msg, the literal value is the message template
 - Assign custom Tivoli Enterprise Console event severity if defined
 - Override the slot values of certain fixed slots (for example, source or subsource)
- User-defined event map takes precedence over those defined in the mapping files in the TECLIB
- Changes to the user-defined event maps are dynamically picked up

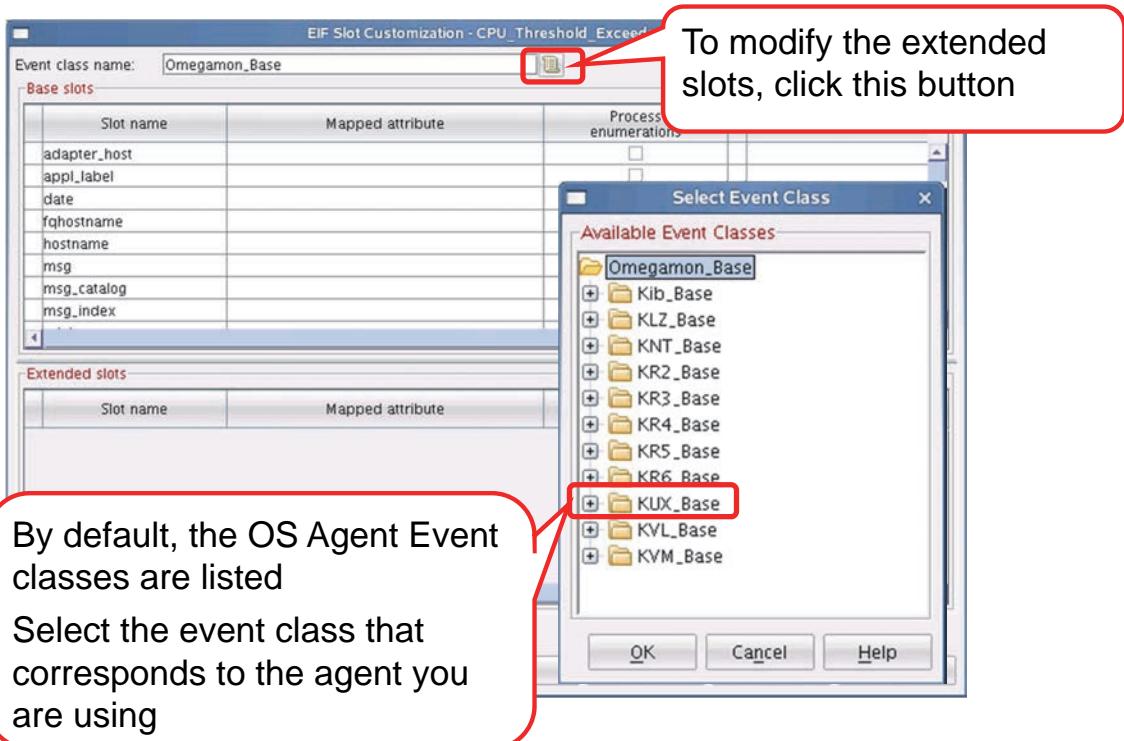
© Copyright IBM Corporation 2013

22

Event slot customization features

You can customize the EIF slot to control what the event looks like in the event repository where the event is forwarded.

EIF extended slot selection



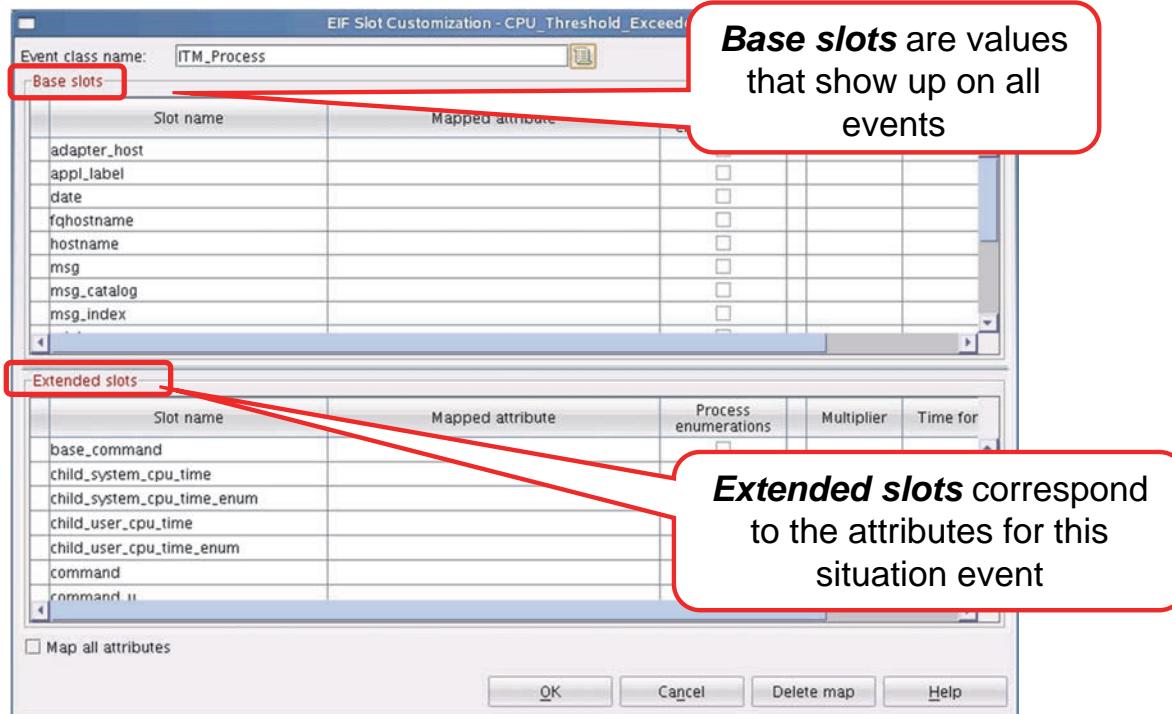
© Copyright IBM Corporation 2013

23

EIF extended slot selection

Select the extended slot that you want to customize so that you can map the attributes in the situation event to the EIF slots.

EIF extended slots and base slots



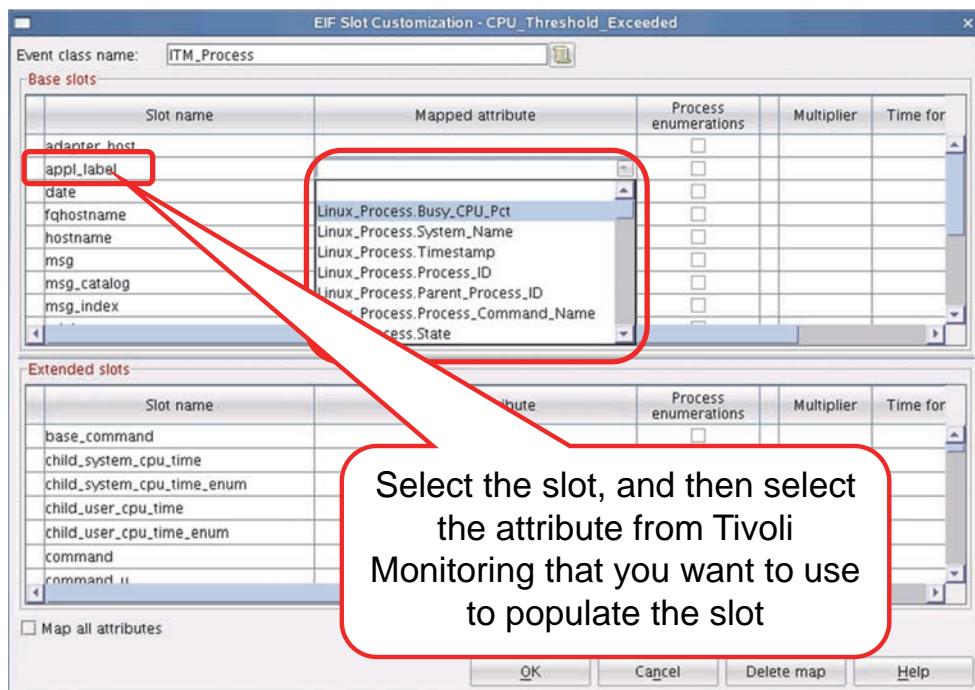
© Copyright IBM Corporation 2013

24

EIF extended slots and base slots

You can update both base and extended slots using the EIF Slot editor.

Modifying EIF slot values



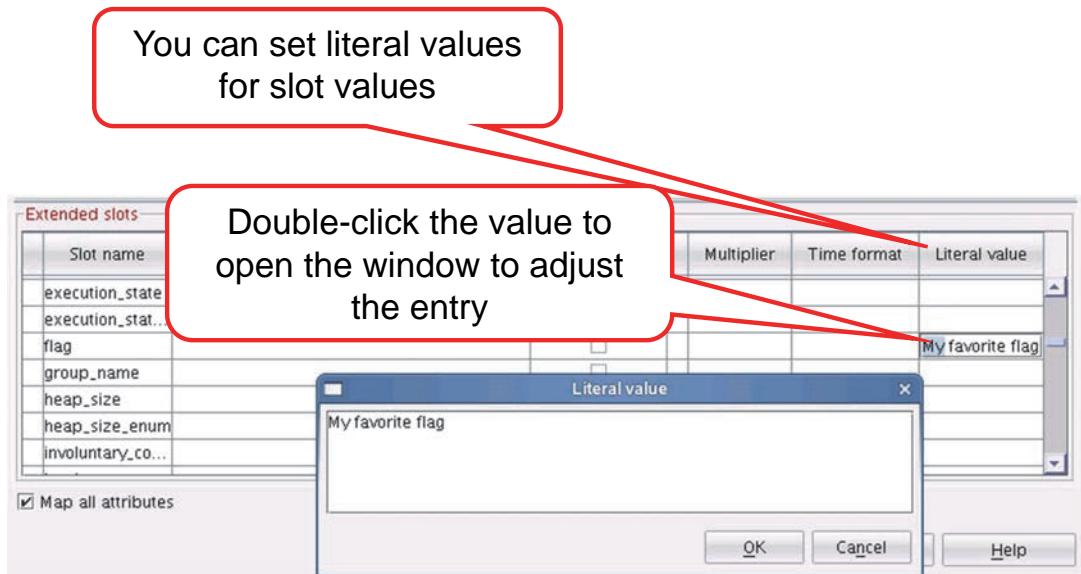
© Copyright IBM Corporation 2013

25

Modifying EIF slot values

Use the menu to assign a specific attribute to a specific slot name.

EIF slot Literal value column



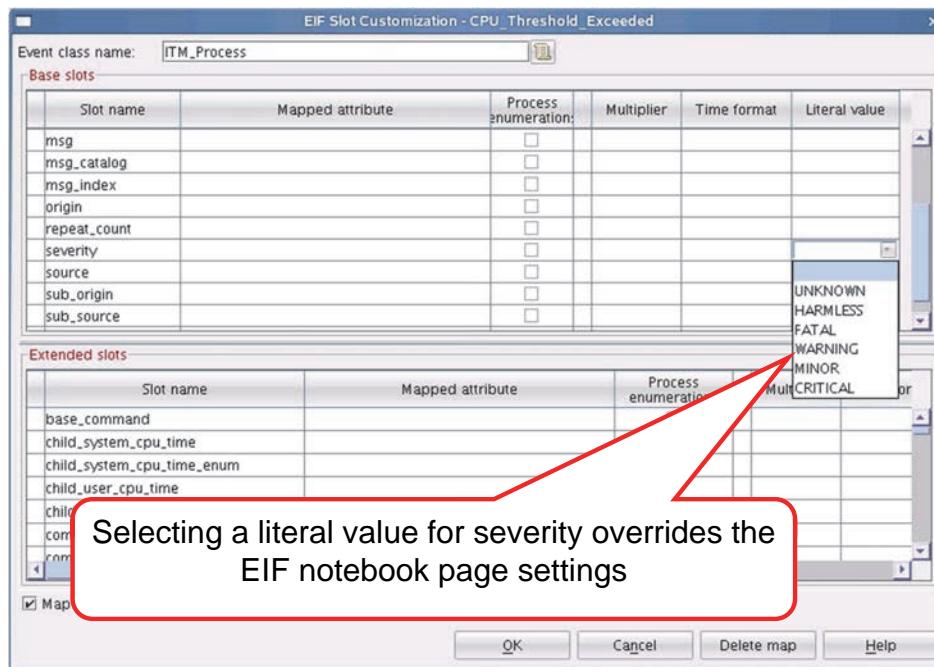
© Copyright IBM Corporation 2013

26

EIF slot Literal value column

You can control the severity of the message that is forwarded.

EIF slot Literal value column (continued)



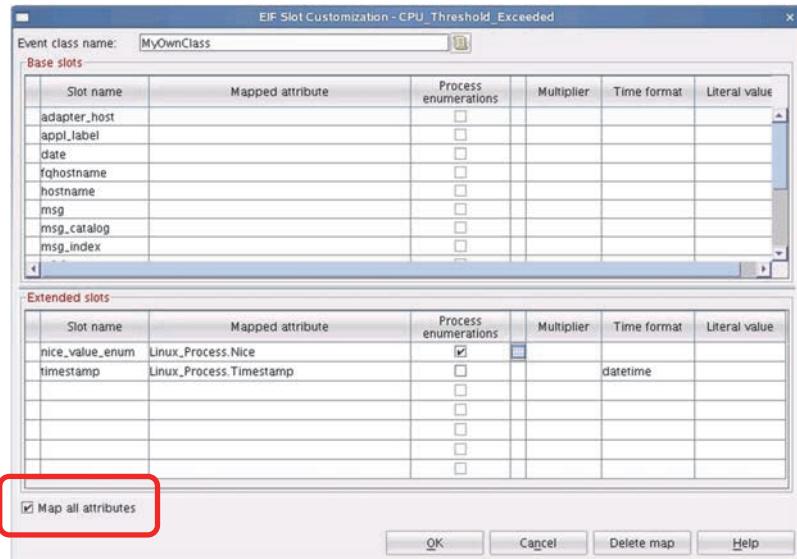
© Copyright IBM Corporation 2013

27

EIF slot Literal value column (continued)

Map all attributes

- If selected, generated event includes all customized slots plus all attributes in event data buffer (subject to event size limitation)
- Useful for cases where only a few slots must be changed



© Copyright IBM Corporation 2013

28

Map all attributes

Typically, you want to accept all the attribute mappings and then adjust only a few. The **Map all attributes** button simplifies that task.

Customizing the message slot

- Specify a customized message by creating a message template in the Literal value field of the msg slot
- Message template consists of fixed text and variable substitution references
- You can use variable substitution only in the message slot
Variable reference is indicated by the syntax \$variable\$
- Variable can reference event attribute, fixed slot value, or situation formula
 - Event attributes: Must be fully qualified, for example, \$AttrTable.AttrName\$
 - Fixed slots: Slots included in all Tivoli Monitoring generated events
Variable names that are not fully qualified are assumed to be a fixed slot reference, except if they are the situation formula references, for example, \$situation_status\$
 - Situation formula: Use \$formula\$ to reference the situation formula

© Copyright IBM Corporation 2013

29

Customizing the message slot

Often, the only column that needs customization is the message slot.

Customizing the message slot (continued)

- If variable reference cannot be resolved, a NULL string is used
- Example:

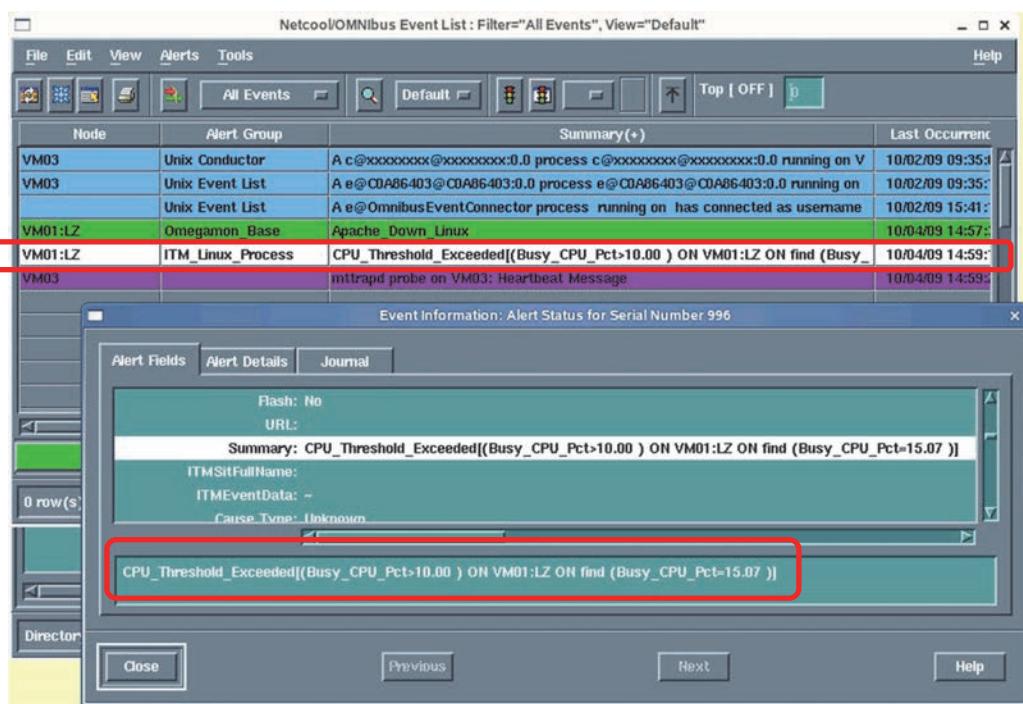
- Message slot template in map

CPU usage exceeded on \$hostname\$. Current CPU usage is
\$Linux_Process.Busy_CPU_Pct\$. Situation \$situation_name\$. (\$formula\$)

- Message slot text in event:

CPU usage exceeded on VM01. Current CPU usage is 11.98.
Situation CPU_Threshold_Exceeded. (|(Busy_CPU_Pct>10.00) ON VM01:LZ ON
java (Busy_CPU_Pct=11.98)|)

Customizing the message slot: Before

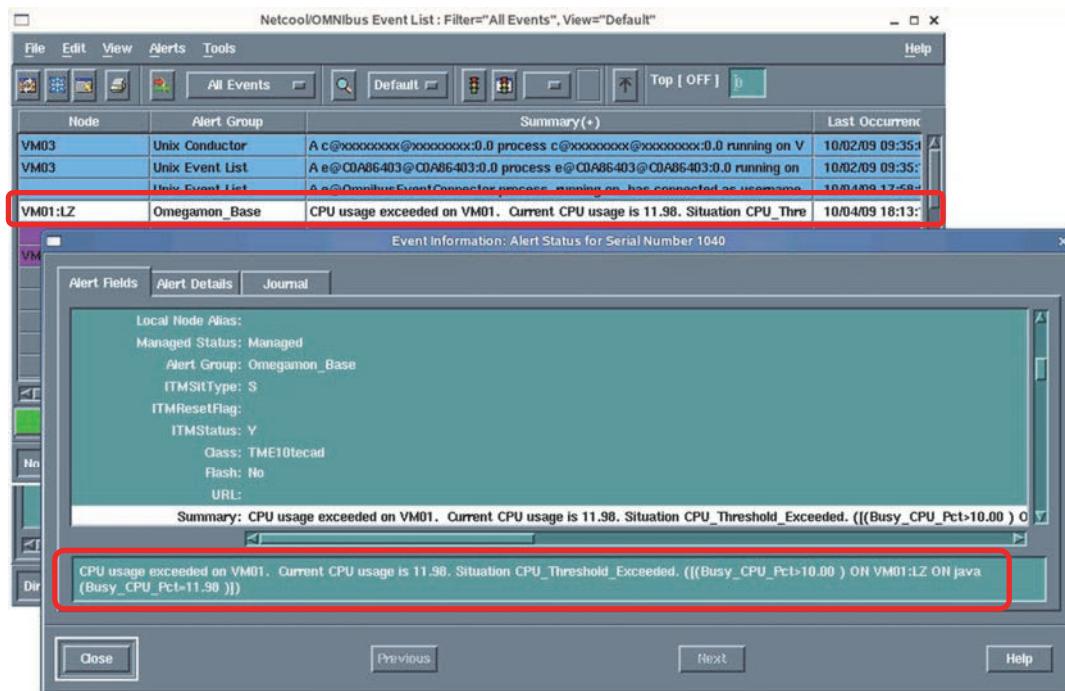


© Copyright IBM Corporation 2013

31

Customizing the message slot: Before

Customizing the message slot: After



© Copyright IBM Corporation 2013

32

Customizing the message slot: After

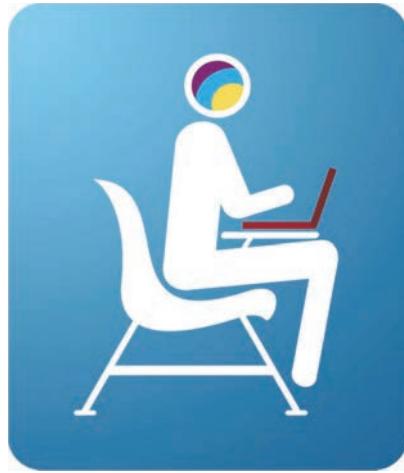
Slot customization window: Problem determination

- If there is any question about why a slot was saved as it was, this portal client trace setting is useful in diagnosing the issue:

ERROR (Unit:EIFSlot detail)

- This trace setting traces information about the definition read or written to the EVNTMAP table and other diagnostics about the control manipulation

Student exercise



© Copyright IBM Corporation 2013

34

Student exercises

Perform the exercises for this unit.

Review questions

1. Can you forward Tivoli Enterprise situation events and Netcool/OMNibus events to the situation event console?
2. What is the reason if the **EIF** tab does not show in the Situation editor?
3. What command do you use to create event destinations that show in the Situation editor on the **EIF** tab?

Review answers

1. Can you forward Tivoli Enterprise situation events and Netcool/OMNIbus events to the situation event console?

No. You can forward situation events to Netcool/OMNIbus, but you cannot forward Netcool/OMNIbus events to the situation event console.

2. What is the reason if the EIF tab does not show in the Situation editor?

*If the monitoring server has not been configured to enable the Tivoli Event Integration Facility, then the **EIF** tab is not displayed.*

3. What command do you use to create event destinations that show in the Situation editor on the EIF tab?

*You use the command **tacmd createEventDest** to create event destinations.*

Summary

Now that you have completed this unit, you can perform the following tasks:

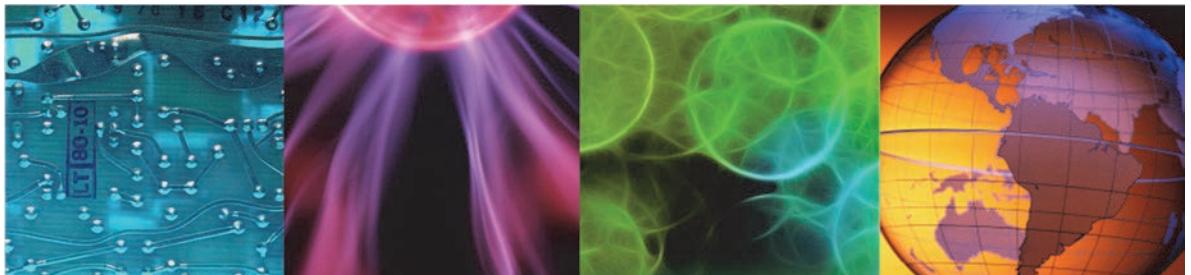
- Describe the ways in which IBM Tivoli Monitoring integrates with Netcool/OMNibus and the Tivoli Enterprise Console
- List the steps involved in forwarding situation events to Netcool/OMNibus
- Forward situation events to Netcool/OMNibus and the Tivoli Enterprise Console
- Customize slot values for situations sent to Netcool/OMNibus and the Tivoli Enterprise Console



9 Command-line interfaces



Command-line interfaces



All files and material for this course (TM063, IBM Tivoli Monitoring 6.3 Advanced Administration) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

What this unit is about

Many functions in IBM Tivoli Monitoring are performed using the Tivoli Enterprise Portal. The command-line interfaces can also be used. This unit introduces you to the commands.

How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

Objectives

When you complete this unit, you can perform the following tasks:

- Run the **tacmd** command to accomplish various tasks
- Issue the UNIX and Linux-only **itmcmd** command
- Manage the authorization policies that control access to resources shown in monitoring dashboards in the IBM Dashboard Application Services Hub
- Issue SOAP requests

Lesson 1. Running the tacmd command

Lesson 1: Running the tacmd command

- The primary command-line interface for IBM Tivoli Monitoring is **tacmd**
- A second command-line interface is **itmcmd**, which is available on UNIX and Linux systems
- Before using the **tacmd** command:
 - On UNIX and Linux systems, change to the default directory, which is usually **/opt/IBM/ITM/bin**
 - On Windows systems, change to the default directory, which is usually **C:\IBM\ITM\bin**
- The **tacmd** family of commands is now optional
See the *IBM Tivoli Monitoring Installation and Setup Guide* for more information about how to ensure that these commands are installed with your monitoring agent

© Copyright IBM Corporation 2013

3

What this lesson is about

This lesson gives a categorized overview of all the **tacmd** commands delivered with IBM Tivoli Monitoring.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the purpose of the **tacmd** command
- List some useful **tacmd** commands

References

IBM Tivoli Monitoring Installation and Setup Guide

There are two command-line interfaces with IBM Tivoli Monitoring. The **tacmd** command is used by Tivoli Monitoring and runs on any operating system. For UNIX and Linux systems the **itmcmd** command is used.

By using the command-line interface, you can embed commands in scripts. These commands perform functions more automatically than if you typed each command manually.

Basic commands

- To learn about available commands and usage:
 - **tacmd help** or **tacmd ?** shows all available **tacmd** commands
 - **tacmd help {command}** or **tacmd ? {command}** shows complete help for a specified command
- Commands require a session established with either the monitoring server or portal server
- For the monitoring server:
 - **tacmd login**: Authenticates a user name and password with a hub monitoring server so that a user can run subsequent commands from the local machine
 - **tacmd logout**: Log off from the hub monitoring server
- For the portal server:
 - **tacmd tepsLogin**: Log in to the portal server; some commands require a portal server log in before being issued
 - **tacmd tepsLogout**: Log off from the portal server

© Copyright IBM Corporation 2013

4

Basic commands

Some commands require access directly to the portal server; others require access to the hub monitoring server. There are two distinct login and logout commands for these purposes. On Linux systems, you typically change your directory to **/opt/IBM/ITM/bin**, the default location of these commands. On Windows systems, the default location is **C:\IBM\ITM\bin**.

Agent management commands

Occasionally, an agent might have to be stopped or started again. These commands are the equivalent of what you use with the Manage Tivoli Enterprise Monitoring Services when you stop or start agents.

Agent management commands

Use the following commands to start and stop agent components, including agents built by the Agent Builder and System Monitor agents

- **tacmd startAgent:** Starts a monitoring agent
- **tacmd stopAgent:** Stops a monitoring agent
- **tacmd restartAgent:** Restarts a monitoring agent
- **tacmd viewAgent:** Shows the details and status of a monitoring agent
- **tacmd updateAgent:** Updates a monitoring agent to a new version

Agent management commands

To use the commands to stop, start, or restart an agent, you must provide the name of a system and the type of agent.

Deployment commands

Depot management

You use these commands to access images from the installation media. After the images are available in the depot, you use these commands to access the images from the depot.

- **tacmd listBundles:** Shows details for component bundles not yet added to a deployment depot
You can run this command only from a monitoring server installation with a depot.
- **tacmd addBundles:** Adds component bundles to a deployment depot
You can run this command only from a monitoring server installation with a depot.
- **tacmd removeBundles:** Removes component bundles from a deployment depot
- **tacmd exportBundles:** Exports one or more deployment bundles to the specified export directory
- To show components that are loaded into the depot, use this command:
 - **tacmd viewDepot:** Shows the components that you can deploy remotely

Agent images are stored on the installation media. Sometimes the images are copied to other accessible locations on disk or on a network drive. These agent images are called **bundles** and are loaded into a depot for deployment.

Remote deployment and managing systems commands

- To remotely install an operating system agent:
 - **tacmd createNode:** Creates a node and starts an OS agent on it
 - **tacmd viewNode:** Shows the versions and patch levels of the systems that are installed on a node or a group of nodes
- To deploy and maintain application agents:
 - **tacmd addSystem:** Adds a new system to the server
 - **tacmd listSystems:** Shows a list of known systems
 - **tacmd configureSystem:** Updates the configuration of a system
 - **tacmd removeSystem:** Remove one or more instances of an agent or uninstall an agent from a managed system
Using the bulk deploy option, the command removes all agents in a deployment and bundle group combination
 - **tacmd cleanMS:** Deletes the entries for offline managed systems
 - **tacmd describeSystemType:** Shows the configuration options available for a system
 - **tacmd deleteApplInstallRecs:** Deletes application supports install records from the Tivoli Monitoring Server.
 - **tacmd listApplInstallRecs:** Shows the application install records.

© Copyright IBM Corporation 2013

7

Remote deployment and managing systems

During a Tivoli Monitoring implementation, you select hosts to be managed. The first step is typically to install the operating system agent. This step creates the node by defining the host name into the physical Navigator and the OS agent node. A typical second step is to install other agents or systems onto this same host.

Managing remote deployment

- To manage remote deployments, use the following commands:
 - **tacmd getDeployStatus**: Shows the status of the asynchronous agent deploy operations
 - **tacmd clearDeployStatus**: Remove entries from the table that stores the status of the asynchronous agent deploy operations
 - **tacmd restartFailedDeploy**: Restarts failed deployments
- To check for prerequisites on the target system before deploying an operating system monitoring agent, use the following command:
 - **tacmd checkPrereq**: Reviews prerequisites on target systems
You can review the results with the **tacmd getdeploystatus** command

Managing system list commands

- These commands work with lists, which are groups of managed systems
- By working with lists of systems, you can avoid repeating similar commands against each individual systems
 - **tacmd createSystemList:** Creates a system list on the server
 - **tacmd viewSystemList:** Shows or exports a system list definition
 - **tacmd editSystemList:** Edits a system list definition
 - **tacmd deleteSystemList:** Deletes a system list from the server
 - **tacmd listSystemList:** Shows a list of known system lists

© Copyright IBM Corporation 2013

9

Managing system lists commands

Grouping managed systems into system lists makes it easier to use a single command for an action against many systems. For example, if you have 20 Linux systems, you can add these systems into a single list. Then, you can make a single deployment operation using this list.

The **listSystemList** command by itself shows all the system lists that have been defined in this enterprise. To see the members of a system list, use the **listSystemList** command followed by **-l** and the name of a specific system list.

Self-describing agent commands

These commands work with self-describing agents

- **tacmd addSdaInstallOptions:*** Adds a version to the Self-Describing Agent (SDA) install option record
- **tacmd clearAppSeedState:** Clears the seed state of an application supports install records
- **tacmd deleteSdaInstallOptions:*** Removes a version from a Self-Describing Agent (SDA) install option record
- **tacmd deleteSdaOptions:** Deletes a configuration option of self describing agent
- **tacmd editSdaInstallOptions:*** Edits the version of a Self-Describing Agent (SDA) install option record
- **tacmd editSdaOptions:** Edits a configuration option of self describing agent
- **tacmd listSdaInstallOptions:*** List the Self-Describing Agent (SDA) install options records
- **tacmd listSdaOptions:** Lists the configuration options of self describing agent
- **tacmd listSdaStatus:*** Lists the Self-Describing Agent (SDA) Enablement information
- **tacmd resumeSda:*** Resumes the Self-Describing Agent (SDA) installation functions
- **tacmd suspendSda:*** Suspends the Self-Describing Agent (SDA) installation functions

* New with IBM Tivoli Monitoring 6.3

© Copyright IBM Corporation 2013

10

Self describing agent commands

These commands are used to enhance the self describing agent functions in IBM Tivoli Monitoring. Self describing agents are enabled with all the application support files ready to be sent to the monitoring server when they connect. The monitoring server must be configured to successfully transfer these files when the self describing agent connects to the monitoring server.

Working with situations and actions

- Enterprise situations send events to the monitoring server
- Situations can forward events to an Event Integration Facility receiver like Netcool/OMNIbus or Tivoli Enterprise Console if a hub monitoring server has been configured to forward events
- Situations can be managed with these commands:
 - **tacmd createSit**: Creates a situation on the server
 - **tacmd editSit**: Edits a situation definition
 - **tacmd deleteSit**: Deletes a situation from the server
 - **tacmd viewSit**: Shows or exports a situation definition
 - **tacmd listSit**: Lists known situations
 - **tacmd manageSit**: Starts or stops the specified situations in the monitoring server
 - **tacmd bulkexportSit**: All or specified types or specified situations are exported from the monitoring server
 - **tacmd bulkimportSit**: All or specified types or specified situation objects are imported to the monitoring server

© Copyright IBM Corporation 2013

11

Working with situations and actions commands

Situations provide the criteria for generating alerts to the situation event console. You can manage situations with these commands.

You can create many situations and use those situations with multiple Tivoli Monitoring environments. You can use the **bulkExportSit** and **bulkImportSit** commands to create a copy to load into another environment.

Situations and attributes override commands

- Monitoring agents are responsible for gathering data on various properties, or attributes, of managed systems
- These commands are available to manage overrides for eligible situations attributes
 - **tacmd listSitAttributes:** Shows a list of situation attribute names that are eligible for use with adaptive monitoring (override) commands for a specific situation
The command distinguishes between attributes that can be a part of a predicate expression and attributes that can be part of a condition expression
 - **tacmd setOverride:** Defines a situation override for a specified situation on a managed system or a list of managed systems
 - **tacmd listOverrides:** Shows the situation overrides defined for a specified situation on a managed system or list of managed systems
 - **tacmd listOverrideableSits:** Shows a list of override-eligible situations for an application
 - **tacmd deleteOverride:** Deletes the situation overrides defined for a specified situation on a managed system or list of managed systems

© Copyright IBM Corporation 2013

12

Situations and attributes override commands

Many situations can have override values. Use this feature to better customize a single situation for different threshold levels, based on selected conditions.

The output from the **listOverrideableSits** command shows each situation that can have overrides applied for a specified agent type.

Situation association commands

- **tacmd createSitAssociation:** Creates one or more situation associations for a Tivoli Enterprise Portal navigator item
- **tacmd deleteSitAssociation:** Dissociates one or more situations from a Tivoli Enterprise Portal navigator item
- **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 importSitAssociations:** Imports all situation associations from an XML file to the Tivoli Enterprise Portal Server
- **tacmd listSitAssociations:** Shows a list of managed systems or managed system lists that are assigned to a Tivoli Enterprise Portal navigator item

© Copyright IBM Corporation 2013

13

Situation association commands

Situations are associated with navigator items to designate where situation events should be displayed in the Navigator Physical view.

Take Action commands

- You can use the Take Action feature to invoke a predefined command or enter a command on a selected managed system
- You can also add action commands to situations to implement simple (reflex) automation when you open an event
- You can use both predefined and customized take action commands
- These commands are used to manage the take action feature:
 - **tacmd createAction:** Creates a new take action command definition on the monitoring server
 - **tacmd viewAction:** Shows a take action command definition
 - **tacmd editAction:** Edits a take action command definition on the monitoring server
 - **tacmd listAction:** Shows a list of take action commands
 - **tacmd executeAction:** Executes the take action command
 - **tacmd executeCommand:** Executes a command on a managed system
 - **tacmd deleteAction:** Deletes a take action command on the monitoring server

© Copyright IBM Corporation 2013

14

Take Action commands

Take Action commands are especially useful when a lengthy command is issued often. You can create a Take Action command once and issue it many times after, without the risk of making typographical errors. People who are not familiar with the command syntax can reference the command by a simple name.

Policy management commands

- A policy is a logical expression that describes a series, or workflow, of automated steps, called **activities**
- You can back up and restore your policies using these commands:
 - **tacmd bulkExportPcy**: All or specified types or specified policies are exported from the monitoring server
 - **tacmd bulkImportPcy**: All or specified types or specified policy objects are imported to the monitoring server

© Copyright IBM Corporation 2013

15

Policy management commands

You can create many policies for use with multiple Tivoli Monitoring environments. Use the **bulkExportPcy** and **bulkImportPcy** commands to create a copy to load into another environment.

Navigators and workspaces

You can back up and restore a custom Navigator view and all workspaces, queries, and situation associations

- **tacmd exportNavigator:** Exports a custom Navigator and all workspaces, queries, and situation associations referred within the custom Navigator to an XML file
- **tacmd importNavigator:** Imports a custom Navigator view, workspaces, queries, and situation associations from an XML file to the portal server
- **tacmd listNavigators:** Shows a list of custom Navigator views assigned to the specified user on the portal server
- **tacmd createSysAssignment:** Assigns one or more managed systems or managed system lists to a Tivoli Enterprise Portal navigator item
- **tacmd deleteSysAssignment:** Deletes one or more managed system assignments from a Tivoli Enterprise Portal navigator item
- **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 importSysAssignments:** Imports all managed system assignments from an XML file to the Tivoli Enterprise Portal Server on with a Tivoli Enterprise Portal navigator item
- **tacmd listSysAssignments:** Shows a list of all situations associated with or eligible for association with a Tivoli Enterprise Portal navigator item

Navigator commands

You can create many customized Navigators, workspaces, queries, and situations to use with multiple Tivoli Monitoring environments. Use the **exportNavigator** and **importNavigator** commands to create a copy to load into another environment.

Workspace management commands

- You can backup workspaces to an XML file and later restore them
- Use these commands to manage the workspaces:
 - **tacmd exportWorkspaces**: Exports one or more workspaces from the portal server to an XML file
 - **tacmd importWorkspaces**: Imports workspaces from an XML file to the portal server
 - **tacmd deleteWorkspace**: Deletes a global or user-customized Tivoli Enterprise Portal workspace from the Tivoli Enterprise Portal Server
 - **tacmd listWorkspaces**: Shows a list of workspaces on the portal server

© Copyright IBM Corporation 2013

17

Workspace management commands

You can create many workspaces to use with multiple Tivoli Monitoring environments. Use the **exportWorkspaces** and **importWorkspaces** commands to create a copy to load into another environment.

Query management commands

- You can back up queries to an XML file and later restore them
- Manage the queries with these commands:
 - **tacmd exportQueries**: Exports one or more portal queries from the portal server to an XML file
 - **tacmd importQueries**: Imports portal queries from an XML file to the portal server
 - **tacmd listQueries**: Shows a list of queries on the portal server

Query management commands

You can create many queries to use with multiple Tivoli Monitoring environments. Use the **exportQueries** and **importQueries** commands to create a copy to load into another environment.

Historical data commands

- You use history commands to configure attribute groups, work with data collections, attributes, and work with baseline values
- These are some of the basic history collection commands:
 - **tacmd histListProduct:** Lists all the products that are available for history configuration
 - **tacmd histConfigureGroups:** Configures the specified attribute groups for history collection
 - **tacmd histUnconfigureGroups:** Unconfigures the history configuration details of the specified attribute groups
 - **tacmd histListAttributeGroups:** Lists all the attribute group names for the specified product that are available for history configuration
 - **tacmd histViewAttributeGroup:** Shows the history configuration details of the specified attribute group

© Copyright IBM Corporation 2013

19

Historical data commands

You can collect historical data for predictive analysis of trends and for problem solving. Many history commands are available to work with the collection of attribute groups for systems.

Historical data collection commands

These commands pertain to managing the collection of historical data

- **tacmd histCreateCollection:** Creates the historical data collection of specified attribute groups
- **tacmd histEditCollection:** Edits the historical data collection of specified attribute groups
- **tacmd histDeleteCollection:** Deletes the historical data collection of specified attribute groups
- **tacmd histViewCollection:** Shows historical data collection configuration information for the specified collection
- **tacmd histListCollections:** Lists historical data collection for the specified managed system
- **tacmd histStartCollection:** Starts the historical data collection of specified attribute groups
- **tacmd histStopCollection:** Stops the historical data collection of specified attribute groups

History data is gathered through the use of named collections. These commands are all used with the management of the data collections.

Historical data baselines commands

- Baseline situation override values can be calculated based on historical data from the Tivoli Data Warehouse
- You use these commands with baselines:
 - **tacmd suggestBaseline**: Calculates 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 acceptBaseline**: Sets a situation overrides based on the baseline (situation override) values calculated 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

© Copyright IBM Corporation 2013

21

Historical data baselines commands

Using baselines, situation overrides can be suggested based on the performance behavior in an environment while examining historical data. A baseline can be suggested and accepted.

Calendar functions commands

- Functions have been provided to manage calendar entries on the monitoring server
- You can use calendar entries with collecting baseline historical data to be used with overrides
- These are the calendar entry commands:
 - **tacmd addCalendarEntry**: Adds a calendar entry to Tivoli Monitoring Services
 - **tacmd viewCalendarEntry**: Shows a calendar entry definition
 - **tacmd editCalendarEntry**: Modifies an existing calendar entry definition on the monitoring server
 - **tacmd deleteCalendarEntry**: Deletes a calendar entry from the monitoring server
 - **tacmd exportCalendarEntries**: Exports all the calendar entries available in the portal server to the specified XML file
 - **tacmd listCalendarEntries**: Shows calendar entry name, type, and data for each calendar entry definition on the monitoring server
 - **tacmd importCalendarEntries**: Imports all the calendar entries available in the specified XML file to the portal server

© Copyright IBM Corporation 2013

22

Calendar functions commands

Sometimes in an environment, you must collect historical data. Calendar functions can create calendar entries to establish these periods of time.

Administering users

- Commands are available to manage portal users in the portal server
- User administration authorization is needed to issue the user management commands
- These are commands to manage individual users:
 - **tacmd createUser**: Creates a new user in the portal server
 - **tacmd editUser**: Edits a specified user definition
 - **tacmd deleteUser**: Deletes a specified user from the portal server
 - **tacmd listUsers**: Lists all the users or the users belonging to a particular group
 - **tacmd viewUser**: Shows the details of a specified user

© Copyright IBM Corporation 2013

23

Administering users with commands

The commands in the slide are used with created portal operators or users. These users can also become members of user groups.

Administering users by user group commands

- To simplify user management, you can use groups that contain one or more users or groups
- Do not confuse these commands with other group commands presented in this unit that are called *user group commands*
- You manage the user groups with these commands:
 - **tacmd createUserGroup:** Creates a new user group in the portal server
 - **tacmd editUserGroup:** Edits a specified user group definition
 - **tacmd viewUserGroup:** Shows the details of a specified user group
 - **tacmd deleteUserGroup:** Deletes a specified user group from the portal server
 - **tacmd listUserGroups:** Lists all existing user groups

Group management

- Groups are used to support deployment, bundles, situations, and data collection
- Do not confuse these commands with user group commands presented in this unit that are called *group commands*
- These are commands used with managing groups:
 - tacmd createGroup**: Creates a new deployment, bundle, or situation group definitions in the agent deployment tables on the monitoring server
 - tacmd deleteGroup**: Deletes a group definition and all group members belonging to the group from the deployment tables on the monitoring server
 - tacmd listGroups**: Shows the name and bundle type for each group definition on the server
 - tacmd editGroup**: Modifies an existing deployment, bundle, or situation group definition in the agent deployment tables on the monitoring server
 - tacmd viewGroup**: Shows the details of a group definition

© Copyright IBM Corporation 2013

25

Group management commands

By using groups, you can simplify distributing data in an environment. Rather than performing an action repeatedly against different systems, the systems can become members of a group. You can invoke the distribution for a single time, and data is directed to all members of the group.

Group member commands

You use these commands to manage members of groups:

- **tacmd addGroupMember**: Adds a group member to the specified group
- **tacmd editGroupMember**: Modifies or defines an alias for an existing situation group member, modifies the properties for a deploy group, or modifies the type, version, platform, or properties for a bundle definition
- **tacmd deleteGroupMember**: Deletes a group member definition from the group
- **tacmd viewGroupMember**: Shows the properties of a specific group member, depending upon the group type of the group member

When groups are created, you can add or remove members from the groups.

Event management

- Event destinations are event management servers that are intended to receive alerts
- These commands are used to manage event destinations
 - **tacmd createEventDest:** Creates a new event destination definition on the monitoring server
 - **tacmd deleteEventDest:** Deletes an event destination server definition from the monitoring server
 - **tacmd editEventDest:** Modifies an existing event destination server definition on the monitoring server
 - **tacmd listEventDest:** Shows the server ID, server name, and server type for every event destination server definition on the server
 - **tacmd viewEventDest:** Shows all properties for the specified event destination definition on the monitoring server

Event management commands

Event servers can display events or alerts from environments. The servers can be defined as event destinations. You managed them with the commands listed in the slide. For the Tivoli Enterprise Console, there is also a command to refresh the Tivoli Enterprise Console configuration files.

Refreshing Tivoli Enterprise Console and Netcool/OMNIbus information commands

- Tivoli Enterprise Console and Netcool/OMNIbus are event server products
- This command is used to refresh Tivoli Enterprise Console and Netcool/OMNIbus configuration and mapping files:
tacmd refreshTECinfo: Refreshes Tivoli Enterprise Console and Netcool/OMNIbus Event Integration Facility configuration and event mapping files

Miscellaneous commands

- Catalog and attribute (cat and atr) files are required to present workspaces, online help, and expert advice for the agent in Tivoli Enterprise Portal.
tacmd refreshCatalog: Refreshes the catalog file
 - This action allows the data server to reread the catalog files and refresh the affinity information
 - It eliminates the need for a Tivoli Enterprise Monitoring Server recycle after support files are installed
- Use this command to configure a user-defined portal server data source:
tacmd configurePortalServer: Configures a portal server data source
 - This command can only be run from a portal server installation
- The PDCollect tool collects the most commonly used information from a system so that IBM service can investigate a problem:
 - It gathers log files, configuration information, and version information
 - Use this command to invoke PDCollect:
tacmd pdcollect: Remotely invoke the PDCollect tool and transfer the compressed file to the local machine

© Copyright IBM Corporation 2013

29

Miscellaneous commands

This slide lists several miscellaneous commands. **PDCollect** is a problem-solving tool. It gathers many sources of information during a problem to avoid making several requests to obtain additional information.

Miscellaneous commands (continued)

- **tacmd getfile**: Transfers a file from a remote managed system to a local path.
- **tacmd putfile**: Transfers a file from a local path to a remote managed system.
- **tacmd listtrace**:* Shows the current RAS1 tracing level on a managed system.
- **tacmd settrace**:* Changes the RAS1 tracing level on a managed system to the specified new value.
- **tacmd setagentconnection**: Updates the agent connection properties and environment variables for the agent.

* New with IBM Tivoli Monitoring 6.3

Lesson 2. Issuing the **itmcmd** commands

Lesson 2: Issuing the **itmcmd** commands

Nearly all **itmcmd** commands are for UNIX and Linux servers. The only exception is the **itmcmd** command for the system monitoring agent.

- **itmcmd agent:** Stops or starts a monitoring agent
- **itmcmd config:** Configures the execution environment for an agent or server
- **itmcmd manage:** Starts the Manage Tivoli Enterprise Monitoring Services application
- **itmcmd server:** Stops or starts the monitoring server
- **itmcmd support:** Configures the monitoring server for a monitoring product

What this lesson is about

This lesson gives a categorized overview of all the **itmcmd** commands delivered with IBM Tivoli Monitoring.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the purpose of the **itmcmd** command
- List some useful **itmcmd** commands

These commands are available to Linux and UNIX environments.

More **itmcmd** commands

- **itmcmd audit:** Manages the size and number of log files in **/opt/IBM/ITM/logs/**
- **itmcmd dbagent:** Stops or starts a distributed database monitoring agent (except DB2)
- **itmcmd dbconfig:** Configures the execution environment for a distributed database agent (except DB2)
- **itmcmd history:** Manages the roll off of history data into delimited text files
- **itmcmd execute:** Runs a user script or command in an IBM Tivoli Monitoring product environment
- **itmcmd resp:** Generates silent response files

You use the **itmcmd** commands to manage servers and agents in Linux and UNIX systems. These commands are not used in other environments. However, there are typically variations of **tacmd** commands that you can use for non-UNIX and non-Linux environments.

Lesson 3. Managing authorization policies

Lesson 3: Managing authorization policies

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

- **tivcmd addtorole:** * To add users or groups to a role
 - **tivcmd copyrole:** * To copy an existing role to a new role
 - **tivcmd createrole:** * To create a new role
 - **tivcmd deleterole:** * To delete a role
 - **tivcmd exclude:** * To exclude a permission from a role
 - **tivcmd grant:** * Assign a permission to a role
- * New with IBM Tivoli Monitoring 6.3

What this lesson is about

This lesson gives a categorized overview of all the tivcmd commands delivered with IBM Tivoli Monitoring.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the purpose of the tivcmd command
- List some useful tivcmd commands

More tivcmd commands

- **tivcmd listdomains:** * To list all the domains in the Authorization Policy Server
 - **tivcmd listobjecttypes:** * To list valid object types and operations
 - **tivcmd listresourcetypes:** * To list valid resource types
 - **tivcmd listroles:** * To list roles
 - **tivcmd login:** * Authenticates a user name and password with a server so that you can run subsequent commands from the local machine
 - **tivcmd logout:** * Logs you off the server
 - **tivcmd removefromrole:** * To remove users or groups from a role
 - **tivcmd revoke:** * Use the revoke command to remove the permission that was (1) previously granted to a role using the grant CLI, or (2) previously denied from a role using the deny CLI
- * New with IBM Tivoli Monitoring 6.3

Lesson 4. Simple Object Access Protocol (SOAP)

Lesson 4: Simple Object Access Protocol (SOAP)

- Platform independent
- Uses XML and HTTP for requests and replies
- Defined using XML standards
- IBM provides many SOAP methods with IBM Tivoli Monitoring web services
- Dynamically query and control IBM Tivoli Monitoring environments with SOAP methods
- SOAP is only available on the hub monitoring server
- **tacmd** uses SOAP
- The SOAP client command-line utility, **kshsoap**, is an http client

What this lesson is about

This lesson covers the implementation of SOAP commands delivered with IBM Tivoli Monitoring. IBM Tivoli Monitoring has functions that use SOAP. A command line is available now to invoke a SOAP request; it is explained in this lesson.

What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the purpose of the **kshsoap** command
- Run various SOAP methods to retrieve data from IBM Tivoli Monitoring

Uses of SOAP methods in Tivoli Monitoring

- Stop or start policies and situations
- Manage and display messages on a Universal Message console
- Retrieve performance attribute data that you can show in charts or reports
- Open and close events
- Make real-time requests for data
- Issue SOAP requests as system commands in the portal
- Forward trapped messages from System Automation for Integrated Operations

Uses of SOAP methods in Tivoli Monitoring

The SOAP methods are useful in IBM Tivoli Monitoring. Many functions that are available in the portal can be achieved using SOAP.

List of SOAP methods in Tivoli Monitoring

- CT_Acknowledge: Sends an event acknowledgement into the IBM Tivoli Monitoring platform
- CT_Activate: Starts a situation or a policy running on the IBM Tivoli Monitoring platform
- CT_Alert: Sends an event into the IBM Tivoli Monitoring platform
- CT_Deactivate: Stops a situation or policy on the IBM Tivoli Monitoring platform
- CT_Email: Sends the output from another CT SOAP method, such as CT_Get, using email through an SMTP server to a defined email address (not available on z/OS)
- CT_Execute: Runs the SOAP request that is stored in a file
- CT_Export: Sends the output from another CT SOAP method, such as CT_Get, to a defined file (not available on z/OS)
- CT_Get: Receives a group of XML objects or individual XML objects from any IBM Tivoli Monitoring platform agent; you can use this to obtain real time data
- CT_Redirect: Reroutes a SOAP request to another registered SOAP method outside of the domain of the IBM Tivoli Monitoring platform
- CT_Reset: Sends an event reset (close event) to the IBM Tivoli Monitoring platform
- CT_Resurface: Resurfaces an acknowledged event in the IBM Tivoli Monitoring platform
- CT_WTO: Sends a Universal Message into the IBM Tivoli Monitoring Platform

© Copyright IBM Corporation 2013

37

List of SOAP methods in Tivoli Monitoring

This slide shows the many SOAP methods used by Tivoli Monitoring.

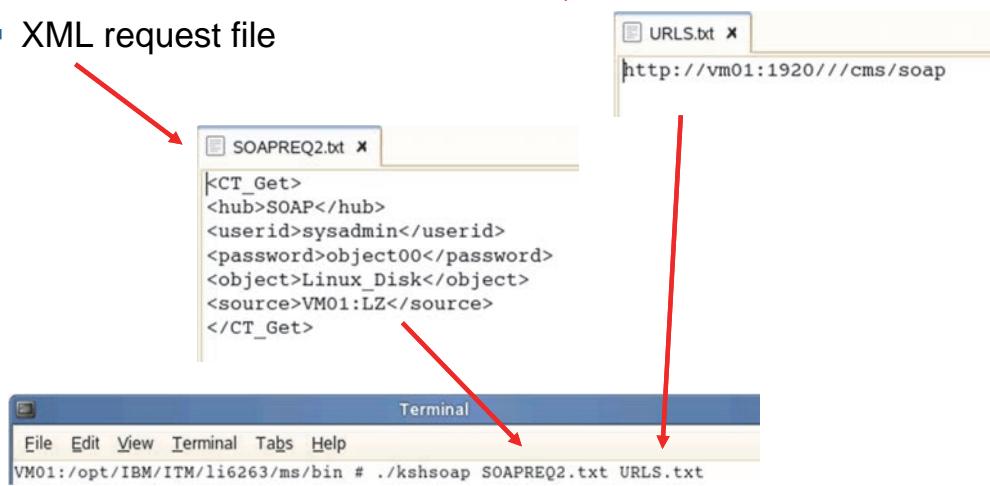
There are two ways to issue SOAP requests:

- Directly through a browser
- By using a command-line interface

The CLI method is presented in this lesson.

SOAP client command-line utility kshsoap

- This CT_Get example illustrates retrieving rows of Linux disk usage information using the SOAP CLI
- **kshsoap** issues a direct SOAP request using two text files:
 - URLs file
 - XML request file



© Copyright IBM Corporation 2013

38

SOAP client command-line utility kshsoap

The command-line interface for SOAP is **kshsoap**. An XML file specifies which method is being used and contains values for many keywords. A second file can contain the URLs of the SOAP servers. The **kshsoap** command is issued with operands that specify the names of these two files.

Command results

The SOAP request is displayed in the terminal window. Only one row is displayed here as an example.

Mount Point	Disk Name	Size (MB)	Disk Used (MB)	Disk Free (MB)	Tc Ino
/	/dev/sda2	39784	5221	34563	0
/proc	proc	0	0	0	0
/sys	sysfs	0	0	0	0
/sys/kernel	reiserfs	0	0	0	0

```

<ROW>
<System_Name>VM01:LZ</System_Name>
<Timestamp>1091029150954000</Timestamp>
<Disk_Name>/dev/sda2</Disk_Name>
<Mount_Point>/</Mount_Point>
<Size dt="number">39784</Size>
<Space_Used dt="number">5221</Space_Used>
<Space_Available dt="number">34563</Space_Available>
<Total_Inodes dt="number">0</Total_Inodes>
<Inodes_Used dt="number">0</Inodes_Used>
<Inodes_Free dt="number">0</Inodes_Free>
<Space_Used_Percent dt="number">13</Space_Used_Percent>
<Inodes_Used_Percent dt="number">0</Inodes_Used_Percent>
<FS_Type>reiserfs</FS_Type>
<Space_Available_Percent dt="number">87</Space_Available_Percent>
<Mount_Point_U>/</Mount_Point_U>
<Inodes_Available_Percent dt="number">100</Inodes_Available_Percent>
</ROW>

```

The row of information matches the top row from the portal Disk Usage view.

© Copyright IBM Corporation 2013

39

Command results

A successful **kshsoap** command shows messages in the terminal window. In this example, the event has been successfully written to the situation event console.

Student exercise



© Copyright IBM Corporation 2013

40

Student exercises

Perform the exercises for this unit.

Review questions

1. Which command do you use to show details for component bundles that are not yet added to a deployment depot?
2. What is accomplished by using the **tacmd exportNavigator** command?
3. Which command is used to start the historical data collection of a specified attribute group?
4. Which two commands do you use to calculate and set baseline situation override values based on historical data from the Tivoli Data Warehouse?

Review answers

1. Which command is used to display details for component bundles not yet added to a deployment depot?

tacmd listBundles

2. What is accomplished by using the **tacmd exportNavigator** command?

A custom Navigator and all its workspaces, queries, and situation associations that are referenced within it are exported to an XML file.

3. Which command is used to start the historical data collection of a specified attribute group?

tacmd histStartCollection

4. Which two commands do you use to calculate and set baseline situation override values based on historical data from the Tivoli Data Warehouse?

tacmd suggestBaseline and tacmd acceptBaseline

Summary

Now that you have completed this unit, you can perform the following tasks:

- Run the **tacmd** command to accomplish various tasks
- Issue the UNIX and Linux-only **itmcmd** command
- Manage the authorization policies that control access to resources displayed in monitoring dashboards in the IBM Dashboard Application Services
- Issue SOAP requests

More about Cloud & Smarter Infrastructure

You can find the latest information about IBM Cloud & Smarter Infrastructure education offerings online at the following location:

www.ibm.com/software/tivoli/education/

Also, if you have any questions about education offerings, send an email to the appropriate alias for your region:

- Americas: tivamedu@us.ibm.com
- Asia Pacific: tivtrainingap@au1.ibm.com
- EMEA: tived@uk.ibm.com

Cloud & Smarter Infrastructure user groups

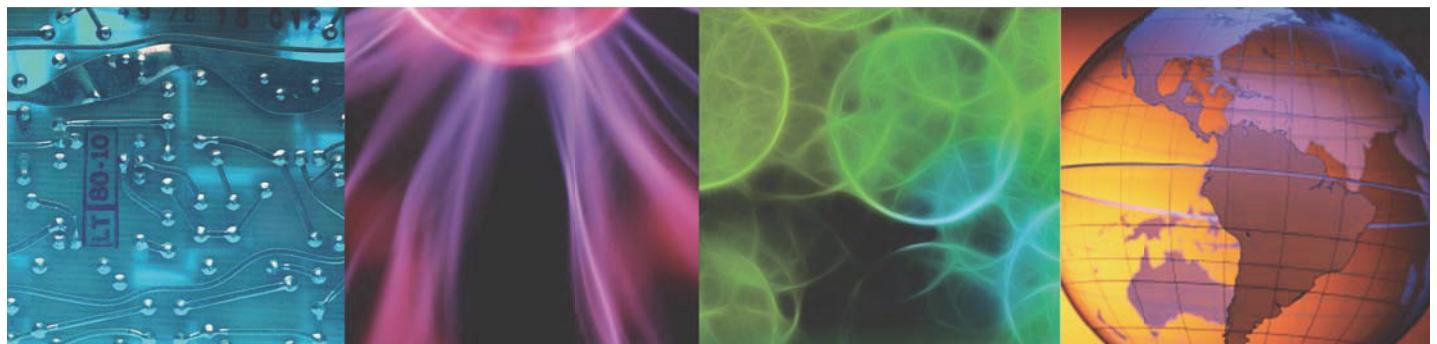
You can get even more out of Cloud & Smarter Infrastructure software by participating in one of the 91 independently run Cloud & Smarter Infrastructure user groups around the world. Learn about online and in-person user group opportunities near you at www.tivoli-ug.org.

Certification

All IBM certifications are based on job roles. They focus on a job a person must do with a product, not just the product's features and functions. Online certification paths are available to guide you through the process for achieving certification in many IBM Cloud & Smarter Infrastructure areas. See ibm.com/tivoli/education for more information about certification.

Special offer for having taken this course: *Now through 31 December 2013:* For completing this course, you are entitled to a 15% discount on your next examination at any Thomson Prometric testing center worldwide. Use this special promotion code when registering online or by telephone to receive the discount: **15CSWR**. (This offer might be withdrawn. Check with the testing center.)

IBM[®]



ibm.com/training

Authorized
IBM | Training



Printed in Ireland