

Course Exercises

Agent Builder 6.3.4 Labs for an IBM Tivoli Monitoring Environment

Course code TV384 ERC 1.0

Click for [Agent Builder 6.3.4 Labs for an IBM Application Performance Management Environment](#)



June 2017 edition

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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.

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.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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.

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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2017.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

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

Contents

Unit 1 Introduction to IBM Agent Builder exercises.....	1-1
Unit 2 Agent creation basics exercises.....	2-1
Exercise 1 Create an agent to monitor specific Windows services	2-1
Create the agent	2-1
Browse Windows services	2-6
Troubleshoot an agent definition problem	2-13
Troubleshoot Agent Builder	2-15
Unit 3 Customizing agents for IBM Tivoli Monitoring exercises.....	3-1
Exercise 1 Install the AB1 agent in an IBM Tivoli Monitoring environment	3-1
Disable IBM Performance Management dashboards	3-2
Install the agent and create the installation scripts	3-3
Install the AB1 agent application support on ITM	3-7
Troubleshoot an Agent Builder agent installation	3-8
Configure, start, and confirm your agent on WIN1	3-9
Exercise 2 Create and import IBM Tivoli Monitoring application support	3-12
Create and tag your custom workspace for export to Agent Builder	3-12
Create a custom situation	3-22
Import your custom workspace and situation into your agent	3-24
Exercise 3 Install and confirm IBM Tivoli Monitoring application support	3-27
Remove your custom workspace and situation	3-27
Create the agent installation scripts	3-30
Reinstall the AB1 application support on ITM	3-30
Reinstall the agent on WIN1 with the script installers	3-31
Confirm the installation of agent and application support components	3-32
Unit 4 Monitoring Windows resources exercises.....	4-1
Exercise 1 Monitor Windows resources	4-1
Prepare to edit your agent	4-1
Add WMI process information to the agent	4-2
Add combined disk information to the agent	4-10
Add DB2 Windows log event information to the agent	4-18
Test the full agent in Agent Builder	4-22
Change the agent version	4-30
Exercise 2 Install and confirm the updated AB1 agent in an IBM Tivoli Monitoring environment	4-31
Create the agent installation scripts	4-31
Reinstall the AB1 application support on ITM	4-32
Reinstall the agent on WIN1 with the script installers	4-33

Confirm the installation of agent and application support components in an IBM Tivoli Monitoring environment	4-33
Unit 5 Monitoring processes and command return codes exercises.....	5-1
Exercise 1 Monitor processes and command return codes	5-1
Create an agent to monitor a process on Windows and Linux	5-2
Create the HTTP server monitor for Windows	5-3
Add the Linux HTTP server process	5-6
Create an agent to monitor a functionality test	5-9
Exercise 2 Install and confirm the AB2 agent in an IBM Tivoli Monitoring environment	5-19
Disable IBM Performance Management dashboards	5-19
Create the agent installation scripts	5-20
Install the AB2 application support on ITM	5-20
Install the AB2 agent on WIN1	5-21
Install the AB2 agent on LIN4	5-22
Confirm the data display in the TEP client	5-24
Unit 6 Monitoring custom data sources exercises.....	6-1
Exercise 1 Create an agent to monitor local and remote systems with scripts and socket connections	6-1
Confirm the script	6-2
Modify AB2 to monitor the script	6-3
Add derived attributes	6-13
Enable the agent to monitor local and remote sockets	6-17
Change the agent version	6-25
Exercise 2 Install and confirm the updated AB2 agent in an IBM Tivoli Monitoring environment	6-26
Create the updated AB2 agent installation scripts	6-26
Reinstall the AB2 application support on ITM	6-26
Reinstall the AB2 agent on WIN1	6-27
Reinstall the AB2 agent on LIN4	6-29
Confirming the script data source on the LIN4 and WIN1 agents	6-32
Confirm the socket data source on the LIN4 and WIN1 agents	6-35
Exercise 3 Create an agent to monitor a log file and a custom Java data source	6-38
Determine the log file data for the agent to gather	6-38
Modify the AB1 agent to monitor the ping.txt log file	6-40
Add derived attributes	6-54
Add a Java API data source	6-60
Change the agent version	6-71
Exercise 4 Install and confirm the updated AB1 agent in an IBM Tivoli Monitoring environment	6-72
Create the updated AB1 agent installation scripts	6-72
Install the updated AB1 application support on ITM	6-72
Reinstall and configure the updated AB1 agent on WIN1	6-72
Confirm the log file data source	6-74
Confirm the Java API data source	6-75
Unit 7 Monitoring remote and optional resources exercises	7-1
Exercise 1 Remotely monitor many resources	7-1
Create an agent to monitor multiple hosts from one agent	7-1

Add a data source to monitor HTTP pages	7-10
Add a data source to monitor network availability	7-13
Change the agent version	7-16
Exercise 2 Install and confirm the updated AB1 agent in an IBM Tivoli Monitoring environment	7-18
Create the updated AB1 agent installation scripts	7-18
Install the updated AB1 application support on ITM	7-18
Install, configure, and start the updated AB1 agent	7-18
Exercise 3 Allow data sources to be optional	7-27
Add monitoring of Common Information Management data sources	7-27
Add monitoring of remote WebSphere Application Server Java Management Extensions	7-36
Add a JDBC data source to the agent	7-47
Change the agent version	7-56
Exercise 4 Install and confirm the updated AB2 agent in an IBM Tivoli Monitoring environment	7-57
Create the updated AB2 agent installation scripts	7-57
Install the updated AB2 application support on ITM	7-57
Reinstall and configure the updated AB2 agent on WIN1	7-58
Reinstall and configure the AB2 agent on LIN4	7-64

Unit 1 Introduction to IBM Agent Builder exercises

This unit has no student exercises.

Unit 2 Agent creation basics exercises

In this unit, you create an agent that is named AB1. It monitors the HTTP server and DB2® services of any target Windows host.

Exercise 1 Create an agent to monitor specific Windows services

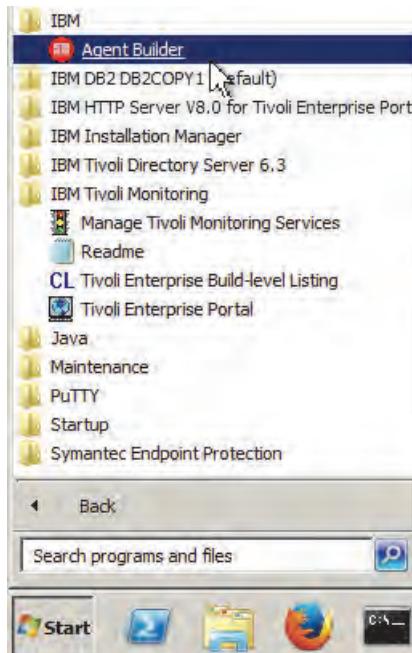
In this exercise, you create a monitoring agent that gathers availability information for Windows DB2 and HTTP services. You must complete this exercise from the WIN1 server.

Create the agent

1. Log in to WIN1 as **Administrator** with password **object00**.
2. Start Agent Builder by using either of the following methods:
 - Double-click the **Agent Builder** icon on the desktop.



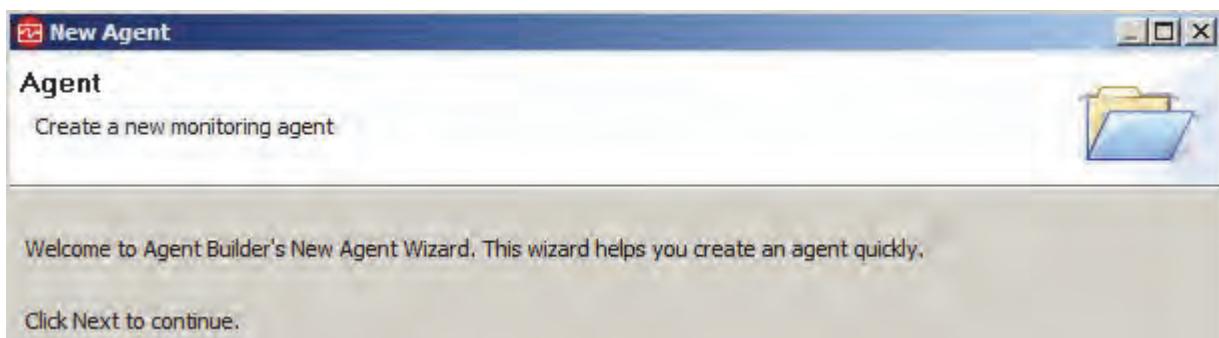
- Select **Start > All Programs > IBM > Agent Builder.**



3. Maximize the Agent Builder application window.
4. Select **File > New > Agent.**

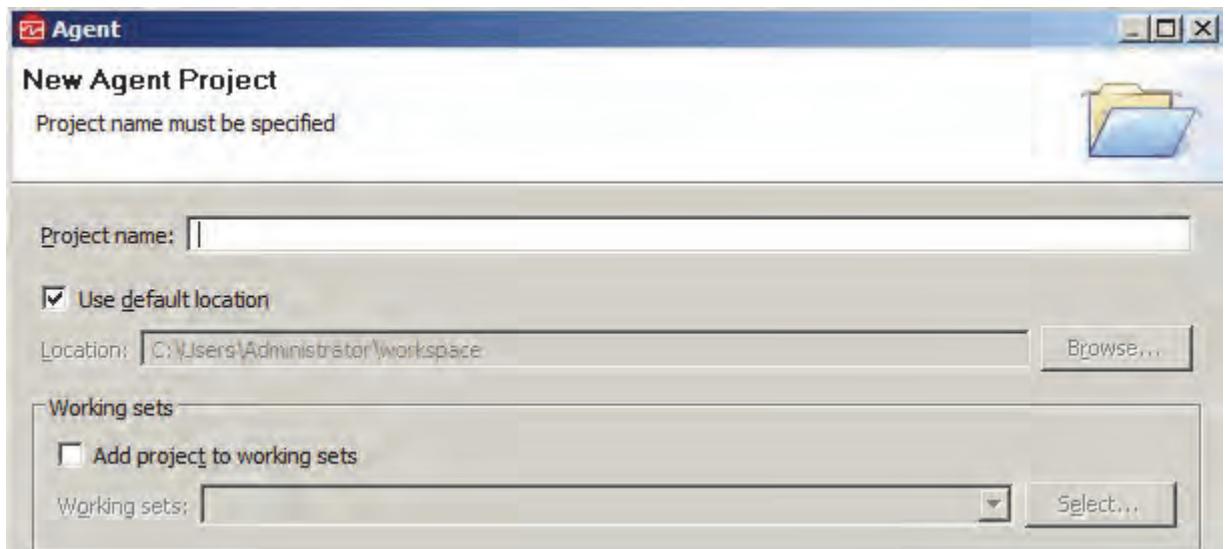


The New Agent wizard opens.

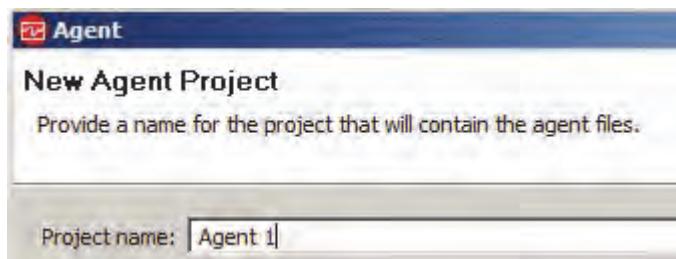


5. Click **Next** to continue with the wizard.

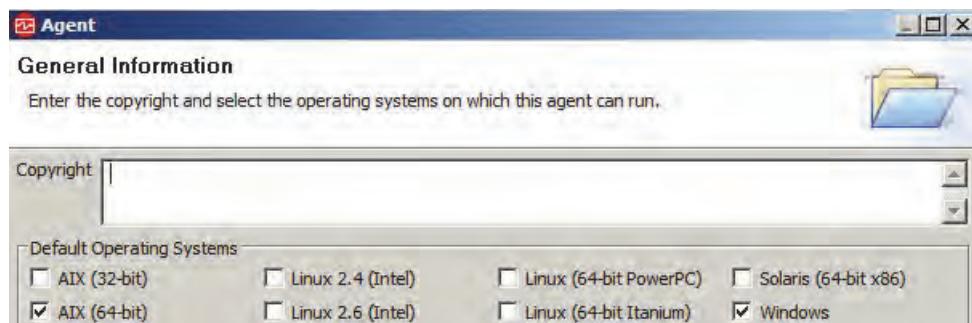
The New Agent Project window opens.



6. Enter **Agent 1** in the **Project name** field and click **Next**.

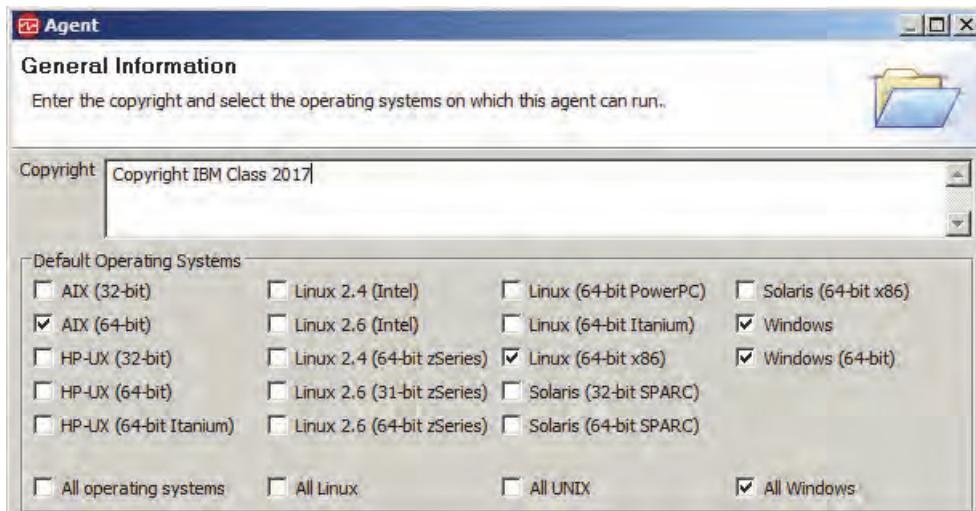


The General Information window opens.



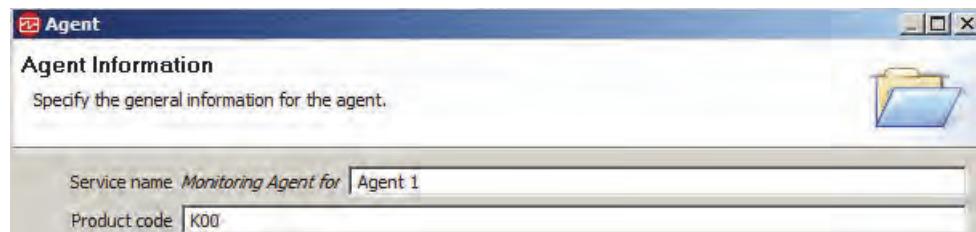
From this window, you can set several general property values for this agent. Enter a copyright statement and identify the specific operating systems on which this agent can be installed. Select the default operating systems on which this agent can be installed.

7. Complete the General Information window.
 - a. Enter **Copyright IBM Class 2017** into the **Copyright** field.
 - b. Keep the default operating systems.



8. Click **Next**.

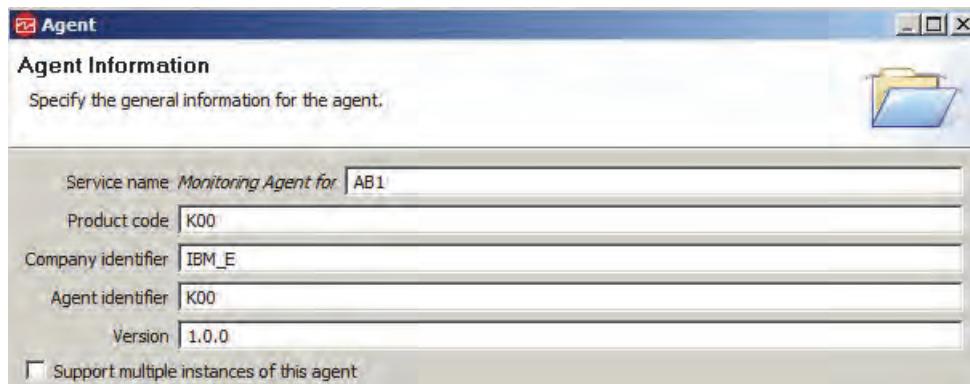
The Agent Information window opens.



Notice that the default **Service name** is the same as the **Project name**, but you can change it. The service name is the name of the service that is created on Windows. It is also the name that is displayed in the IBM Performance Management and Manage Tivoli Monitoring Services utilities for managing installed agents.

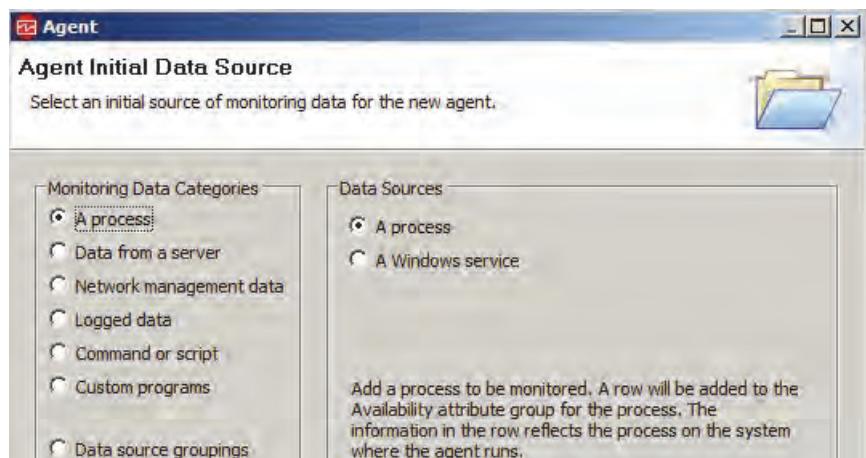
9. Change the Service name to **AB1**.
10. Change the Company identifier to **IBM_E**.

The Agent Information window looks like this screen capture.



11. Click Next.

The Agent Initial Data Source window opens.

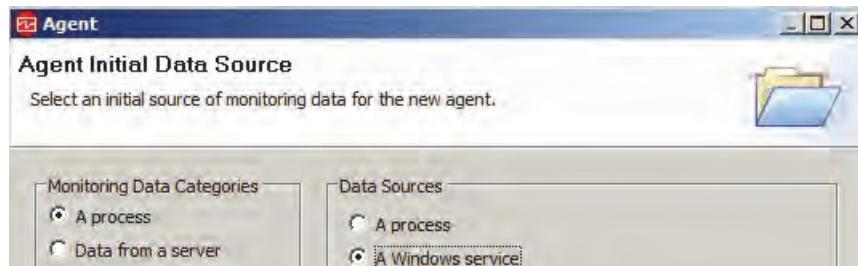


12. Click the different options under **Monitoring Data Categories and browse the different types of data and data groupings you can monitor.**

How each data source is configured can differ greatly, but some data sources work in similar ways. You work with each data source and see their similarities and dissimilarities in later exercises.

For this exercise, you want your agent to monitor several Windows services.

13. Select **A process from **Monitoring Data Categories** and **A Windows service** from **Data Sources**.**



14. Click Next.

The Service Monitor window opens.



From this window, you can enter the display name and the service name of the service you want to monitor. You can also browse for a list of services and select one or more services from the list.



Note: If you plan to browse, do not enter a display name because it is overwritten.

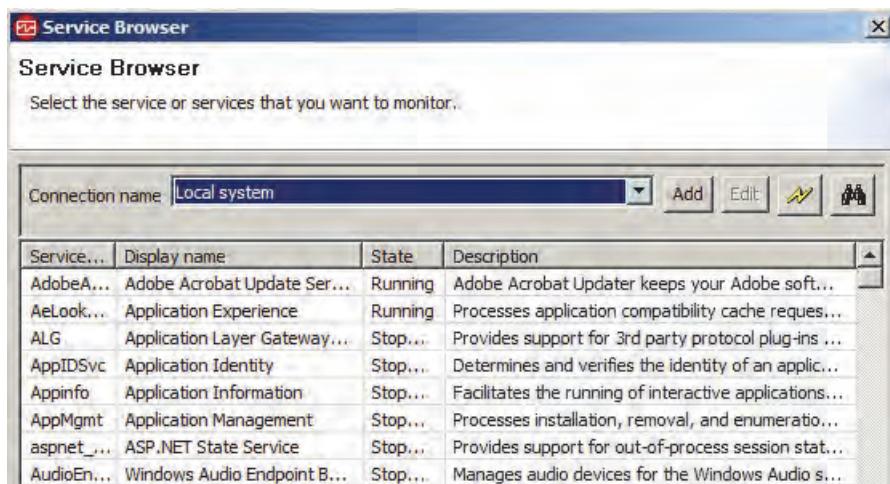
Browse Windows services

You select a single service first.

15. Configure the agent to monitor the DB2 service.

- In the Agent wizard, click **Browse**.

The Service Browser opens.



From this browser, you see all of the services that are installed on this system.

- b. Locate and click the **DB2** service and click **OK**.

Service Name	Display name	State	Description
CryptSvc	Cryptographic Services	Running	Provides four management services: Catalog
DB2	DB2 - DB2COPY1 - DB2	Running	Allows applications to create, update, control
DB2DAS00	DB2DAS - DB2DAS00	Running	Supports local and remote database administra
DB2GOVERNO...	DB2 Governor (DB2COPY1)	Stop...	Collects statistics for applications connected t

The Service Monitor window is displayed.



Both the display name and the service name are entered. You can also connect to and list the services that are installed on remote hosts.

16. Browse processes on a remote system.
- Click **Browse** again to return to the **Service Browser**.
 - Click **Add**.

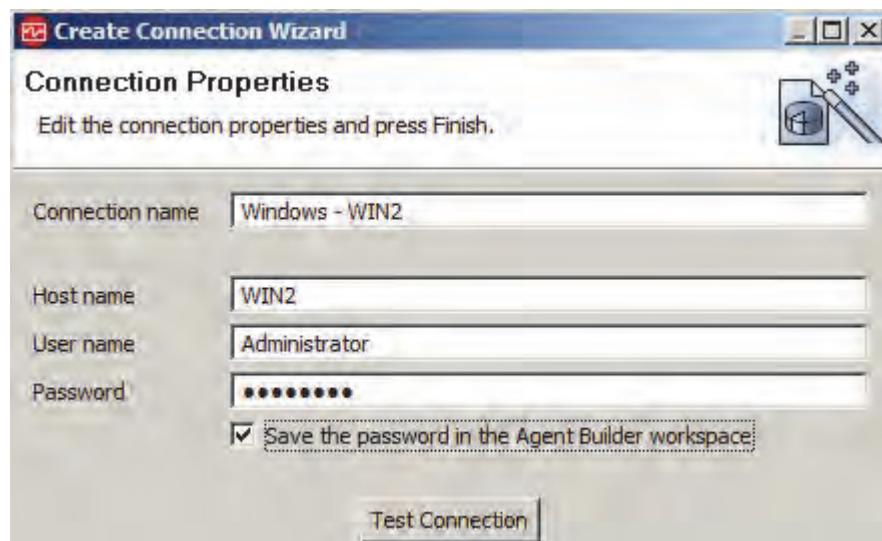


- Select **Windows systems** and click **Next**.



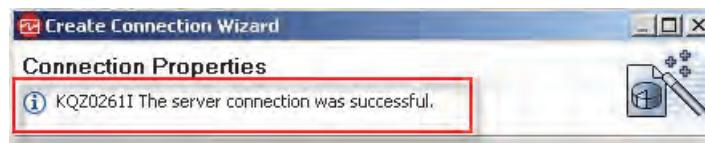
d. Enter the following values:

- ◆ Connection name: **Windows - WIN2**
- ◆ Host name: **WIN2**
- ◆ User name: **Administrator**
- ◆ Password: **object00**
- ◆ Save the password: **Checked**



e. Click **Test Connection**.

f. Confirm that the connection was successful.

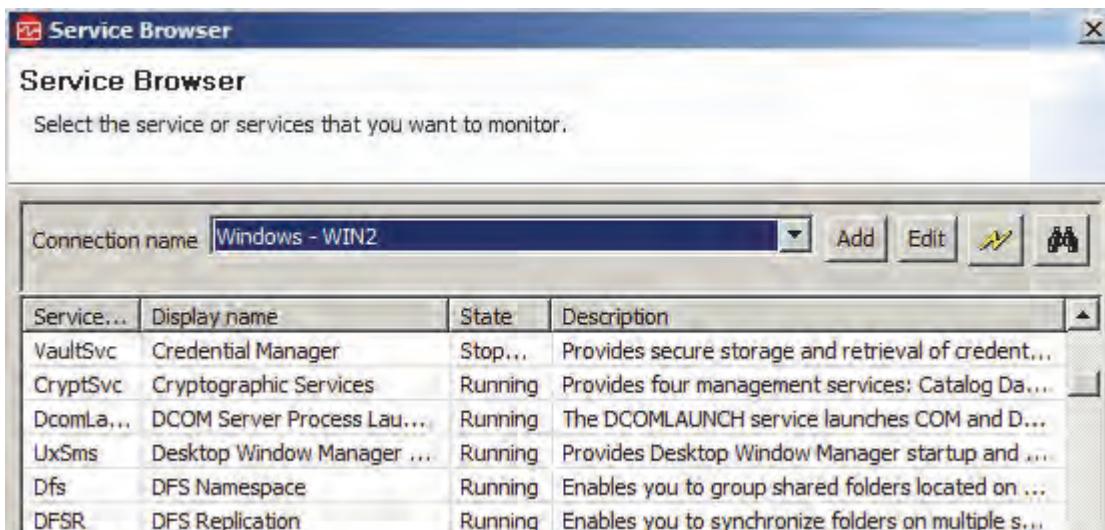


g. Click **Finish**.

You are returned to the Service Browser and the services currently running on WIN2 are displayed.

h. Click the **Display Name** column header to sort the list of services by their display name.

- If you scroll down the list, you find that DB2 is not installed on WIN2.



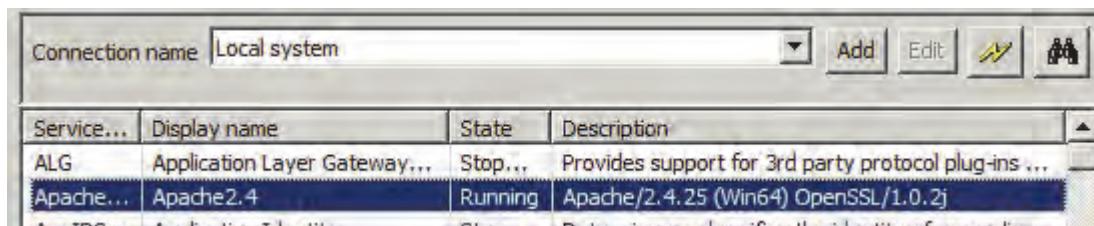
Note: The purpose of browsing a remote data source is not to identify a target host to monitor. Instead, the purpose is to identify data that you want to monitor that is not found on the local host. You don't browse WIN2 to identify WIN2 as a target to monitor, but to select the services that run on it. In this instance, browsing to WIN2 was for demonstration purposes only, and the services you want to monitor are not on it. You must select these services on WIN1.

17. Select the HTTP and DB2 services to monitor:

- Select **local system** (WIN1) from the **Connection name** menu to return to services that run on the local host.
- Select all services that start with DB2 by clicking the first DB2 service, holding down the Shift key, and clicking the last DB2 service.



- c. Scroll up and select the **Apache2.4** service by holding down the CTRL key and clicking the service.



- d. Scroll down and confirm all the DB2 services are still selected.

- e. Click **OK**.

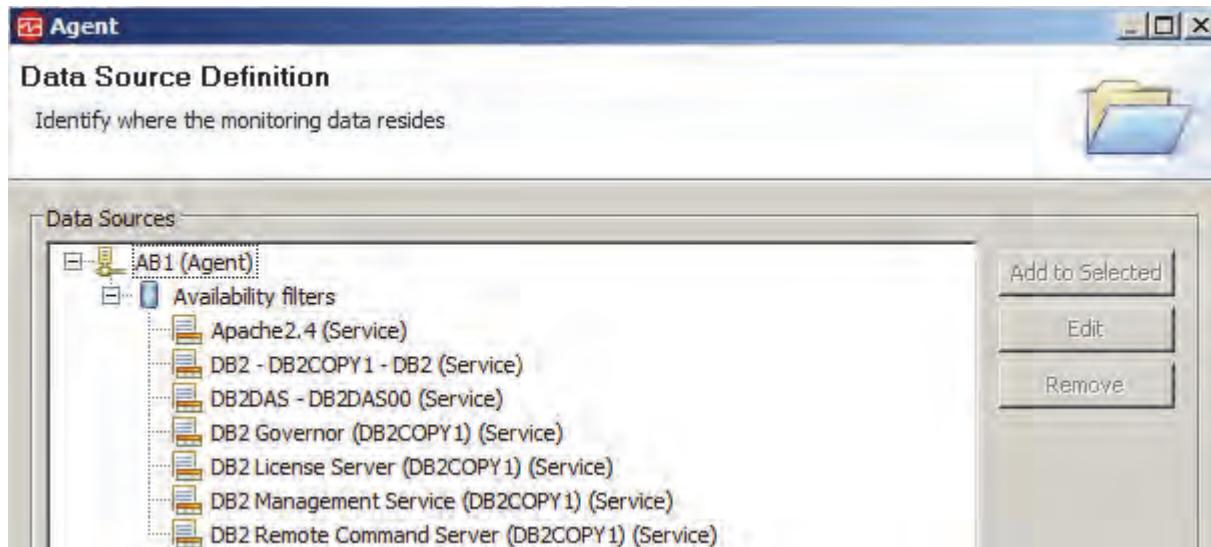
The Service Monitor window is displayed. The **Display name** and **Service name** fields contain a phrase that indicates that you selected more than one service to monitor.



18. Click **Next**.

The Data Source Definition window opens. Your Tivoli Enterprise Portal display name of your agent is displayed, and one attribute group named **Availability filters** is shown.

19. If it is not expanded, click the plus sign (+) beside **Availability filters** to expand the availability filters.



Each of the services that is monitored is listed here.

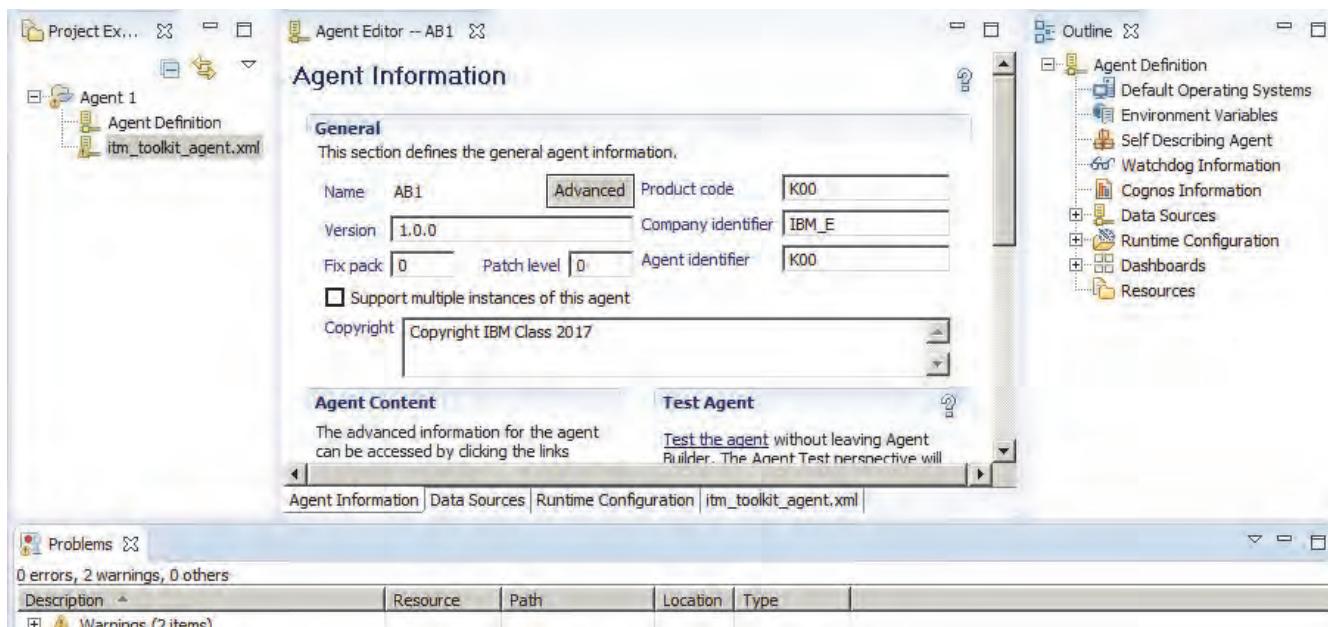
From this window, you can continue to modify the data sources this agent monitors. For example, you can perform the following tasks:

- Add data sources to monitor with this agent. (**Hint:** Highlight **AB1** to add any type of data source, or highlight **Availability filters** to limit the new data source to availability filters.)
- Remove any availability filter or the entire **Availability filters** attribute group.
- Edit an availability filter.

As defined by this scenario, the basic agent definition is complete.

20. Click **Finish**.

You exit the Agent wizard, and the Agent Builder workspace is displayed.

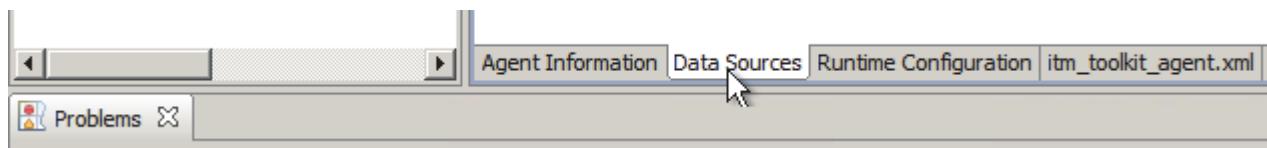


Using the Project Explorer on the left, you can browse your agent projects and their components. The **itm_toolkit_agent.xml** component, which holds your basic agent definition, can be opened in the Agent Editor on the right.

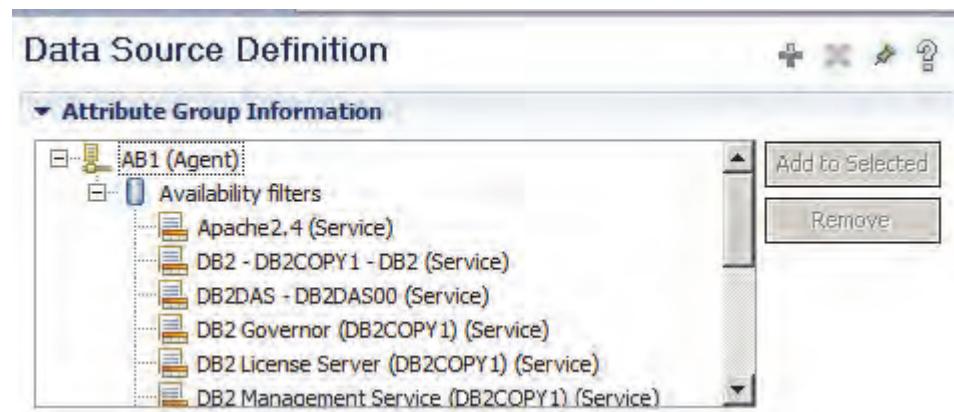
You can use the Agent Editor to modify your agent without restarting the Agent wizard.

21. Browse the other agent definition tabs.

- Click the **Data Sources** tab at the bottom of the Agent Editor window.

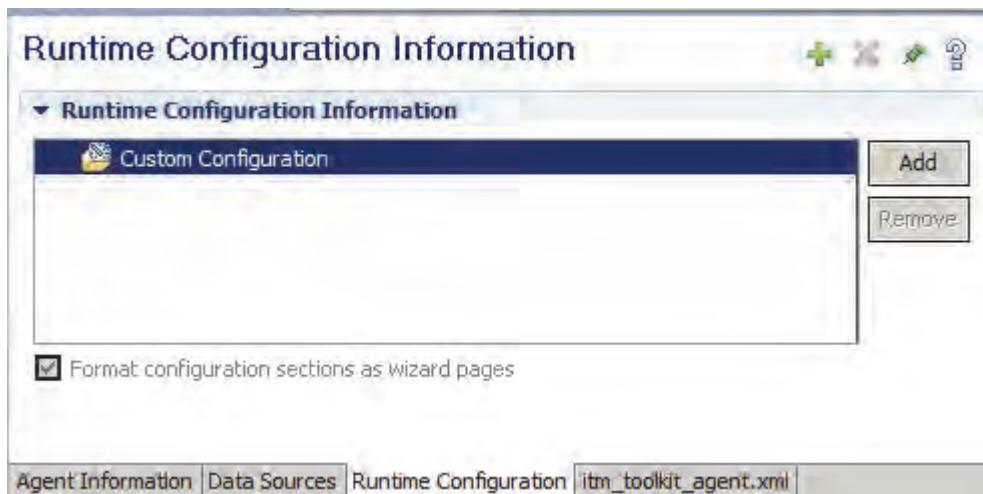


- b. Expand the **Availability filters** attribute group, and click one of the services.



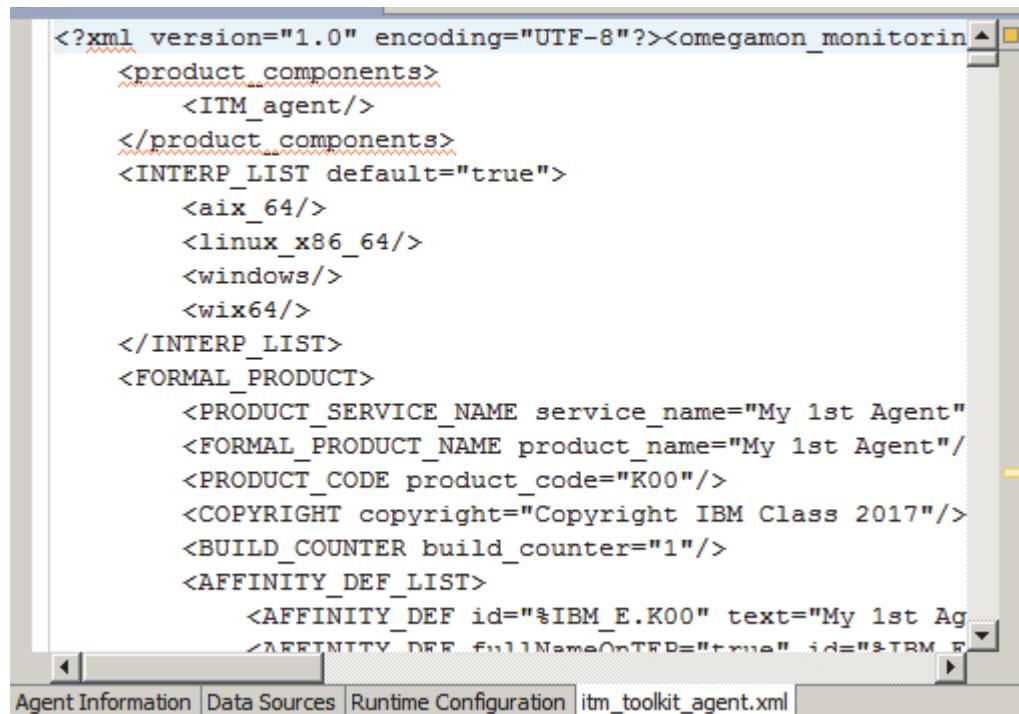
This tab contains all of the data source information that you entered with the Agent wizard. As with the wizard, you can use this tab to edit, remove, and add attributes and attribute groups. You can also edit the availability filter metadata, such as the display name.

- c. Click the **Runtime Configuration** tab at the bottom of the Agent Editor window.



From this tab, you can configure the agent to prompt for data when the agent is configured, such as a local host or directory name. For this exercise, you do not work with runtime configuration.

- d. Click the **itm_toolkit_agent.xml** tab at the bottom of the Agent Editor window.



```

<?xml version="1.0" encoding="UTF-8"?><omegamon_monitorin
<product_components>
    <ITM_agent/>
</product_components>
<INTERP_LIST default="true">
    <aix_64/>
    <linux_x86_64/>
    <windows/>
    <wix64/>
</INTERP_LIST>
<FORMAL_PRODUCT>
    <PRODUCT_SERVICE_NAME service_name="My 1st Agent"
    <FORMAL_PRODUCT_NAME product_name="My 1st Agent"/>
    <PRODUCT_CODE product_code="K00"/>
    <COPYRIGHT copyright="Copyright IBM Class 2017"/>
    <BUILD_COUNTER build_counter="1"/>
    <AFFINITY_DEF_LIST>
        <AFFINITY_DEF id="%IBM_E.K00" text="My 1st Ag
    <AFFINITY_DEF id="STRM_E
</AFFINITY_DEF_LIST>
</FORMAL_PRODUCT>

```

The XML code defines product components, an interpretation list, and a formal product section with service and product details.

The agent information that is displayed in the other tabs is written to this .xml file, which is stored in your Agent Builder workspace directory.



Note: Editing the data in the .xml file edits the data in the tabbed windows, but it is not a recommended or supported practice.

- e. Browse the .xml file and see whether you can identify any of the agent data you entered in this exercise.

Troubleshoot an agent definition problem

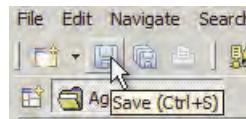
In this section, you create a problem within your agent definition and then use the **Problems** pane to troubleshoot the issue.

22. Click the **Agent Information** tab in the **Agent Editor**.

23. Change the **Company identifier** from **IBM_E** to **IBM_Education**.

Company identifier	IBM_Education
--------------------	---------------

24. Click the **Save** icon to save your changes.



Notice the error in the **Problems** pane at the bottom of the Agent Builder application.

Problems					
1 error, 2 warnings, 0 others					
Description	Resource	Path	Location	Type	
+ Errors (1 item)					
+ Warnings (2 items)					

Hint: If the Problems pane is not visible, drag the resize bar just above the **Problems** tab.

25. Click the plus (+) next to the error to expand the display and see the error.

Problems					
1 error, 2 warnings, 0 others					
Description	Resource	Path	Location	Type	
- Errors (1 item)					
KQZ0197E The combined length of company identifier and ag	itm_toolkit_...	/Agent 1	line 21	Agent Build...	

The window lists the error description, the resource where the problem occurs, and the location where the error exists.

26. Place your cursor on the **Description** field to display the full description.

Description	Resource	Path	Location	Type
- Errors (1 item)				
KQZ0197E The combined length of company identifier and agent identifier cannot exceed 11 characters.				Agent Build...
+ Warnings (2 items)				

Changing the Company identifier made the property too long. Double-clicking the error in the **Problems** tab takes you to an **Agent Editor** tab where you can locate and fix the error.



Important: Ignore the warnings in the Problems pane for now.

27. Click the **Data Sources** tab to take you away from the **Agent Information** tab.

28. Double-click the error in the **Problems** tab.

You are returned to the **Agent Information** tab where the problem can be fixed.

29. Change the **Company identifier** back to **IBM_E** and save your agent project.

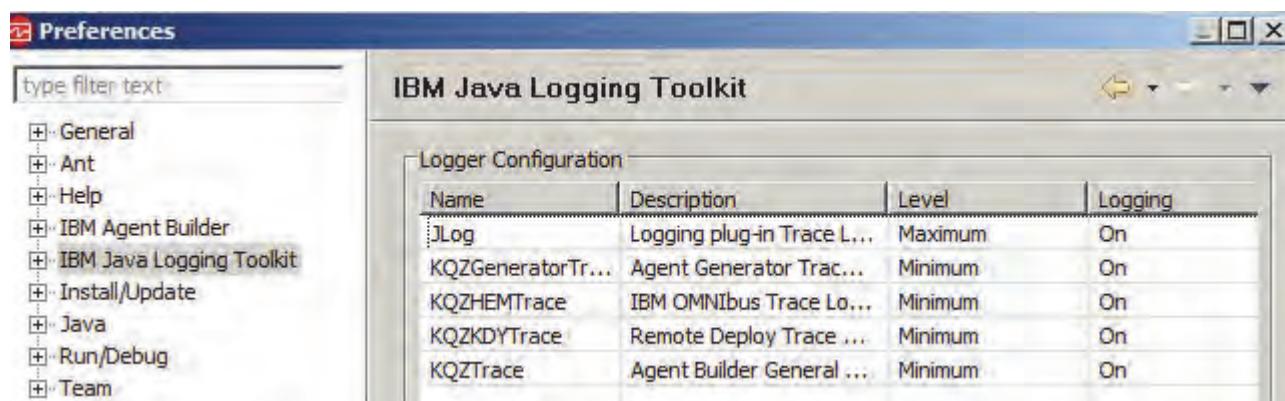


30. Notice that the error in the Problems pane closes.

Troubleshoot Agent Builder

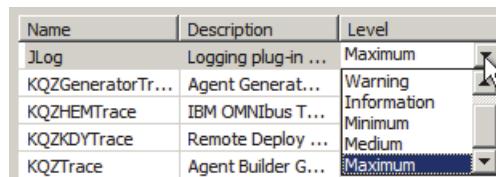
In this section, you see where you can configure and view logging for the Agent Builder application if you must troubleshoot the Agent Builder application.

31. View and configure logging in the Preferences window.
- In Agent Builder, select **Window > Preferences**.
 - Click **IBM Java Logging Toolkit**.



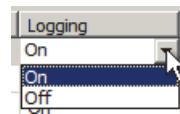
Notice the current logger configurations for Agent Builder.

- Click **Maximum** in the **Level** field for **JLog**.
 - Click the menu button to the right of Maximum.
- A list is displayed. You can use this list to change the log level.



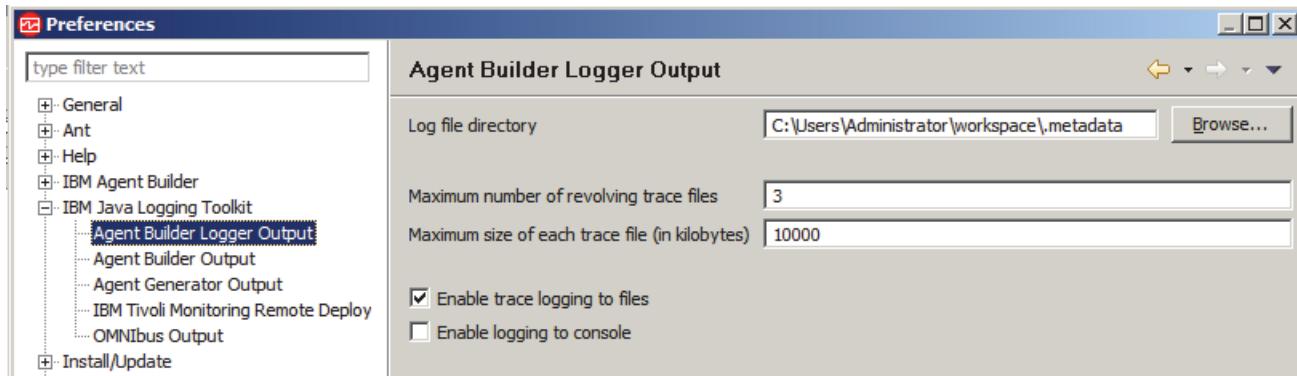
- Click in the **Logging** field for **JLog** and its menu button.

Another menu is displayed.



You can use this menu to turn logging on and off. To see detailed information for each log, you must expand the **IBM Java Logging Toolkit** entry on the left.

- f. Expand **IBM Java Logging Toolkit** and click each subcomponent's logging configuration.



You can now select the individual logs and adjust settings for each one. You can set the logging directory, the number of revolving trace files, and the maximum size of the trace file. You can also enable or disable logging to file or the console.

Notice the default location of each log file.

- g. Click **OK** to close the Preferences window.

32. Browse the log files.

- a. Using Windows Explorer, access the default log files directory:
C:\Users\Administrator\workspace\.metadata
- b. Browse the contents of any log file and look for any errors or warnings.

You successfully created an agent in Agent Builder. In the next units, you learn how to modify this agent for and install into an IBM Performance Management and an IBM Tivoli Monitoring environment.

Unit 3 Customizing agents for IBM Tivoli Monitoring exercises

In this unit, you install the AB1 agent in an IBM Tivoli Monitoring environment. You then customize the agent to include custom IBM Tivoli Monitoring application support.

Exercise 1 Install the AB1 agent in an IBM Tivoli Monitoring environment

In this exercise, you install your AB1 agent into an IBM Tivoli Monitoring environment. This task requires the installation of the agent and the application support for various IBM Tivoli Monitoring components, such as the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal desktop client.

1. Confirm that your Tivoli Monitoring environment is running.
 - a. Log in to the ITM server as **Administrator** with password **object00**.
 - b. Open **Windows Services** and confirm that all **DB2** services set to start automatically are started.

Services (Local)					
DB2 - DB2COPY1 - DB2	Name	Description	Status	Startup Type	Log On As
Stop the service	Cryptographic Services	Provides fo...	Started	Automatic	Network S...
Restart the service	DB2 - DB2COPY1 - DB2	Allows appl...	Started	Automatic	.\db2admin
	DB2 - DB2COPY1 - DSRDBM01	Allows appl...	Manual	Local System	
	DB2 Governor (DB2COPY1)	Collects st...	Manual	Local System	.\db2admin
	DB2 License Server (DB2COPY1)	Monitors D...	Manual	Local System	
	DB2 Management Service (DB2COPY1)	Manages D...	Started	Automatic	Local System
	DB2 Remote Command Server (DB2COPY1)	Supports r...	Started	Automatic	.\db2admin
	DB2DAS - DB2DAS00	Supports lo...	Started	Automatic	.\db2admin
	DCOM Server Process Launcher	The DCOM...	Started	Automatic	Local System

- c. If they are not started, start them now.
- d. Close Windows Services.
- e. Select **Start > Manage Tivoli Monitoring Services** to start the Manage Tivoli Enterprise Monitoring Services (MTMS) utility.

- f. Ensure that the following components started successfully:
- ◆ Tivoli Enterprise Portal Server
 - ◆ Monitoring Agent for Windows OS
 - ◆ Tivoli Enterprise Monitoring Server

Service/Application	Task/SubSystem	Configured	Status	Configuration	Startup	Account
Eclipse Help Server	HELPSVR	Yes	Started	up-to-date	Auto	LocalSystem
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A	N/A
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A
Tivoli Enterprise Portal Server	KFWSRV	Yes (TEMS)	Started	up-to-date	Auto	LocalSystem
Warehouse Summarization and Pr... Warehouse Summarization and Pr...	Primary	Yes (TEMS)	Started	up-to-date	Auto	LocalSystem
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	up-to-date	Auto	LocalSystem
Warehouse Proxy	Primary	Yes (TEMS)	Started	up-to-date	Auto	LocalSystem
Performance Analyzer	Primary	Yes (TEMS)	Started	up-to-date	Auto	LocalSystem
Tivoli Enterprise Monitoring Autom... Tivoli Enterprise Monitoring Autom...	KAS1	Yes (TEMS)	Started	up-to-date	Auto	LocalSystem
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Started	up-to-date	Auto	LocalSystem

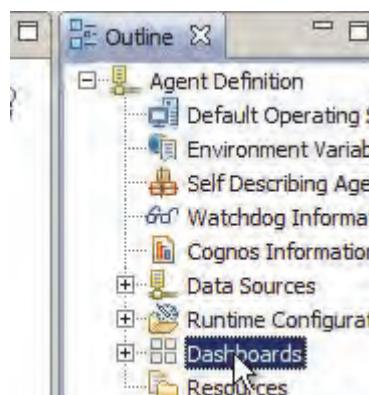
- g. If they are not started, start them now.



Hint: Double-click the component name.

Disable IBM Performance Management dashboards

2. In Agent Builder on WIN1, with the AB1 agent open, select **Dashboards** in the **Outline** window on the right.



The Dashboards tab opens.

3. Select **No dashboard presence for this agent**.

Dashboards

Overview

The Dashboards page lets you identify data from your agent that can be displayed on the IBM Performance Management console. Run the [Dashboard Setup wizard](#) to select an overall status attribute, other displayable attributes, and attributes that identify the software server being monitored. You can run the Dashboard Setup wizard again if you want to select different attributes or quickly see the attributes that are currently selected.

Dashboard Components

- Show agent components in the dashboard
- No dashboard presence for this agent

4. Save your agent project.

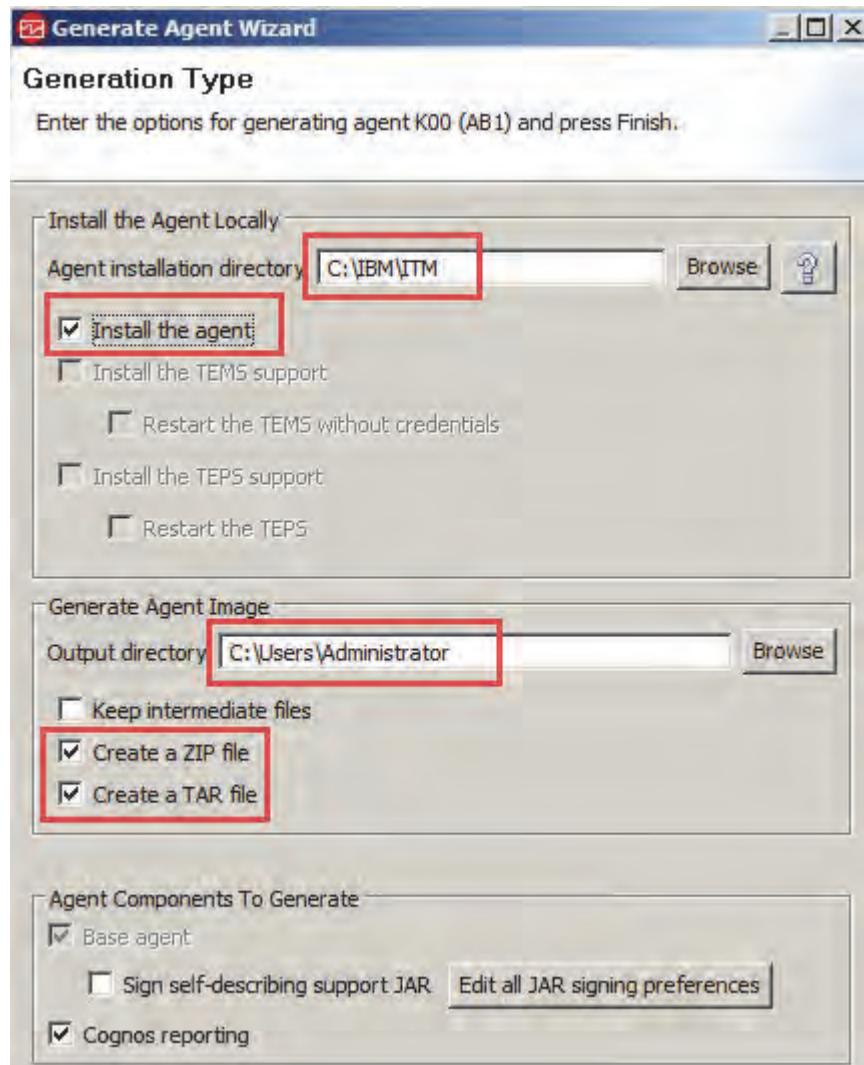


Note: The warnings in the Problems panel are removed.

Install the agent and create the installation scripts

5. Select **Agent Editor > Generate Agent** from the main menu.
6. Confirm that the Agent installation directory is **C:\IBM\ITM**.
7. Select **Install the agent**.

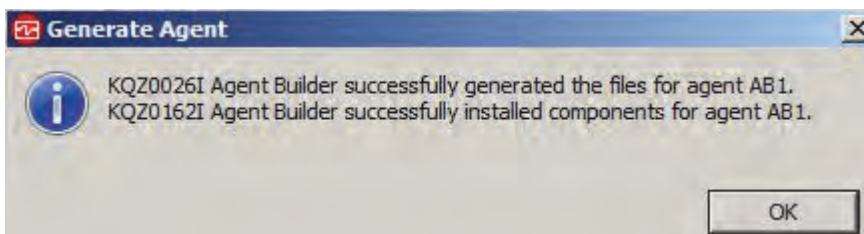
8. Ensure that **Create a ZIP file** and **Create a TAR file** are checked.



Notice the directory where the files are created. You go to this directory to access the generated files. Keep the default output directory for this exercise.

9. Click **Finish**.

A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated and the agent is installed.



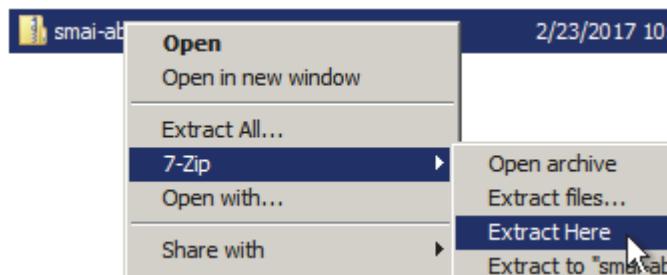
10. Click **OK** to close the confirmation window.

11. Open a command prompt or Explorer window and go to **C:\Users\Administrator** and confirm that the **smai-ab1-01.00.00.00.tgz** and **smai-ab1-01.00.00.00.zip** files are created.

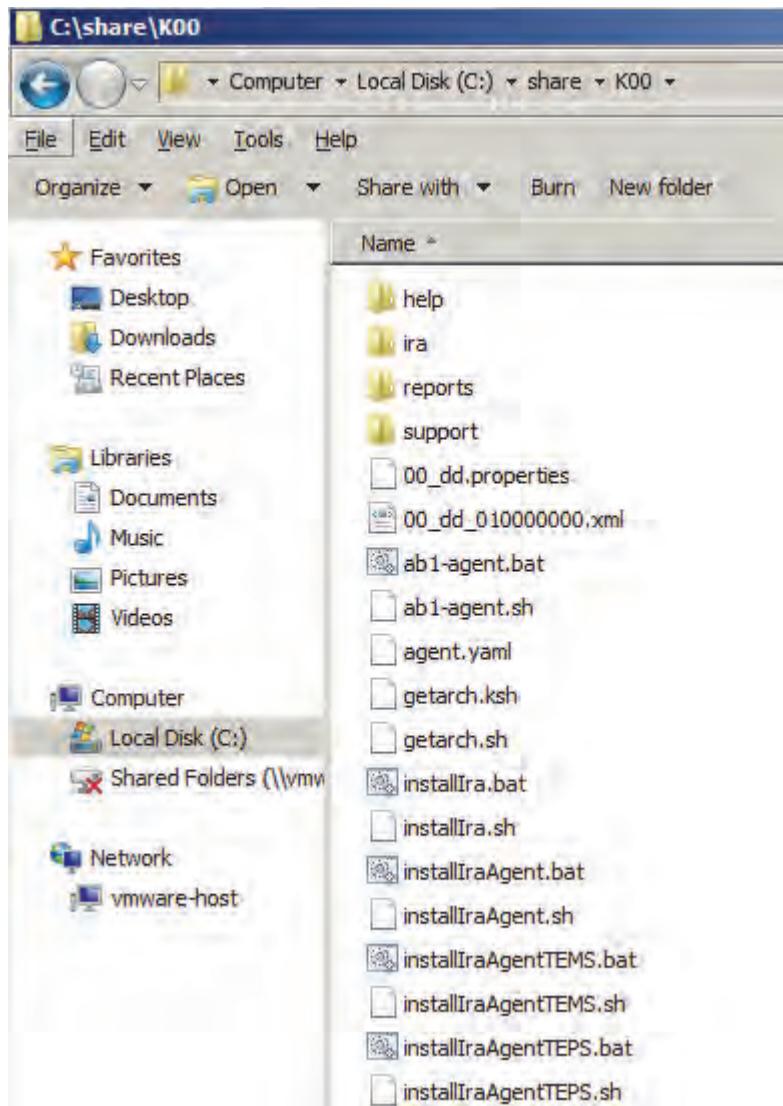
The **smai-ab1-01.00.00.00.tgz** and **smai-ab1-01.00.00.00.zip** files contain the same files that you can use to install the agent and the application support for all platforms, including Windows, UNIX, and Linux. Because of the archive formats, the files in the .zip file can be used for installing components on Windows and the .tgz file can be used to install components on UNIX and Linux.

You install an agent with the installation scripts in a later exercise.

12. Create a directory that is named **K00** in the **C:\share** directory.
13. Copy the **smai-ab1-01.01.00.00.zip** file into the **C:\share\K00** directory.
14. From Windows Explorer, right-click the **C:\Users\Administor\smai-ab1-01.01.00.00.zip** file and select **7-Zip > Extract Here**.



The agent installation files are extracted.



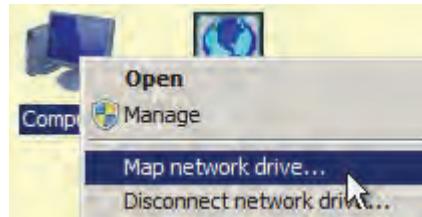
- **installIraAgent.bat/.sh** installs the agent.
- **installIraAgentTEMS.bat/.sh** installs the Tivoli Enterprise Monitoring Server support.
- **installIraAgentTEPS.bat/.sh** installs the Tivoli Enterprise Portal Server and Tivoli Enterprise Portal support.
- **installIra.bat/.sh** installs everything, including the agent and application support on the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop client.



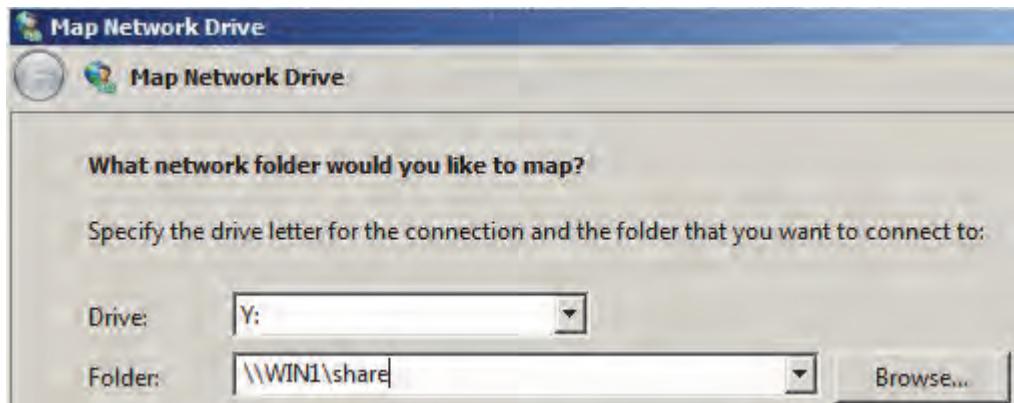
Note: You can run each script without arguments to get a usage statement.

Install the AB1 agent application support on ITM

15. On ITM, exit the TEP client if it is open. To avoid display problems in the Tivoli Enterprise Portal, close the TEP client during an application support installation.
16. Open a command prompt and change to the **Y:\K00** directory.
If the Y: drive is not available, map the Y: drive to \\WIN1\share.
 - a. On the ITM desktop, right-click **Computer** and select **Map network drive**.



- b. Set the drive to **Y:** and enter **\\\WIN1\share** in **Folder**.



- c. Click **Finish**.

The WIN1 share opens in Windows Explorer

To install the IBM Tivoli Monitoring application support without installing the agent, you run the `installIraAgentTEMS` and `installIraAgentTEPS` scripts.

17. Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00
installIraAgentTEPS.bat C:\IBM\ITM
```

```
Y:\>cd K00
Y:\K00>installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00
Validating user...
KUIC00007I: User sysadmin logged into server on https://ITM:3661.
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.

Install of K00 TEMS Support successful.

Y:\K00>installIraAgentTEPS.bat C:\IBM\ITM
Online help for this agent will not be available until the Help Server is restarted,
which also requires restarting the TEPS.

Install of K00 TEPS Support successful.

Y:\K00>
```

The full installation takes 7 - 10 minutes to complete. Status information is displayed for each command.



Important: If you receive a notification that the Tivoli Enterprise Monitoring Server or Tivoli Enterprise Portal Server installation failed, ensure that the following services are running and that they are started in the following order. Then, do the installation again.

- DB2 - DB2Copy1
- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server

You completed the installation of the AB1 agent (K00) agent and application support on the ITM server.

Troubleshoot an Agent Builder agent installation

In this section, you browse the agent installation log files.

18. Check the IBM Tivoli Monitoring installation log files for the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

- a. From Windows Explorer, open the following directory:

C:\ibm\ITM\TMAITM6_64\logs

In this directory, any log file that starts with your agent's product code, **K00**, is related to your agent. Any of those files with the word **install** in it is related to your agent installation.

- b. Open **K00install.log** in Notepad and review the contents.
- c. Open and review any other installation log files that you are interested in.

Configure, start, and confirm your agent on WIN1

19. On WIN1, configure and start the **Monitoring Agent for AB1** agent in the Manage Tivoli Enterprise Monitoring Services (MTMS) utility.

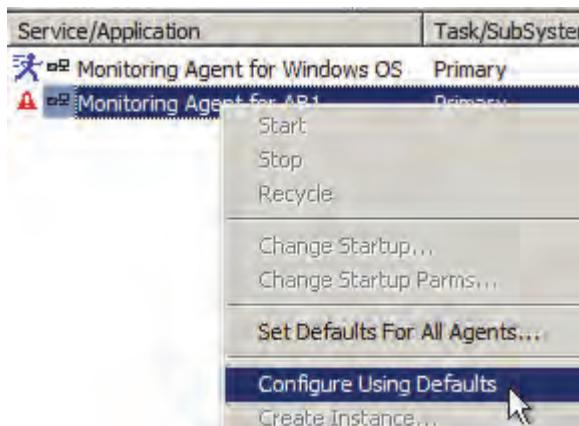
- a. Select **Start > All Programs > IBM Tivoli Monitoring > Manage Tivoli Monitoring Services**.



The Manage Tivoli Monitoring (MTMS) utility opens.

Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]											
Action	Options	View	Windows	Help							
Service/Application	Task/SubSystem	Configured	Status	Configuration	Startup	Account	Desktop	HotStandby	Version		
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	up-to-date	Auto	LocalS...	Yes	No	06.30.07.00		
Monitoring Agent for AB1	Primary	No	N/A						01.00.00.00		

- b. Notice the following items:
 - ◆ The name of your agent in the MTMS utility is **Monitoring Agent for AB1**.
 - ◆ The agent is not configured or started.
 - ◆ The agent version is **1.00.00**.
- c. Right-click your new agent and click **Configure Using Defaults**.



The agent is configured and the status icon changes.

Service/Application	Task/SubSystem	Configured	Status
Monitoring Agent for Windows OS	Primary	Yes (TEMPS)	Started
Monitoring Agent for AB1	Primary	Yes (TEMPS)	Stopped

- d. Double-click **Monitoring Agent for AB1** to start it.

The status icon changes again.

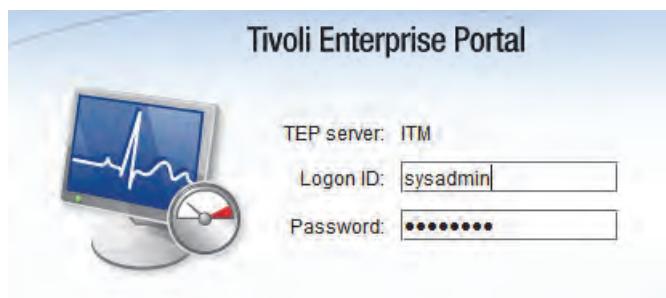
Service/Application	Task/SubSystem	Configured	Status
Monitoring Agent for Windows OS	Primary	Yes (TEMPS)	Started
Monitoring Agent for AB1	Primary	Yes (TEMPS)	Started



Note: Keep the MTMS utility open; you use it again later.

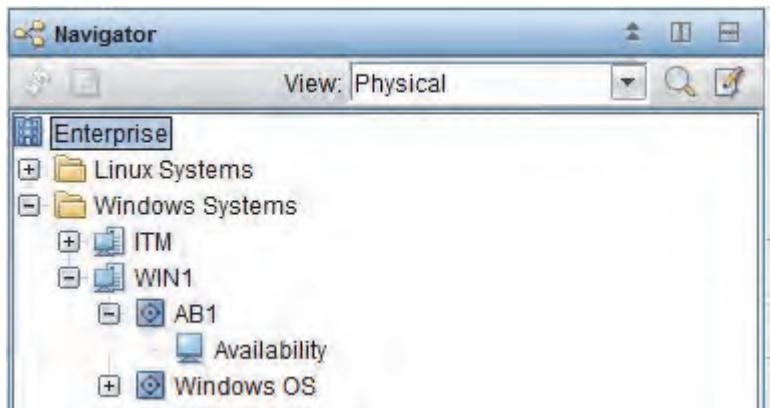
20. Confirm that your **AB1** agent is displayed in the Tivoli Enterprise Portal under **WIN1** and that it is gathering data.

- On ITM, double-click the Tivoli Enterprise Portal icon  on the desktop to start the Tivoli Enterprise Portal desktop client.
- Log in as **sysadmin** with password **object00**.



- If prompted, permanently accept the security certificate.

- d. In the Navigator, expand **Windows Systems > WIN1 > AB1**.



In the Navigator, your agent is called **AB1**, which is the value that is entered in the **Display name** field in Agent Builder.

21. Confirm that the default attribute group and workspace for this agent exist and contain data.

- a. Click **Availability** in the Navigator.

Report					
Node	Timestamp	Application Component	Name	Status	Full N:
WIN1:00	03/03/17 21:13:10	Apache2.4	Apache2.4	UP	Apache2.4
WIN1:00	03/03/17 21:13:10	DB2 - DB2COPY1 - DB2	DB2	UP	DB2 - DB2COPY1 - DB2
WIN1:00	03/03/17 21:13:10	DB2DAS - DB2DAS00	DB2DAS00	UP	DB2DAS - DB2DAS00
WIN1:00	03/03/17 21:13:10	DB2 Governor (DB2COPY1)	DB2GOVERNOR_DB2COPY1	DOWN	DB2 Governor (DB2COP
WIN1:00	03/03/17 21:13:10	DB2 License Server (DB2COPY1)	DB2LICD_DB2COPY1	DOWN	DB2 License Server (DB
WIN1:00	03/03/17 21:13:10	DB2 Management Service (DB2COPY1)	DB2MGMTSVC_DB2COPY1	UP	DB2 Management Servic
WIN1:00	03/03/17 21:13:10	DB2 Remote Command Server (DB2COPY1)	DB2REMOTECMD_DB2COPY1	UP	DB2 Remote Command

The Report view contains the service monitoring data.

- b. Scroll to the right to see all of the data that is collected by each service.

Three types of data sources are availability monitors: Windows services, processes, and command return codes. These three data filters differ from all the other data sources in that their data is always displayed in the same Attribute Group, **Availability**, in the Tivoli Enterprise Portal. The Type column distinguishes between the three different data types, and some availability data sources do not use all of the columns.

For example, notice that the Windows services do not use the last two columns.

Processor Time	Command Line	Functionality Test Status	Functionality Test Message
0	"C:\Apache24\bin\httpd.exe" -k runservice	N/A	N/A
0	C:\PROGRA~1\IBM\SQLLIB\bin\db2syscs.exe	N/A	N/A
0	"C:\Program Files\IBM\SQLLIB\bin\db2dasrrm.exe"	N/A	N/A
0	N/A	N/A	N/A
0	N/A	N/A	N/A
0	"C:\Program Files\IBM\SQLLIB\BIN\db2mgmtsvc.exe"	N/A	N/A

Only command return code data sources use these columns.

This basic test of your agent confirmed that the agent and its application support components that are installed correctly. It also confirmed that the agent gathers the data that you want and it displays the data correctly in the Tivoli Enterprise Portal.

You are not limited to this default workspace. You can add custom workspaces, situations, and Take Actions to an agent project and they are installed with your agent's application support. This ability is covered later in this class.

Exercise 2 Create and import IBM Tivoli Monitoring application support

In this exercise, you modify the default workspaces of the AB1 node by adding a custom view. You also create a custom situation for your agent.

Create and tag your custom workspace for export to Agent Builder

To create custom workspaces for import into an Agent Builder agent, you must modify the Tivoli Enterprise Portal startup environment and enter the Tivoli Enterprise Portal workspace administration mode (Admin mode). In this section, you modify the TEP client environment to tag new workspaces for export to Agent Builder. You enable the Admin mode later in this exercise.

Enable the TEP client to tag workspaces for export to Agent Builder

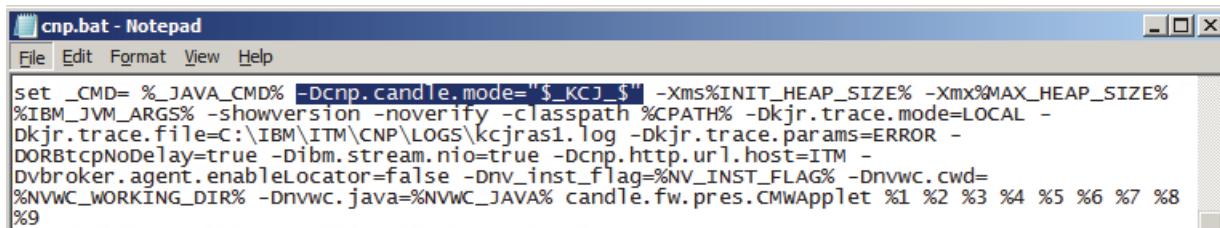
1. On ITM, exit the TEP client.
2. Modify the **set _CMD** line in the TEP client startup script to include the following variable, which identifies importable workspaces:

```
-Dcnp.candle.mode="$_KCJ_$"
```

 - a. Edit the following file on the ITM server:

```
C:\IBM\ITM\CNP\cnp.bat
```
 - b. Edit the **set _CMD** line by inserting the following options:

```
-Dcnp.candle.mode="$_KCJ_$"
```



The screenshot shows a Windows Notepad window titled "cnp.bat - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main text area contains the following batch script code:

```
set _CMD= %_JAVA_CMD% -Dcnp.candle.mode="$_KCJ_$" -Xms%INIT_HEAP_SIZE% -Xmx%MAX_HEAP_SIZE%  
%IBM_JVM_ARGS% -showversion -noverify -classpath %CPATH% -Dkjr.trace.mode=LOCAL -  
Dkjr.trace.file=C:\IBM\ITM\CNP\LOGS\kcjras1.log -Dkjr.trace.params=ERROR -  
DORBtcpnodeDelay=true -Dibm.stream.nio=true -Dcnp.http.url.host=ITM -  
Dvbroker.agent.enableLocator=false -Dnv_inst_flag=%NV_INST_FLAG% -Dnvwc.cwd= %NVWC_WORKING_DIR% -Dnvwc.java=%NVWC_JAVA% candle.fw.pres.CMWApplet %1 %2 %3 %4 %5 %6 %7 %8  
%
```

Changing this value tags workspaces that are created in the TEP Admin mode so that they can be exported into Agent Builder.

- c. Save your changes and exit the editor.
3. Start the **TEP client** and log in as **sysadmin**.

Create a custom workspace

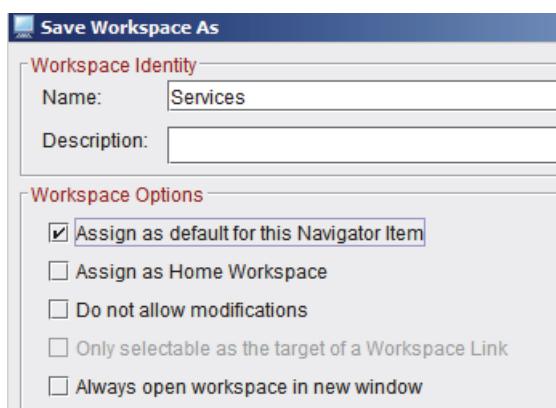
4. Locate and open the **Availability** workspace for **AB1**.
5. Right-click **Availability** and click **Workspace**.



The current workspace is listed as Availability, but no actual Availability workspace exists. This workspace is a mock workspace. The Tivoli Enterprise Portal automatically generates it based on the single query that is associated with the Availability Navigator item.

You must create an actual workspace for Availability.

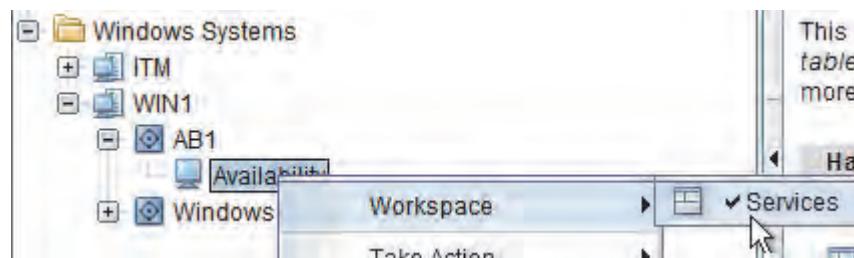
6. Save a copy of the temporary workspace name it **Services**, and set it as the new default workspace.
 - a. Select **File > Save Workspace As**.
 - b. Enter **Services** in the **Name** field.
 - c. Select **Assign as default for this Navigator Item**.



- d. Click **OK** to save your changes.

Services is now displayed in the upper left corner of the Tivoli Enterprise Portal.

- e. Right-click **Availability** in the Navigator and highlight **Workspace**.



You have only one workspace, Services, and the Availability workspace no longer exists. This behavior is the default for new agents. They are not true workspaces until you create them in the Tivoli Enterprise Portal.

7. Make the following changes to the Report view of the Services workspace.



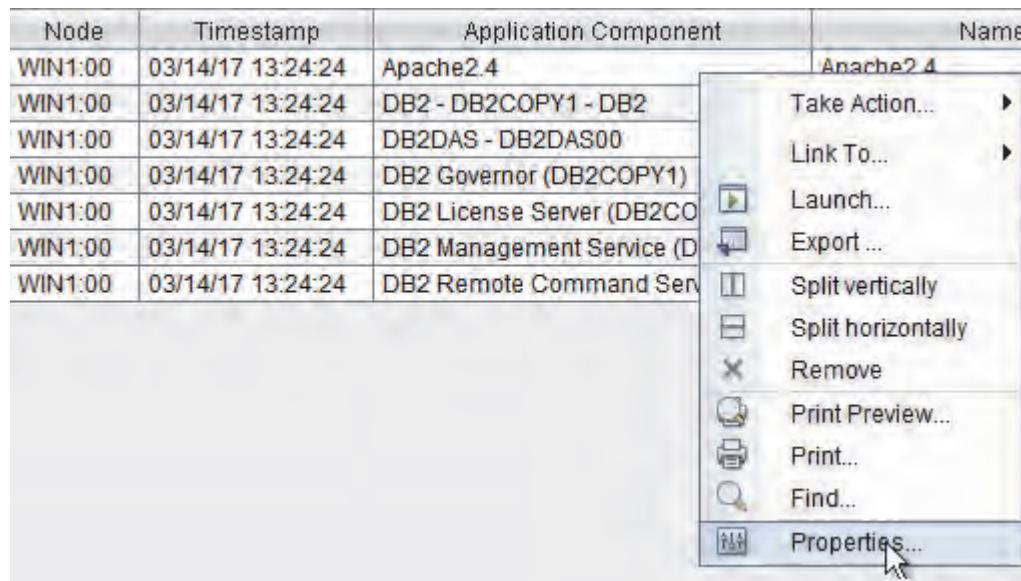
Hint: Detailed steps are included on the next page beginning with Step a.

- Change the title to **Services**.
- Remove the following columns:
 - Node
 - Name
 - Full Name
 - Type
 - Functionality Test Status
 - Functionality Test Message
- Move the columns so that the first four columns are in the order that is shown in the following example.

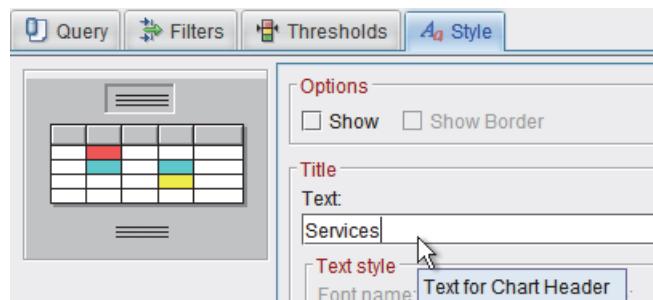
Services				
Timestamp	PID	Application Component	Status	
03/03/17 ...	1420	Apache2.4	UP	
03/03/17 ...	2968	DB2 - DB2COPY1 - DB2	UP	
03/03/17 ...	1620	DB2DAS - DB2DAS00	UP	
03/03/17 ...	0	DB2 Governor (DB2COPY1)	DOWN	

If you look at the default view, you see that many columns contain redundant data or data that is not applicable to monitoring service availability. The Availability workspace can hold data from three different types of sources: Windows services, processes, and the command return codes, where unique data might be shown in these columns. You can remove the columns from the view because your agent is gathering only Windows Services data.

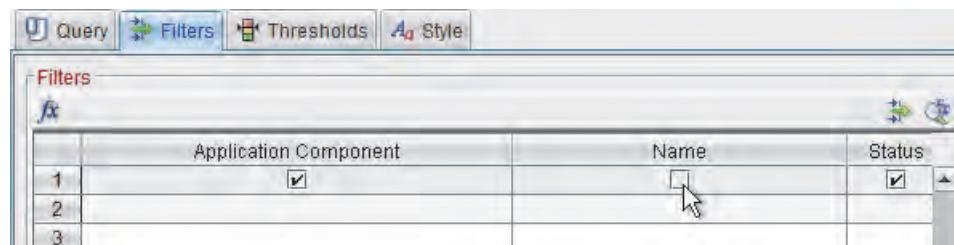
- a. Right-click the **Report** view and click **Properties**.



- b. Click the **Style** tab.
c. Change **Report** in the **Title Text** field to **Services**.



- d. Click the **Filters** tab.
e. Remove the unwanted columns by clearing the check box under the Node, Name, Full Name, Type, Functionality Test Status, and Functionality Message columns.



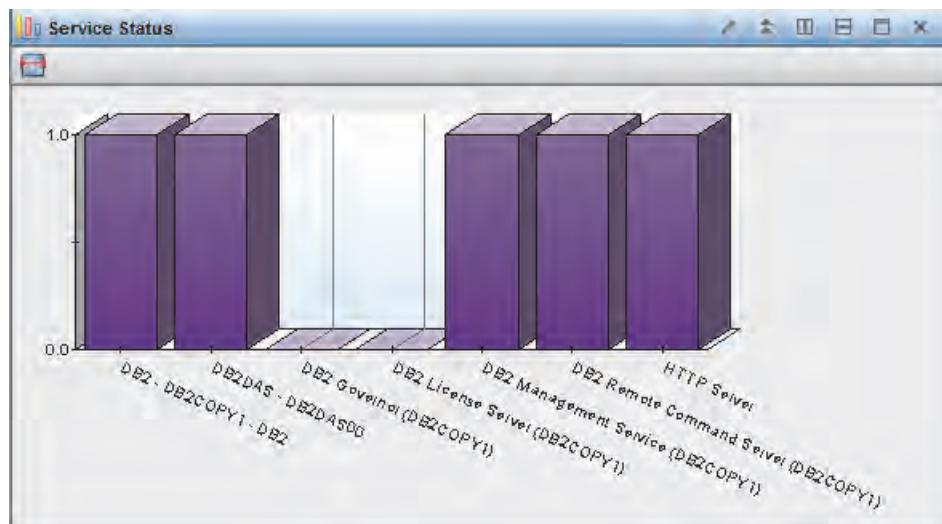
- f. Drag the four columns to the order shown in the following example:

Filters					
	Node	Timestamp	PID	Application Component	Status
1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2					

- g. Click **OK** to close the properties window and view your changes.

Your view looks like the Services view that was shown at the start of this step.

8. Replace the undefined view at the upper right with the following bar chart that shows the status of the monitored services.



- a. Click the **Bar Chart** icon from the toolbar menu.

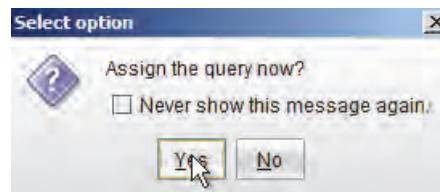


- b. Click the undefined view with the **Bar Chart** icon.

This screenshot shows a browser window with the following details:
- Title bar: "This view has not been defined"
- Address bar: "Location: http://itm:15200"
- Content area:

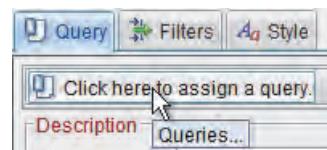
- A large bold text: "This view has not been defined".
- An explanatory message: "This is the default workspace for this Navigator item, and no table view. You can enter a URL in the address text box to see more views as described in these topics:"
- An icon of a bar chart with a red arrow pointing to it, located next to the explanatory text.

- c. Click **Yes** to assign a query.



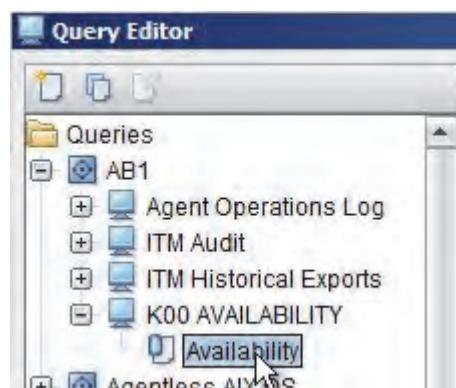
The Properties window opens.

- d. Click **Click here to assign a query**.



The Query Editor opens.

- e. Locate and click the **Availability** query for your **AB1** agent.



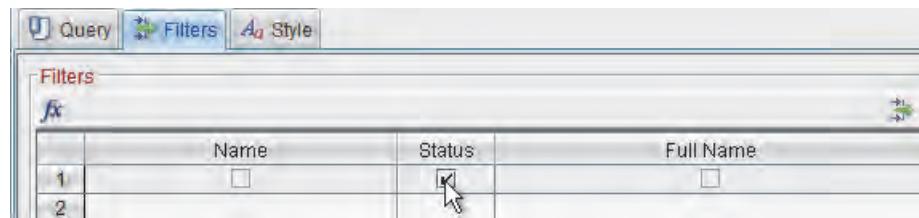
- f. Click **OK** to select this query.

The Properties window is displayed.

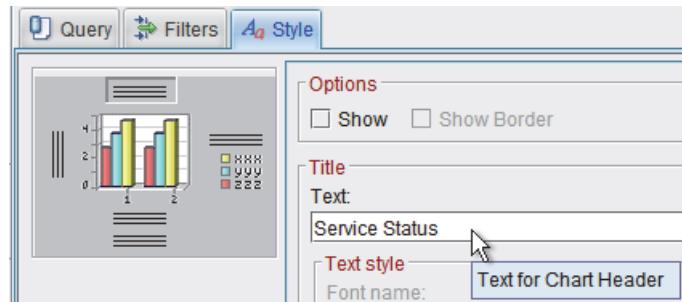


- g. Click the **Filters** tab.

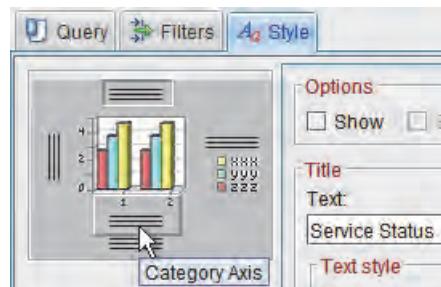
- h. Locate and select the check box below **Status**.



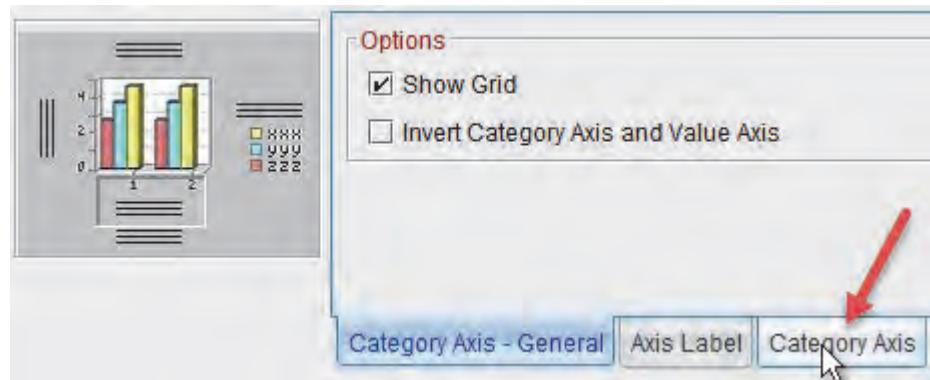
- i. Click the **Style** tab.
j. Change **Bar Chart** in the **Title Text** field to **Service Status**.



- k. Click the **Category Axis style** section.

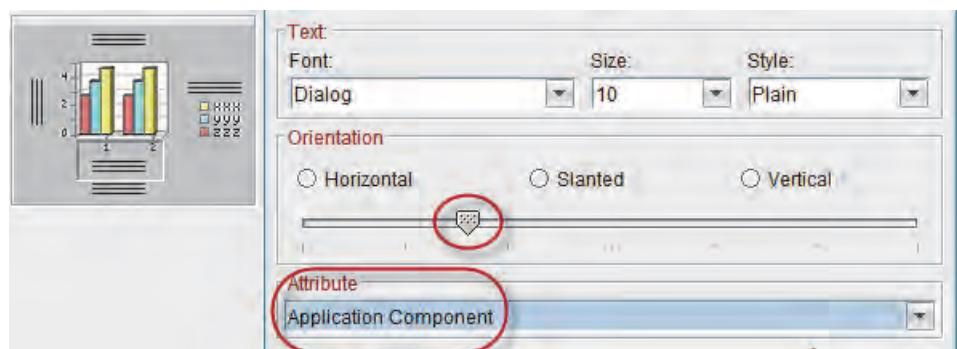


- l. Click the **Category Axis** tab.

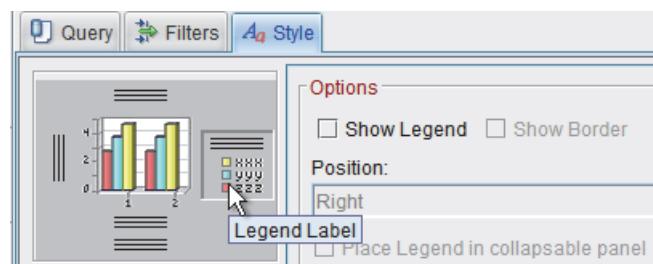


- m. Place the **Orientation** bar between **Horizontal** and **Slanted**.

- n. Select **Application Component** from the **Attribute** field.

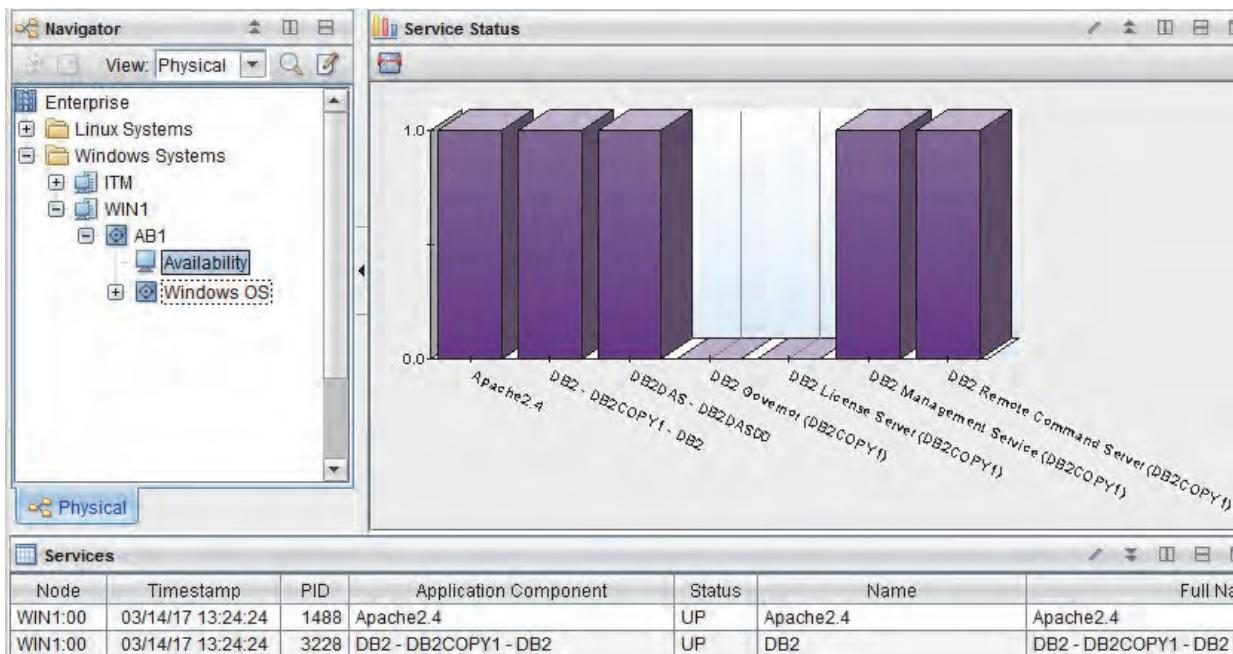


- o. Click the **Legend Label** style section and clear **Show Legend**.



- p. Click **OK** to close the **Properties** windows and view your changes.

Your final workspace looks like the following example.



9. Select **File > Save Workspace** to save your changes to the workspace.

Publish your workspace

The custom workspace that you created is assigned only to you. To be imported into an Agent Builder agent, a workspace must be published in the TEP Admin mode. In this section, you enter the TEP Admin mode and resave your workspace, which publishes it for general use. You then turn off the TEP Admin mode.

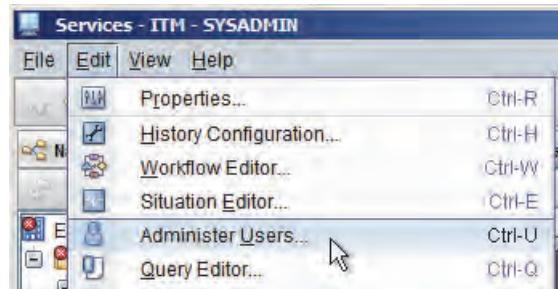


Note: You can activate the TEP Admin mode before creating the workspace. This method of publishing workspaces after you create them outside of TEP Admin mode is considered a Tivoli Enterprise Portal best practice. It helps to ensure that you do not accidentally publish incomplete workspaces or workspaces other than the ones you intended.

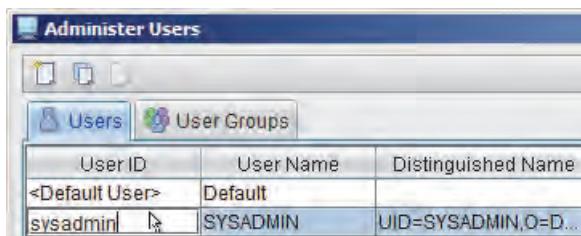
10. If you are not already there, open your **Services** workspace by going to the **Availability** node.

11. Enter the TEP Admin mode.

- Select **Edit > Administer Users**.

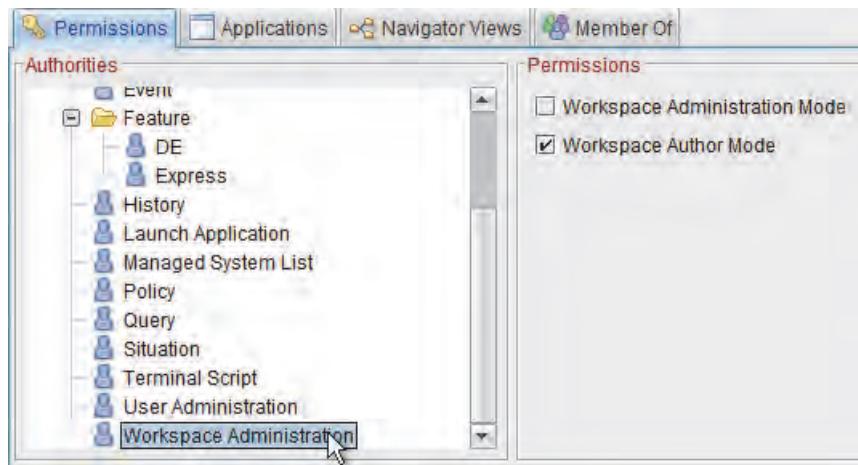


- Click **sysadmin** on the **Users** tab.



Important: Be sure that you click **sysadmin** and do not leave **Default** selected.

- c. Locate and click **Workspace Administration** in the left pane of the **Permissions** tab.



- d. On the right side of the window, select the **Workspace Administration Mode** check box.



- e. Click **OK** to save your changes and close the Administer Users window.

If you followed the steps correctly, ***ADMIN MODE*** is displayed in the desktop title bar.



If it is not displayed, repeat this step to activate it.

12. Resave the workspace by selecting **File > Save Workspace**.

Your Services workspace is now published and tagged for import into Agent Builder.



Note: Do not exit the TEP Admin mode now. It must be active during the import process.

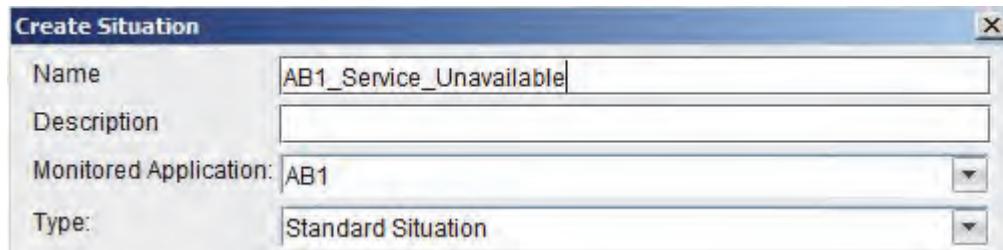
Create a custom situation

13. Create a situation with the following values:

- Association: **Availability**
 - Name: **AB1_Service_Unavailable**
 - Formula: **Status = DOWN**
 - Sampling Interval: **2 min.**
 - State: **Minor**
 - Distribution: **TivE_AB1** (This setting ensures that the situation distributes to any host where your agent is installed.)
- a. Right-click **Availability** under **AB1** in the Navigator and click **Situations**.
The Situation Editor opens.
 - b. Click the **Create new Situation** icon.



- c. Type **AB1_Service_Unavailable** in the **Name** field and click **OK**.

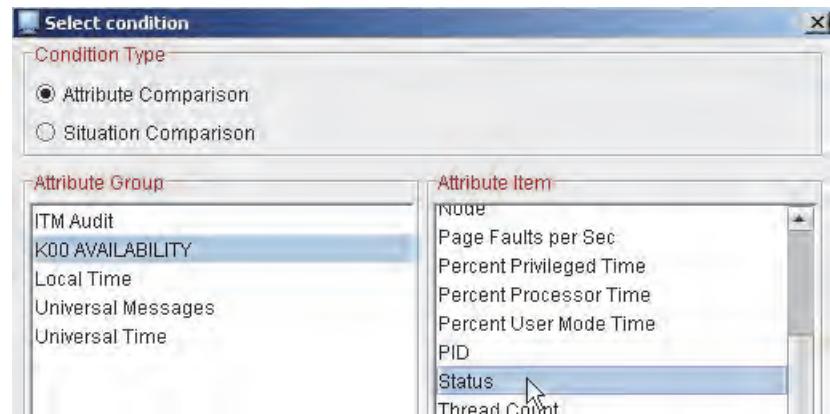


Name	AB1_Service_Unavailable
Description	
Monitored Application:	AB1
Type:	Standard Situation

The **Select condition** window opens.

- d. Ensure that **K00AVAILABILITY** is selected in the **Attribute Group** window.

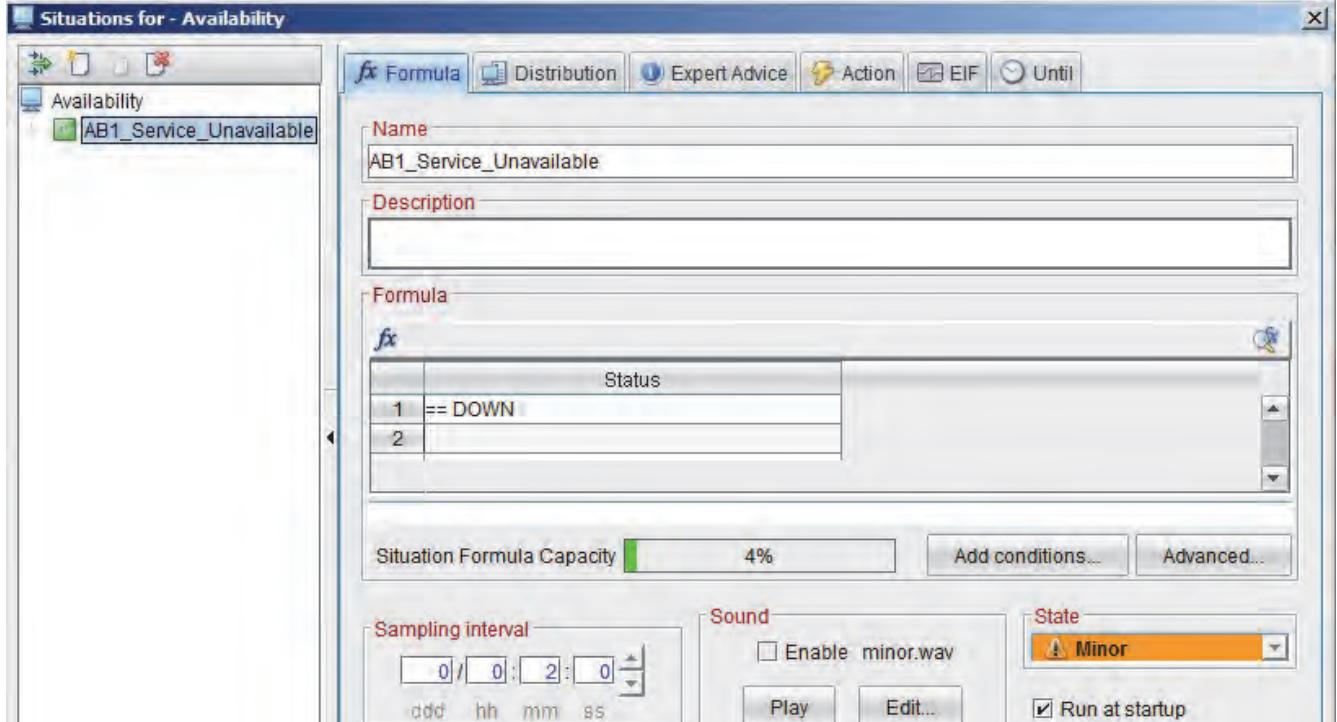
- e. Select **Status** from the **Attribute Item** window and click **OK**.



You are returned to the Situations for window with your new situation open for editing.

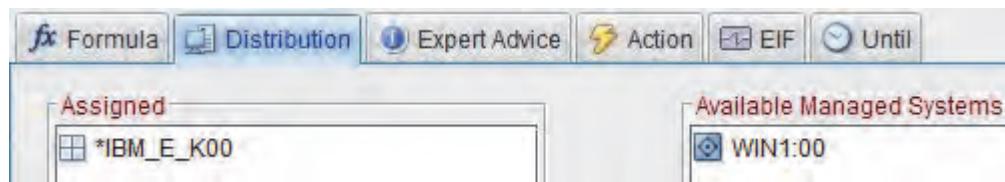
- f. Click the first field under **Status** and select **Down** from the menu.
g. Change the **Sampling interval** to **2 minutes**.
h. Select **Minor** from the **State** menu.

Your situation looks like the following example.



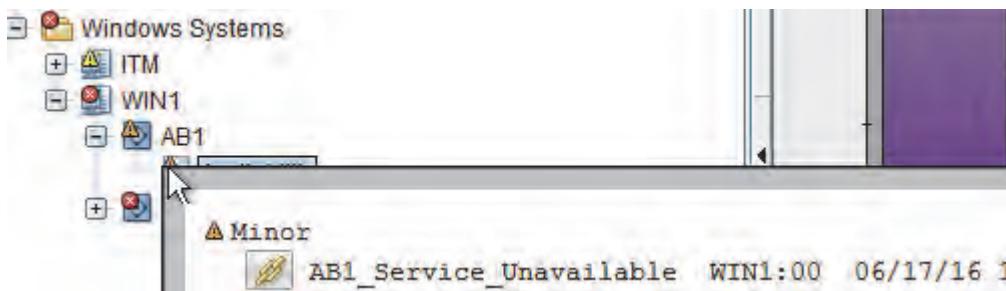
- i. Click the **Distribution** tab.

- j. Remove **WIN1** from the **Assigned** column and move **IBME_K00** to the **Assigned** column.



- k. Click **OK** to save your changes and close the Situation Editor.

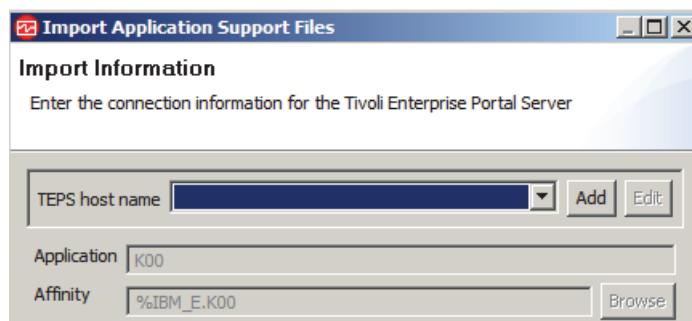
Soon after completing the situation, you receive an event notification because two of the DB2 services are down by default.



Import your custom workspace and situation into your agent

14. In Agent Builder on WIN1, right-click the **Agent 1** project in the Project Explorer and select **IBM Corporation > Import Application Support Files**.

The Import Information window opens.



15. Click **Add** to the right of the **TEPS hostname** field.

The Create Connection Wizard window opens.

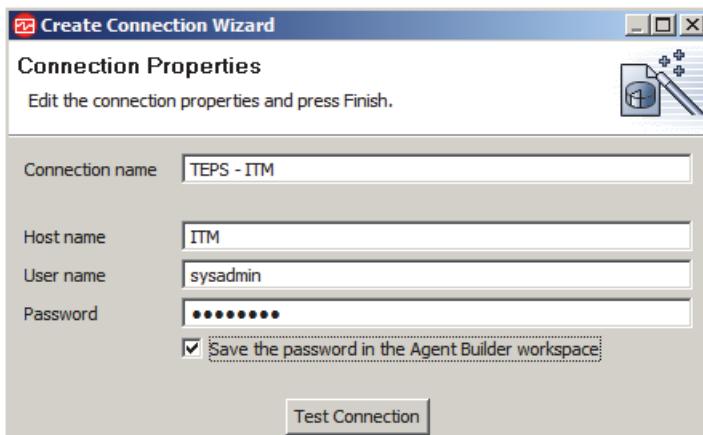


16. Select **Tivoli Enterprise Portal Server Managed System** and click **Next**.

The Connection Properties window opens.

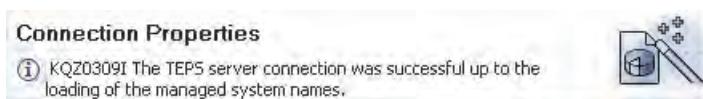
17. Set the following property values:

- Host name: **ITM**
- User name: **sysadmin**
- Password: **object00**
- Save the password: **Selected**



18. Click **Test Connection** to test the connection properties.

19. Confirm that you have a successful connection.

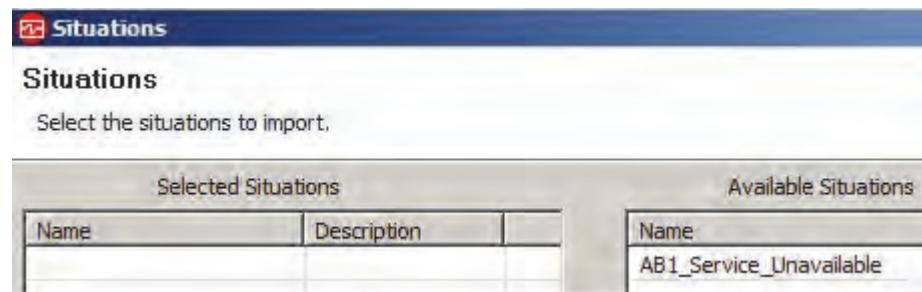


20. Click **Finish** to save your changes and close the wizard.

You are returned to the Import Information window with ITM in the **TEPS host name** field.

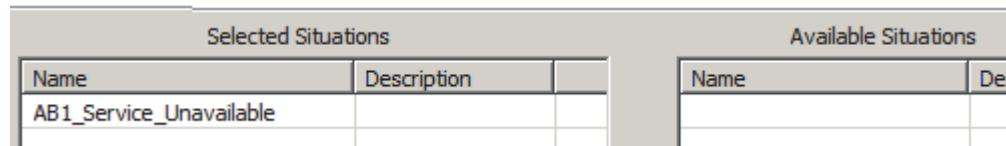
21. Click **Finish** to start importing application support from the ITM server.

The Situations window opens.



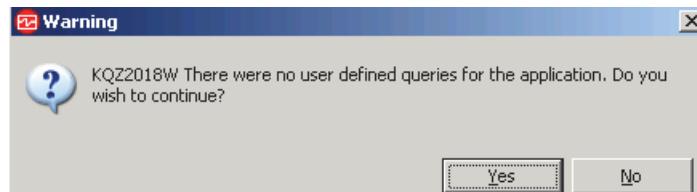
Notice the AB1_Service_Unavailable situation in the Available Situations column.

22. Move the **AB1_Service_Unavailable** situation into the Selected Situations column.



23. Click **OK**.

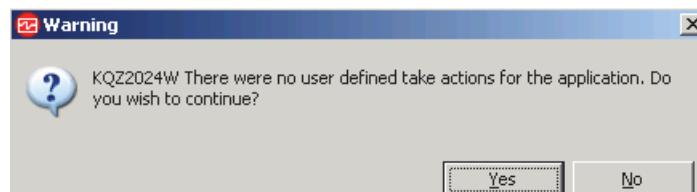
A warning message is displayed.



This warning is expected because you did not create any queries.

24. Click **Yes**.

Another warning message opens.

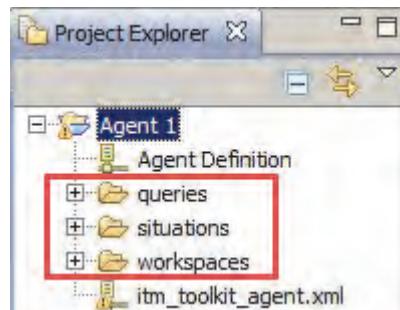


This warning is also expected because you did not create any new Take Actions.

25. Click **Yes**.

After a progress bar shows that the export is completed, you are returned to the Agent Builder interface.

You now see three new directories in the Project Explorer under **Agent 1**.



Note: After you import application support files, modifying the company identifier, agent identifier, or product code will invalidate any imported application support, and you must re-create them.

26. If necessary, save your agent project.

Exercise 3 Install and confirm IBM Tivoli Monitoring application support

In this exercise, you reinstall your AB1 agent and the custom application support that you added.

Remove your custom workspace and situation

To properly test your application support, you install the agent and application support on systems that do not already contain the items that are installed. For this exercise, to create a blank system on which to test your agent, you undo the work that you have done. You delete the custom workspaces and situation that you created on the ITM server. You uninstall your agent. After that, you reinstall your agent and confirm that it creates the application support components that you imported into the agent in Agent Builder.

1. On ITM in the Tivoli Enterprise Portal, if you exited the TEP Admin mode, reenter the **TEP Admin mode**. See the previous exercise, [“Publish your workspace”](#) on page 3-20 for detailed steps.
2. Delete the **Services** workspace that is attached to **AB1 > Availability**.
 - a. Open the **Services** workspace on the **Availability** node.
 - b. Select **File > Delete Workspace**.
 - c. Click **Yes** to confirm the deletion.

The Services workspace that you published is deleted, but the personal workspace you created originally still exists. In [Step 5](#) on page 3-28, you delete your personal Services workspace outside of the TEP Admin mode.

3. Exit the TEP Admin mode in the Tivoli Enterprise Portal.
 - a. In the Tivoli Enterprise Portal, select **Edit > Administer Users**.
 - b. Click **sysadmin** on the **Users** tab.

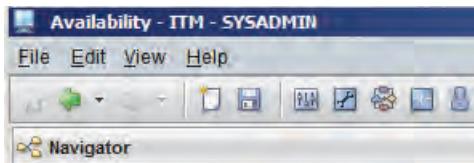


Note: Ensure that you select **sysadmin** and clear **Default**.

- c. Click **Workspace Administration** in the left pane of the **Permissions** tab.
- d. On the right side of the window, clear the **Workspace Administration Mode** check box and click **OK**.



If you followed the steps correctly, ***ADMIN MODE*** is no longer displayed in the desktop title bar.



The workspace name in the upper left is now called Availability.

4. Open the personal **Services** workspace by completing one of the following steps in the Tivoli Enterprise Portal Navigator:
 - Right-click **Availability** and select **Workspace > Services**.
 - Click any node to move away from **Availability** and then click **Availability** to return.

The personal Services workspace opens.

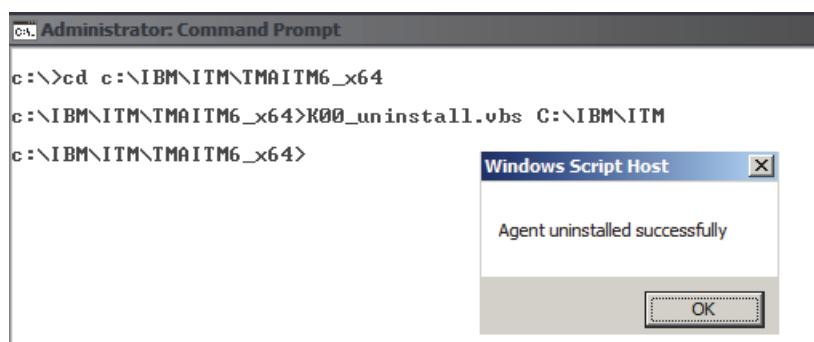
5. Delete the personal **Services** workspace that is attached to **AB1 > Availability**.
 - a. Select **File > Delete Workspace**.
 - b. Click **Yes** to confirm the deletion.

The personal Services workspace that you created is deleted.

6. Delete the **AB1_Service_Unavailable** situation.
 - a. Open the **Situation Editor**.
 - b. Locate and click your **AB1_Service_Unavailable** situation under **AB1** in the Navigator.
 - c. Right-click the situation and click **Delete Situation**.
 - d. Click **Yes** to confirm the deletion.
 - e. Click **OK** to exit the Situation Editor.

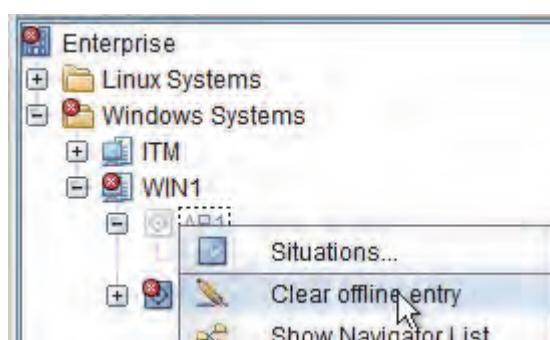
7. Uninstall the AB1 agent on WIN1 by running the following commands from a command prompt:

```
cd C:\IBM\ITM\TMAITM6_x64  
k00_uninstall.vbs C:\IBM\ITM
```



8. Wait until you receive a prompt that indicates the agent was successfully uninstalled.
9. Click **OK** to close the prompt.

10. In Tivoli Enterprise Portal on ITM, right-click **AB1** and click **Clear offline entry**.



The agent and its application support are removed.

11. Exit the **TEP client**. To avoid display problems in Tivoli Enterprise Portal, close the TEP client during an application support installation.

Create the agent installation scripts

12. In Agent Builder on WIN1, with the **AB1** agent open, select **Agent Editor > Generate Agent** from the main menu.

13. Clear the **Install the Agent** option.

14. Ensure that **Create a ZIP file** and **Create a TAR file** are checked.

Notice the directory where the files are created. You go to this directory to access the generated files. Keep the default output directory for this exercise.

15. Click **Finish**.

16. Click **Yes** to replace the K00 image.

17. Click **Yes** again to overwrite the files.

A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated.

18. Click **OK** to close the confirmation window.

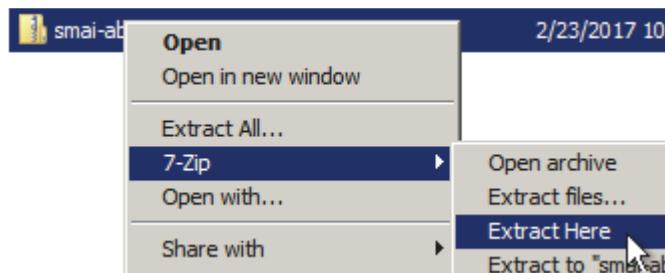


Note: Keep Agent Builder open; you use it again later.

19. Delete the contents of the **C:\share\K00** directory.

20. Copy the new **C:\Users\Administor\smai-ab1-01.00.00.00.zip** file into **C:\share\K00** directory.

21. Extract the files from **smai-ab1-01.00.00.00.zip** into **C:\share\K00** with **7-Zip**.



Reinstall the AB1 application support on ITM

In this section, you reinstall the updated AB1 application support onto the ITM server.

22. If needed, re-create the drive map to **\WIN1\share**.

23. Open a command prompt and change to the **Y:\K00** directory.

To install the IBM Tivoli Monitoring application support without installing the agent, run the `installIraAgentTEMS` and `installIraAgentTEPS` scripts.

24. Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00
installIraAgentTEPS.bat C:\IBM\ITM
```

```
Y:\>cd K00
Y:\K00>installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00
Validating user...
KUIC00007I: User sysadmin logged into server on https://ITM:3661.
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.

Install of K00 TEMS Support successful.

Y:\K00>installIraAgentTEPS.bat C:\IBM\ITM
Online help for this agent will not be available until the Help Server is restarted, which also requires restarting the TEPS.

Install of K00 TEPS Support successful.

Y:\K00>
```

The full installation takes 7 - 10 minutes to complete. Status information is displayed for each command.

You completed the reinstallation of the AB1 agent (K00) agent and application support on the ITM server.

Reinstall the agent on WIN1 with the script installers

In this section, you reinstall the updated AB1 on WIN1 with the script installers.

25. Open a command prompt and run the following commands:

```
cd \share\K00
installIraAgent.bat C:\IBM\ITM

C:\Users\Administrator>cd \share\K00
C:\share\K00>installIraAgent.bat C:\IBM\ITM
Installing agent into C:\IBM\ITM
Installing K00 ....
Install of K00 Agent successful.

C:\share\K00>_
```

26. Reconfigure the agent.

- Locate the new agent in the MTMS utility. If necessary, refresh the view.
- Right-click the agent and select **Configure using Defaults**.

27. Start the agent.

You completed the reinstallation of the AB1 agent (K00) agent on the WNI1 server.

Confirm the installation of agent and application support components

28. On the ITM server, your Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server must be running. If they are not, start them now.



Note: If you restart either process, processor utilization on the ITM server might spike as the Tivoli Enterprise Monitoring Server starts and Tivoli Enterprise Portal Server connects to the Tivoli Enterprise Monitoring Server. Wait until processor utilization normalizes before starting the TEP client.

29. Restart the TEP client and log in as **sysadmin**.
30. If prompted at any time, click **Navigator update pending** to update the navigator with the new navigator items.
31. Confirm that your **Services** workspace is the default for **Availability** and that your **AB1_Service_Unavailable** situation was added. You also get a situation event for your situation.

Unit 4 Monitoring Windows resources exercises

Exercise 1 Monitor Windows resources

Your company sells products online. The application has a web server front end that pulls information from a DB2 database. Every component for this application runs on 15 different Windows servers. You need to build a custom agent that you can install on each server. The agent monitors the specific software and hardware components of this application.

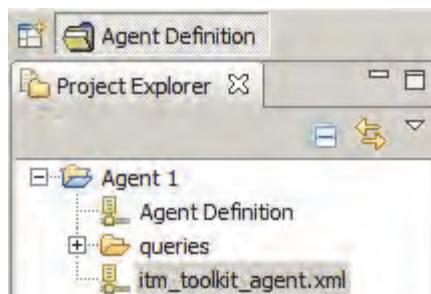
In this exercise, you modify your AB1 agent to monitor the following items:

- The web server that provides the HTML portions of the application
- The DB2 database services and process that provide the database of this application
- The logical disk space where the database is stored
- Windows systems events for events that are related to the HTTP server and DB2

Prepare to edit your agent

In this section, you start Agent Builder and open your AB1 agent.

1. On WIN1, launch Agent Builder if it is not already running.
2. If prompted, keep the default workspace directory and click **OK**.
3. Maximize the Agent Builder application window.
4. If the **AB1** agent is not already open, expand **Agent 1** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.



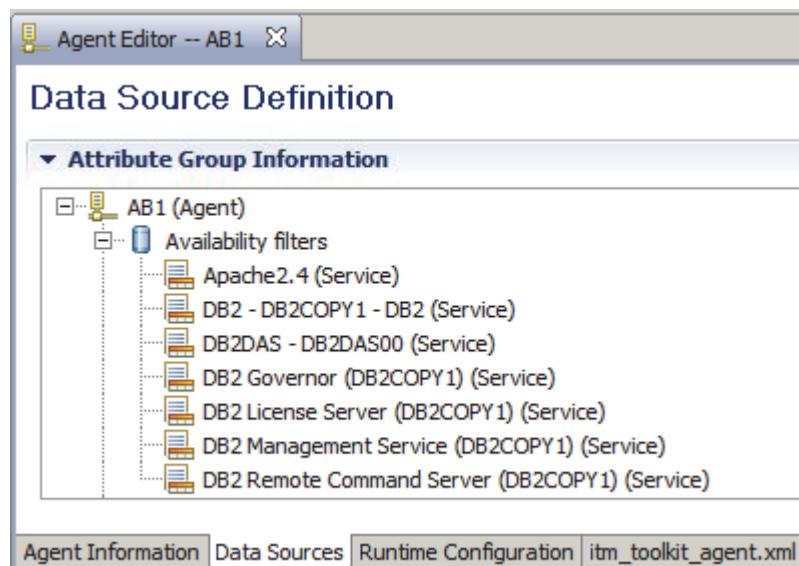
Add WMI process information to the agent

In this section, you do the following tasks:

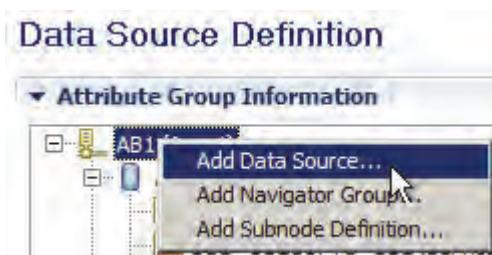
- Add Windows process information from WMI to your agent
- Change the default attribute group name
- Remove several attributes from the group

Complete the following steps:

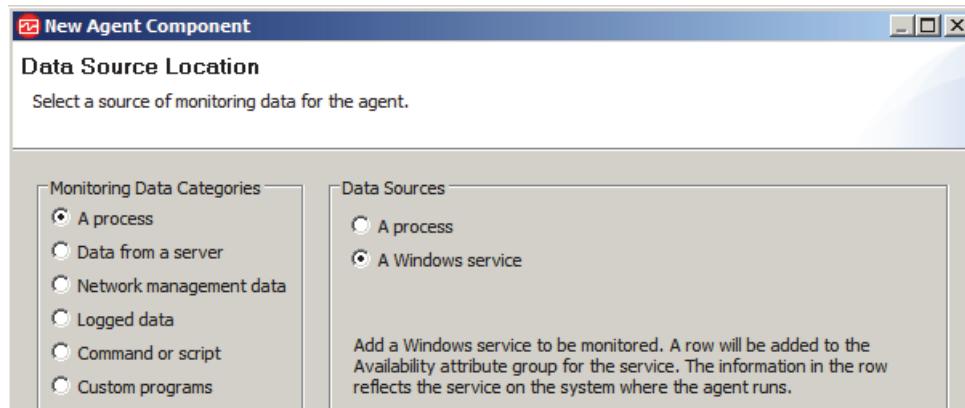
5. Click the **Data Sources** tab.



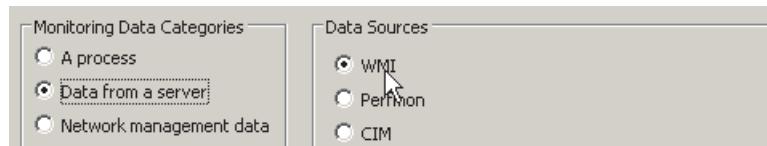
6. Right-click **AB1 (Agent)** under **Attribute Group Information** and click **Add Data Source**.



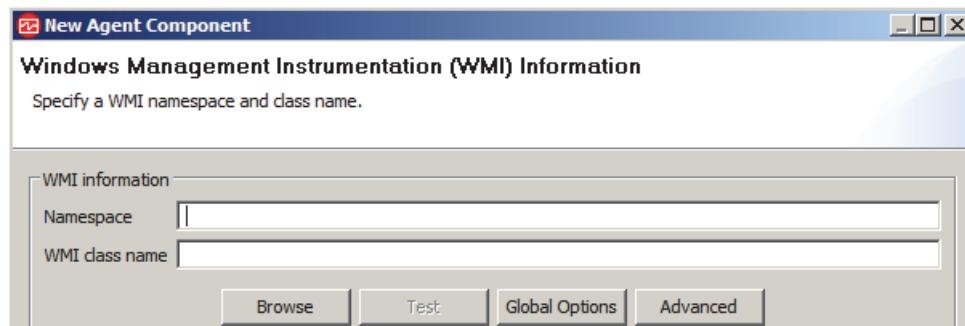
The Data Source Location window opens.



7. Select **Data from a server**.
8. Verify that **WMI** is selected and click **Next**.



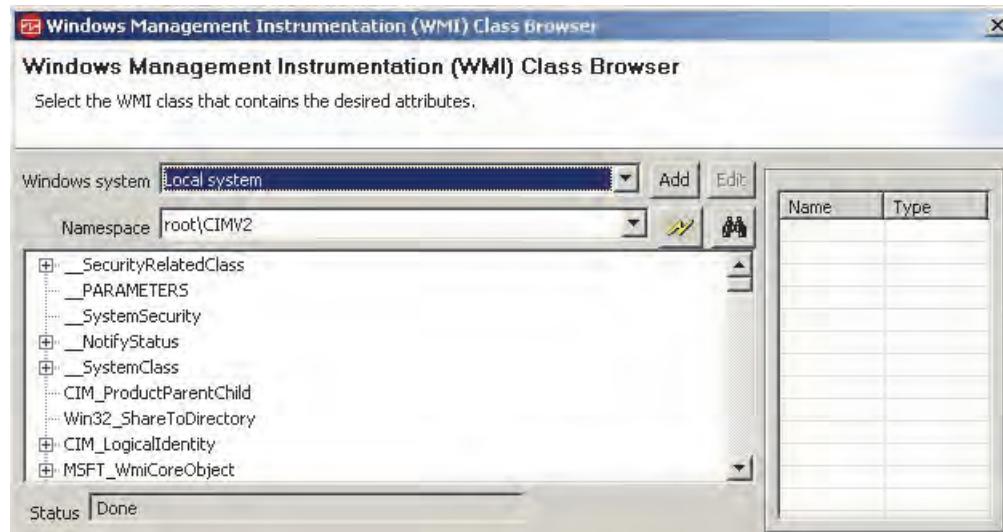
The Windows Management Instrumentation (WMI) Information window opens.



To create a successful agent, you must know the information that needs to be monitored or how to identify the data. For WMI, you must know the namespace and class that provide the data that you want. You can either enter the information manually in the fields that are shown or you can browse a running Windows system for the WMI information.

9. Click **Browse**.

The Windows Management Instrumentation (WMI) Class Browser opens.



10. Verify that **Windows system** is **local system** and **Namespace** is **root/CIMV2**.

11. Click the **Search** icon

12. Enter **Process** in the **Search phrase** field and click **Search**.



13. Locate and select **Win32_Process**.

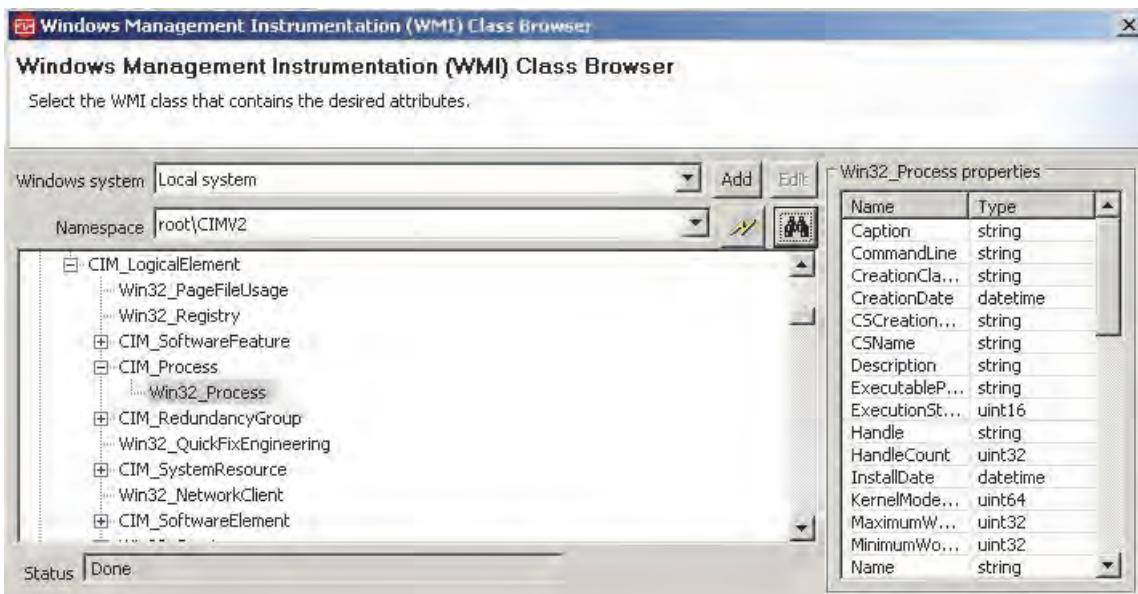
14. Click **OK**.





Important: Be careful when selecting **Win32_Process**. Several items have similar names, such as **Win32_Processor**.

You are returned to the Class Browser window.

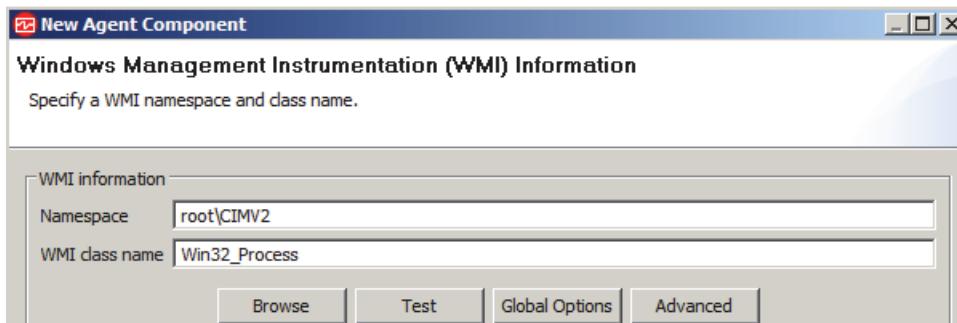


Notice the **Win32_Process** properties on the right. This data can be gathered from this WMI class. Agent Builder creates a table structure, called an **attribute group**, to hold this class and field or column, called an **attribute**, for each of these properties when they are monitored. Later you see where you can edit this attribute list by removing attributes that you do not want the agent to monitor.

15. Click **OK** to select the **Win32_Process** class and all of its properties.

The Class Browser window closes.

The Windows Management Instrumentation (WMI) Information window is displayed with the **Namespace** and **WMI class name** fields completed.

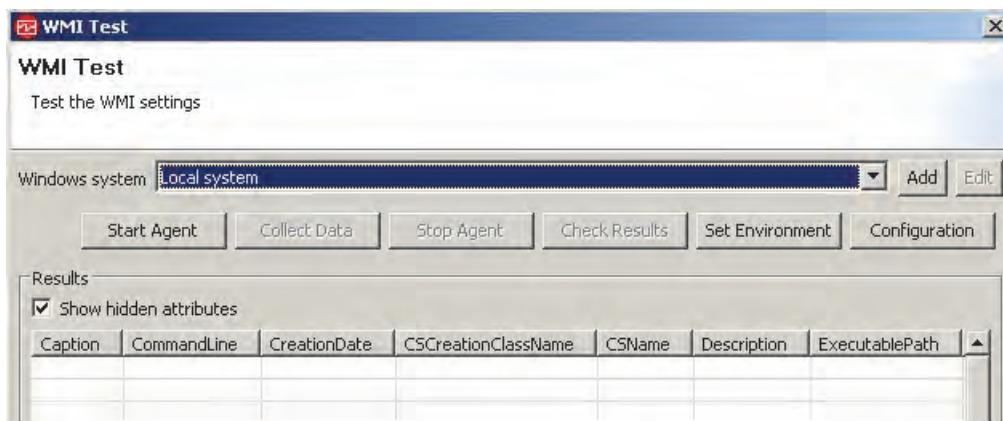


Many data sources have a test feature to confirm that the data source can gather the target data and to configure the attributes.

16. Test this individual attribute group.

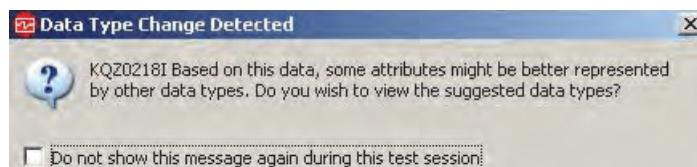
- a. Click **Test**.

The WMI Test window opens.

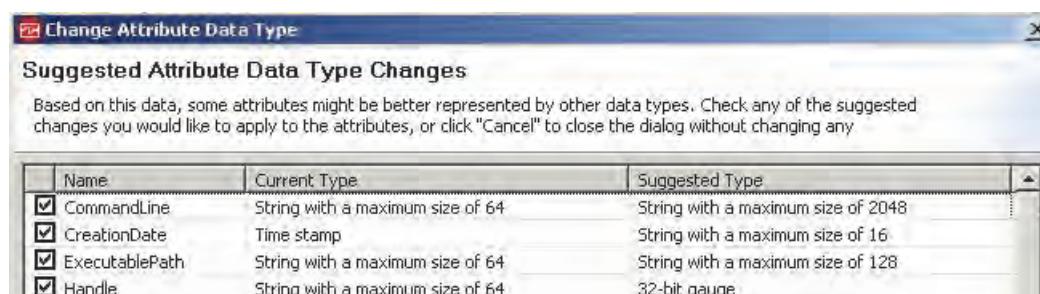


The default of the tester is to gathering data from the local system.

- b. Click **Start Agent** and **Collect Data**.



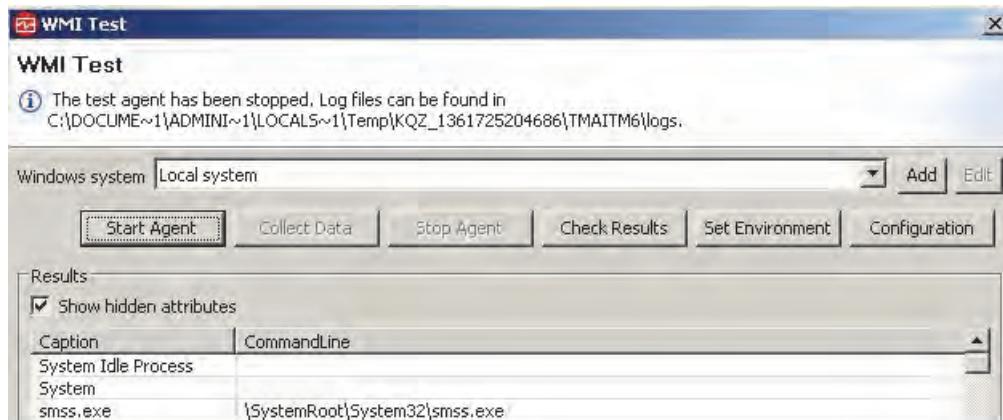
- c. Read the **Data Type Change Detected** window and click **Yes**.



The window suggests several data type changes. Clear individual attributes that you do not want changed. Click **OK** to save all checked changes. Click **Cancel** to make none of the suggested changes.

- d. Click **OK** to apply all of these suggested data type changes.

The WMI Test window reopens with data that is gathered from the local host.



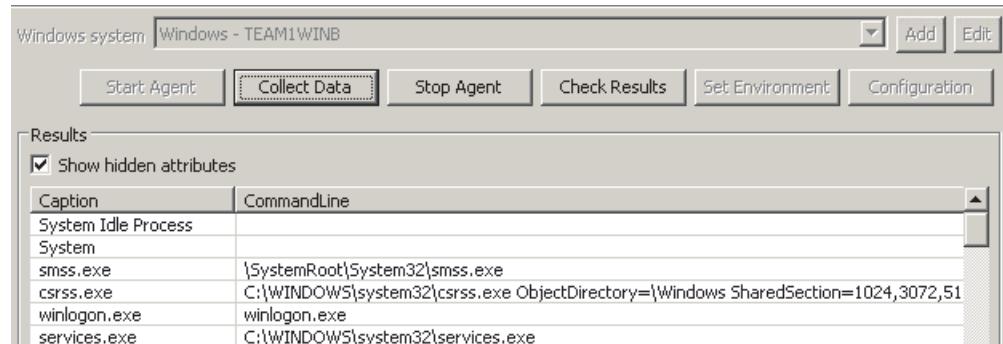
The tester can also gather data from a remote system.

- e. Click **Stop Agent**, if it is available.
- f. Select **Windows - WIN2** from the **Windows systems** menu.



Important: If the WIN2 connection does not exist, return to [Unit 2, Exercise 1, Step 16](#) on page 2-7 for instructions on creating the connection definition to WIN2.

- g. Click **Start Agent** and **Collect Data**.



- h. Confirm that the tester gathers process data from WIN2.

- i. Click **OK** to close the WMI Tester.

You modified your agent to gather the **Win32_Process** data from WMI and tested the individual data source with the tester utility.

17. Click **Finish** to close the **New Agent Component** window.

The **Data Source Definition** tab on the Agent Editor is displayed.

Attribute group information

- AB1 (Agent)
- Availability filters
- Win32_Process (WMI)

WMI Class Information

Attribute group name: Win32_Process

Help: Data gathered from WMI class Win32_Process.

Produces a single data row Can produce more than one data row Produces large number of data rows

Add this attribute group to a reporting category Performance

Agent Information | Data Sources | Runtime Configuration | itm_toolkit_agent.xml

Note: If you modified the agent for IBM Performance Management, you also see the Apache_Status and DB2_Status filter data sources.

With the **Win32_Process** attribute group selected, you can see and edit its WMI class information. The attribute group name acquired the WMI class name. This name is displayed in the Tivoli Enterprise Portal navigator. You can edit it without affecting the data that it retrieves.

18. Change **Win32_Process** to **My_App_Processes**.

- Click **Win32_Process** in the **Attribute Group Information** navigator.
- In the **WMI Class Information** section, replace the attribute group name **Win32_Process** with **My_App_Processes**. Press Enter to activate the change.

Attribute group information

- AB1 (Agent)
- Availability filters
- My_App_Processes (WMI)

WMI Class Information

Attribute group name: My_App_Processes

Help: Data gathered from WMI class Win32_Process.

The attribute group name changes in the upper display also.

19. Click the plus sign (+) next to **My_App_Processes (WMI)** to expand the attribute group and list its attributes.



Notice that the attribute names are based on the WMI names that you saw when you searched for **Process** in the WMI browser. In this window, you can remove unwanted attributes from this attribute group. You can also add attribute groups, remove attribute groups, and edit attributes and attribute groups.

20. Click an attribute, such as **CreationClassName**.

WMI Attribute Information

Attribute name	CreationClassName
Help	Attribute CreationClassName from /
<input type="checkbox"/> Hidden - can only be used in derived attribute	

You can edit the attribute metadata, such as the attribute name, the maximum size, whether the data is hidden, and whether it is a key attribute.

21. Remove the following attributes:

- CreationClassName
- OSCreationClassName
- Quota* (Total of 4)
 - a. Scroll down the list and select one or more attributes.



Hint: Press Ctrl and click to select more than one attribute at a time.

- b. Click **Remove**.
- c. Repeat until you remove all six attributes.

22. Select **File > Save** or click the **Save** icon to save your agent project.



You modified **AB1** to gather Windows process information from WMI. Continue to the next section without closing Agent Builder.

Add combined disk information to the agent

In this section, you do the following tasks:

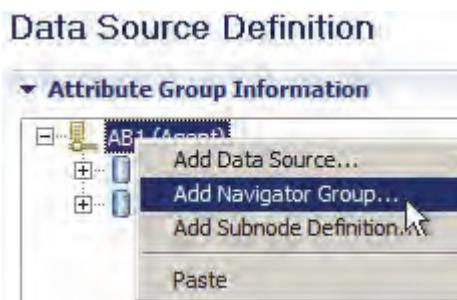
- Add logical disk information from both Perfmon and WMI to your agent
- Create a Navigator node that is called **Disk_Storage** and place both attribute groups under it
- Combine the logical disk attribute groups from Perfmon and WMI into one new attribute group that merges both sets of data

Complete the following steps:

23. From the **Data Source** tab, create a navigator group that is called **Disk_Storage** and add the Perfmon logical disk attributes.
- a. Right-click **AB1 (Agent)** under **Data Sources** and click **Add Navigator Group**.



Hint: Collapsing all objects in the Attribute Group Information navigator might help in locating **AB1**.



The Navigator Group Information window opens.



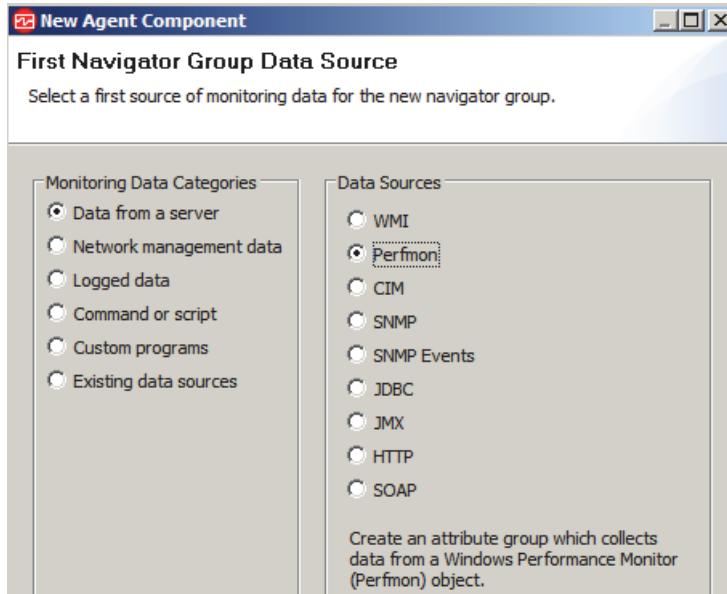
- b. Enter the following information:
- ◆ Navigator group name: **Disk_Storage**
 - ◆ Help: **Logical disk storage information about AB1**

A screenshot of the 'Navigator Group Information' dialog box. The 'Navigator group name' field contains 'Disk_Storage' and the 'Help' field contains 'Logical disk storage information about AB1'.

- c. Click **Next**.

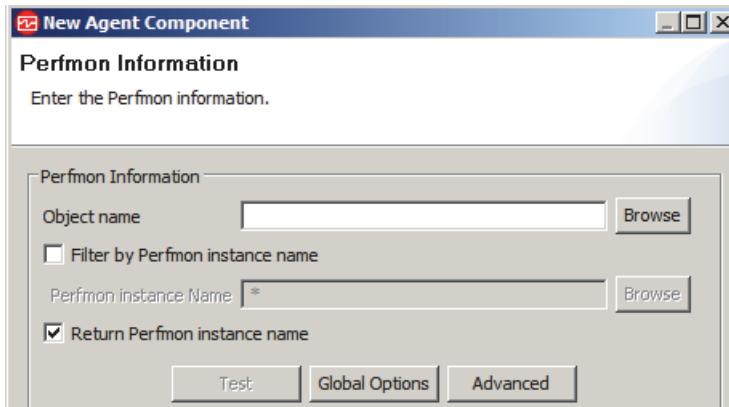
The First Navigator Group Data Source window opens. You now add the Perfmon LogicalDisk attributes.

- d. Select **Data from a server** under **Monitoring Data Categories** and select **Perfmon** under **Data Sources**.



- e. Click **Next**.

The Perfmon Information window opens.



- f. Click **Browse**.

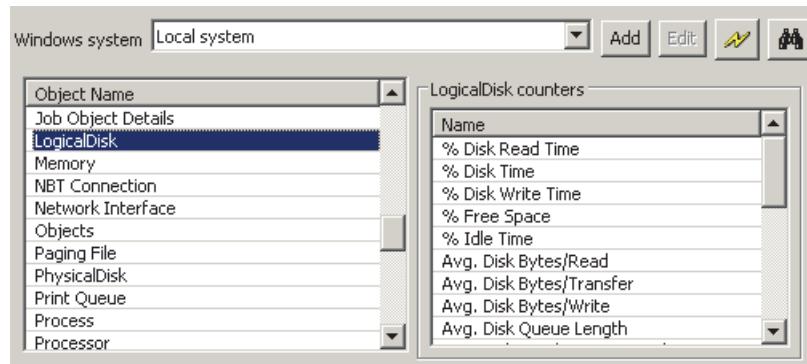
The Performance Monitor (Perfmon) Object Browser opens and displays Perfmon data.



As with the WMI browser, you can connect to remote Windows hosts and browse the Perfmon database. However, you do not browse a remote host in this exercise.

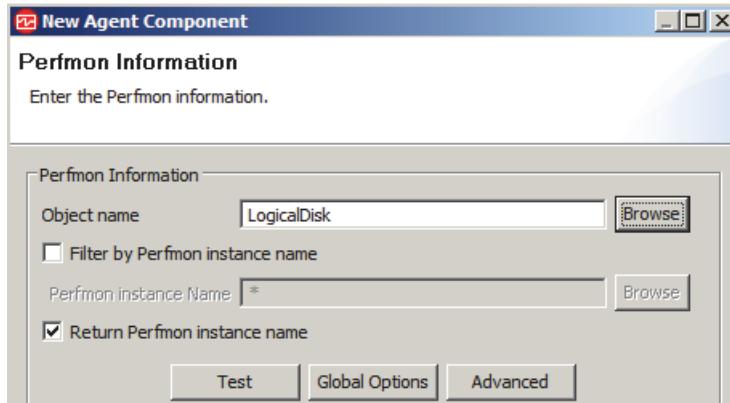
- g. If **Windows - WIN2** is still selected in the **Windows system** field, select **Local system**.

- h. Locate and click **LogicalDisk** from the **Object Name** column.



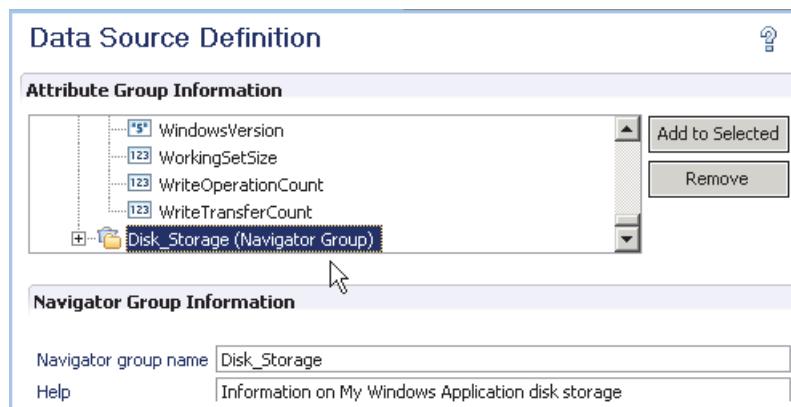
Notice all the counters that are contained in this object.

- i. Click **OK** to identify the LogicalDisk object and all of its counters the agent monitors.
The Perfmon Information window opens.



- j. Click **Finish**.

The Agent Editor **Data Source Definition** tab is displayed.



k. Expand **Disk_Storage (Navigator Group)**.



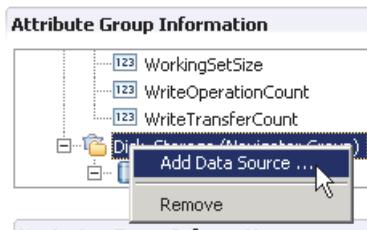
l. Expand **Logical Disk (PerfMon)**.



Now you can see the Navigator Group, the Attribute Group, and the Attributes that the agent gathers and displays in the management console. From here, you can add and remove attributes and attribute groups. You can also edit any of these new components. For this exercise, you keep each of the Perfmon components as they are.

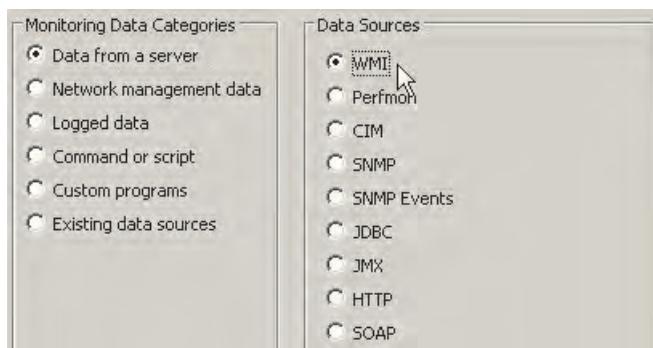
24. Add the **WMI Win32_LogicalDisk** attributes to the **Disk_Storage Navigator Group**.

a. Right-click **Disk_Storage (Navigator Group)** and click **Add Data Source**.



The Navigator Group Data Source window opens.

b. Verify that **Data from a server** under **Monitoring Data Categories** and **WMI** under **Data Sources** are selected.

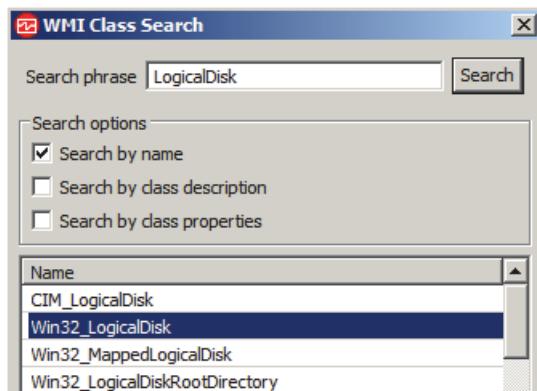


c. Click **Next**.

The Windows Management Instrumentation (WMI) Information window opens.

d. Click **Browse**.

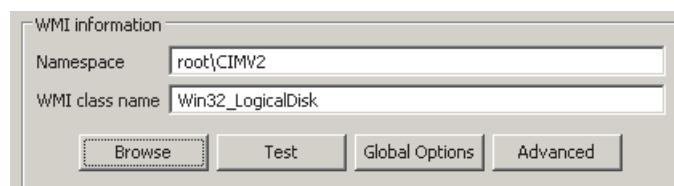
- e. Verify that **Windows system** is **Local system** and **Namespace** is **root\CIMV2**.
- f. Click the **Search** icon (binoculars).
- g. Enter **LogicalDisk** in the **Search phrase** field and click **Search**.
- h. Click **Win32_LogicalDisk** and click **OK**.



The Class Browser window opens.

25. Click **OK** to select the **Win32_LogicalDisk** class and all of its properties to be monitored by this agent.

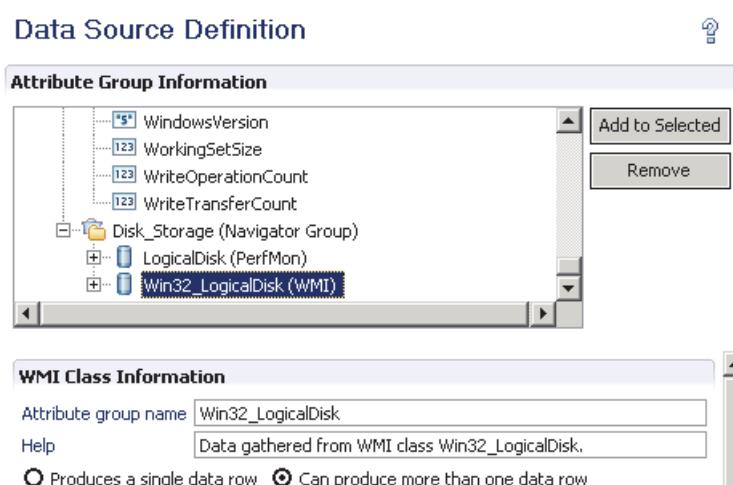
The Class Browser window closes and the Windows Management Instrumentation (WMI) Information window is displayed with the **Namespace** and **WMI class name** fields completed.



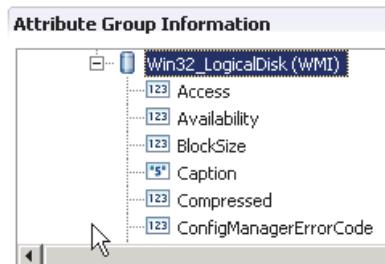
You modified your agent to gather the **Win32_LogicalDisk** data from WMI.

26. Click **Finish**.

The Agent Editor **Data Source Definition** tab is displayed.

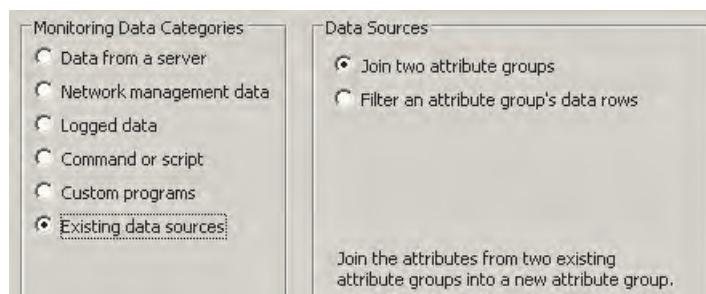


27. Expand **Win32_LogicalDisk (WMI)**.



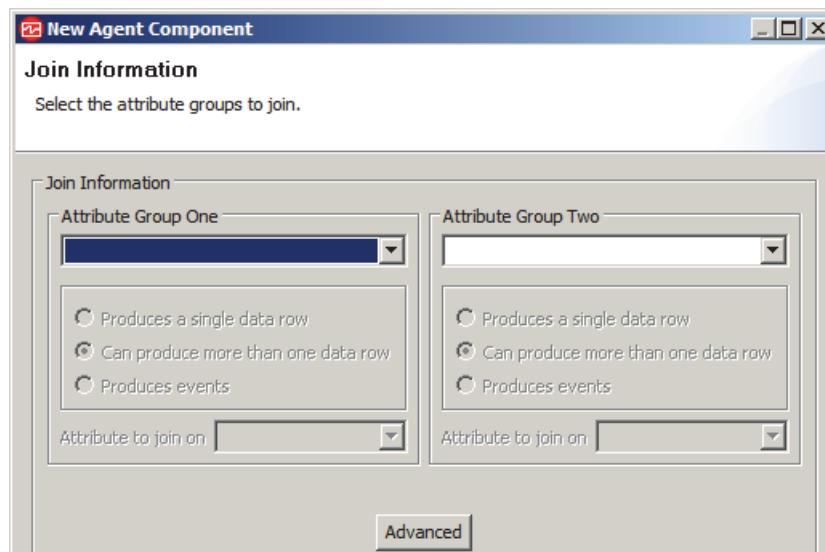
28. Combine the **LogicalDisk** data from Perfmon and the **Win32_LogicalDisk** data from WMI into a new attribute group called **LogicalDiskCombined**.

- Right-click **Disk_Storage (Navigator Group)** and click **Add Data Source**.
- Select **Existing data sources** under **Monitoring Data Categories** and ensure that **Join two attribute groups** is selected.

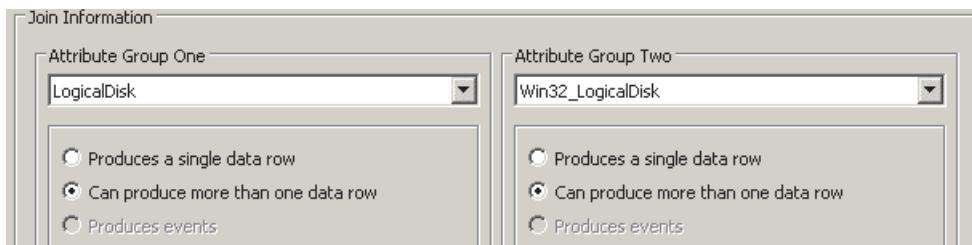


- Click **Next**.

The Join Information window opens.



- d. Select **LogicalDisk** for **Attribute Group One** and **Win32_LogicalDisk** for **Attribute Group Two**.



- e. Browse the attribute names under both **Attribute to join on** menus that look for common attributes. Notice that both have the attribute **Name**.

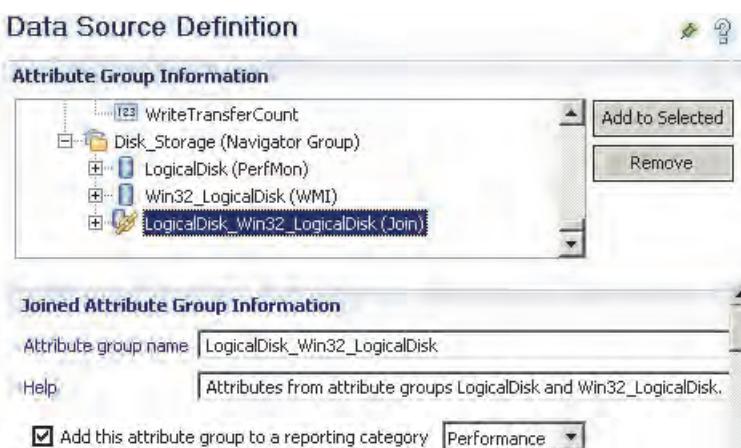
To join two data sources, you must ensure that all attribute names are unique across both data sources. You change the name of one of these attributes later in this exercise.

- f. Select **Name** as the attribute to join on for both attribute groups.



- g. Click **Finish** to create the attribute group.

The **Data Source Definition** tab of the Agent Editor is displayed.



- h. Change the attribute group name **LogicalDisk_Win32_LogicalDisk** to **LogicalDiskCombined**.

Data Source Definition

Attribute Group Information

	SupportsDiskQuotas
	SupportsFileBasedCompression
*	SystemCreationClassName
*	SystemName
123	VolumeDirty
*	VolumeName
*	VolumeSerialNumber
+	LogicalDiskCombined (Join)

Join Attribute Group Information

Attribute group name:	LogicalDiskCombined
Help:	Attributes from attribute groups LogicalDisk and Win32_LogicalDisk.

- Expand **LogicalDiskCombined** and notice that it has the combined attributes.

Notice that **Name** attribute from the **LogicalDisk** attribute group is changed to **Name_LogicalDisk**.



Agent Builder does this change automatically because an attribute group cannot have two attributes of the same name. This change did not occur because you selected the two attributes for the join, but because the two attributes have the same value: **Name**. This change occurs automatically in joins for all attributes with the same attribute name.

- Save your agent project.

You successfully added the disk storage information from Perfmon and WMI to your agent. To complete this agent, you must add the Windows log events for DB2 and the HTTP Server.

Add DB2 Windows log event information to the agent

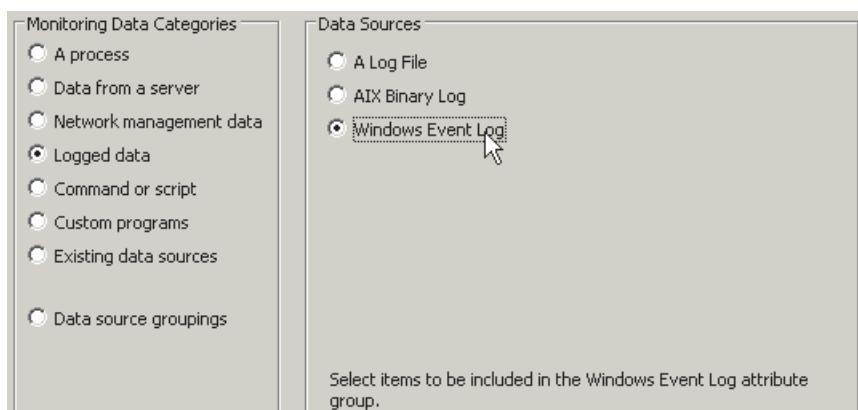
Complete the following steps to add Windows Log Event information that is filtered for DB2.

- Click **AB1 (Agent)** under **Attribute Group Information** and click **Add to Selected**.



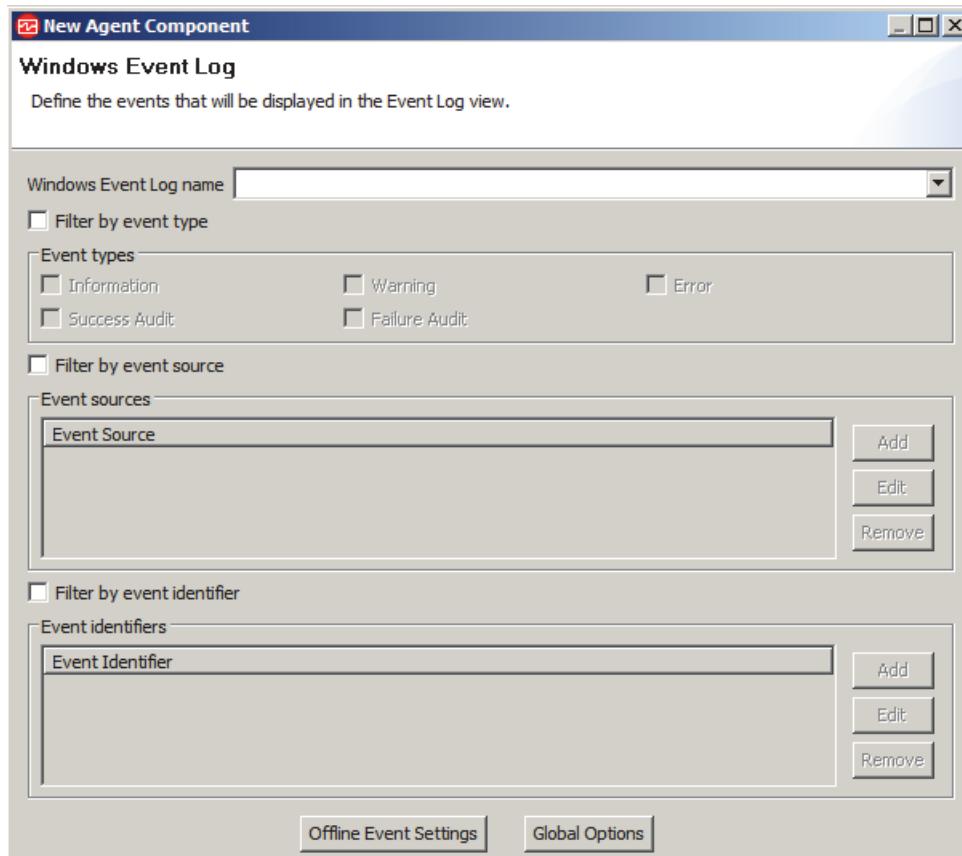
The Data Source Location window opens.

- Click **Logged Data** under **Monitoring Data Categories** and click **Windows Event Log** under **Data Sources**.



32. Click **Next**.

The Windows Event Log window opens.



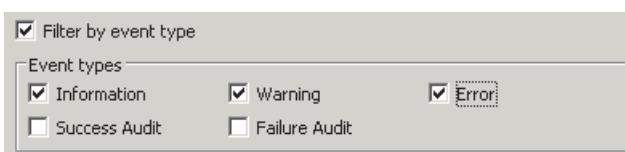
33. Create a filter that reports DB2 Information, Warning, and Error events from the Application log.

- a. Select **Application** from the **Windows Event Log name** list.



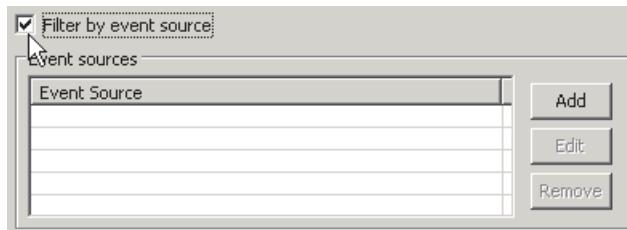
You must specify which event types the agent retrieves. Limit yours to Information, Warning, and Error.

- b. Select **Filter by event type** and select **Information, Warning, and Error**.



Your agent returns all Information, Warning, and Error events from the Application log. Now you need to limit the events to just DB2.

c. Select **Filter by event source**.



From this pane, you can add one or more event sources.

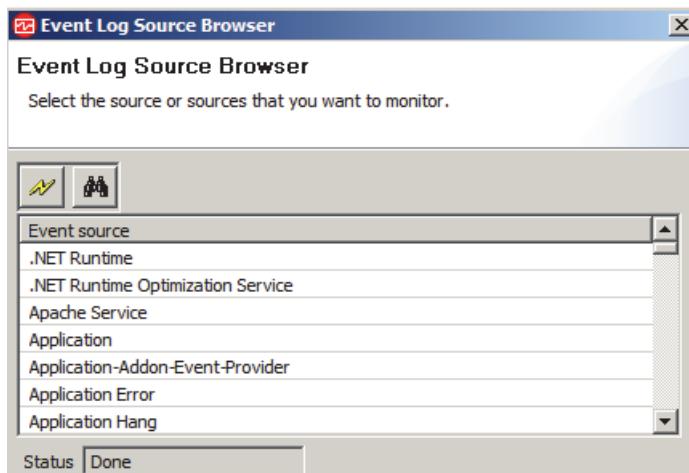
d. Click **Add**.

The Event Source window opens.



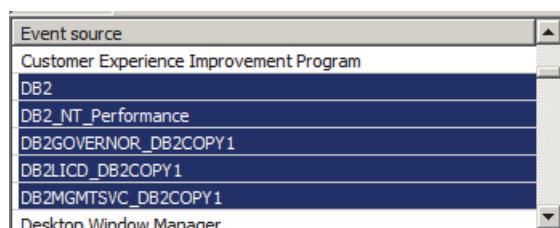
e. Click **Browse**.

The Event Log Source Browser opens with a list of event sources that are found on this host.



There is no connection to remote hosts with this browser. You can select one or more event sources.

f. Select all event sources that start with **DB2**.

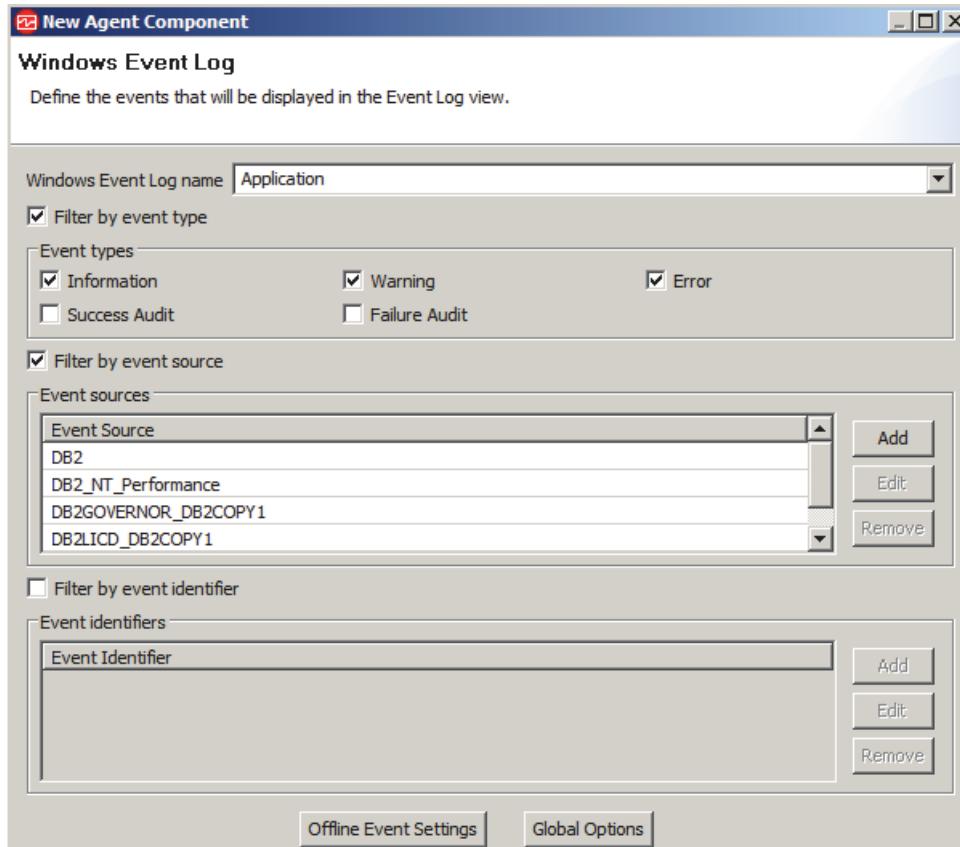




Hint: Press Shift and click to select contiguous items. Press CTRL and click to select noncontiguous items.

- g. Click **OK** to save your event source selections.

Your final configuration should look like the following example.



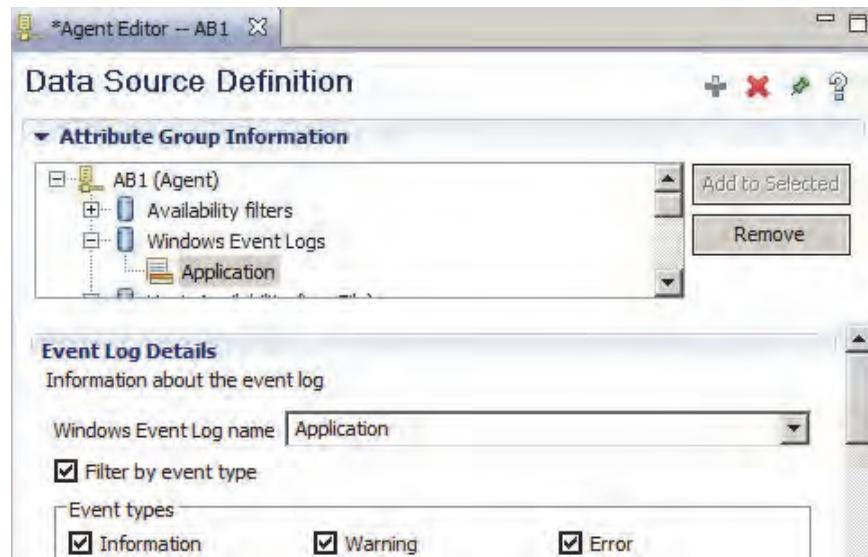
34. Configure the data source to collect offline events.

- a. Click **Offline Event Settings**.



- ◆ **Do not collect any offline events:** Events that are generated while the agent is shut down are not processed. This option is the default option.

- ◆ **Collect all offline objects:** All events that are generated while the agent is shut down are processed.
 - ◆ **Specify custom collection settings:** You can enter a value to throttle the processing of old events that are based on a time value, or number of events, or both. By using this option, you ensure that the management server is not overloaded with events when the agent starts.
- b. Click **Collect all offline events**.
 - c. Click **OK** to save your changes and close the bookmark settings window.
35. Click **Finish** to close the Windows Event Log window.
The Agent Editor **Data Source** tab is displayed.
36. Locate and expand the **Windows Event Logs** attribute group and confirm the single Application attribute.



37. Save your agent project.

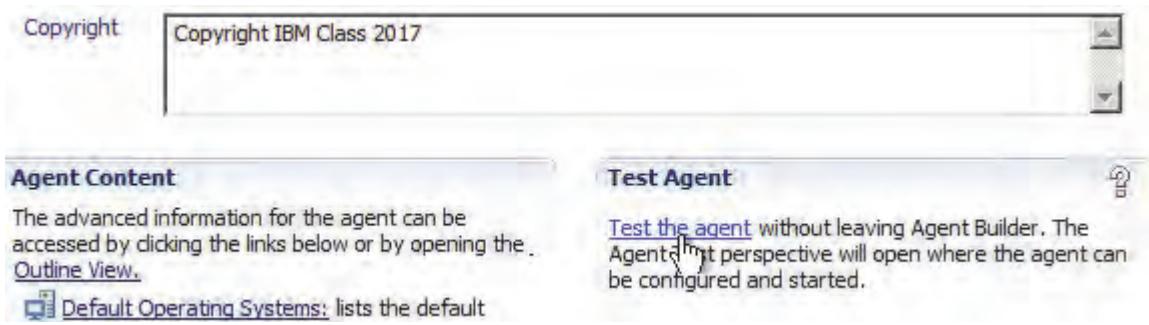
You added Windows Event Log filters for DB2 and HTTP server events to your AB1 agent. You also completed all the agent modifications that are part of this scenario. You now install and test your agent.

Test the full agent in Agent Builder

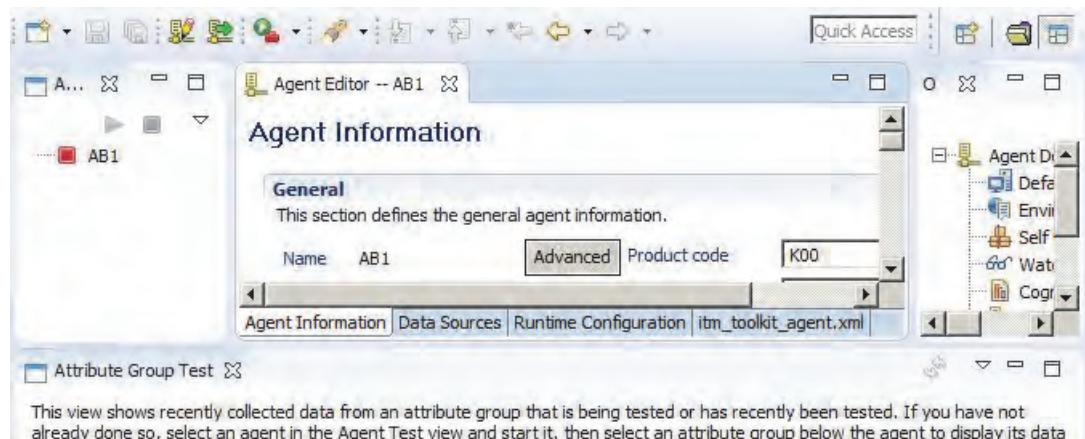
In this section, you test your full agent while in Agent Builder and confirm that all data sources retrieve data.

38. Click the **Agent Information** tab.

39. Click **Test the agent**.



The Agent Test perspective opens.



The Agent Test perspective replaces the Project Explorer window with the Agent Test window and the Problems window with the Attribute Group Test window. It keeps the Agent Editor and Outline windows.

40. Move between the two perspectives.

- Click the **Agent Definition** perspective button on the upper right to return to that perspective.



The Project Explorer and Problems windows replace the Agent Test and Attribute Group Test windows.

- Click the **Agent Test** perspective button on the upper right to return to that perspective.



The Agent Test and Attribute Group Test windows reopen. You can keep the Agent Test perspective open while you edit the agent definition.

- c. With the Agent Test perspective open, click the **Data Sources** tab at the bottom of the **Agent Editor** window.



The Data Sources page opens in the Agent Editor, while the Agent Test and Attribute Group Test windows remain visible. In this way, you can edit the agent definition while keeping the Agent Test and Attribute Group Test windows visible. This perspective can be helpful when defining agent components based on data gathered by the agent.

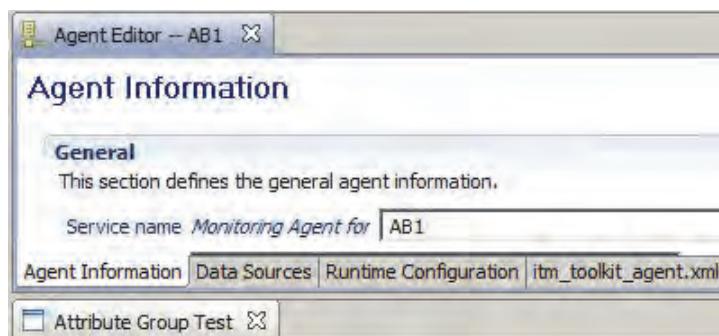
If you modify the Agent Test perspective, such as closing the Agent Test or Attribute Group Test windows, you can restore the perspective to its default layout.

41. Modify and reset the Agent Test perspective.

- a. Click the close icon on the Agent Test window.

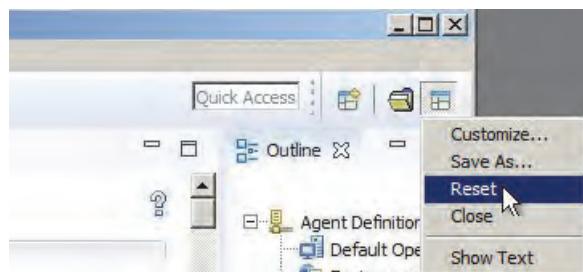


The Agent Test window closes.



In this step, you modified the Agent Test perspective by removing the Agent Test window. Any time that you open this perspective, it will not include the Agent Test window.

- b. Right-click the **Agent Test perspective** button and click **Reset**.

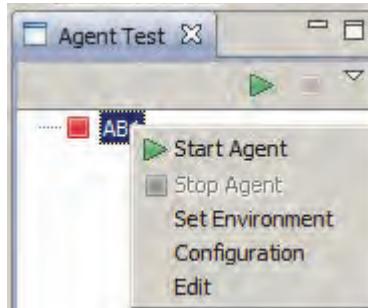


- c. Click **Yes** to confirm resetting the perspective.

The perspective is reset to its default layout and the Agent Test window reopens. Now any time you open this perspective, it will be in its default configuration.

Return to testing your data sources.

42. Right-click **AB1** in the Agent Test window and browse the menu options.



From this menu, you can do the following actions:

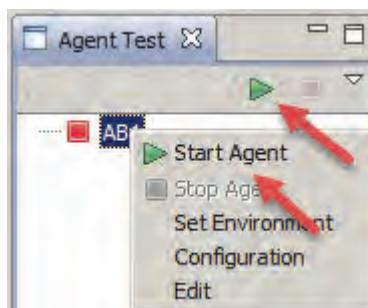
- Set environment variables the agent needs.
- Configure the agent and enter any needed runtime properties.
- Edit the agent.
- Start the agent and test all data sources.



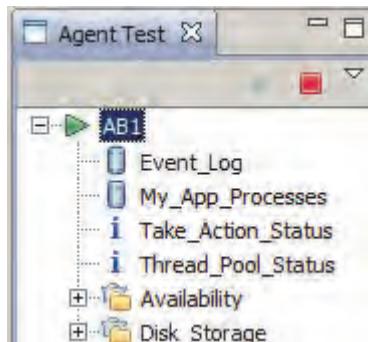
Note: Edits that are made to a running agent are not visible until the agent is restarted.

This agent does not need any environment variables or configuration properties.

43. Start the agent by either clicking **Start Agent** from the menu or the **Start Agent icon** (



A window indicates that the agent is starting. When the agent starts, its attribute groups are shown as child processes of the agent in the Agent Test view.



The attribute group icon (⌚) the attribute groups.

The status attribute groups that give information about the agent (Performance Object Status, Thread Pool Status, and Take Action Status) are also shown as child processes of the agent in the Agent Test view. The information icon indicates the status attribute groups.

The stop agent icon becomes available when the agent is started.

If your agent has subnodes or navigator groups, they are shown as nodes in the Agent Test view. Subnode definitions are shown under the agent. A subnode instance node is shown under the subnode definition node. Attribute groups and navigator groups are shown under the subnode instance node.



Hint: You can start and run more than one agent at the same time.

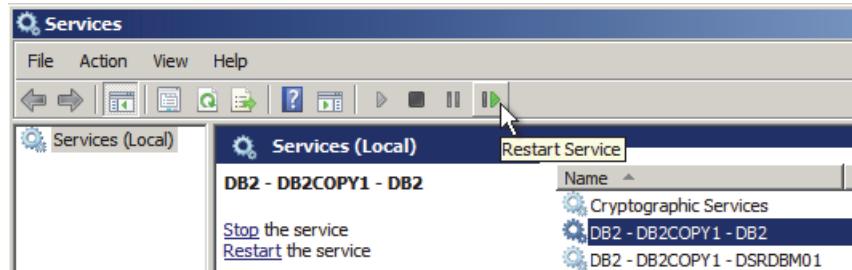
44. Confirm the **Event_Log** data source.

a. Click **Event_Log**.

The screenshot shows the Agent Editor interface for agent AB1. On the left, the Agent Test window displays the structure of the agent with nodes like AB1, Event_Log, My_App_Processes, Take_Action_Status, Thread_Pool_Status, Availability, and Disk_Storage. On the right, the Agent Information panel is open, showing the General section with fields for Service name (set to Monitor), Product code (set to knn), and Agent Information (set to Data). Below the Agent Information panel is the Attribute Group Test panel, which contains a table with four columns: Log_Name, Event_Source, Event_Type, and Event_ID. The table displays a single row with the timestamp 0 data rows returned at Feb 24, 2017 3:30:57 PM.

The initial Attribute Group Test view does not contain data, but you see valid column heading and no error messages.

- To generate some events, restart the **DB2 - DB2COPY1 - DB2** service in the Windows Services utility.



- Refresh the Attribute Group Test view by clicking **AB1** and clicking **Event_Log** again.



Hint: It might take a minute for data to show. Repeat this step until it does.

Attribute Group Test					
7 data rows returned at Jul 11, 2015 10:44:45 AM.					
Log_Name	Event_Source	Event_Type	Event_ID	Event_Category	Message
Application	DB2	Informational	1	None	DB2STOP : SQL
Application	DB2	Warning	5	None	2015-07-11-10
Application	DB2	Informational	1	None	db2start : SQL1

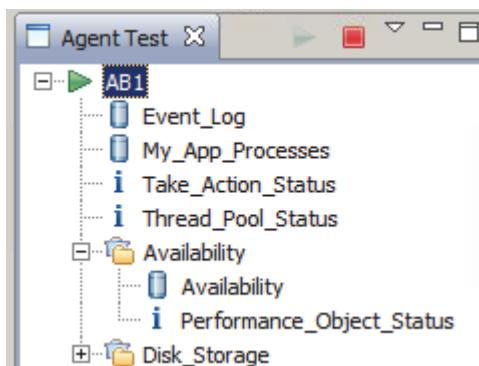
45. Confirm the **My App Processes** data source.

- Click **My App Processes**.

Caption	CommandLine
System Idle Process	
System	
smss.exe	%SystemRoot%\System32\smss.exe
rsrss.exe	%SystemRoot%\System32\rsrss.exe

The **My App Processes** attribute group pulls process information from WMI, but it is not filtered for the HTTP server and DB2. You might want to edit the view to filter out everything except DB2 and HTTP server process information.

- b. Scroll to the right to see all of the data that is collected by each process. As with all agents that are created with Agent Builder, the default workspace is a single view with all of the data that is displayed in a table.
46. Confirm the Availability data sources.
- a. Expand **Availability** in the Agent Test window.



An Availability attribute group and a Performance_Object_Status node are displayed below the Availability node.

- b. Click the **Availability** attribute group.

Attribute Group Test			
7 data rows returned at Feb 24, 2013 11:23:41 AM.			
Application_Component	Name	Status	Full_Name
DB2 - DB2COPY1 - DB2	DB2	UP	DB2 - DB2COPY1 - DB2
DB2DAS - DB2DAS00	DB2DAS00	UP	DB2DAS - DB2DAS00
DB2 Governor (DB2COPY1)	DB2GOVERNOR_DB2COPY1	DOWN	DB2 Governor (DB2COPY1)
DB2 License Server (DB2COPY1)	DB2LICD_DB2COPY1	DOWN	DB2 License Server (DB2COPY1)
DB2 Management Service (DB2COPY1)	DB2MGMTSVC_DB2COPY1	UP	DB2 Management Service (DB2COPY1)
DB2 Remote Command Server (DB2COPY1)	DB2REMOTECMD_DB2COPY1	UP	DB2 Remote Command Server (DB2COPY1)
HTTP Server	IHSforTivoliEnterprisePortalServer	UP	IHS for Tivoli Enterprise Portal Server

The Attribute Group Test view opens with the Windows Service availability data you added to the agent in a previous exercise.

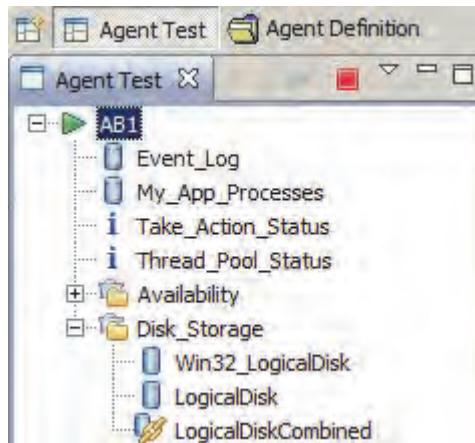
- c. Click **Performance_Object_Status**.

Attribute Group Test				
5 data rows returned at Mar 6, 2017 8:29:11 PM.				
Query_Name	Object_Name	Object_Type	Object_Status	Error
Event_Log	Event Log	NT_EVENT_LOG	ACTIVE	NO_E
My_App_Processes	root\CMIV2:Win32_Process	WMI	ACTIVE	NO_E
Win32_LogicalDisk	root\CMIV2:Win32_LogicalDisk	WMI	ACTIVE	NO_E
LogicalDisk	LogicalDisk	PERFMON	ACTIVE	NO_I

This attribute group monitors the availability of the performance object data sources that you added in this exercise. An entry is added data source for each.

47. Confirm the **Disk_Storage** data sources.

- Expand **Disk_Storage** in the **Agent Test** window.



The two source data sources, **LogicalDisk** and **Win32_LogicalDisk**, and the combined data source, **LogicalDiskCombined**, are shown.

- Click **WIN32_LogicalDisk** and **LogicalDisk** under **Disk_Storage** and confirm that they retrieve data.

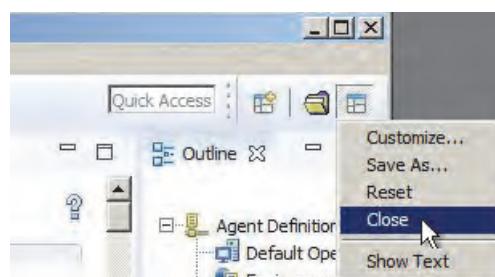


Important: Combined data sources, such as **LogicalDiskCombined**, do not show data in the full agent test utility and might stop the agent.

48. Stop the agent by clicking the **Stop Agent** icon.



49. Close the **Agent Test** perspective by right-clicking the **Agent Test** perspective icon and clicking **Close**.



You successfully tested your full agent in Agent Builder.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

50. Select to the **Agent Information** tab.

51. Change the version for 1.0.0 to **1.0.1**.

Agent Information

General
This section defines the general agent information.

Name	AB1	Advanced	Product c
Version	1.0.1	Company	
Fix pack	0	Patch level	0
		Agent ide	



Important: If your initial version number is not 1.0.0, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

52. Save your agent project.

Exercise 2 Install and confirm the updated AB1 agent in an IBM Tivoli Monitoring environment

In this exercise, you reinstall your AB1 agent on the WIN1 server and then confirm that data is being gathered for the new attribute groups you added.

1. On ITM, exit the TEP client, if it is open. To avoid display problems in Tivoli Enterprise Portal, close the TEP client during an application support installation.
2. On WIN1, if the MTMS utility is not already running, open it and stop the **Monitoring Agent for AB1** agent.



Note: Before reinstalling an agent, stop the current agent.

Create the agent installation scripts

3. In Agent Builder on WIN1, with the **AB1** agent open, select **Agent Editor > Generate Agent** from the main menu.
4. Clear all options under **Install the Agent Locally**.
5. Ensure that **Create a ZIP file** and **Create a TAR file** are checked.

Notice the directory where the files are created. You go to this directory to access the generated files. Keep the default output directory for this exercise.

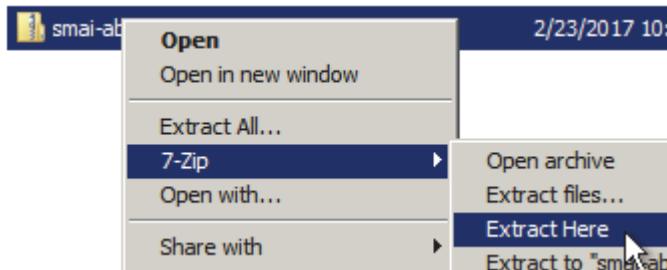
6. Click **Finish**.
7. Click **Yes** to replace the K00 image.
A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated.
8. Click **OK** to close the confirmation window.



Note: Keep Agent Builder open; you use it again later.

9. Delete the contents of the **C:\share\K00** directory.
10. Copy the new **C:\Users\Administor\smai-ab1-01.01.00.00.zip** file into **C:\share\K00** directory.

11. Extract the files from **smai-ab1-01.01.00.00.zip** into **C:\share\K00** with **7-Zip**.



Reinstall the AB1 application support on ITM

In this section, you reinstall the updated AB1 application support onto the ITM server.

12. On ITM, exit the TEP client. To avoid display problems in the Tivoli Enterprise Portal, close the TEP client during an application support installation.

13. If needed, re-create the drive map to **\WIN1\share**.

14. Open a command prompt and change to the **Y:\K00** directory.

To install the IBM Tivoli Monitoring application support without installing the agent, you run the **installIraAgentTEMS** and **installIraAgentTEPS** scripts.

15. Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00  
installIraAgentTEPS.bat C:\IBM\ITM
```

```
Y:\>cd K00  
Y:\K00>installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00  
Validating user...  
KUIC00007I: User sysadmin logged into server on https://ITM:3661.  
The requested service has already been started.  
More help is available by typing NET HELPMSG 2182.  
Install of K00 TEMS Support successful.  
Y:\K00>installIraAgentTEPS.bat C:\IBM\ITM  
Online help for this agent will not be available until the Help Server is restarted,  
which also requires restarting the TEPS.  
Install of K00 TEPS Support successful.  
Y:\K00>
```

The full installation takes 7 - 10 minutes to complete. Status information is displayed for each command.

You completed the reinstallation of the AB1 agent (K00) agent and application support on the ITM server.

Reinstall the agent on WIN1 with the script installers

In this section, you reinstall the updated AB1 on WIN1 with the script installers.

16. On WIN1, open a command prompt and run the following commands:

```
cd \share\K00  
installIraAgent.bat C:\IBM\ITM
```

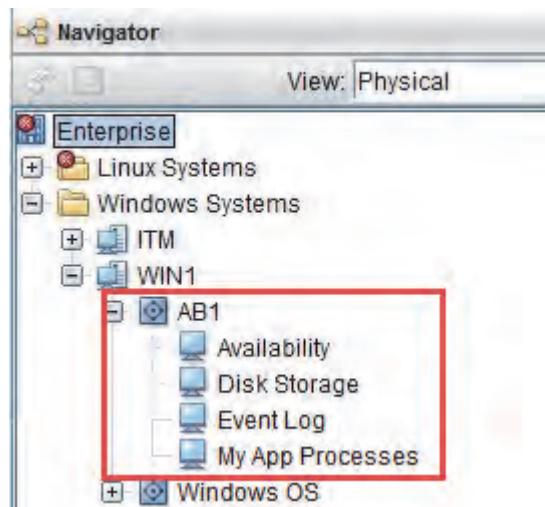
```
C:\Users\Administrator>cd \share\K00  
C:\share\K00>installIraAgent.bat C:\IBM\ITM  
Installing agent into C:\IBM\ITM  
Installing K00 .....  
Install of K00 Agent successful.  
C:\share\K00>_
```

17. Start the agent.

You completed the reinstallation of the AB1 agent (K00) agent on the WIN1 server.

Confirm the installation of agent and application support components in an IBM Tivoli Monitoring environment

18. On ITM, restart the TEP client and confirm that your **AB1** Navigator items are displayed.



Note: If the new Navigator items are not displayed or are displayed incorrectly, you might have to restart the Tivoli Enterprise Portal Server.

19. Confirm that the **My App Processes (WMI)** workspace exists and that it contains the HTTP server and DB2 process data.
- Click **My App Processes** in the Navigator.

Node	Timestamp	Caption	CommandLine
WIN1:00	03/06/17 20:32:33	System Idle Process	
WIN1:00	03/06/17 20:32:33	System	
WIN1:00	03/06/17 20:32:33	smss.exe	\SystemRoot\System32\smss.exe
WIN1:00	03/06/17 20:32:33	csrss.exe	%SystemRoot%\system32\csrss.exe Object
WIN1:00	03/06/17 20:32:33	csrss.exe	%SystemRoot%\system32\csrss.exe Object

The **My App Processes** attribute group pulls process information from WMI, but it is not filtered for the HTTP server and DB2. You might want to edit the view to filter out everything except DB2 and HTTP server process information.

- Scroll to the right to see all of the data that is collected by each process. As with all agents that are created with Agent Builder, the default workspace is a single view with all of the data that is displayed in a table.
20. Confirm that the **Event Log** attribute group is listed after the agent and that its workspace contains data.
- Click **Event Log**.

Node	Time Generated	Log Name	Event Source	Event Type	Event ID	Event Category

The initial Report view might not contain data, but you see valid column heading and no Tivoli Enterprise Portal error messages.

- On WIN1, restart the **DB2 - DB2COPY1 - DB2** service again to generate some events.
- In TEP client on ITM, refresh the **Event Log** workspace. Repeat as needed until data is displayed.

Node	Log Name	Event Source	Event Type	Event ID	Event Category	
WIN1:00	Applicati...	DB2	Informational	1	None	db2start : SQL1063N
WIN1:00	Applicati...	DB2	Warning	5	None	2017-03-06-20.30.35
WIN1:00	Applicati...	DB2MGMTSVC_DB2COPY1	Informational	1	None	mgmt_svcCopyAllSub
WIN1:00	Applicati...	DB2	Informational	1	None	DB2STOP : SQL1064
WIN1:00	Applicati...	DB2	Warning	5	None	2017-03-06-20.30.32

21. Confirm the collection of offline events by the **Event Log** attribute group.

- a. Record the Time Generated value of the top event in the Event Log Report (or the current time on the ITM server).

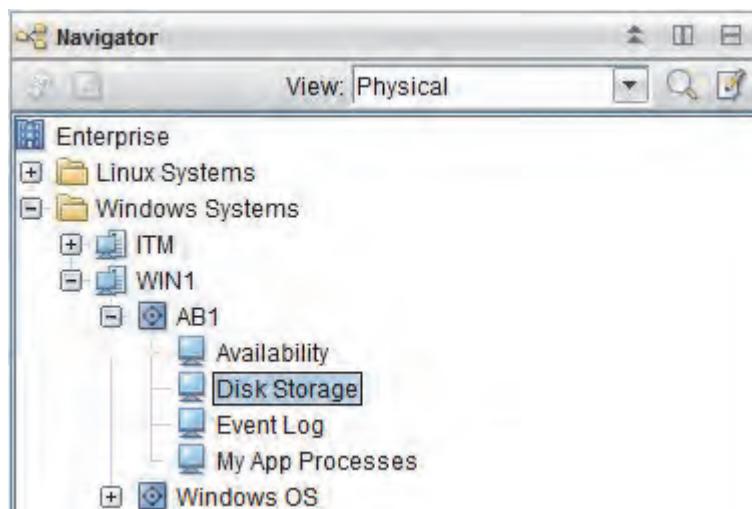
- a. On WIN1, stop the **AB1** agent.

- b. Wait at least 1 minute, and restart the **DB2 - DB2COPY1 - DB2** service again to generate some offline events.

- c. Start the AB1 agent.

- d. Return to the TEP client on ITM and confirm that the **Event Log** workspace shows events with a Time Generated value during the period that the agent was down.

22. Click **Disk_Storage** in the Navigator.



Notice that no data is in the default Report view, and that a **Request failed during execution** error is shown. More than one query is associated with DiskStorage. The Tivoli Enterprise Portal automatically generates a temporary workspace only when one query is associated with a Navigator item. You must create the view to display the **LogicalDiskCombined** data.

23. Customize the **Disk_Storage** Navigator item to show the **LogicalDiskCombined** data in the **Report** view.

- a. Right-click the **Report** view (not the title bar) and click **Properties**.

The Properties window opens.

- b. Click the **Click here to assign a query** button.

The Query Editor opens.

- c. Locate and click the following query in the Navigator:

AB1 > K00 LOGICALDISKCOMBINED > LogicalDiskCombined

Description: Attributes from attribute groups LogicalDisk and

Data Source: TEMS HUB_ITM HUB_ITM

Last Modified:

- Last Modified on: Mon, 03/06/2017 08:12 PM
- Last Modified by: SYSADMIN

	fx Node	fx Timestamp	fx Name LogicalDisk
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	== \$NODE\$		
3			
4			

- d. Click **OK** to select the query and close the Query Editor.
- e. Click **OK** to close the Properties window.

You are returned to the DiskStorage workspace. Your Report view looks like the following image.

Report							
Node	Timestamp	Name LogicalDisk	Pct Disk Read Time	Pct Disk Time	Pct Disk Write Time	Pct F	
WIN1:00	03/06/17 20:33:14	C:	0	0	0	13	

You can also add views for Perfmon LogicalDisk and WMI Win32_LogicalDisk queries to compare their data to what is in the combined view.

- f. Save your changes to this workspace.



Important: For the remainder of the class, you do not import any custom application support components into your agent. Also, you do not activate the TEP Admin mode whenever you create custom workspaces, such as modifying the DiskStorage workspace. However, typically you activate the TEP Admin mode to publish workspaces that can be imported into an agent for final distribution.

Looking at your Navigator, notice that you did not obtain the expected Performance Object Status node, which is typically created for Performance data sources such as Perfmon and WMI. This behavior is to be expected when you combine Availability monitors and Performance monitors. The Performance Object Status attribute group is associated with the Availability node and can be displayed there.

24. Click **Availability** and notice that you no longer have the custom workspace that you created. The application support for the updated AB1 agent is installed on top of the previous application support.
25. Create an Availability workspace, adding the original Windows Services data that you created in [Unit 2, “Agent creation basics exercises”](#) and the Performance Object Status data in the upper right. Do not re-create the Service Status view.

The screenshot shows the TEP interface with two main windows:

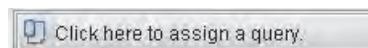
- Navigator** window (left): Shows a tree structure under "Enterprise". Under "Windows Systems", there is a node "WIN1" which contains "AB1". "AB1" has two children: "Availability" and "Disk Storage". A toolbar at the top of the Navigator window includes icons for Refresh, New, and Search.
- Performance Object Status** window (right): A table showing performance data. The columns are Node, Timestamp, Query Name, and Object Name. The data includes various system components like Event Log, Host_Availability, URL_Objects, Managed_URLs, Managed_Nodes, My_App_Processes, JavaAPIData, SNMP_nodes, Win32_LogicalDisk, and LogicalDisk, all timestamped at 04/12/17 06:17:55.

- a. Click **Availability** in the TEP Navigator.

The custom workspace that you created in the previous unit is gone. The Report view in the default workspace shows no data, and an error message is displayed.

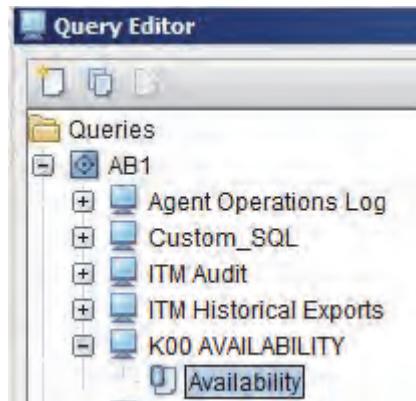
The Availability Navigator item changed because of the addition of performance monitors to your agent. The Availability Navigator item no longer is just the Availability attribute group. It is now a Navigator Group that is called Availability that contains two attribute groups: Availability and Performance Object Status. The workspace that you created is not there because it was attached to the Availability attribute group, which is no longer displayed in the Navigation tree. The Tivoli Enterprise Portal Server does not create a default view for this node because more than one query is assigned.

- b. Right-click the **Report** view and click **Properties**.
- c. Click the **Click here to assign a query** button.



The Query Editor window opens.

- d. Locate and click **AB1 > K00AVAILABILITY > Availability**.



- e. Click **OK** to close the **Query Editor**.

- f. In the **Style** tab, rename the view title to **Services**.

- g. Click **OK** to close the Properties - Availability window.

The Availability workspace is displayed with the Windows Service availability data that is displayed in the Services view.

Services						
Node	Timestamp	Application Component	Name	Status		
WIN1:00	03/06/17 20:35:55	Apache2.4	Apache2.4	UP	A	
WIN1:00	03/06/17 20:35:55	DB2 - DB2COPY1 - DB2	DB2	UP	D	
WIN1:00	03/06/17 20:35:55	DB2DAS - DB2DAS00	DB2DAS00	UP	D	
WIN1:00	03/06/17 20:35:55	DB2 Governor (DB2COPY1)	DB2GOVERNOR_DB2COPY1	DOWN	D	

- h. Create a table view of the Performance Object Status data. Click the **Table** icon from the toolbar.



Your mouse pointer changes to the **Table** icon.

- i. Click the **Table** icon on the undefined upper-right view.

A confirmation window opens.

- j. Click **Yes** to confirm that you are changing this view.

Another confirmation window opens.

- k. Click **Yes** to assign a query.

The Properties window opens.

- l. Click the **Click here to assign a query** button.

The Query Editor opens.

m. Locate and click the following query in the Navigator:

AB1 > K00 PERFORMANCE OBJECT STATUS> Performance Object Status

n. Click **OK** to select the query and close the Query Editor.

o. In the **Style** tab, rename the view title to **Performance Object Status**.

p. Click **OK** to close the Properties window.

Your final workspace looks like the one shown at the beginning of [Step 25](#) on page 4-37.

26. Save your changes to this workspace.

Performance Object Status			
Node	Timestamp	Query Name	Object Name
WIN1:00	03/06/17 20:38:50	Event_Log	Event Log
WIN1:00	03/06/17 20:38:50	My_App_Processes	root\CMIV2:Win32_Process
WIN1:00	03/06/17 20:38:50	Win32_LogicalDisk	root\CMIV2:Win32_LogicalDisk
WIN1:00	03/06/17 20:38:50	LogicalDisk	LogicalDisk
WIN1:00	03/06/17 20:38:50	LogicalDiskCombined	Joined Queries: LogicalDisk and Win32_Log

The Performance Object Status attribute group is displayed for any agent that contains performance monitors. It displays the status of the performance objects that are collected for this application. Notice that a row for each of the new data sources you added is displayed. Also, notice that the objects are active and that no error codes are being returned.

27. Scroll to the right on the **Performance Object Status** view.

The test of your agent confirms that the agent and its application support components that are installed correctly. It also confirms that the agent gathers the data that you wanted and that it displays the data correctly in the Tivoli Enterprise Portal.

Unit 5 Monitoring processes and command return codes exercises

In the first exercise in this unit, you create a single, multiplatform agent that can be installed on both Windows and Linux. The agent monitors related but different items on each operating system.

You add availability monitoring of the HTTP server process so that when the agent is installed on Windows, it monitors the Windows HTTP server process. But, when the agent is installed on Linux, it monitors the Linux HTTP server process.

Furthermore, the HTTP server produces a process ID (PID) file whenever it is running. Using availability monitoring of a command return code, you can create platform-specific commands that confirm the existence of this file.

Exercise 1 Monitor processes and command return codes

Your company has a large farm of web servers, which run on both Windows and Linux. You want to create a single agent that can be installed on both operating systems and can return information that indicates the availability of the HTTP server processes.

In this exercise, you complete the following tasks:

- Create an agent to monitor the availability of the following items:
 - The HTTP server process on both Windows and Linux
 - The HTTP server .pid file on Windows and Linux
- Install the agent and application support with installation scripts for both Windows and Linux.

Create an agent to monitor a process on Windows and Linux

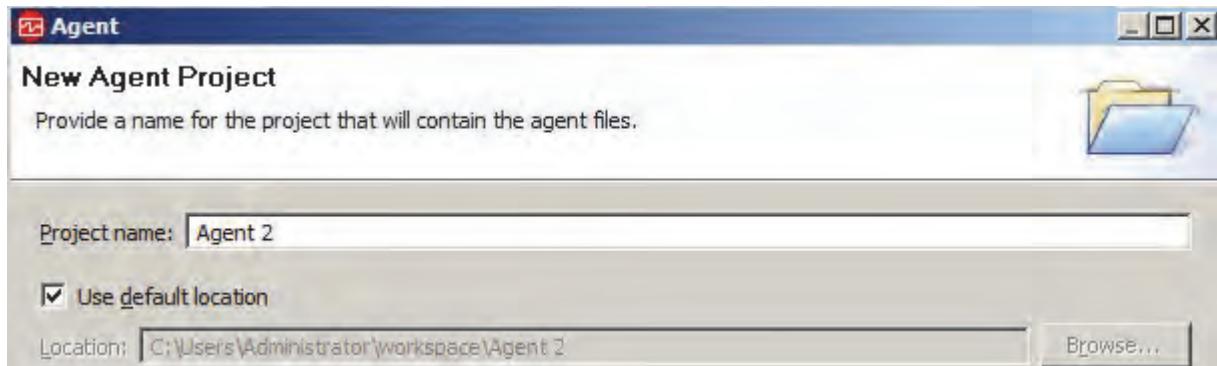
In this exercise, you create an agent to monitor the availability of the HTTP server process on both Windows and Linux.

1. On the WIN1 server, if the **Agent Builder** application is not already started, start it now.
2. Close the **AB1 Agent Editor** window if it is open.
3. Select **File > New > Agent**.

4. Click **Next**.

The New Agent Project window opens.

5. Enter **Agent 2** in the **Project name** field and click **Next**.



The General Information window opens.

6. Confirm that copyright is set to **Copyright IBM Class 2017**.



Notice the default operating systems onto which this agent is allowed to install.

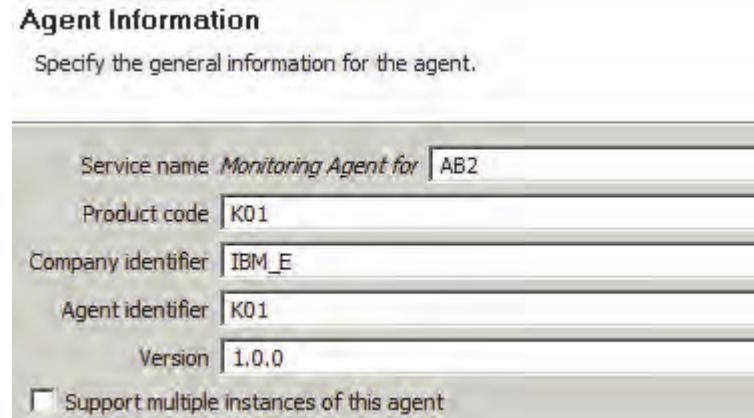
7. Click **Next**.

The Agent Information window opens.

8. Change the Service name to **AB2**.

9. Confirm the following agent information:

- Product code: **K01**
- Company identifier: **IBM_E**
- Agent identifier: **K01**
- Version: **1.0.0**

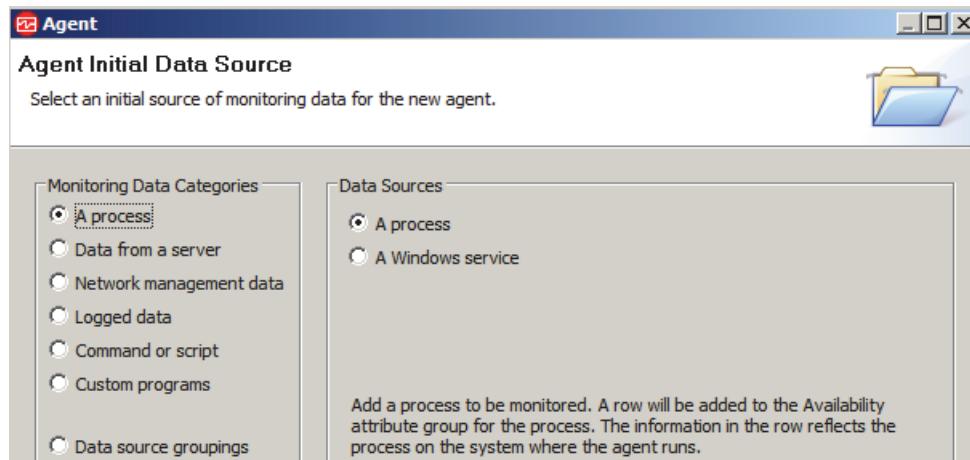


10. Click **Next**.

The Agent Initial Data Source window opens.

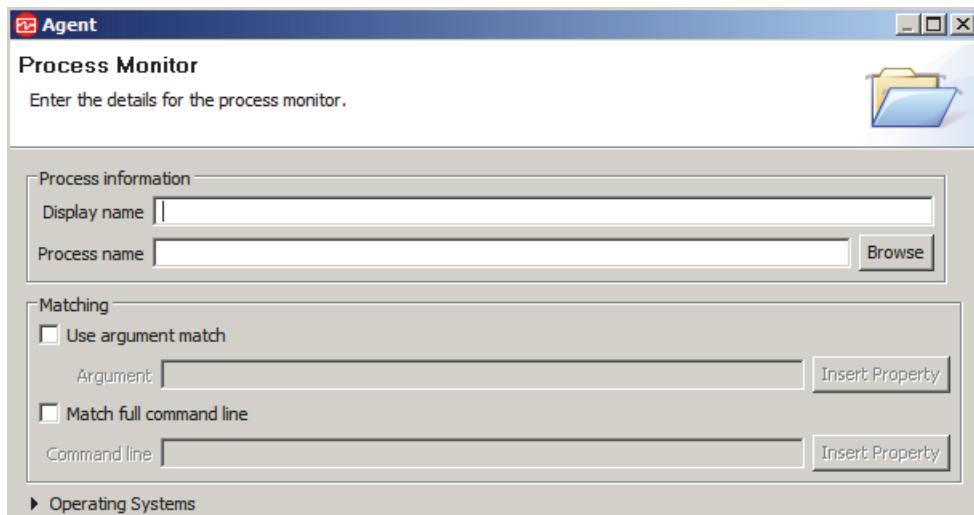
Create the HTTP server monitor for Windows

11. In the Agent wizard, select **A process** under both **Monitoring Data Categories** and **Data Sources**.



12. Click **Next**.

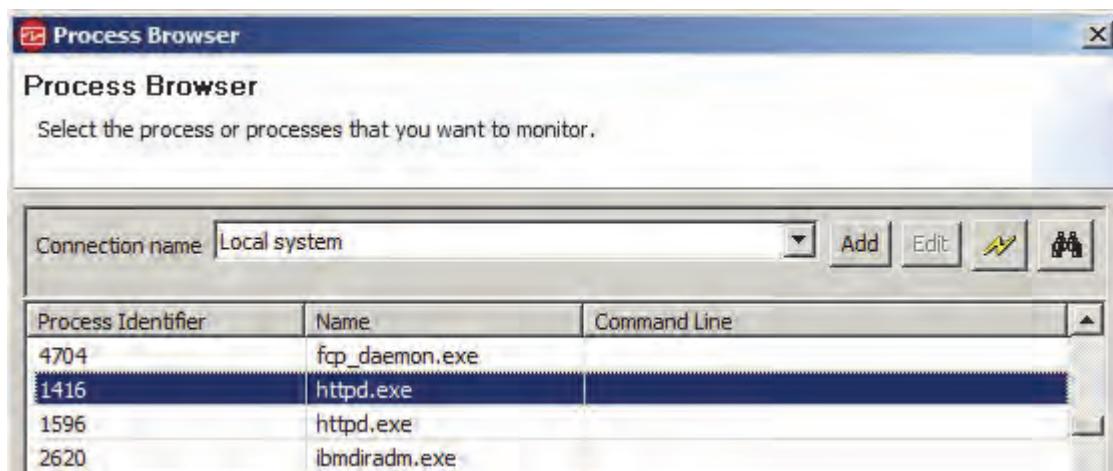
The Process Monitor window opens.



13. Enter **HTTP_Win** in the **Display name** field.



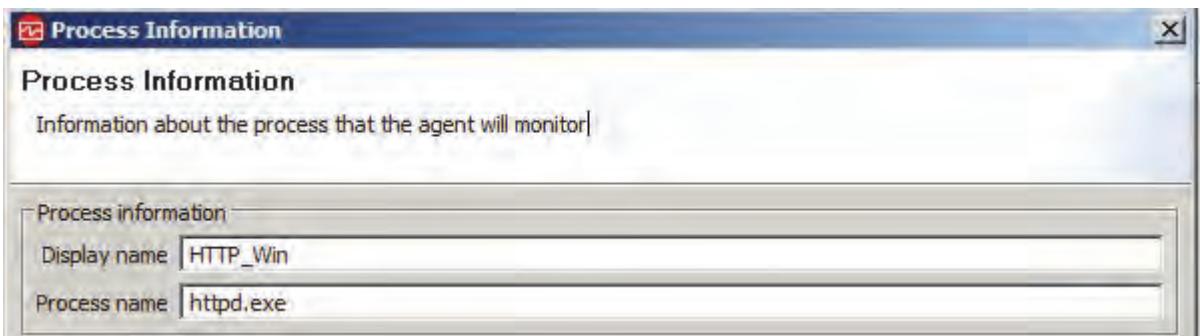
14. Click **Browse** to the right of the **Process name** field.
15. Click the **Name** column to sort by process name.
16. Locate and click one of the **httpd.exe** process rows and click **OK**.



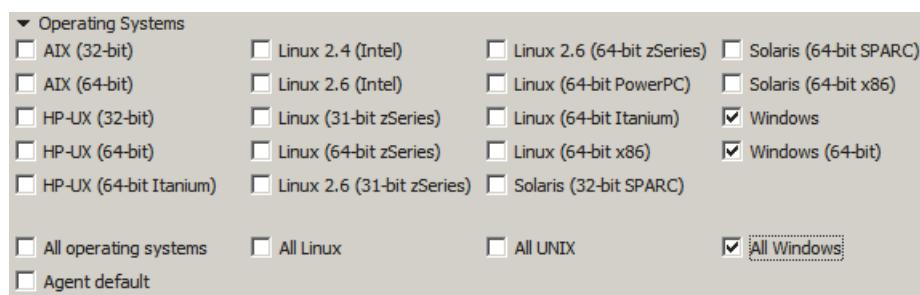
Hint: Click the Name column header to sort by the process name.

You return to the Process Monitor window.

17. Verify that the process name, **httpd.exe**, is entered.



18. Expand **Operating Systems**, clear **Agent default**, and select **All Windows**.

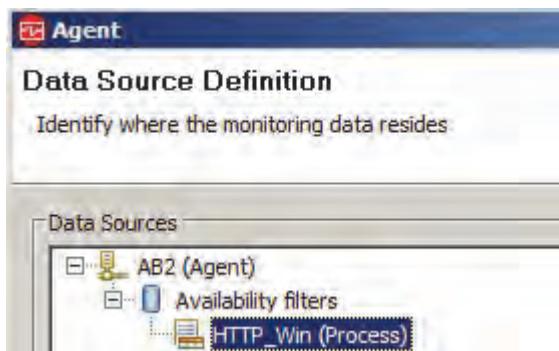


This step causes the agent to monitor only the httpd.exe process when the agent is install on a Windows system.

19. Click **Next**.

The Data Source Definition window opens.

20. If necessary, expand **Availability filters**.



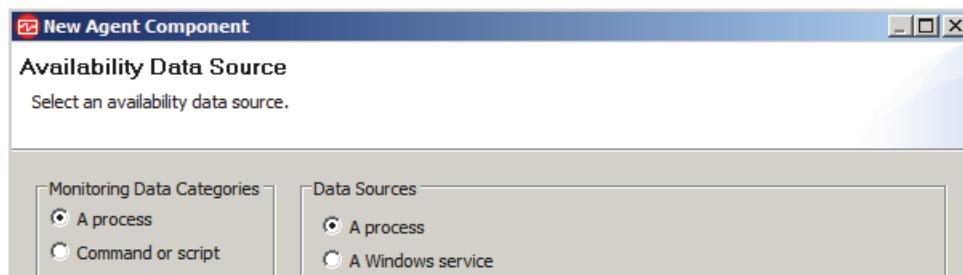
The HTTP_Win availability filter that you created is displayed.

Add the Linux HTTP server process

21. Select **Availability filters** and click **Add to Selected**.



The Availability Data Source window opens.



22. Ensure that **A process** is selected under both **Monitoring Data Categories** and **Data Sources**.

23. Click **Next**.

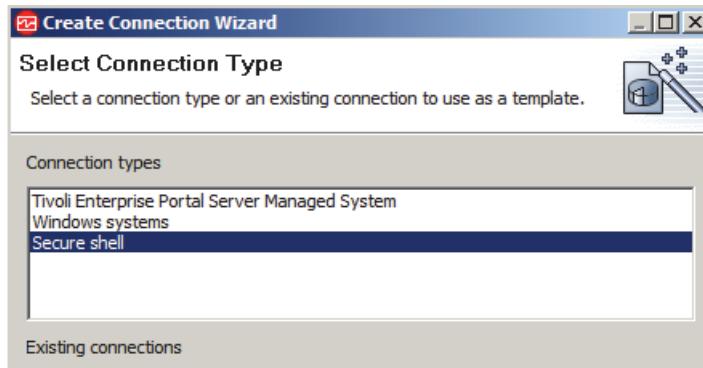
24. Enter **HTTP_Lin** in the **Display name** field.



25. Click **Browse** to the right of the **Process name** field.

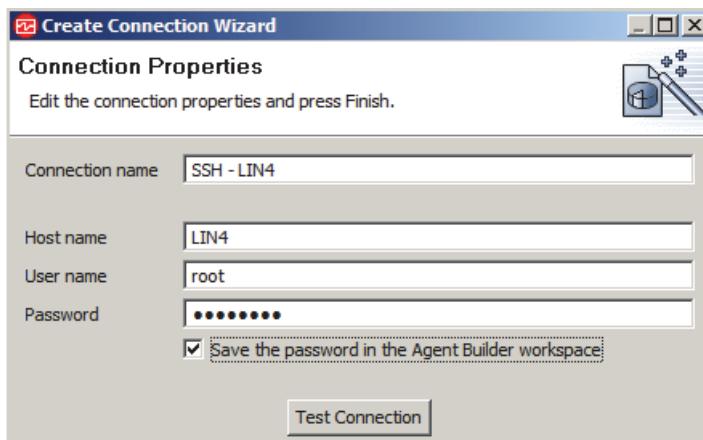
26. Click **Add** to add a connection.

27. Select **Secure Shell** and click **Next**.



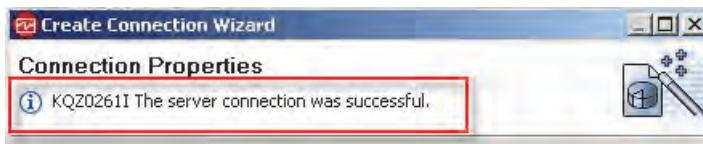
28. Enter the following values:

- Host name: **LIN4**
- User name: **root**
- Password: **object00**
- Save the password: **Checked**



29. Click **Test Connection**.

30. Confirm that the connection was successful.



31. Click **Finish**.

You are returned to the Service Browser. The services currently running on LIN4 are displayed.

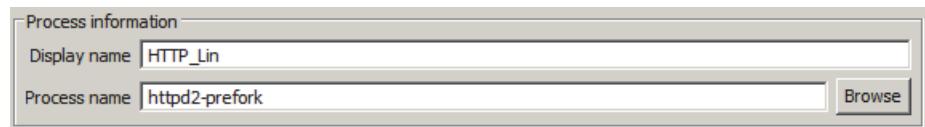
32. Locate and click one of the **httpd2-prefork** (Apache HTTP Server) process rows and click **OK**.

Process Identifier	Name	Command Line
1837	haldd-runner	haldd-runner
5945	haveged	/sbin/haveged -w 1024 -v 1
6718	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...
6719	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...
6720	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...
6721	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...

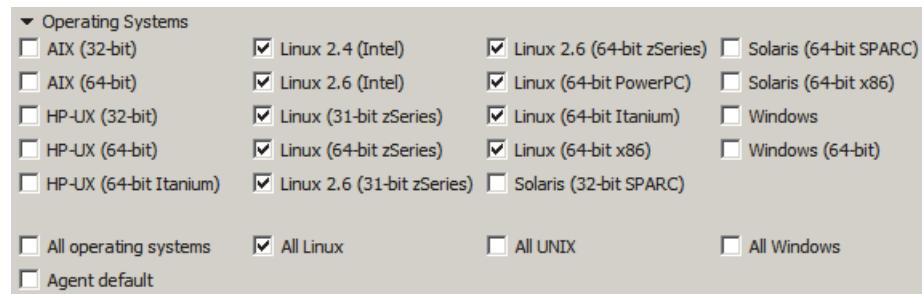


Hint: Click the Name column header to sort by the process name.

The Process Monitor window is displayed. Notice that the process name is listed.



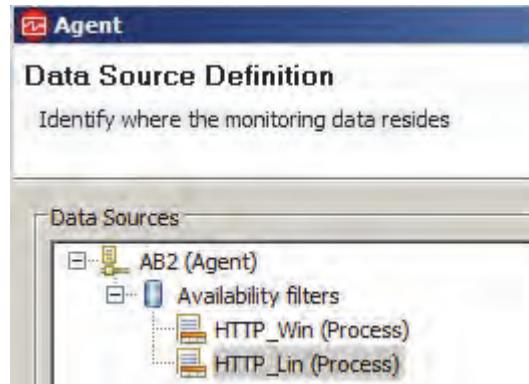
33. Expand **Operating Systems**, clear **Agent default**, and select **All Linux**.



This step causes the agent to monitor only the httpd2-prefork process when the agent is installed on a Linux system.

34. Click **Finish** to close the Process Monitor window.

Your two configured processes are displayed in the Data Source Definition window.



35. Click **Finish** to close the Data Source Definition window and save your changes.



Note: You might continue with adding the next data source without exiting the Agent wizard. Clicking **Finish** is a precaution to save your current work.

Your AB2 agent information is displayed in the Agent Builder interface.



36. Explore the **Agent Information**, **Data Sources**, and **itm_toolkit.xml** tabs.

Create an agent to monitor a functionality test

In this section, you add a command return code functionality test to your agent. You add a command that is run on the agent's host and then monitor the return codes from that command.



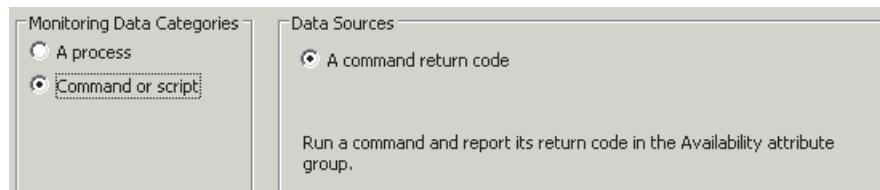
Hint: Save your agent periodically.

37. Click the **Data Sources** tab.

38. Right-click **Availability filters** and click **Add Availability Filter**.

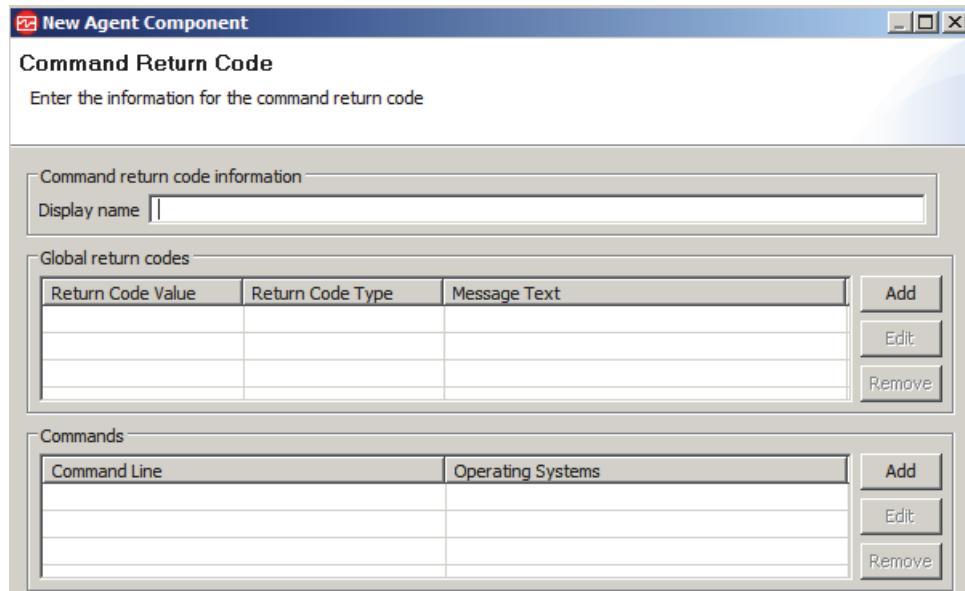
The Data Source Location window opens.

39. Click **Command or script** under **Monitoring Data Categories** and ensure that **A command return code** is selected under **Data Sources**.



40. Click **Next**.

The Command Return Code window opens.



41. Enter **HTTP PID File** in the **Display name** field.

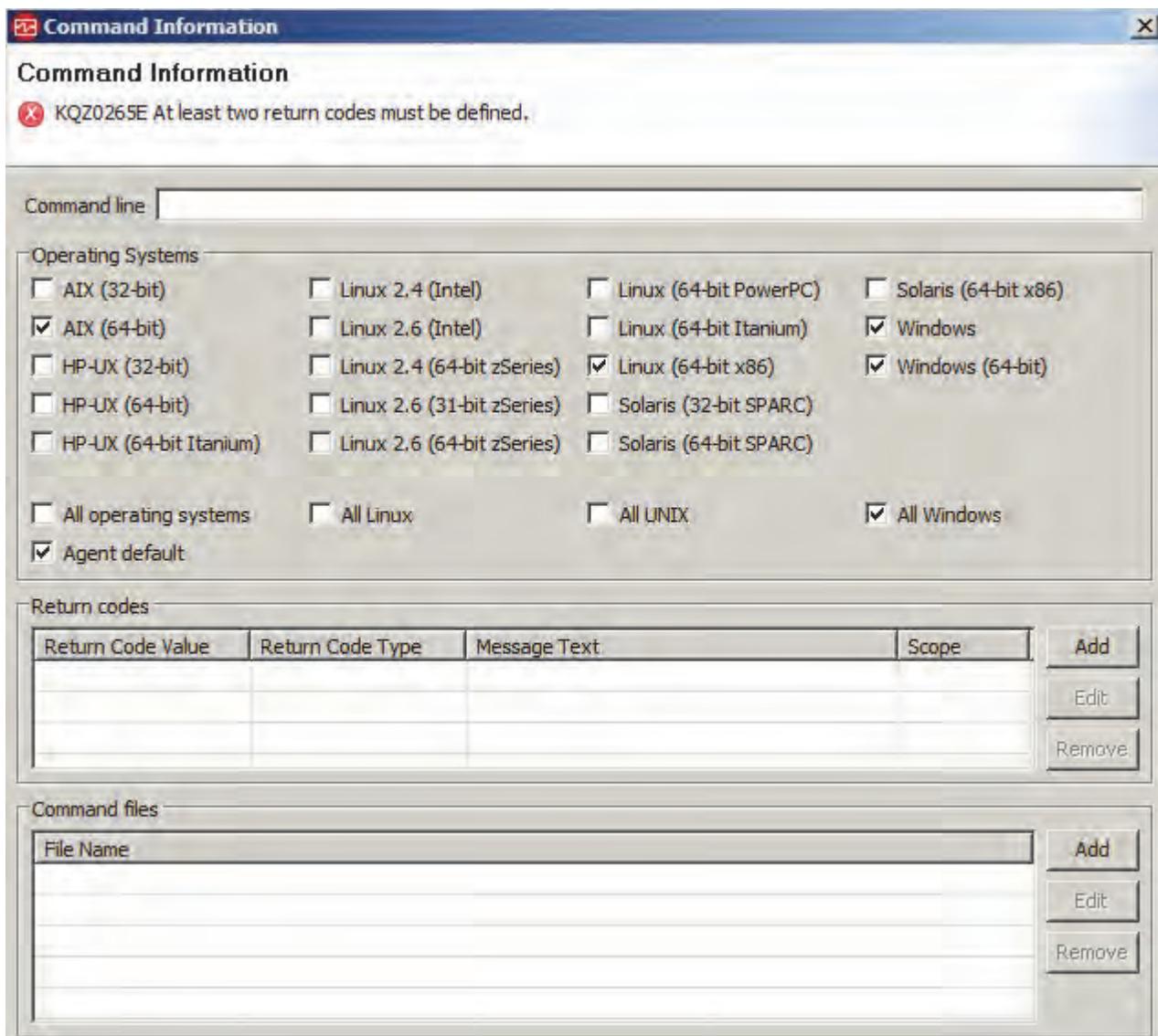
Command return code information	
Display name	HTTP PID File

42. Create the following Windows command.

Commands	
Command Line	Operating Systems
cmd.exe /c dir "C:\Apache24\logs\httpd.pid"	Windows, Windows (64-bit)

a. Click **Add** in the **Commands** sections.

The Command Information window opens.



b. Enter the Windows command in the **Command line** field.

cmd.exe /c dir "C:\Apache24\logs\httpd.pid"

Command line cmd.exe /c dir "C:\Apache24\logs\httpd.pid"



Note: Consider testing the command in a command prompt before adding it to the agent. The command must successfully display the **httpd.pid** file.

```
C:\ Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cmd.exe /c dir "C:\Apache24\logs\httpd.pid"
Volume in drive C has no label.
Volume Serial Number is BC46-ADCF

Directory of C:\Apache24\logs

05/04/2016  04:10 PM                6 httpd.pid
               1 File(s)           6 bytes
               0 Dir(s)   5,986,361,344 bytes free

C:\Users\Administrator>
```

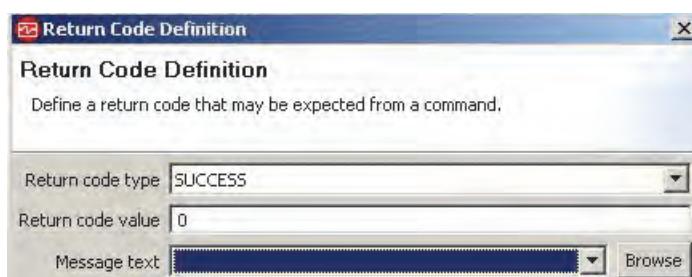
- c. Clear **Agent default**, and select **All Windows**.

43. Create the return codes for this Windows command, as shown in the following image:

Return codes		
Return Code Value	Return Code Type	Message Text
0	OK	PID file exists
1	NOT_RUNNING	PID file does not exist

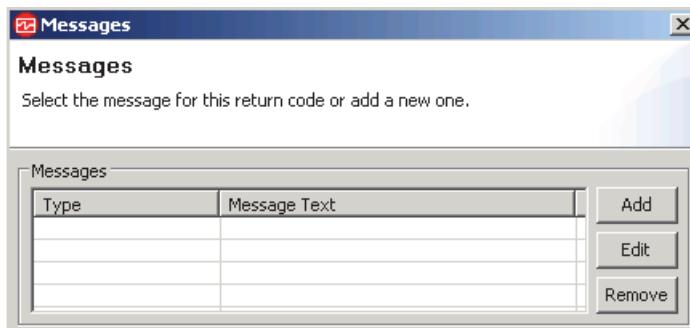
Add | Edit | Remove

- a. Click **Add** next in the Return codes section.
The Return Code Definition window opens.
- b. Select **Success** from the **Return code Type** menu and enter **0** in the **Return code value** field.



- c. Click **Browse** next to the **Message text** menu.

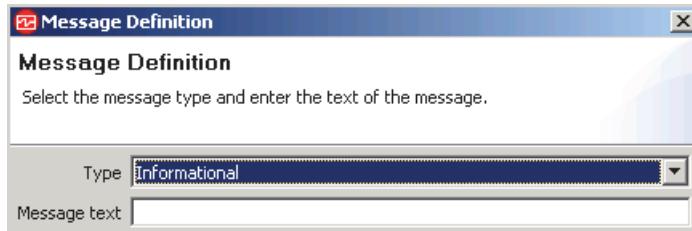
The Messages window opens.



No messages are defined in this agent now. You create two messages: one for when the PID file exists and one for when it does not exist.

- d. Click **Add**.

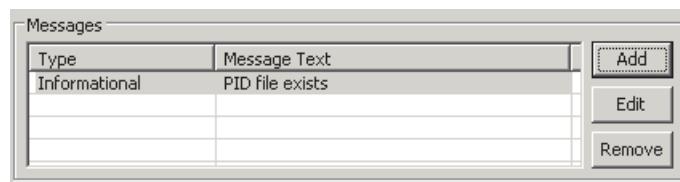
The Message Definition window opens.



- e. Select **Informational** from the **Type** menu and enter **PID file exists** in the **Message text** field. Click **OK**.



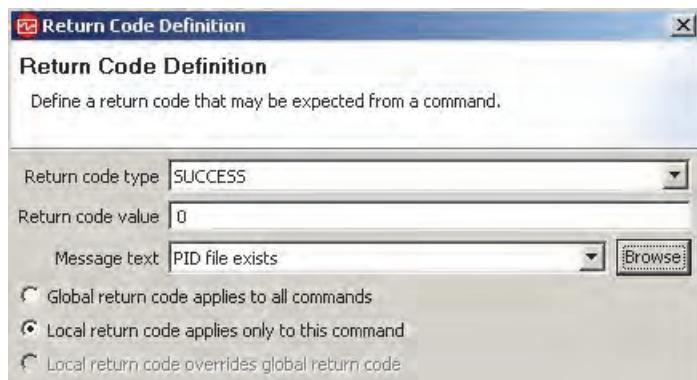
Your new message is displayed in the Messages window.



You can add the other message now, but for this exercise, you create the second message when you create the second return code.

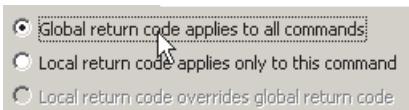
- f. Ensure that the **PID File exists** message is selected and click **OK**.

The Return Code Definition window is displayed.



Notice that this return code is defined locally and applies only to this command. From this menu, you can define the return code globally so that other commands might use it. Because this code also means success because the PID file exists, you can use it for the Linux command when you define it globally.

- g. Click **Global return code applies to all commands**.



Important: When you repeat the previous step for the second command return code, keep that return code defined locally.

- h. Click **OK** to save your return code definition.

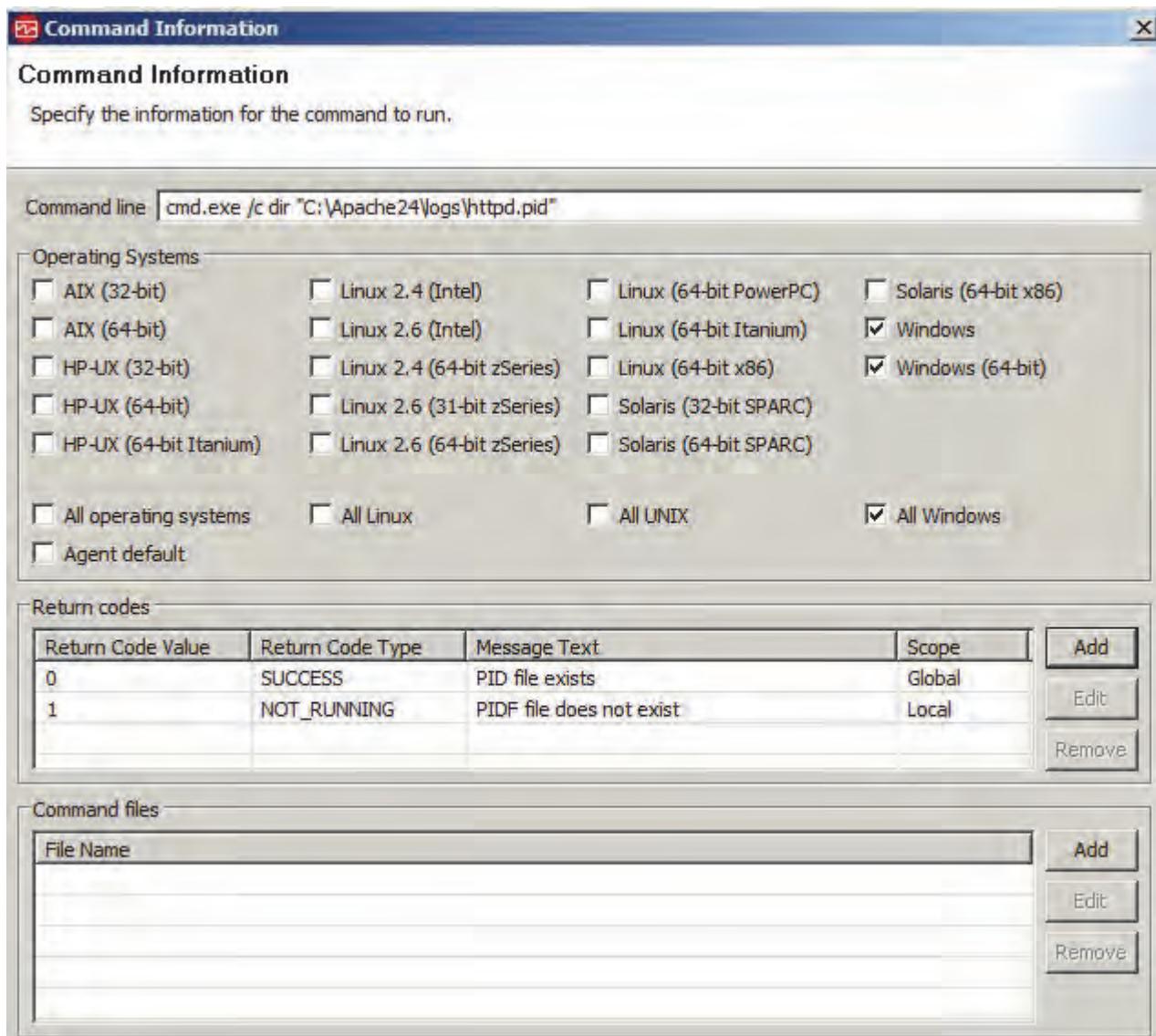
The Command Return Code window is displayed.

Return codes			
Return Code Value	Return Code Type	Message Text	Scope
0	SUCCESS	PID file exists	Global

Add **Edit** **Remove**

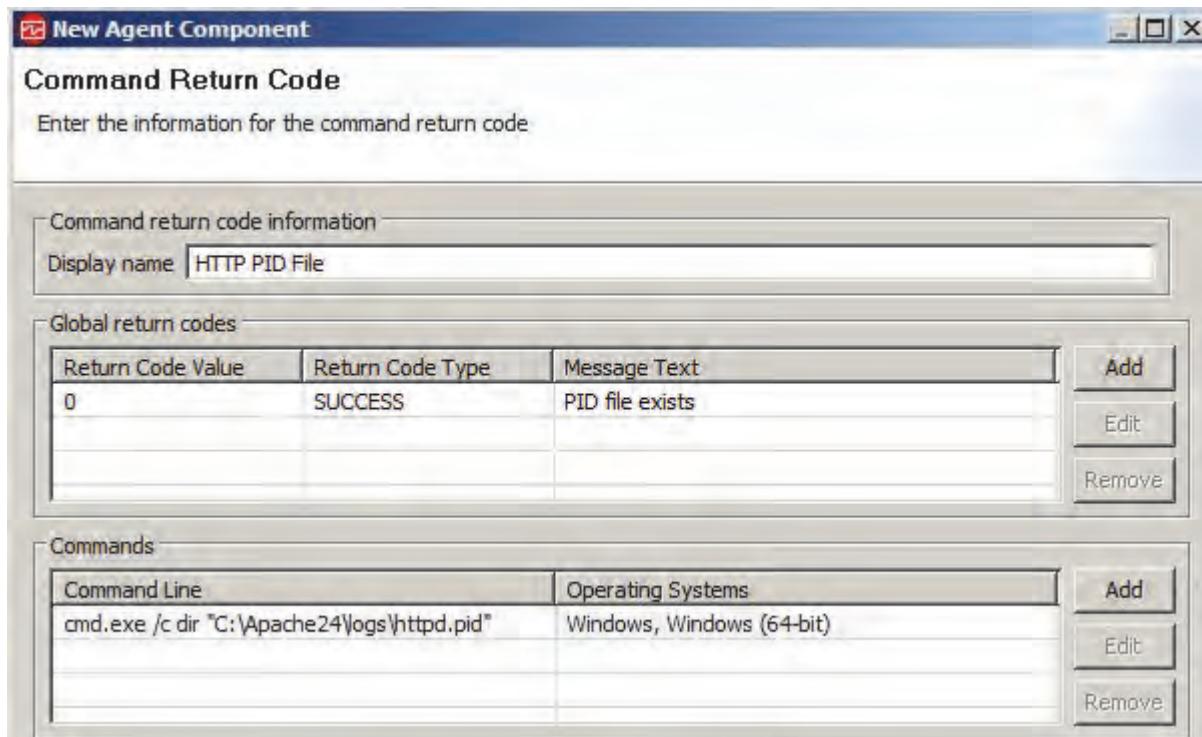
- i. Repeat these steps to create the second return code definition according to the following specifications:
- ◆ Return Code type: **NOT_RUNNING**
 - ◆ Return code: **1**
 - ◆ Message Type: **Error**
 - ◆ Message Text: **PID file does not exist**
 - ◆ Local return code applies only to this command

Your final Command Information window for this command looks like the following example:

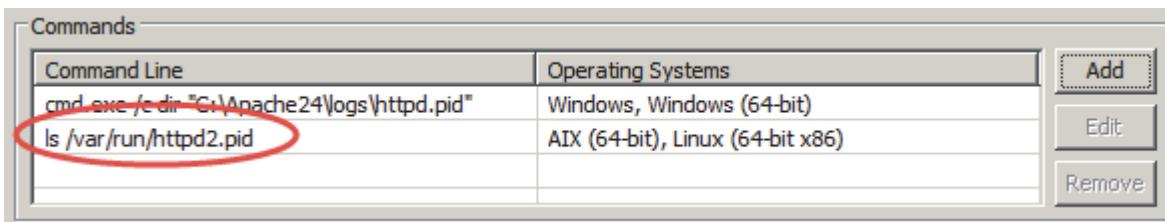


- j. Click **OK** to save the Windows command and close the Command Information window.

The Windows command is displayed in the Command Return Code section and the global return code is shown in the Global return codes section.



44. Create the Linux command that is shown in the following image.



a. Click **Add** in the Commands section to add the Linux command.

b. Enter the Linux command in the **Command line** field.

`ls /var/run/httpd2.pid`

Command line | ls /var/run/httpd2.pid

Again, test the command before adding it to your agent.

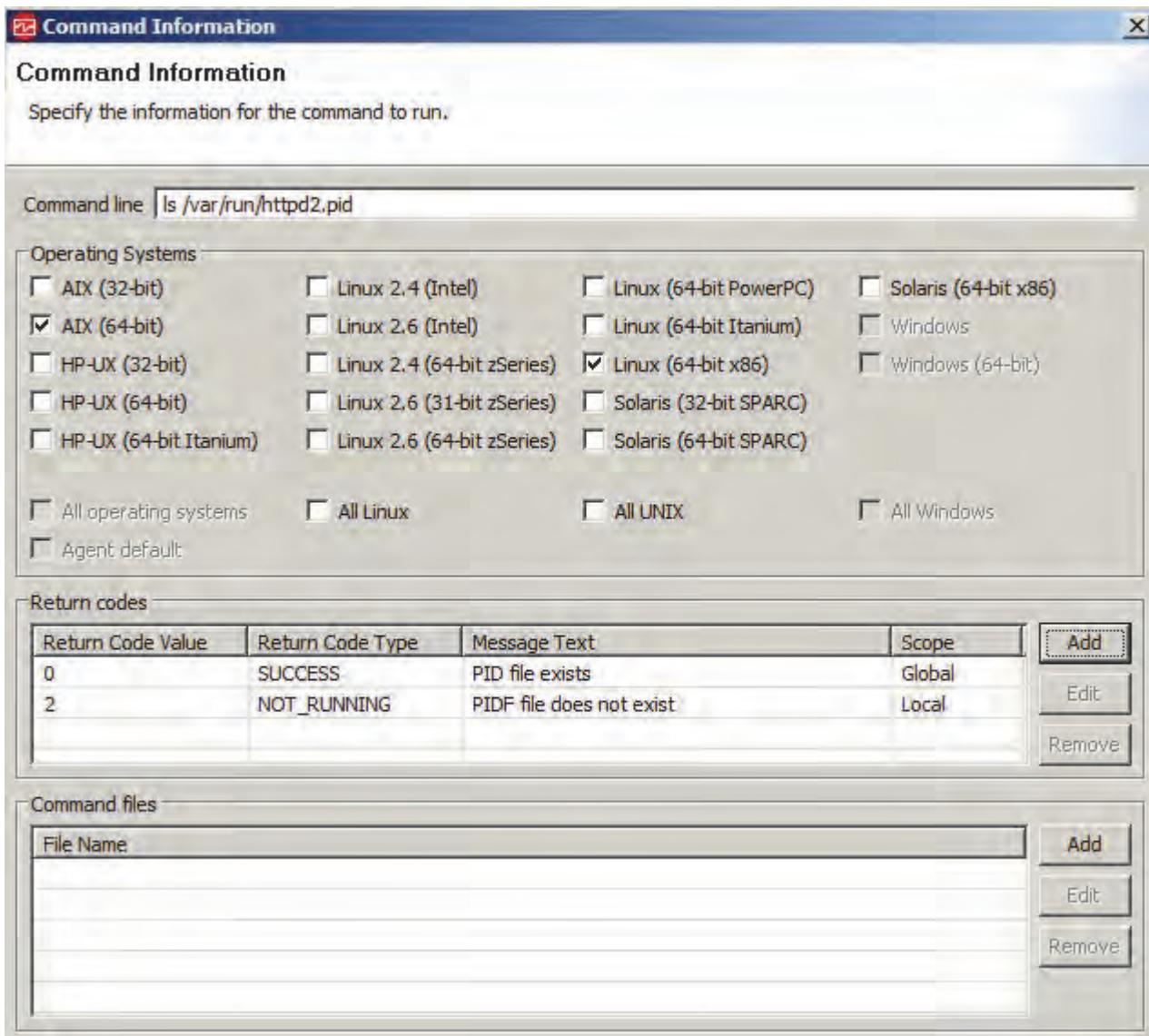
45. Keep the default Linux and UNIX systems under **Operating Systems**.

Because you already defined the 0 return code globally, it is already applied to this command, and you do not have to define it now. You do must define the failed return code for this command.

46. Repeat the previous steps to create the local failed return code on Linux according to the following specifications:

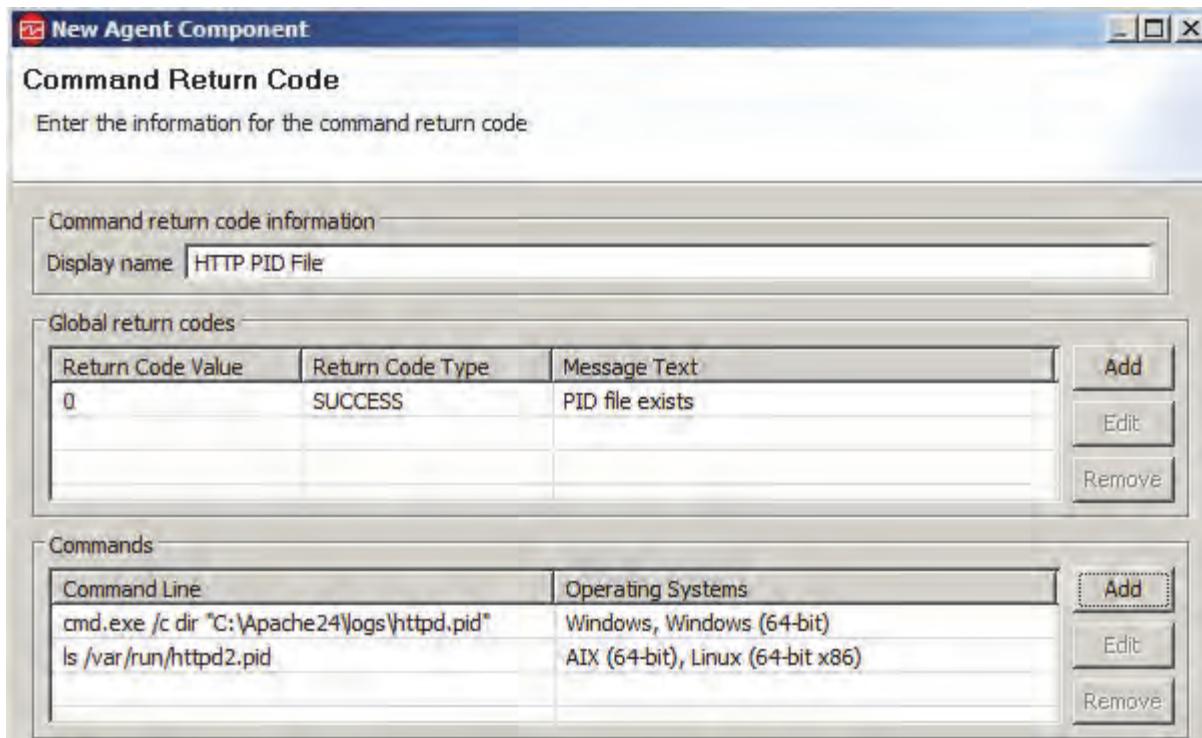
- Return Code type: **NOT_RUNNING**
- Return code: **2**
- Message Type: **Error**
- Message Text: **PID file does not exist**

Your final command window looks like the following example.



47. Click **OK** to save this Linux command.

The Command Return Code window is displayed.

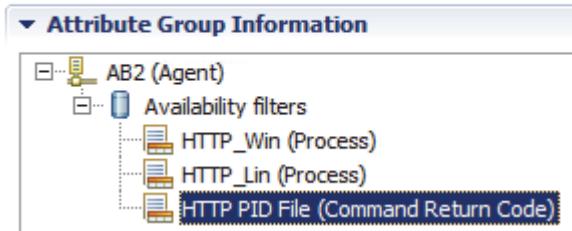


48. Click **Finish** to close the Command Return Code window.

The **Data Source Definition** tab is displayed.

49. If needed, expand **Availability Filters** and confirm your new command return code availability filter.

Data Source Definition



50. Save your agent project.

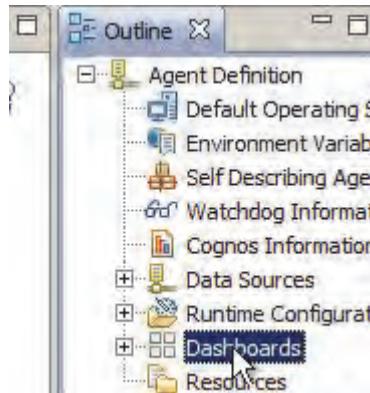
You successfully added a command return code data source to your agent.

Exercise 2 Install and confirm the AB2 agent in an IBM Tivoli Monitoring environment

In this exercise, you test your AB2 agent and confirm that data is being gathered in an IBM Tivoli Monitoring environment.

Disable IBM Performance Management dashboards

1. In Agent Builder on WIN1, with the **AB2** agent open, select **Dashboards** in the **Outline** window on the right.



The **Dashboards** tab opens.

2. Select **No dashboard presence for this agent**.

Overview

The Dashboards page lets you identify data from your agent that can be displayed on the IBM Performance Management console. Run the [Dashboard Setup wizard](#) to select an overall status attribute, other displayable attributes, and attributes that identify the software server being monitored. You can run the Dashboard Setup wizard again if you want to select different attributes or quickly see the attributes that are currently selected.

Dashboard Components

Show agent components in the dashboard
 No dashboard presence for this agent

3. Save your agent project.



Note: The warnings in the Problems panel are removed.

Create the agent installation scripts

4. Select **Agent Editor > Generate Agent** from the main menu.
5. Clear the **Install the Agent** option.
6. Ensure that **Create a ZIP file** and **Create a TAR file** are checked.

Notice the directory where the files are created. You go to this directory to access the generated files. Keep the default output directory for this exercise.

7. Click **Finish**.

A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated.

8. Click **OK** to close the confirmation window.



Note: Keep Agent Builder open; you use it again later.

9. Create a directory that is named **K01** in the **C:\share** directory.
10. Copy the new **smai-ab2-01.00.00.00.zip** file into **C:\share\K01** directory.
11. Extract the files from **smai-ab2-01.00.00.00.zip** into **C:\share\K01** with **7-Zip**.

Install the AB2 application support on ITM

In this section, you install the AB2 application support onto the ITM server.

12. On ITM, exit the TEP client. To avoid display problems in the Tivoli Enterprise Portal, close the TEP client during an application support installation.
13. If needed, re-create the drive map to **\WIN1\share**.
14. Open a command prompt and change to the **Y:\K01** directory.

To install the IBM Tivoli Monitoring application support without installing the agent, you run the **installIraAgentTEMS** and **installIraAgentTEPS** scripts.

15. Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00
installIraAgentTEPS.bat C:\IBM\ITM
```

```
Y:\K00>cd ..\k01
Y:\K01>installIraAgentTEMS.bat C:\IBM\ITM -h itm -u sysadmin -p object00
Validating user...
KUIC00007I: User sysadmin logged into server on https://itm:3661.
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.

Install of K01 TEMS Support successful.

Y:\K01>installIraAgentTEPS.bat C:\IBM\ITM
Online help for this agent will not be available until the Help Server is restarted,
which also requires restarting the TEPS.

Install of K01 TEPS Support successful.

Y:\K01>
```

The full installation takes 7 - 10 minutes to complete. Status information is displayed for each command.

You completed the installation of the AB2 agent (K01) application support on the ITM server.

Install the AB2 agent on WIN1

In this section, you install the AB2 agent on WIN1 with the script installers.

16. On WIN1 open a command prompt and run the following commands:

```
cd \share\K01
installIraAgent.bat C:\IBM\ITM
```

```
C:\share\K00>cd ..\k01
C:\share\K01>installIraAgent.bat c:\IBM\ITM
Installing agent into c:\IBM\ITM
Installing K01 ....
Install of K01 Agent successful.

C:\share\K01>
```

17. Configure the agent.

- Locate the new agent in the MTMS utility. If necessary, refresh the view.

Service/Application	Task/SubSystem	Configured	Status
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started
Monitoring Agent for AB2	Primary	No	
Monitoring Agent for AB1	Primary	Yes (TEMS)	Started

- Right-click the agent and select **Configure using Defaults**.

18. Start the agent.

You completed the installation of the AB2 agent (K01) agent on the WNI1 server.

Install the AB2 agent on LIN4

19. Access LIN4 and log in as **root** with password **object00**.

20. Open a terminal window and run the following commands to confirm that the installation scripts are visible to LIN4:

```
mount /mnt/share
ls /mnt/share
cd /mnt/share/K01
ls
```

21. Confirm the **installIraAgent.sh** script exists.

```
lin4:~ # mount /mnt/share
lin4:~ # ls /mnt/share
K00 K01
lin4:~ # cd /mnt/share/K01
lin4:/mnt/share/K01 # ls
01_dd_010000000.xml installIraAgent.bat      ira
01_dd.properties    installIraAgent.sh        k01unix.dsc
ab2-agent.bat       installIraAgentTEMS.bat   K01WINNT.dsc
ab2-agent.sh        installIraAgentTEMS.sh   K01WIX64.dsc
agent.yaml         installIraAgentTEPS.bat   reports
getarch.ksh         installIraAgentTEPS.sh  smai-ab2-01.00.00.00.zip
getarch.sh          installIra.bat           support
help               installIra.sh
lin4:/mnt/share/K01 #
```

22. Install the agent onto this computer by running the following command:

```
./installIraAgent.sh /opt/IBM/ITM
```

```
lin4:/mnt/share/K01 # ./installIraAgent.sh /opt/IBM/ITM/
Installing k01 .....
copying agent config files now.....
Install of K01 Agent successful.
To configure the agent, run:
  /opt/IBM/ITM//bin/itmcmd config -A 01
To start the agent, run:
  /opt/IBM/ITM//bin/itmcmd agent start 01
To stop the agent, run:
  /opt/IBM/ITM//bin/itmcmd agent stop 01
lin4:/mnt/share/K01 #
```

23. Configure the agent by running the following command:

```
cd /opt/IBM/ITM/bin/  
./itmcmd config -A 01
```

```
lin4:/mnt/share/K01 # cd /opt/IBM/ITM/bin/  
lin4:/opt/IBM/ITM/bin # ./itmcmd config -A 01  
Agent configuration started...
```

```
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
```

- a. Press Enter to confirm that this agent connects to a Tivoli Enterprise Monitoring Server.
- b. Press Enter to accept each of the following default values:
 - ◆ ip.pipe as the Network Protocol
 - ◆ Not to install a Network Protocol 2
- c. Enter **ITM** as the TEMS Host Name and press Enter.

```
Network Protocol [ip, sna, ip.pipe, ip.spipe, ip6, ip6.pipe or ip6.spipe] (Default is: ip.pipe):
```

```
Now choose the next protocol from one of these:
```

- ip
- sna
- ip.spipe
- ip6
- ip6.pipe
- ip6.spipe
- 0 for none

```
Network Protocol 2 (Default is: 0):
```

```
TEMs Host Name for IPv4 (Default is: lin4): ITM
```

- d. Press Enter to accept each of the following default values:
 - ◆ Port of 1918
 - ◆ Null as the KDC_PARTITION name
 - ◆ No to configuring a connection to a secondary TEMS
 - ◆ 0 for an Optional Primary Network Name
 - ◆ Disable HTTP

A prompt that indicates the agent configuration is complete.

```
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):

Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):
Enter Optional Primary Network Name or @ for "none" (Default is: @):
Disable HTTP? [1=YES, 2=NO] (Default is: 2):
Agent configuration completed...
Would you like to restart the component to allow new configuration to take effect? [1=Yes, 2=No] (Default is: 1):
```

You receive a prompt that indicates the agent configuration is complete and prompted to restart the agent.

24. Press Enter to restart the agent.

```
Would you like to restart the component to allow new configuration to take effect? [1=Yes, 2=No] (Default is: 1):
Stopping Monitoring Agent for AB2
Monitoring Agent for AB2 stopped
Starting Monitoring Agent for AB2
Monitoring Agent for AB2 started
Lin4:/opt/IBM/ITM/bin #
```

25. Start the MTMS utility and confirm the agent status by running the following command:

```
/opt/IBM/ITM/bin/itmcmd manage &
```

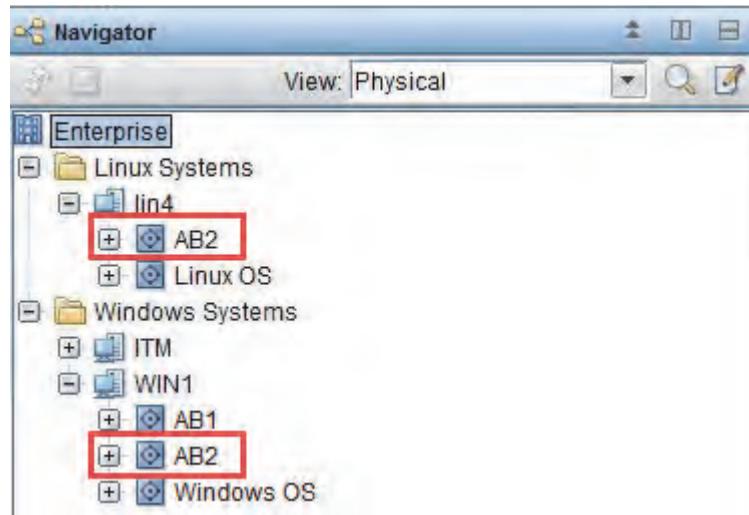
Manage Tivoli Enterprise Monitoring Services				
Actions Options View Help				
Service	Version	Platform	Configured	Status
Monitoring Agent for AB2	V01.00.00.00	Linux x86_64 R...	Yes	Started
Monitoring Agent for Linux OS	V06.30.02.00	Linux x86_64 R...	Yes	Started

You successfully installed the agent on LIN4.

Confirm the data display in the TEP client

26. On ITM, restart the TEP client and log in as **sysadmin** with password **object00**.
27. If prompted at any time, click **Navigator update pending** to update the navigator with the new navigator items.

28. Expand the tree to display the subnodes of WIN1 and LIN4.



29. If **AB2** is not displayed under either host, wait 3 - 5 minutes for Navigator updates. If updates are not displayed, restart the agent in the MTMS utility on that host. If the LIN4 AB2 agent is not displayed, reconfigure it and confirm that you set the HUB_TEMS to ITM.

30. Confirm the LIN4 AB2 data.

- Expand **AB2** under LIN4 and click **Availability**.
- Confirm that you see valid data in the Report view for the HTTP Server processes and the HTTP Server PID file on LIN4.

Report							
Node	Timestamp	Application Component	Name	Status	Full Name	Type	
lin4:01	03/06/17 21:27:25	HTTP PID File	ls /var/run/httpd2.pid	PASSED	N/A	FUNCTIONALITY	
lin4:01	03/06/17 21:27:25	HTTP_Lin	httpd2-prefork	UP	/usr/sbin/httpd2-prefork	PROCESS	
lin4:01	03/06/17 21:27:25	HTTP_Lin	httpd2-prefork	UP	/usr/sbin/httpd2-prefork	PROCESS	
lin4:01	03/06/17 21:27:25	HTTP_Lin	httpd2-prefork	UP	/usr/sbin/httpd2-prefork	PROCESS	
lin4:01	03/06/17 21:27:25	HTTP_Lin	httpd2-prefork	UP	/usr/sbin/httpd2-prefork	PROCESS	
lin4:01	03/06/17 21:27:25	HTTP_Lin	httpd2-prefork	UP	/usr/sbin/httpd2-prefork	PROCESS	
lin4:01	03/06/17 21:27:25	HTTP_Lin	httpd2-prefork	UP	/usr/sbin/httpd2-prefork	PROCESS	

- Test your monitors by running the following command on LIN4 to stop its HTTP Server:
`/etc/init.d/apache2 stop`

Note: It can take several minutes for the status change to show.

- d. On ITM, refresh the **Availability** workspace.

Report						
Node	Timestamp	Application Component	Name	Status	Full Name	
lin4:01	03/06/17 21:29:06	HTTP PID File	ls /var/run/httpd2.pid	FAILED	N/A	
lin4:01	03/06/17 21:29:06	HTTP_Lin	httpd2-prefork	DOWN	N/A	

- e. Scroll right to see the functionality test message information.

Percent Processor Time	Command Line	Functionality Test Status	Functionality Test Message
0	N/A	NOT RUNNING	K010002E: PIDF file does not exist
0	N/A	N/A	N/A

- f. Run the following command to restart the HTTP Server on LIN4:

```
/etc/init.d/apache2 start
```

31. Repeat the previous step and confirm the WIN1 AB2 data by starting and stopping the **Apache2.4** service in Windows Services.gedit

Unit 6 Monitoring custom data sources exercises

In this unit, you modify your agents to gather and monitor data with custom instrumentation.

In the first solution, you modify the AB2 agent to deploy and run a script that generates monitoring data. The agent then monitors and returns the data that is generated by the script. The key to this solution is to create a script or command that generates the target data in the correct format. You also enable the agent to establish a socket connection with an application and monitor the application.

In the second solution, you modify the AB1 agent to monitor a log file. The agent must be able to parse the log file to pull the IP address, packets sent, packets received, and packets lost. Then the agent creates an attribute that states the status of the host that is based on the total number of packets received. Specifically, it states the host status is Good if four packets are received and Failure if no packets are received. Additionally, you add a Java API data source that enables the agent to monitor an application or resource with custom Java code.

Exercise 1 Create an agent to monitor local and remote systems with scripts and socket connections

Agent Builder can create an agent that carries one or more scripts, runs each script on the host, and returns the output data.

In this exercise, you do the following tasks:

- Create an agent that distributes a script, runs the script, and returns the output data to the agent
- Create an agent that listens to a socket connection and monitors data and error codes sent to that connection
- Create the installation archive files with command line interface (CLI) commands

Confirm the script

To return the script output correctly, the data must be placed in a delimited format with no unnecessary data. This task requires that a script that generates and formats the data properly be used by the agent.

Start with a command for the initial formatting of the data. Then, use a script that parses the basic data into only the data you want returned to the monitoring system.

In this exercise, you use a script that uses the VMSTAT command. VMSTAT is a good command for this exercise because it displays interesting data in a space-delimited format, which reduces the amount of extra parsing that you must do on the data.

1. On LIN4, open a terminal window and run the following command:

```
vmstat 1 2
```

```
Lin4:~ # vmstat 1 2
procs -----------memory----- swap-----io--- system--cpu-----
r b swpd   free   buff   cache   si   so    bi    bo   in    cs us sy id wa st
0 0      0 1261992 17132 380340   0   0   144    7 114 154 1 1 82 16 0
0 0      0 1261992 17132 380352   0   0     0   0 101 144 0 0 100 0 0
Lin4:~ #
```

The VMSTAT reports information about processes, memory, paging, block IO, traps, and processor activity. The first report provides averages since the last restart. More reports provide information about a sampling period of length delay. Because you are not interested in the average since the last restart, your command requests two reports with a 1-second delay between them, making the second report a good current interval.

For this exercise, you reduce the data that you gathered to just the processor information on the far right. The columns *us*, *sy*, *id*, and *wa* show the percentage of time the processor spent on user processes and system processes, idle, or waiting.

To parse the data correctly, you need a command or script that removes the first two rows and then everything except those four data points. A script for this task is provided.

2. Change to the following directory:

```
/root/AB_Files
```

3. Run the following command to see the contents of your script:

```
cat script1.sh
```

```
lin4:~/AB_Files # cd /root/AB_Files/
lin4:~/AB_Files # ls
AgentGenerator.log  SocketTestDL.pl  SocketTestDR.pl  SocketTestEL.pl  SocketTest.pl
script1.sh          SocketTestDL.pl~  SocketTestDR.pl~  SocketTestER.pl
lin4:~/AB_Files # cat script1.sh
vmstat 1 2 | tail -1 | awk '{print $13","$14","$15","$16}'
lin4:~/AB_Files #
```

The tail section outputs only the last row. The *awk* section produces the 13th - 16th values, delimited with a comma.

- Run the following command:

```
./script1.sh
```

```
lin4:~/AB_Files # ./script1.sh  
8,8,67,17
```

This example has a properly formatted script output for your agent: delimited data points. Notice that the delimiting character is a comma. You now have a script that you can build script data source around.

- Copy the script to the WIN1 share with the following command:

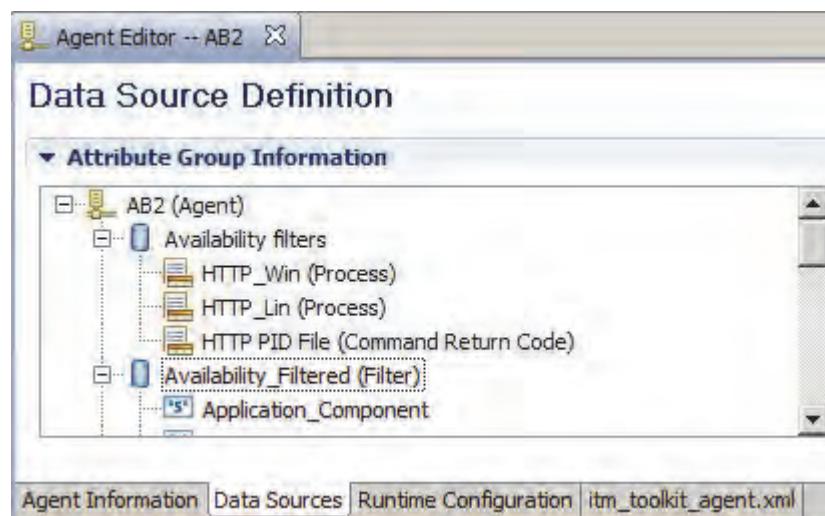
```
cp script1.sh /mnt/share
```

You confirmed the script and made it available to Agent Builder on WIN1.

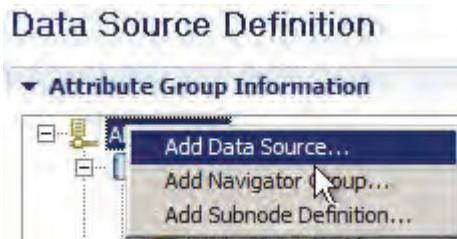
Modify AB2 to monitor the script

Complete the following steps:

- On WIN1, start Agent Builder, if it is not already running.
- If the **AB2** agent is not already open, expand **Agent 2** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
- Click the **Data Sources** tab.

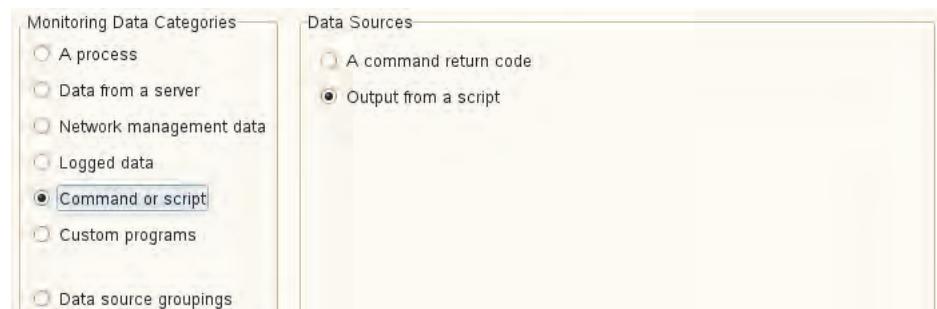


9. Right-click **AB2 (Agent)** under **Attribute Group Information** and click **Add Data Source**.



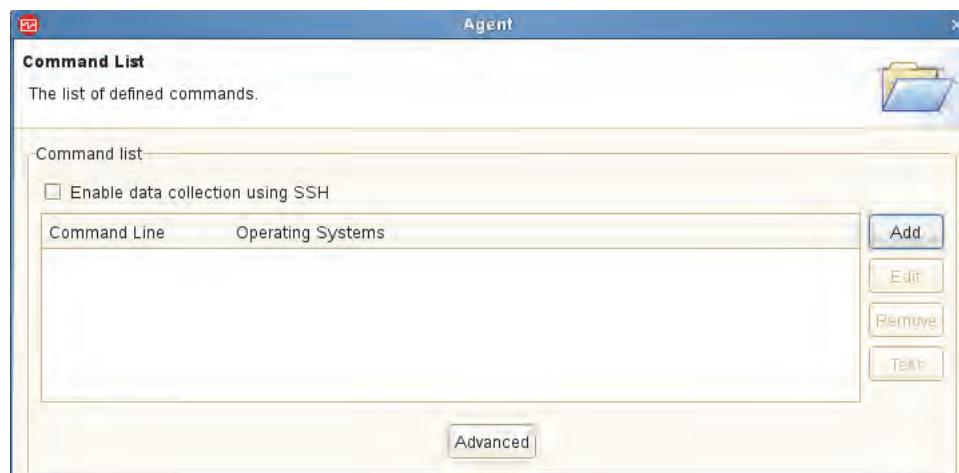
The New Agent Component window opens to Data Source Location.

10. Click **Command or script** under **Monitoring Data Categories** and **Output from a script** under **Data Sources**.



11. Click **Next**.

The Command List window opens.



From this window, you can add one or more scripts to this agent.

12. Click **Add**.

13. Enter **script1.sh** in the command line.



Notice the semicolon in the Separator section.

14. Change the separator from a semicolon to a comma.



15. Keep the **Agent default** operating systems.

16. Click **Add** to the right of the **Command files** section.

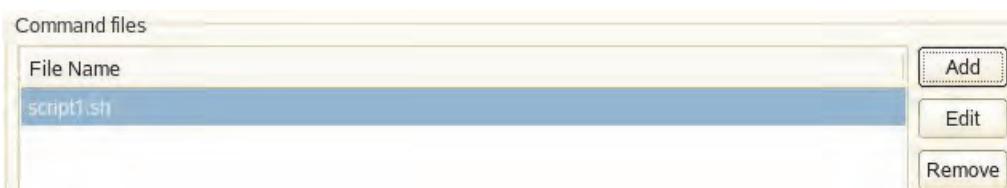
17. Navigate to and locate the following file:

C:\share\script1.sh

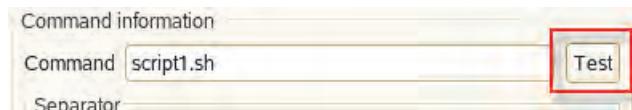


18. Click **script1.sh** and click **Open**.

The Script Information window opens with the **script1.sh** file that is displayed in the Command files section.



19. Click **Test**.



20. If prompted, click **Yes** to stop the agent.

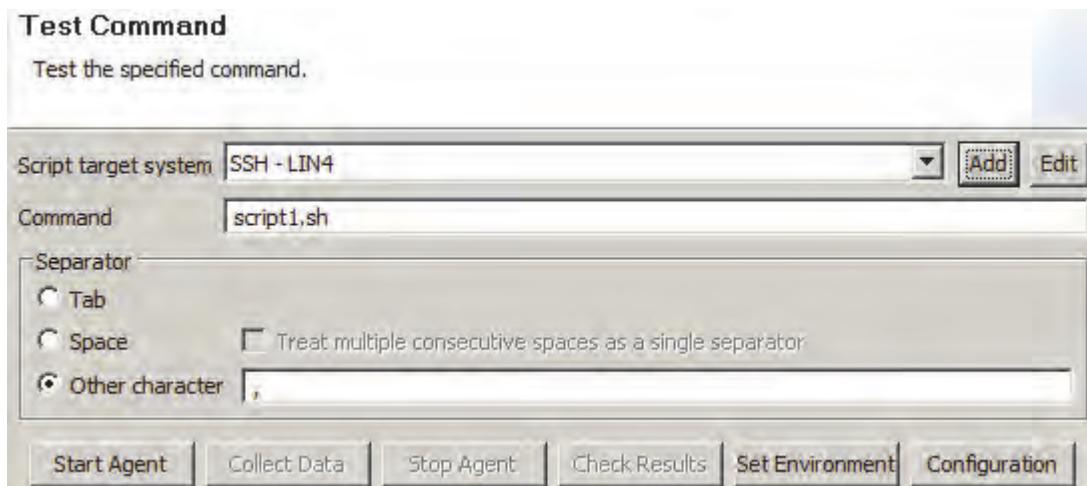
The Test Command window opens.



21. Select **SSH - LIN4** from the **Script target system** menu.



22. Ensure that the target system is LIN4, and the command name and separator are correctly listed.



23. Click **Start Agent**.

24. Click **Collect Data**.

A status indicator is displayed.

25. Select **No** to the prompt to view suggested data types.

The script results are captured and attributes are defined to hold the data.

Results			
<input checked="" type="checkbox"/> Show hidden attributes			
Attribute_1	Attribute_2	Attribute_3	Attribute_4
12	5	75	9

Notice that this test was successful and attributes are correctly created to match the data that is retrieved. Had your script test been unsuccessful, you might find errors in the lower part of the Test Command window and you might see the attributes that are created to be inadequate. For example, you might find two or more data points that are combined into one attribute.



Note: If you do not test a script in the Agent Builder tester, the Agent wizard will not define attributes and you must create them manually outside of the wizard.

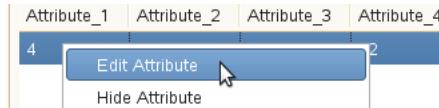
Up to this point in this course, you created agents with predefined attribute groups and attributes. Because Agent Builder cannot know the attributes that are gathered, you must define them now.

26. Click **Stop Agent**.

27. Edit the four attributes for the four values that are generated by this script.

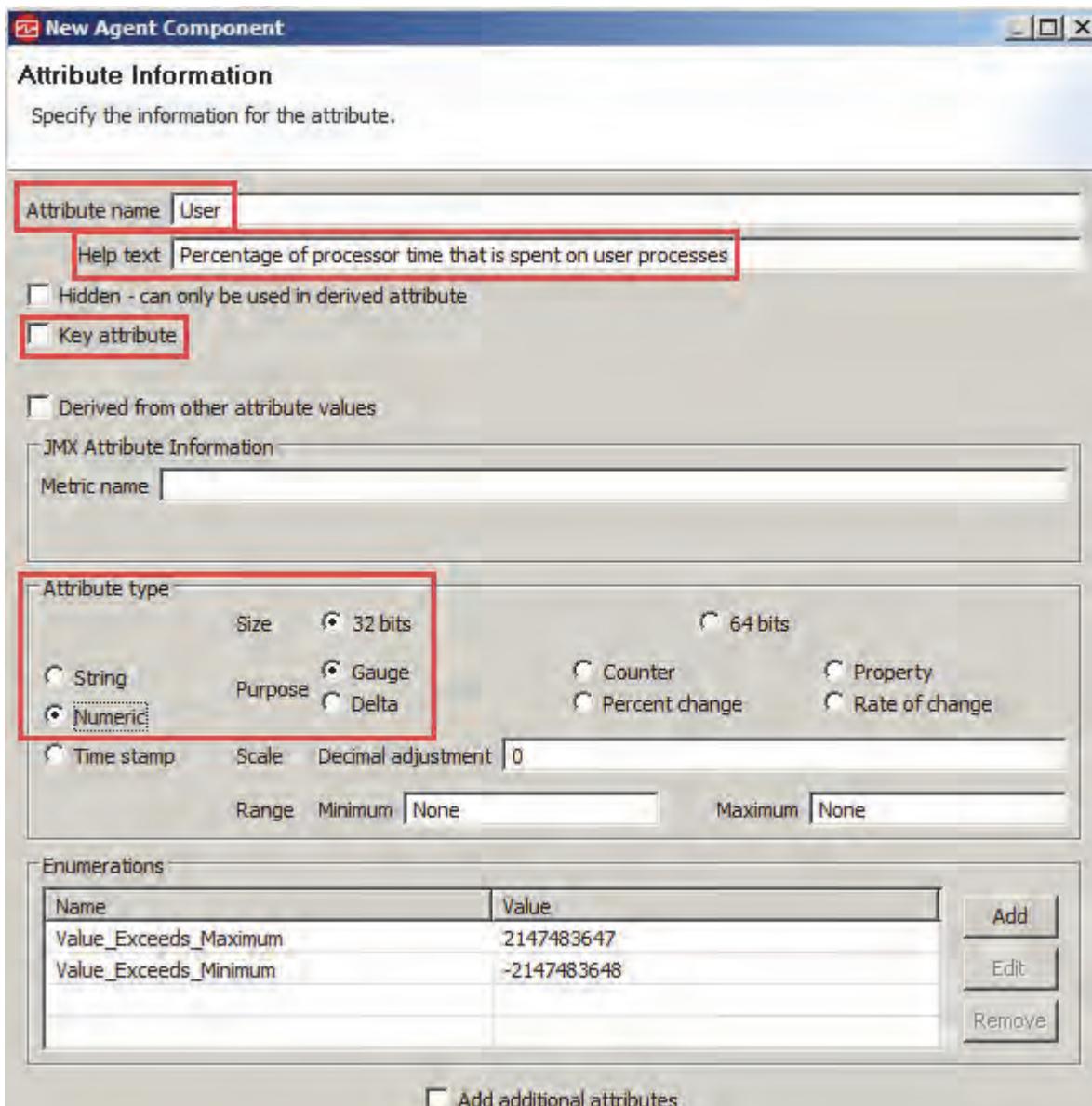
Current Attribute	New Name	Help text	Key?	Type
1	User	Percentage of processor time that is spent on user processes	No	Numeric, Gauge, 32-bit
2	System	Percentage of processor time that is spent on system processes	No	Numeric, Gauge, 32-bit
3	Idle	Percentage of processor time that is spent idle	No	Numeric, Gauge, 32-bit
4	Wait	Percentage of processor time that is spent waiting	No	Numeric, Gauge, 32-bit

- a. Right-click the cell in the attribute column and click **Edit Attribute** or click the column header.



The Attribute Information window opens.

- b. Complete the Attribute Information window based on the information in the preceding table.



- c. Click **OK** after you finish.

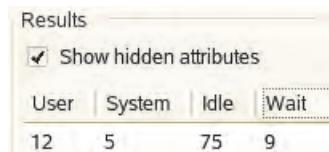
The new column name is displayed in the Test Command window.

Results				
<input checked="" type="checkbox"/> Show hidden attributes				
User	Attribute_2	Attribute_3	Attribute_4	
12	5	75	9	

- d. Select **No** to the prompt to view suggested data types.

- e. Repeat these steps for each attribute.

When completed, you see the following results in your Results pane.



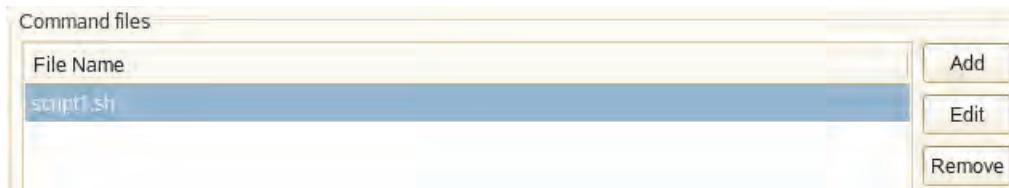
In the Test Command window, you successfully tested your script and you created the attributes that are needed to capture the data.



Important: Clicking **OK** from the Test utility exits the utility and saves your attribute modifications. Clicking **Cancel** exits without saving the attribute definitions.

28. Click **OK** to close the **Test Command** window.

The Command Information window is displayed with the **script1.sh** script in the Command Files section.

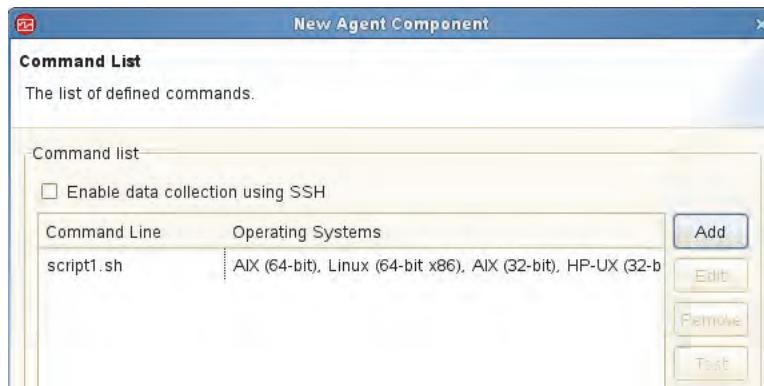


29. Select **All operating systems**.



30. Click **OK** to save your command and close the **Command Information** window.

The Command List window is displayed with the **script1.sh** script added.



You can add more scripts to this agent, edit an existing script, or remove an existing script. Your agent uses the only this script.

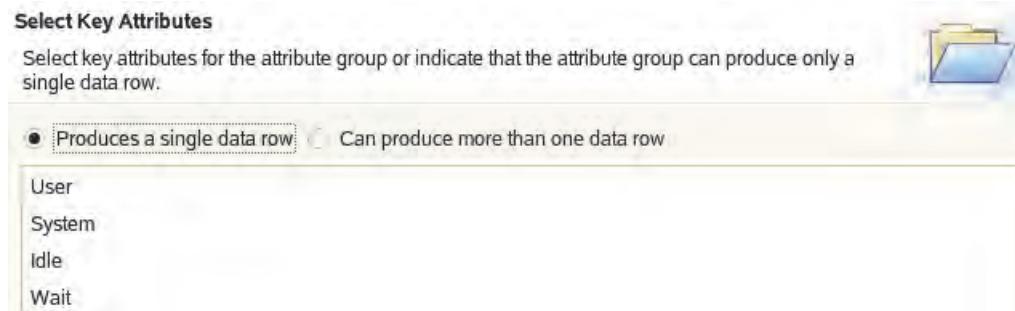
31. Click **Enable data collection using SSH** to enable the agent to collect the data remotely over SSH.



32. Click **Next**.

The Select Key Attributes window opens.

33. Click **Produces a single data row**.



Both the **Next** and **Finish** buttons are active. You completed enough steps to create this agent. If you want to continue working in the wizard to modify this agent, such as adding more data sources, you click **Next**.

34. Click **Finish**.

You are returned to the Data Sources tab showing your new Script data source.

The screenshot shows the 'Data Source Definition' dialog box. In the top right corner, there are four icons: a green plus sign, a red minus sign, a green checkmark, and a blue question mark. Below these are two buttons: 'Add to Selected' and 'Remove'. The main area is titled 'Attribute Group Information' with a dropdown arrow. Inside, there is a tree view with three items: 'Availability filters', 'Availability_Filtered (Filter)', and 'script1_sh (Script)'. To the right of the tree view are two buttons: 'Add to Selected' and 'Remove'. Below this section is a 'Command List' area. It contains a text input field labeled 'Attribute group name' with the value 'script1_sh'. Underneath it is a 'Help' field containing the text 'Data gathered using script script1.sh.'. There are three radio buttons: one selected (filled with a dot) and two empty. The first radio button is labeled 'Produces a single data row'. The second is 'Can produce more than one data row'. The third is 'Produces large number of data rows' with a checked checkbox. At the bottom of the dialog box is a tab bar with four tabs: 'Agent Information', 'Data Sources' (which is selected and highlighted in blue), 'Runtime Configuration', and 'itm_toolkit_agent.xml'.

35. Save your agent project.

Notice that the attribute group name has the script name.

36. Modify the agent by renaming the attribute group to Processor Utilization.

- Click **script1_sh (Script)** in the **Attribute Group Information** pane.
- Replace **script1.sh** with **Processor_Utilization** in the **Attribute group name** field and press Enter to activate the change.

The screenshot shows the 'Data Source Definition' dialog box with the 'Attribute Group Information' pane open. The tree view now shows 'AB2 (Agent)' expanded, with 'Processor_Utilization (Script)' selected and highlighted with a blue border. In the 'Command List' area, the 'Attribute group name' field has been changed to 'Processor_Utilization'. A red arrow points from the text 'Processor_Utilization' in the 'Attribute group name' field towards the selected item in the tree view.

37. Save your agent project.

You successfully created an agent that deploys and monitors data that is created by a script.

Add derived attributes

In this section, you add derived attributes that do the following tasks:

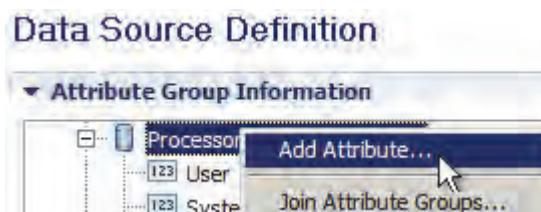
- Get the host name from an environment variable of the monitored host
- Resolve the IP address from the host name
- Concatenate the HostName and IP address fields

38. Add a derived attribute that identifies the agent host according to the following specifications:

- Attribute name: **HostName**
- Help text: **Capture the host name from the hostname environment variable**
- Attribute Type: **String, Maximum size: 64**
- Derived Attribute Formula:

`getenv("hostname")`

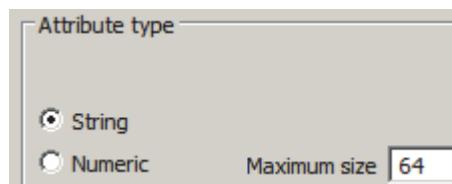
- a. In the **Data Sources** tab, right-click **Processor_Utilization (Script)** attribute group and select **Add Attribute**.



- b. Enter the **Attribute name** and **Help text** values.

Attribute name	HostName
Help text	Capture the host name from the hostname environment variable

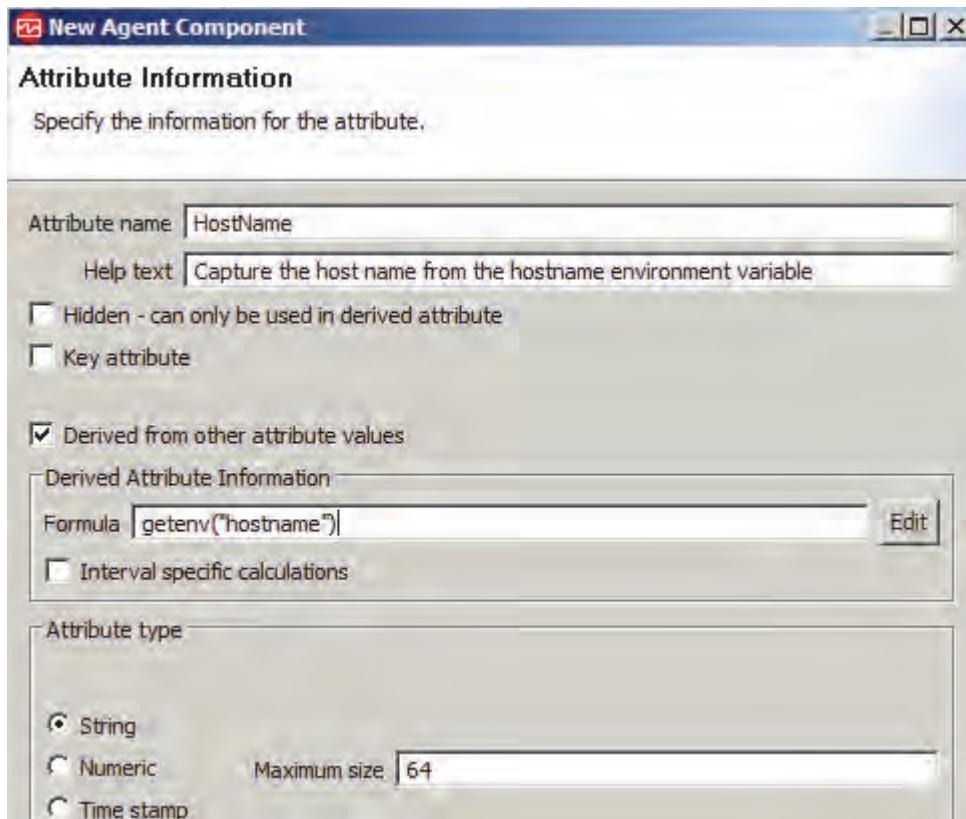
- c. Select **String** in **Attribute Type**.



- d. Select **Derived from other attribute values** and enter the formula.

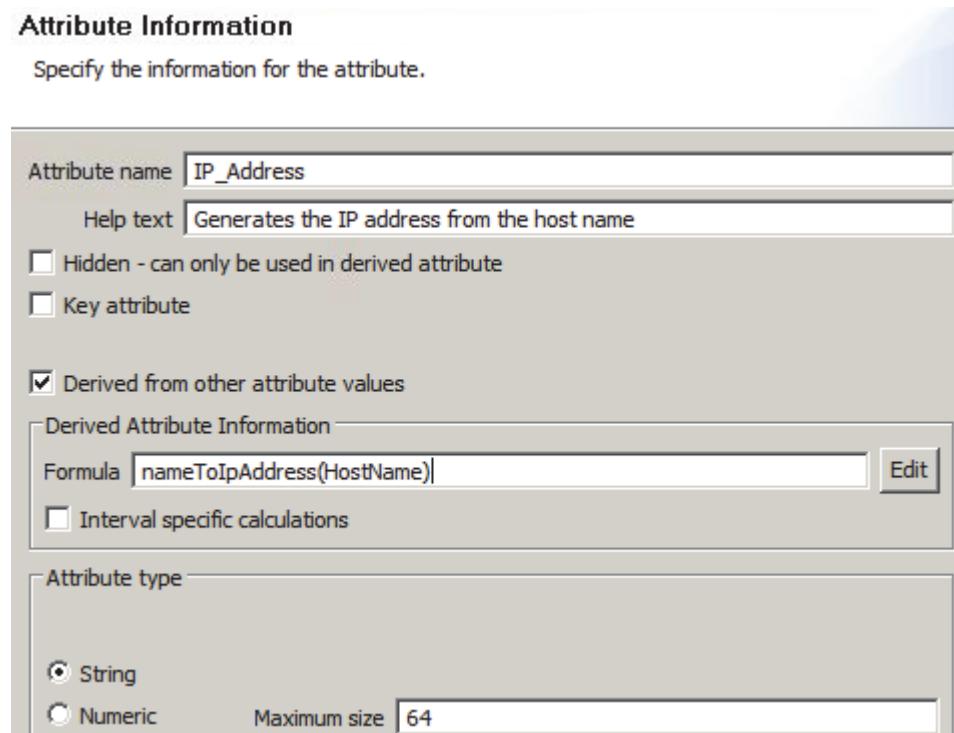
<input checked="" type="checkbox"/> Derived from other attribute values	
Derived Attribute Information	
Formula <input type="text" value="getenv('hostname')"/>	<input type="button" value="Edit"/>
<input type="checkbox"/> Interval specific calculations	

The Attribute Information window looks like the following screen capture.



- e. Click **Finish** to close the Attribute Information window.
39. Add an attribute to Processor_Utilization that derives the IP address from the host name captured by the HostName attribute. Complete the attribute as shown here:
- Attribute name: **IP_Address**
 - Help text: **Generates the IP address from the host name**
 - Attribute Type: **String, Maximum size: 64**
 - Derived Attribute Details Formula:
`nameToIpAddress (HostName)`

The Attribute Information window looks like the following screen capture.



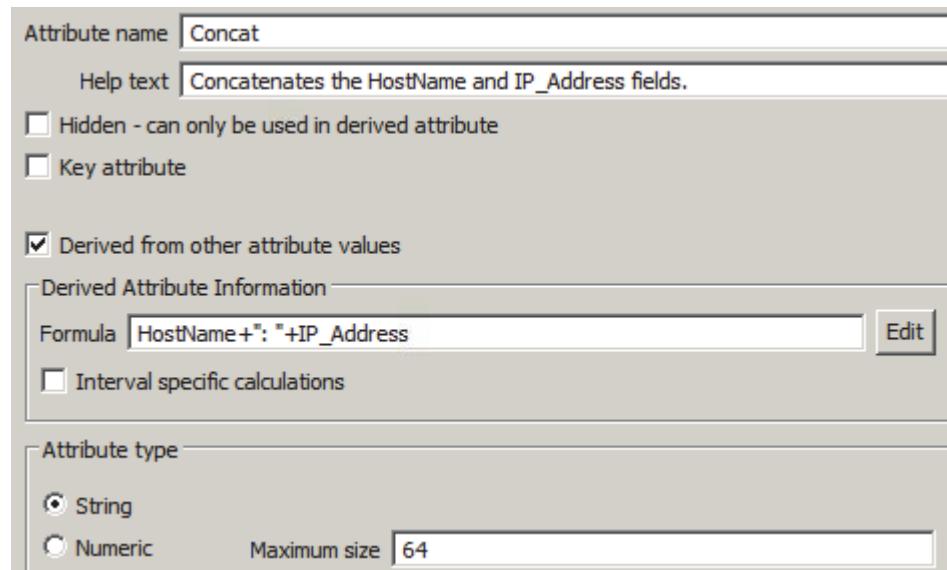
Hint: Click **Finish** to close the Attribute Information window or select **Add additional attributes** to create the derived attribute in the next step.

40. Add a derived attribute that concatenates the HostName and IP_Address attributes according to the following specifications:

- Attribute name: **Concat**
- Help text: **Concatenates the HostName and IP_Address attribute values.**
- Attribute Type: **String, Maximum size: 64**
- Derived Attribute Details Formula:

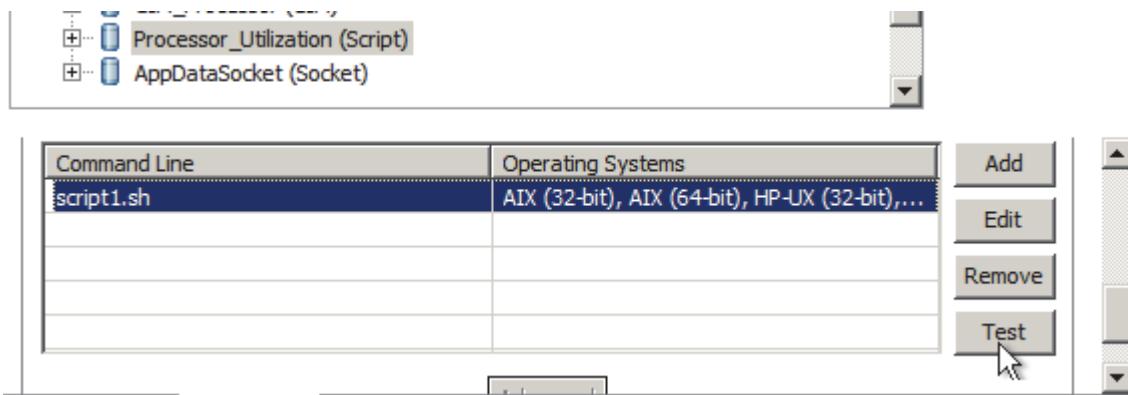
HostName+": "+IP_Address

The Attribute Information window looks like the following screen capture.



41. Test Processor_Utilization and confirm that it returns one row of data and generates the derived attributes.

- a. Select **Processor_Utilization**.
- b. Scroll down in **Command List** and select **script1.sh**.
- c. Click **Test**.



- d. Select **SSH - LIN4** in **Script target system**.
- e. Click **Start Agent** and **Collect Data**.

The results looks like this screen capture.

Test Command

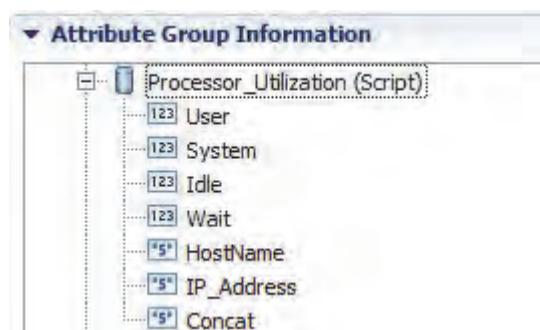
1 data row returned at Feb 27, 2017 5:50:43 PM.

The screenshot shows the 'Test Command' configuration window. The 'Script target system' is set to 'SSH - LIN4'. The 'Command' field contains 'script1.sh'. Under 'Separator', the 'Other character' option is selected with a colon (:) entered. Below the command area are two buttons: 'Start Agent' and 'Collect Data'. The 'Results' section includes a checked checkbox for 'Show hidden attributes'. A table displays the data row: User (1), System (0), Idle (99), Wait (0), HostName (WIN1), IP_Address (192.168.1.103), and Concat (WIN1: 192.168.1.103).

f. Click **OK** to close the Test Command window.

g. Click **OK** to close the Command Information window.

The final Processor_Utilization data source looks like the following screen capture:



42. Save your agent project.

Enable the agent to monitor local and remote sockets

In this section, you add a data source that receives data from an application through a socket connection. You install the agent locally and remotely to the monitored application to see the different configuration requirements.

The application sends four data points: a message and three numeric values.

Message	MsgCode	MsgCnt	QueueSize
A message from perl host name	1	2	123
More from perl	456	123	789

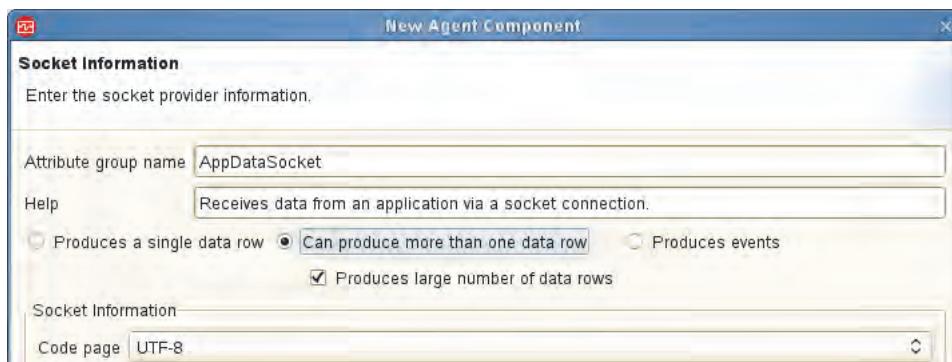
Furthermore, the agent must be able to process two error codes: one informational and the other an error.

43. On the **Data Sources** tab, right-click **AB2 (Agent)** and select **Add Data Source**.
44. On the Agent Initial Data Source page or the Data Source Location page, click **Custom programs** in the Monitoring Data Categories area.
45. In the Data Sources area, click **Socket**.



46. Click **Next**.

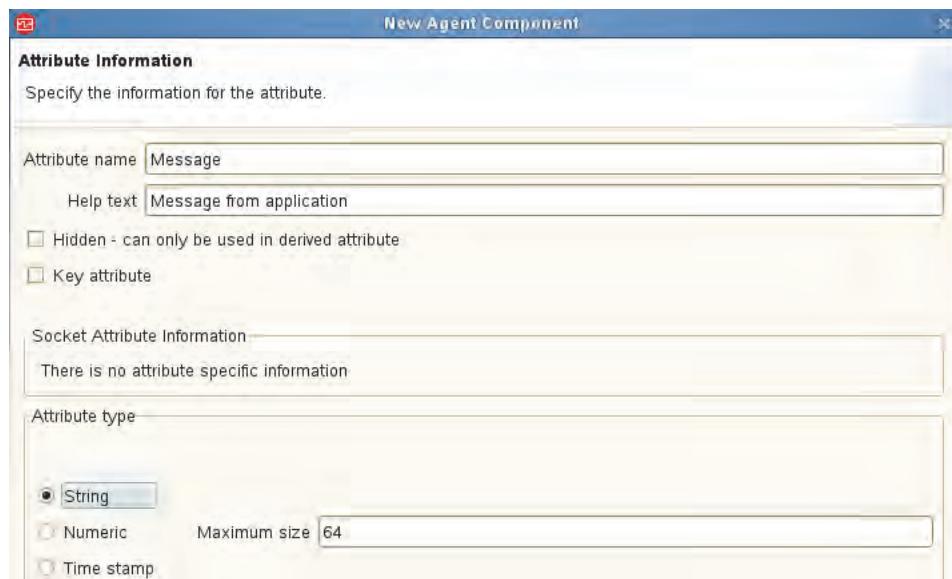
47. On the Socket Information page, enter the following information about the first attribute.
 - Attribute group name: **AppDataSocket**
 - Help text: **Receives data from an application via a socket connection.**
 - Can produce more than one data row: **Selected**
 - Code page: **UTF-8**



48. Click **Next**.

49. On the Attribute Information page, enter the following information about the first attribute:

- Attribute name: **Message**
- Help text: **Message from application**
- Hidden: **Not selected**
- Key attribute: **Not selected**
- Attribute type: **String**
- Maximum size: **64**



50. Click **Next**.



51. Create the following error codes:

Display value	Internal value	Error type	Message text
APPINFO	1001	Informational	Informational message.
APPERROR	1002	Error	Error message.

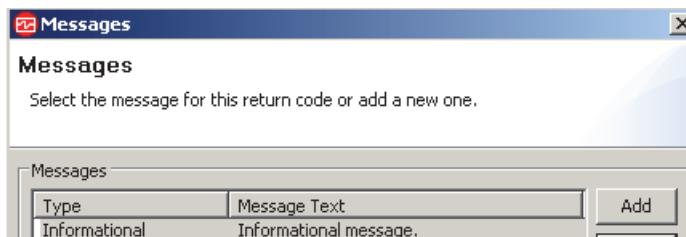
- a. Click **Add**.
- b. Enter the **Display value** and the **Internal value**.



- c. Click **Browse**.
- d. Click **Add**.
- e. Set the message type in the **Type** list and enter the message text in the **Message text** field.



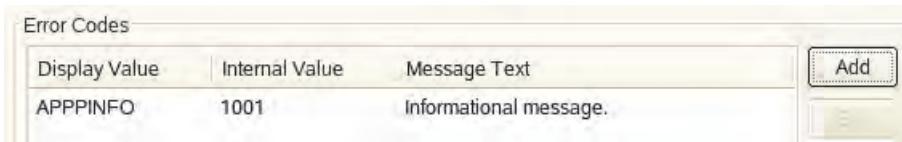
- f. Click **OK** to save the message definition.



- g. Click **OK** to close the Messages window.

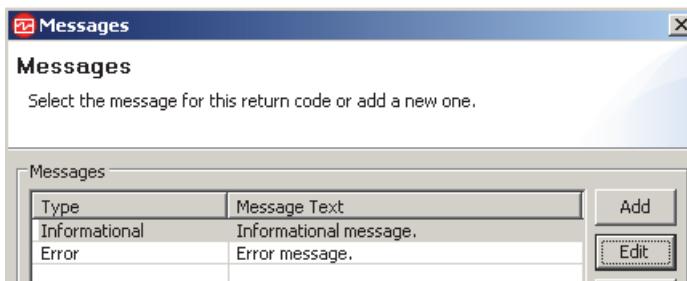


- h. Click **OK** to save the error code.

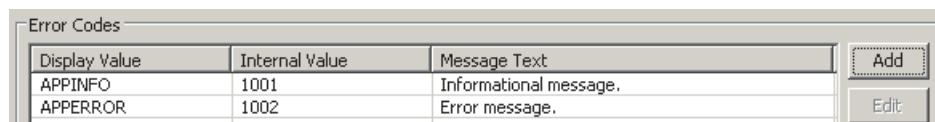


- i. Repeat these steps for the second message.

When you create the second message, the Messages window looks like the following screen capture.



When you create the second error code, the Error Codes section of the Global Socket Data Source Information window looks like the following screen capture.



52. Click **Finish**.

The wizard closes and the **Data Sources** tab opens.

Data Source Definition

Attribute Group Information

- AB2 (Agent)
 - Availability filters
 - Availability_Filtered (Filter)
 - AppDataSocket (Socket)**
 - Processor_Utilization (Script)

53. Save your agent project.

54. Add the following three attributes to the AppDataSocket data source.

Attribute name	Help text	Key attribute	Attribute type
MsgCode	Message code.	Yes	Numeric, Gauge, 32-bit
MsgCnt	Message count.	No	Numeric, Gauge, 32-bit
QueueSize	Message queue size.	No	Numeric, Gauge, 32-bit

- Right-click **AppDataSocket (Socket)** and click **Add Attribute**.
- Enter the attribute information and click **Finish**.
- Repeat for remaining two attributes.



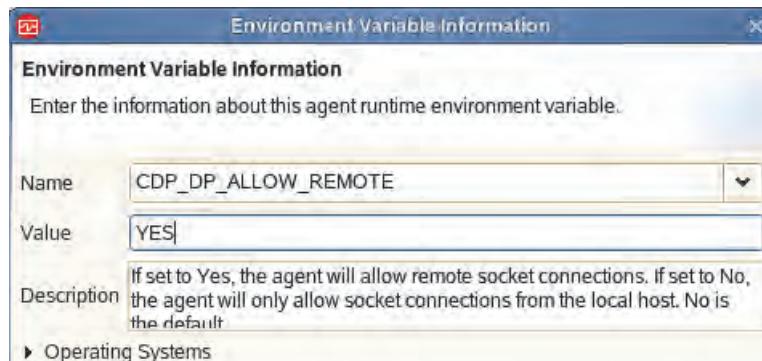
55. Enable the agent to receive socket connections from remote clients.

- Go to the Agent Information page and select **Environment Variables**.



Hint: If necessary, scroll down.

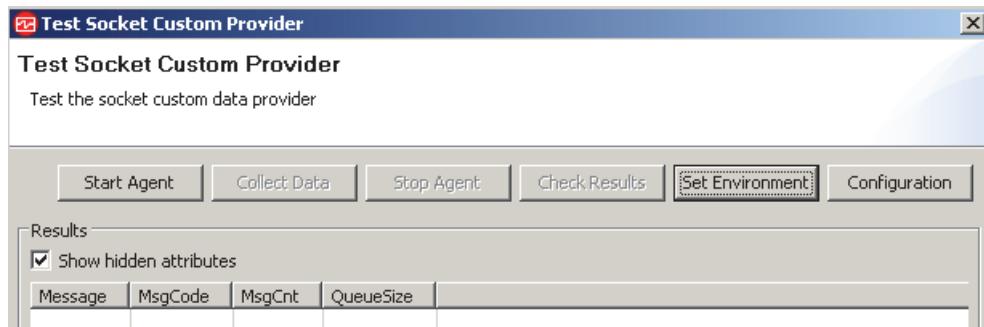
- Click **Add**.
- In the **Name** field, select **CDP_DP_ALLOW_REMOTE** from the list of environment variables.
- Set the **Value** field to **YES**.



- Click **OK** to save.

56. Test the AppSocketData data source.

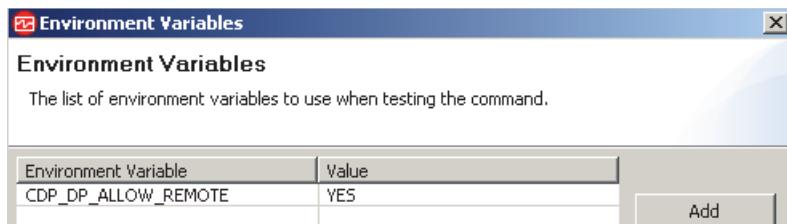
- Go to the **Data Sources** tab, click **AppDataSocket (Socket)**.
- Click **Test**. If necessary, scroll down.



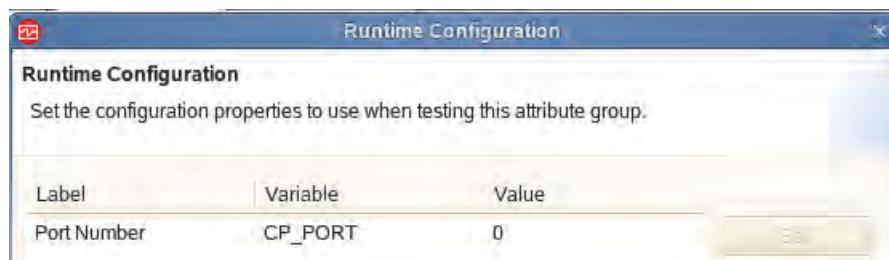
- Click **Set Environment** and confirm the CDP_DP_ALLOW_REMOTE variable is set to **Yes**.



Note: You might set this variable here for the test session.



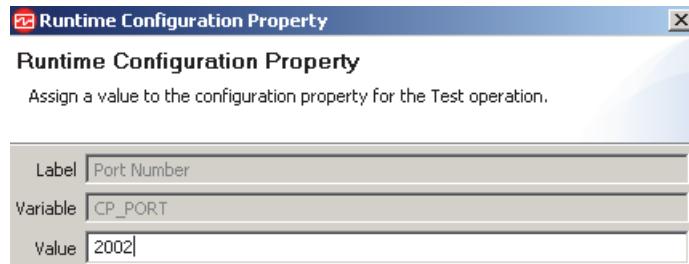
- Click **OK** to close the Environment Variables window.
- Click **Configuration**.



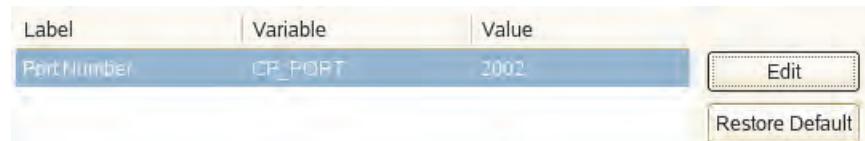
A CP_PORT value of 0 means that the agent uses any available port and writes that port number to the **k02_cps.properties** file. For the test to work, either your application must be able to read the **k02_cps.properties** file to send its data to that port, or you must configure the agent to listen on the port that the application sends to.

- Click the **Port Number** row and click **Edit**.

- g. Change the port value to **2002**.



- h. Click **OK** to save the new port number.



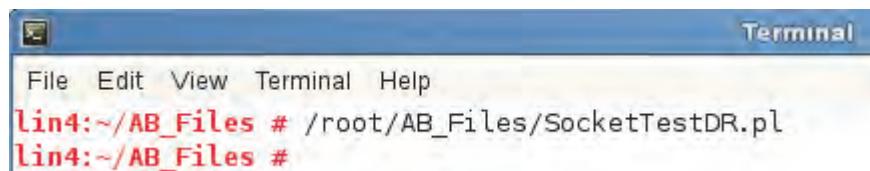
- i. Click **OK** to close the Runtime Configuration window.

- j. Click **Start Agent**.

The monitored application is on LIN4 and must connect to this agent here on WIN1.

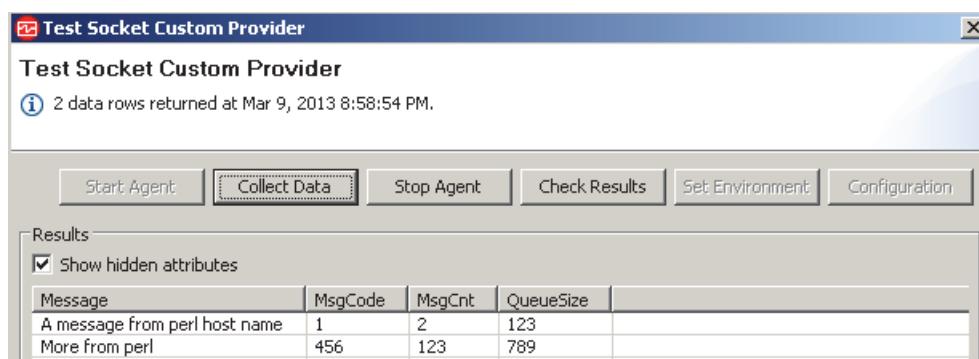
- k. On LIN4, open a terminal window and run this command.

```
/root/AB_Files/SocketTestDR.pl
```



SocketTestDR.pl sends two rows of data to WIN1 through port 2002.

- l. Return to Agent Builder on WIN1 and click **Collect Data**.



- m. Confirm that two rows of data are returned with values in each column that matches the **SocketTestDR.pl** script output.

- n. Click **OK** to close the test window.

57. Save your agent project.

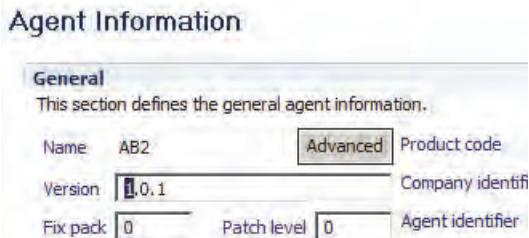
You successfully added a socket data source and enabled it to gather and monitor data from an application with a socket connection.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

58. Select the **Agent Information** tab.

59. Change the version for 1.0.0 to **1.0.1**.



Important: If your initial version number is not 1.0.1, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

60. Save your agent project.

Exercise 2 Install and confirm the updated AB2 agent in an IBM Tivoli Monitoring environment

In this exercise, you test your Custom Script and Socket agent and confirm that data is being gathered in an IBM Tivoli Monitoring environment.

Create the updated AB2 agent installation scripts

1. In Agent Builder on WIN1, generate the AB2 agent installation scripts. (See “[Create the agent installation scripts](#)” on page 20 for detailed steps.)
2. Delete the contents of the **C:\share\K01** directory.
3. Copy the .zip installation archive in the **C:\Users\Administrator** directory into the **C:\share\K01** directory.
4. Extract the .zip installation archive into the K01 directory.



Hint: From Windows Explorer, right-click the **smai-ab2-01.01.00.00.zip** file and select **7-Zip > Extract Here**.

You successfully generated your installation archives and extracted an installer to the WIN1 share.

Reinstall the AB2 application support on ITM

In this section, you install the updated AB2 application support onto the ITM server.

5. On ITM, exit the TEP client. To avoid display problems in the Tivoli Enterprise Portal, close the TEP client during an application support installation.
6. If needed, re-create the drive map to **\WIN1\share**.
7. Open a command prompt and change to the **Y:\K01** directory.

8. Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00  
installIraAgentTEPS.bat C:\IBM\ITM
```

```
Y:\K00>cd ..\k01  
Y:\K01>installIraAgentTEMS.bat C:\IBM\ITM -h itm -u sysadmin -p object00  
Validating user...  
KUIC00007I: User sysadmin logged into server on https://itm:3661.  
The requested service has already been started.  
More help is available by typing NET HELPMSG 2182.  
Install of K01 TEMS Support successful.  
Y:\K01>installIraAgentTEPS.bat C:\IBM\ITM  
Online help for this agent will not be available until the Help Server is restarted,  
which also requires restarting the TEPS.  
Install of K01 TEPS Support successful.  
Y:\K01>
```

The full installation takes 7 - 10 minutes to complete. Status information is displayed for each command.

You completed the installation of the updated AB2 agent application support on the ITM server.

Reinstall the AB2 agent on WIN1

In this section, you install the updated AB2 agent on WIN1 with the script installers.

9. On WIN1 open a command prompt and run the following commands:

```
cd \share\K01  
installIraAgent.bat C:\IBM\ITM
```

10. Configure the agent.

- a. Locate the AB2 agent in the MTMS utility. If necessary, refresh the view.
- b. Confirm that it is the correct version.

- c. Right-click the agent and select **Reconfigure**.

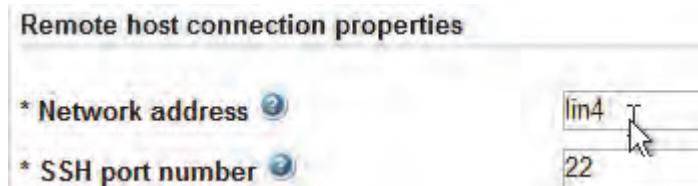


You are prompted to configure the TEMS connection.

- d. Ensure that **ITM** is the target TEMS and click **OK**.

The Agent Configuration window opens.

- e. Enter **LIN4** in **Network address**.



- f. Click **Next**.

- g. Enter the following values in the Password section:

- ◆ Username: **root**
- ◆ Password: **object00**
- ◆ Confirm password: **object00**

The credentials used to make the connection

Username	<input type="text" value="root"/>
Password	<input type="password" value="*****"/>
Confirm Password	<input type="password" value="*****"/>

- h. Click **Next**.

- i. Enter **2002** in the **Port Number** field.

Socket Data Source

Port Number	<input type="text" value="2002"/>
-------------	-----------------------------------

- j. Click **OK** to close the agent configuration window and save your changes.
11. Start the agent.

You completed the installation of the updated AB2 agent on the WIN1 server.

Reinstall the AB2 agent on LIN4

In this section, you install the updated AB2 agent onto the LIN4 system from the WIN1 share directory.

12. On LIN4, change to the **/mnt/share/K01** directory where you extracted the agent on the WIN1 server.
13. Install the agent on this computer by running the following command:

```
./installIraAgent.sh /opt/IBM/ITM
```

```
Lin4:~/AB Files # cd /mnt/share/K01
Lin4:/mnt/share/K01 # ./installIraAgent.sh /opt/IBM/ITM/
Installing k01 .....
Processing. Please wait...
Stopping Monitoring Agent for AB2 ...
Product Monitoring Agent for AB2 was stopped gracefully.
Agent stopped...
copying agent config files now....
Processing. Please wait...
Starting Monitoring Agent for AB2 ...
Monitoring Agent for AB2 started
Install of K01 Agent successful.
To configure the agent, run:
  /opt/IBM/ITM//bin/itmcmd config -A 01
To start the agent, run:
  /opt/IBM/ITM//bin/itmcmd agent start 01
To stop the agent, run:
  /opt/IBM/ITM//bin/itmcmd agent stop 01
Lin4:/mnt/share/K01 #
```

14. Configure the agent on LIN4.
 - a. Open a terminal window and start the agent configuration by running the following commands:

```
cd /opt/IBM/ITM/bin
./itmcmd config -A 01
```

```
Lin4:/mnt/share/K01 # cd /opt/IBM/ITM/bin/
Lin4:/opt/IBM/ITM/bin # ./itmcmd config -A 01
Agent configuration started...
Edit "Monitoring Agent for AB2" settings? [ 1=Yes, 2=No ] (default is: 1):
```

- b. Press **Enter** to confirm that you want to configure the agent.

You are prompted for the host name or IP address of the system where the monitoring script (**script1.sh**) is run.

- c. Enter **LIN4** and press Enter.

```
SSH Remote Connection :  
Remote host connection properties  
  
Host name or IP address of remote system  
Network address (default is: ): lin4
```

- d. Press Enter to accept port 22.

```
The port number used for SSH communication.  
SSH port number (default is: 22): █
```

- e. Type **1** and press Enter to set a password.

```
The type of authentication to use to make the connection  
Authentication Type [ 1=Password, 2=Public Key ] (default is: ): 1
```

- f. Press Enter to select **No** to disconnecting after each collection interval.

```
Disconnect from the remote system after each data collection interval. By default, the SSH communication socket will establish an authenticated session at the first connection and will leave the connection active until the agent is stopped.  
Disconnect from the remote system after each collection interval [ 1=Yes, 2=No ] (default is: 2):
```

- g. Press Enter to select **No** to deleting the script after each collection interval.

```
Delete the script(s) from the remote system after each data collection interval.  
By default, the script(s) will be uploaded to the remote system at the first connection and will only be refreshed if the local copy changes. The script(s) will be removed from the remote system immediately before the agent is stopped.  
Remove script(s) from the remote system after each collection interval [ 1=Yes, 2=No ] (default is: 2):
```

- h. Enter **root** for the user name and press Enter.

```
Password :  
The credentials used to make the connection  
  
Username for the remote system  
Username (default is: ): root█
```

- i. Enter **object00** twice for the password.

```
Password used for the remote system.  
Enter Password (default is: ):  
  
Re-type : Password (default is: ):  
█
```

You are prompted for the socket data source port. Since the socket application runs locally and it is configured to read the **k02_cps.properties** file, you can use the default port of 0, which uses any available port.

- j. Do not enter a value and press Enter to keep the default value of 0.

```
Socket :  
Socket Data Source
```

The port that the agent will use to listen on for data from socket clients. A value of 0 indicates an ephemeral port will be used.

Port Number (default is: 0):

You are asked if the agent connects to a TEMS.

- k. Enter 1 and press Enter to answer Yes.

```
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1): 1
```

- l. Press Enter to accept each of the following default values:

- ◆ ip.pipe as the Network Protocol
- ◆ Not to install a Network Protocol 2

- m. Confirm that **ITM** is the **TEMS Host Name** and press Enter.

```
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):  
  
Network Protocol [ip, sna, ip.pipe, ip.spipe, ip6, ip6.pipe or ip6.spipe] (Default is: ip.pipe):  
  
Now choose the next protocol from one of these:  
- ip  
- sna  
- ip.spipe  
- ip6  
- ip6.pipe  
- ip6.spipe  
- 0 for none  
Network Protocol 2 (Default is: 0):  
TEMS Host Name for IPv4 (Default is: ITM):
```

n. Press Enter to accept each of the following default values:

- ◆ Port of 1918
- ◆ Null as the KDC_PARTITION name
- ◆ No to configuring a connection to a secondary TEMS
- ◆ 0 for an Optional Primary Network Name
- ◆ Disable HTTP

```
IP.PIPE Port Number (Default is: 1918):  
Enter name of KDC_PARTITION (Default is: null):  
  
Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):  
Enter Optional Primary Network Name or 0 for "none" (Default is: 0):  
Disable HTTP? [1=YES, 2=NO] (Default is: 2):  
Agent configuration completed...  
Would you like to restart the component to allow new configuration to take effect? [1=Yes, 2=No] (Default is: 1): █
```

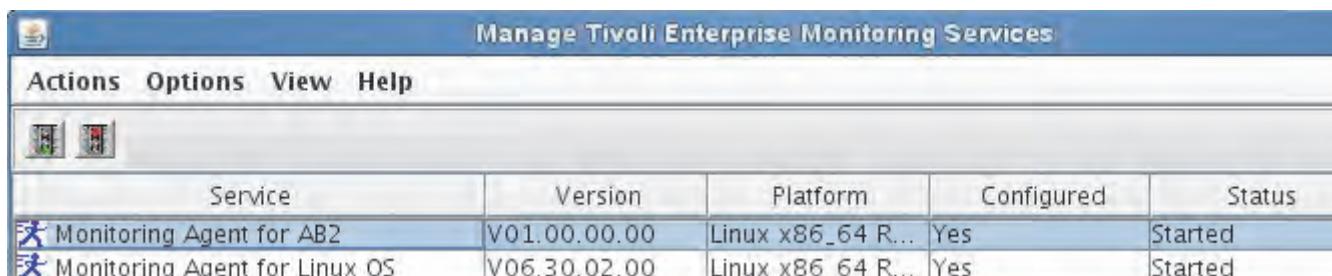
You receive a prompt that indicates the agent configuration is complete and prompted to restart the agent.

15. Press Enter to restart the agent.

```
Would you like to restart the component to allow new configuration to take effect? [1=Yes, 2=No] (Default is: 1):  
Stopping Monitoring Agent for AB2  
Monitoring Agent for AB2 stopped  
Starting Monitoring Agent for AB2  
Monitoring Agent for AB2 started  
lin4:/opt/IBM/ITM/bin # █
```

16. Start the MTMS utility and confirm the agent status by running the following command:

```
/opt/IBM/ITM/bin/itmcmd manage &
```



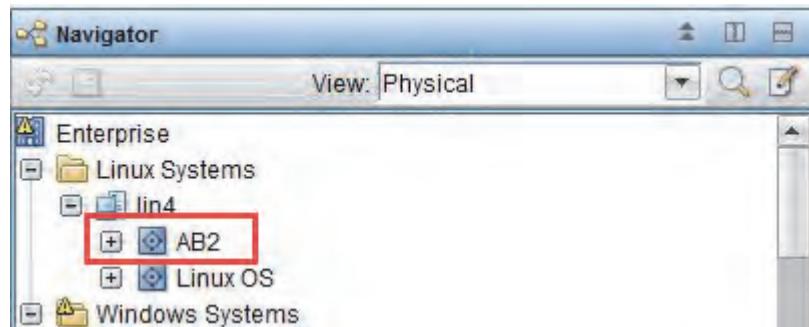
Manage Tivoli Enterprise Monitoring Services				
Actions Options View Help				
Service	Version	Platform	Configured	Status
Monitoring Agent for AB2	V01.00.00.00	Linux x86_64 R...	Yes	Started
Monitoring Agent for Linux OS	V06.30.02.00	Linux x86_64 R...	Yes	Started

You successfully installed the agent on LIN4.

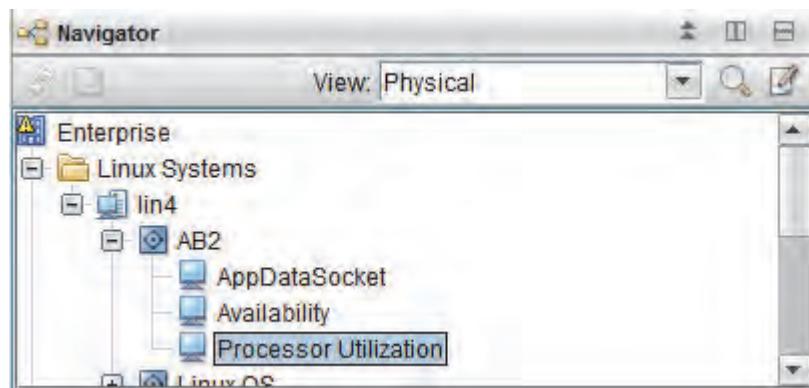
Confirming the script data source on the LIN4 and WIN1 agents

17. On the ITM server, restart the TEP client and log in as **sysadmin**.

18. If prompted at any time, click **Navigator update pending** to update the navigator with the new navigator items.
19. Expand the Navigator tree to display **LIN4** under Linux Systems.



20. If AB2 is not displayed, wait a few minutes for a Navigator update. If the Navigator does not update in 3 minutes, restart the agent on LIN4. If agent still does not show, reconfigure the agent and ensure that you set the Tivoli Enterprise Monitoring Server host name correctly to ITM.
21. After AB2 is visible, expand it and click **Processor Utilization**.



If the AB2 nodes do not display correctly, restart the TEP client or Tivoli Enterprise Portal Server.

22. Confirm that the data in the Report view is valid.

Node	Timestamp	User	System	Idle	Wait	HostName	IP Address	Concat
lin4:01	03/08/17 18:57:30	0	0	100	0	lin4	192.168.1.107	lin4: 192.168.1.107

23. Wait 30 seconds and press F5 to refresh the workspace and post new data.
24. If no data is displayed, check out the Performance Object Status for any errors that are generated by the script data source.

25. Roll your mouse over the heading of each attribute you define.

Report								
Node	Timestamp	User	System	Idle	Wait	HostName	IP Address	
lin4:01	03/08/17 18:57:30	0	0	100	0	lin4	192.168.1.10	

Notice that the help text that you entered for each attribute is displayed,

26. Repeat the steps in this section and confirm that Processor Utilization script data is displayed correctly for the AB2 agent on the WIN1 server.

This view has not been saved
This is the default workspace for this node. To save this workspace, enter a URL in the address text box to the right and click Save.

Hands-on practice and guidelines
[Tutorial: Defining a workspace](#)

Node	Timestamp	User	System	Idle	Wait	HostName	IP Address	Concat
WIN1:01	03/08/17 19:00:24	0	0	100	0	WIN1	192.168.1.103	WIN1: 192.168.1.103

Note: Remember, this agent runs on the WIN1 server and monitors the processor utilization on the LIN4 server.

You successfully defined an agent that deploys the **script1.sh** file, runs it, and returns that data to Tivoli Enterprise Portal.

Confirm the socket data source on the LIN4 and WIN1 agents

27. Create monitoring data for the socket data source of your agents. Use the Perl scripts on LIN4 to send data to the AppDataSocket data source on both the LIN4 and WIN1 servers.

- On LIN4, run the following commands to send monitoring data through a socket to the WIN1 and LIN1 agents:

- ◆ /root/AB_Files/SocketTestDL.pl
- ◆ /root/AB_Files/SocketTestDR.pl

```
lin4:~/AB_Files # /root/AB_Files/SocketTestDL.pl
lin4:~/AB_Files # /root/AB_Files/SocketTestDR.pl
lin4:~/AB_Files #
```

The DL script sends monitoring data to the local host (LIN4) and determines the correct port by reading the k01_cps.properties file that is generated by the agent. The DR script sends monitoring data to the WIN1 host through port 2002.

- If you are interested in the commands that are used to determine the port and send the data, open each script in a text editor. They represent the work that you must do within your application to enable it to send data to a socket data source.



Note: To see an example of the coding that is required to send a message from an application to an Agent Builder agent with a socket data provider, open each script in a text editor.

28. Return to the ITM server and confirm that the AB2 agent on LIN4 and WIN1 gathers the socket data.

Node	Timestamp	Message	MsgCode	MsgCnt	QueueSize
WIN1:01	03/08/17 20:31:41	A message from perl host name	1	2	123
WIN1:01	03/08/17 20:31:41	More from perl	456	123	789

You successfully built an agent to gather and monitor data from an application with a socket connection.

Exercise 3 Create an agent to monitor a log file and a custom Java data source

In this exercise, you do the following tasks:

- Create an agent that extracts data from a log file
- Create an agent that gathers data from a Java application with the Java API data source
- Create a custom runtime configuration property that captures the location of the log file to be monitored
- Install the agent and application support

You complete all steps in this exercise on the WIN1 server.

Determine the log file data for the agent to gather

In this section, you look at the log file from which the agent extracts data to better understand why the agent is configured the way that it is. Complete the following steps:

1. On the WIN1 server, open a command prompt and change to the following directory:
C:\AB_Files\Scripts
2. Use the **type** command to display the contents of the **pinginterval.bat** file.

```
C:\Users\Administrator>cd \AB_Files\Scripts
C:\AB_Files\Scripts>type pinginterval.bat
:START
c:\windows\system32\ping -w 100 LIN4>>c:\AB_Files\Output\ping.txt
c:\windows\system32\ping -w 100 WIN1>>c:\AB_Files\Output\ping.txt
GOTO START
C:\AB_Files\Scripts>_
```

This script pings two hosts, appends the results to a mock log file (C:\AB_Files\Output\ping.txt) and repeats this process until stopped.

3. Run the **pinginterval.bat** file from a command prompt for 15 seconds to gather log data and then cancel (Ctrl + C, and Y) the batch file.

```
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 LIN4 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 WIN1 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>GOTO START
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 LIN4 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 WIN1 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>GOTO START
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 LIN4 1>>c:\AB_Files\Output\ping.txt
^CTerminate batch job (Y/N)? Y
```

4. Find the **ping.txt** file in C:\AB_Files\Output and view its contents.

You monitor the **ping.txt** log file. A sample of the content of the **ping.txt** log file that is created by **pinginterval.bat** is shown here.

```

ping.txt - Notepad
File Edit Format View Help
Pinging lin4.ibm.edu [192.168.1.107] with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time=1ms TTL=64
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

Pinging win1.ibm.edu [192.168.1.103] with 32 bytes of data:
Reply from 192.168.1.103: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

The circled rows contain the data that you want to extract; specifically, the IP address, packets sent, packets received, and packets lost.



Note: Even though **ping.txt** is not a true log file and Agent Builder has a data source specifically built to monitor devices with ICMP ping, **ping.txt** provides an excellent target for learning how to monitor a log file. With it, you can see some of the complex parsing abilities you can use in an agent.

The goal of this exercise is to create an agent that pulls specific information to help you monitor the status of the two hosts LIN4 and WIN1.

The data that you want the agent to extract from the log file is shown in the following screen capture.

Node	Timestamp	IP Address	Sent	Received	Lost	Status	Avg Lost	HostName	Concat	Path
ITM:03	07/12/15 18:24:31	192.168.1.107	4	4	0	Good	0	lin4.ibm.edu	lin4.ibm.edu: 192.168.1.107	C:\IBM\ITM\GSK8_x64\lib64
ITM:03	07/12/15 18:24:28	192.168.1.103	4	4	0	Good	0	win1.ibm.edu	win1.ibm.edu: 192.168.1.103	C:\IBM\ITM\GSK8_x64\lib64
ITM:03	07/12/15 18:24:25	192.168.1.107	4	4	0	Good	0	lin4.ibm.edu	lin4.ibm.edu: 192.168.1.107	C:\IBM\ITM\GSK8_x64\lib64
ITM:03	07/12/15 18:24:22	192.168.1.103	4	4	0	Good	0	win1.ibm.edu	win1.ibm.edu: 192.168.1.103	C:\IBM\ITM\GSK8_x64\lib64

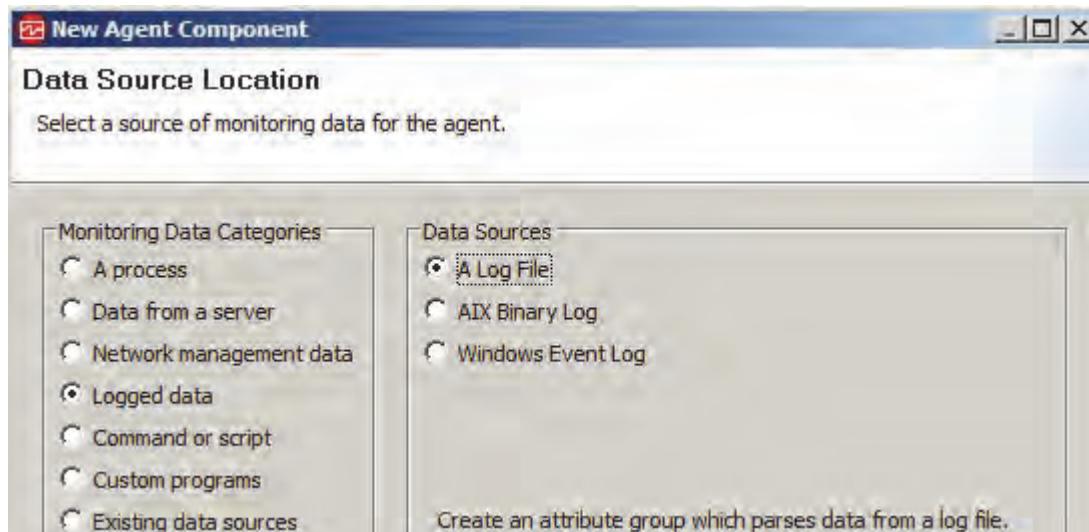
The agent must do the following parsing tasks to the log file:

- Recognize multiple lines in the log file as one record
- Extract data that is not delimited in a consistent way
- Ignore all the unwanted lines of data

Modify the AB1 agent to monitor the ping.txt log file

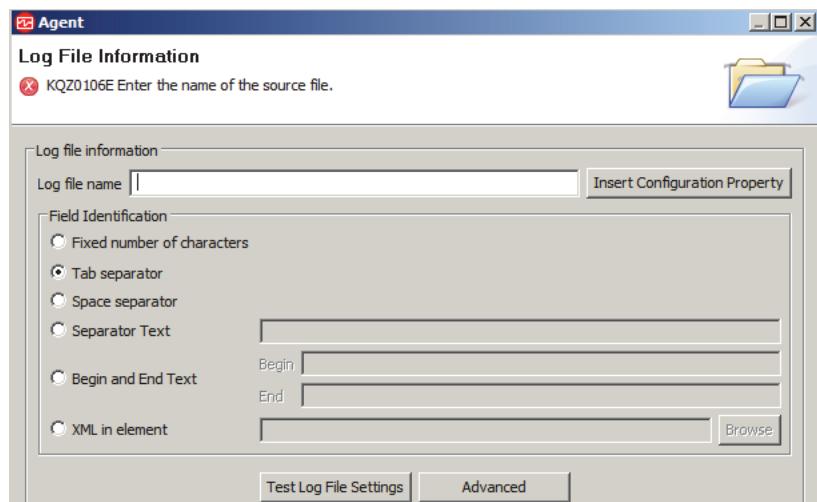
Complete the following steps:

5. If Agent Builder on the WIN1 server is not already open, start it.
6. Close the **AB2 Agent Editor** window if it is open.
7. If the **AB1** agent is not already open, expand **Agent 1** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
8. Click the **Data Sources** tab.
9. Right-click **AB1 (Agent)** under **Attribute Group Information** and click **Add Data Source**.
10. Select **Logged data** under Monitoring Data Categories and **A Log File** under Data Sources.



11. Click **Next**.

The Log File Information window opens.



First, you identify the log file to monitor. Enter the log file name in one of the following ways:

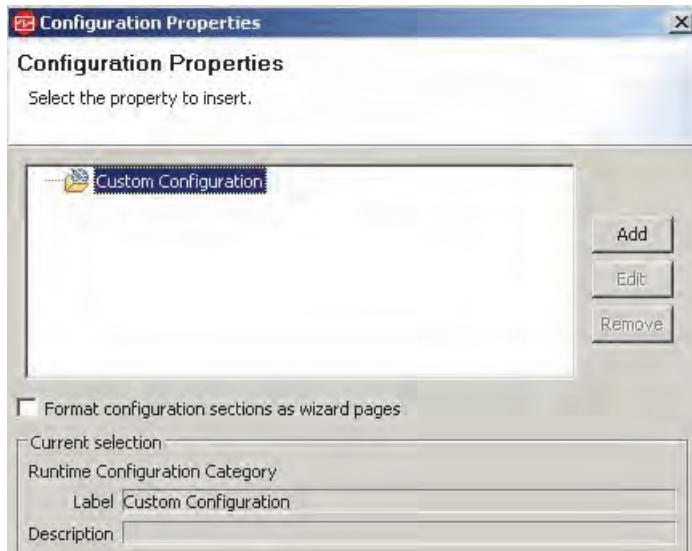
- **Full path and file name:** If the log file on each host to be monitored is the same name and in the same location, enter the full path and file name in the **Log file name** field.
- **Different paths or file names per platform:** If the file name or path to the file differs for different operating systems, you can identify the different files names and paths in the Advanced Log File Attribute Group Information window. Each log file entry requires a unique label.
- **Entered at agent configuration:** Part or all the log file path and name can come from a runtime configuration property that is entered in the agent configuration panes after the agent is installed on a specific host. To do this, click **Insert Configuration Property**.
- **Dynamic File Name:** For applications that create output files with dynamic file names, you can configure the **File log name** field to match patterns. For example, **IN{#####}.log** matches file names that start with IN followed by six numeric characters and file extension **.log**.
- **Regular Expression:** Like dynamic file naming, you can use regular expressions to define a pattern to select the correct log file or files to monitor.

You use a simple full path and file name.

12. Create and insert a runtime configuration property that inserts the location of the **ping.txt** log file. This property is set during agent configuration with a value that is appropriate for the agent host.

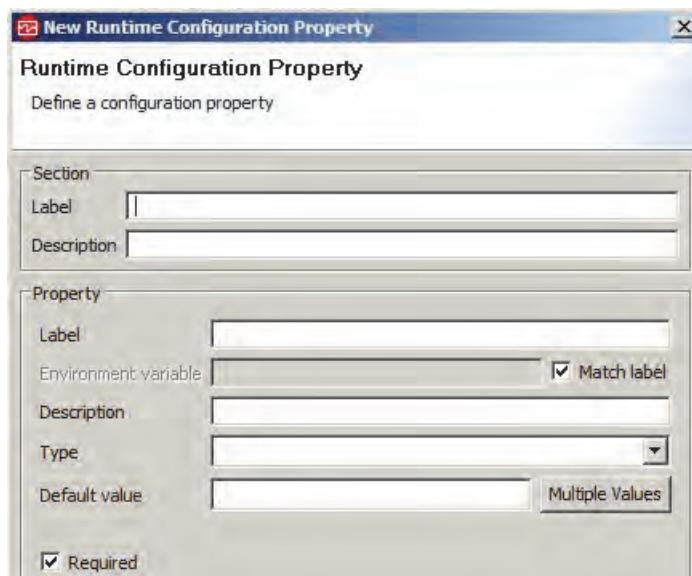
- a. Click in the **Log file name** field and click **Insert Configuration Property**.

The Configuration Properties window opens.



- b. Click **Add**.

The Runtime Configuration Property window opens.



- c. Enter the following information:

◆ Section

Label: **Log_File**

Description: **Runtime configuration properties for log file monitoring**

◆ Property

Label: **File_location**

Description: **Location of the log file**

Type: **String**

◆ Default value: **C:\AB_Files**



Note: For this exercise, the default value is intentionally entered incorrectly. You correct the value when you install and configure the agent.

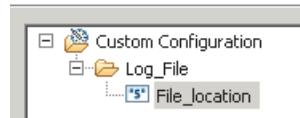
Section		
Label	Log_File	
Description	Runtime configuration properties for log file monitoring	
Property		
Label	File_location	
Environment variable	K00_FILE_LOCATION	<input checked="" type="checkbox"/> Match label
Description	Location of the log file	
Type	String	
Default value	C:\AB_Files	<input type="button" value="Multiple Values"/>
<input checked="" type="checkbox"/> Required		

- d. Click **OK** to save the property.

Your File_location property is listed in the Configuration Properties window.

Configuration Properties

Select the property to insert.



- e. Ensure the **File_location** property is selected and click **OK**.

The Log File Information window is displayed with the File_location property inserted into the **Log file name** field.

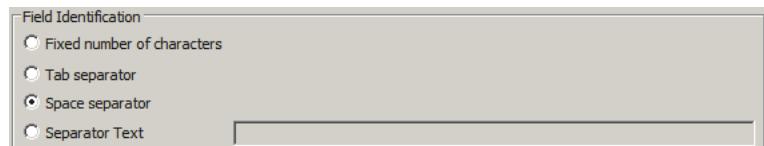
Log file information
Log file name \${K00_FILE_LOCATION}
Field Identification

13. Type \ping.txt in the **Log file name** field after the **File_location** property.

Log file information
Log file name \${K00_FILE_LOCATION}\ping.txt
Field Identification

Field Identification defines the default method of identifying each data point to be retrieved. The default is **Tab**, but the **ping.txt** file uses spaces.

14. Click **Space separator**.



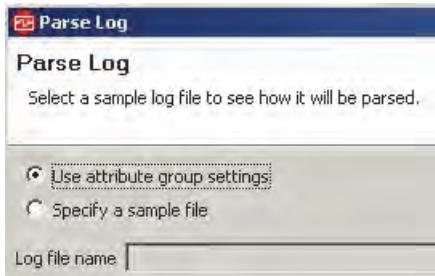
15. Test your log file filter.

- Click **Test Log File Settings**.



Note: If **Test Log File Settings** is unavailable, you missed a setting, such as the space separator from [Step 14](#) on page 6-42.

The Parse Log window opens. Notice that your log file name and path are not displayed.



If you entered a specific directory path and file name, it is listed in the **Log file name** field. You do not have to test the exact path and file name that is used by the agent. If the log file you want to test is different from the one shown in the **Log file name** field, click **Specify a sample file** and change the value to the appropriate path and file name. In this lab exercise, you browse and locate the **ping.txt** file that you created earlier in this exercise.

- Click **Specify a sample file** and **Browse**.
- Locate and click the **C:\AB_Files\Output\ping.txt** file.



- Click **Open** to select the file and close the browser.

You are returned to the Parse Log window.



- Click **Start Agent**.

A status window opens and closes. You are notified that the agent is started.

The test agent has been started. Log files can be found in C:\DOCUMENTS\ADMINISTRATOR\LOCALS\Temp\KQZ_1352920363564\TMAITM6\logs.

f. Click **Collect Data**.

g. If prompted, click **No** to view suggested data types.

The resulting data is displayed.

The screenshot shows the 'Parse Log' application interface. At the top, it displays 'Parse Log' and a message indicating '2702 data rows returned at May 12, 2016 6:23:17 PM.' Below this, there are two radio button options: 'Use attribute group settings' (unchecked) and 'Specify a sample file' (checked). A 'Log file name' input field contains 'C:\AB_Files\Output\ping.txt' with a 'Browse' button to its right. A row of buttons includes 'Start Agent' (disabled), 'Collect Data' (highlighted in blue), 'Stop Agent', 'Check Results', 'Set Environment', and 'Configuration'. The main area is titled 'Results' and contains a table with the following data:

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7
Pinging	lin4.ibm.edu	[192.168.1.107]	with	32	bytes	of
Reply	from	192.168.1.107:	bytes=32	time<1ms	TTL=641	
Reply	from	192.168.1.107:	bytes=32	time<1ms	TTL=641	

The parser treats each line as a record (row) and creates an attribute that is based on the space separator. The initial goal is to ensure that you can find the line that you want and that all of the data you want is on one line. Find rows that contain the data you want.

The screenshot shows the 'Parse Log' application interface with the 'Results' table. Four specific rows are highlighted with a red border: the first row ('Pinging'), the second row ('Reply'), the third row ('Reply'), and the fourth row ('Approximate'). These rows represent the data extracted from the log file.

Reply	from	192.168.1.107:	bytes=32	time<1ms	TTL=641	
Ping	statistics	for	192.168.1.107:			
Packets:	Sent	=	4,	Received	=	4,
Approximate	round	trip	times	in	milli-seconds:2	

Unless defined otherwise, each row of a log file is considered an independent record. For the **ping.txt** log file, the data that you want is spread across two lines.

```
-- Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging lin4.ibm.edu [192.168.1.107] with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

For this log file, you must define a record as consisting of at least two rows. But notice all the extra lines of data that does not represent exact data that you want to return. Not only must you define how to identify a record, but you also must configure the agent to ignore unwanted data. One way is to expand your record definition to include more rows as part of the record.

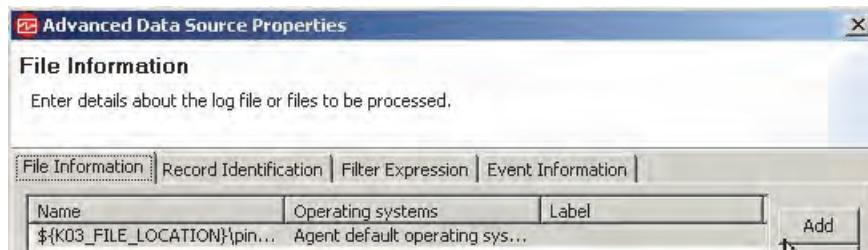
For example, this image shows that the **ping.txt** file uses ten text lines for a record, from Pinging to Minimum.

```
-- Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging lin4.ibm.edu [192.168.1.107] with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging win1.ibm.edu [192.168.1.103] with 32 bytes of data:
Reply from 192.168.1.103: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

16. Configure the agent to identify ten rows as one record.

- Click **Cancel** to exit the Parse Log utility without saving any attribute configurations.
- Click **Advanced** at the bottom of the window.

The Advanced Data Source Properties window opens to the **File Information** tab.

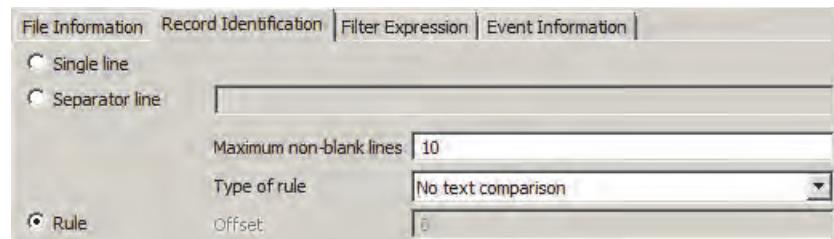


In the **File Information** tab, you enable the agent to process different log files on different operating systems. The other options define how the agent reads the log file and processes the records.

- c. Click the **Record Identification** tab.

The **Record Identification** defines a record.

- d. Click **Rule** in **Record Identification**.



- e. Enter **10** in **Maximum non-blank lines**.

- f. Click **OK** to save these settings.

The Log File Information window is displayed. The log file definition is not yet completed, but you can start testing it.

17. Test your log file filter again (see [Step 15](#) for detailed steps).

The resulting data is displayed.

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7	Attribute_8
Pinging	lin4.ibm.edu	[192.168.1.107]	with	32	bytes	of	data:
Pinging	win1.ibm.edu	[192.168.1.103]	with	32	bytes	of	data:
Pinging	lin4.ibm.edu	[192.168.1.107]	with	32	bytes	of	data:

All the data elements are on a single row in the parser.

18. Scroll right to find the data you want.

```
time<1ms TTL=641 Reply from 192.168.1.107: bytes=32 time<1ms TTL=641 Ping statistics for 192.168.1.107: Packets: Sent = 4, R
time<1ms TTL=1282 Reply from 192.168.1.103: bytes=32 time<1ms TTL=1282 Ping statistics for 192.168.1.103: Packets: Sent = 4,
time<1ms TTL=643 Reply from 192.168.1.107: bytes=32 time<1ms TTL=643 Ping statistics for 192.168.1.107: Packets: Sent = 4, R
time<1ms TTL=1284 Reply from 192.168.1.103: bytes=32 time<1ms TTL=1284 Ping statistics for 192.168.1.103: Packets: Sent = 4,
```

Ensure that you find rows that contain Ping statistics with all your data points on the same row, including the IP address, Sent, Received, and Lost packets.

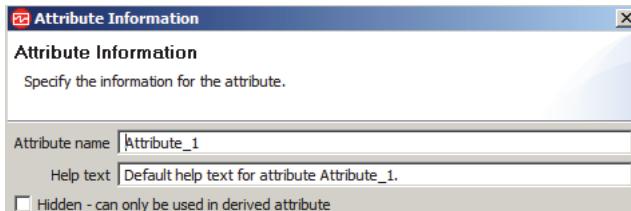
Notice any extra lines of data that do not represent your record or data you want to return. There are many ways to configure the agent to ignore that data.

Next, you define the first four attributes to filter for the correct data. It is best to work from left to right. You want the IP address to display in Attribute_1. The idea is to configure Attribute_1 to ignore data from the start of the record to the data you want in this attribute.

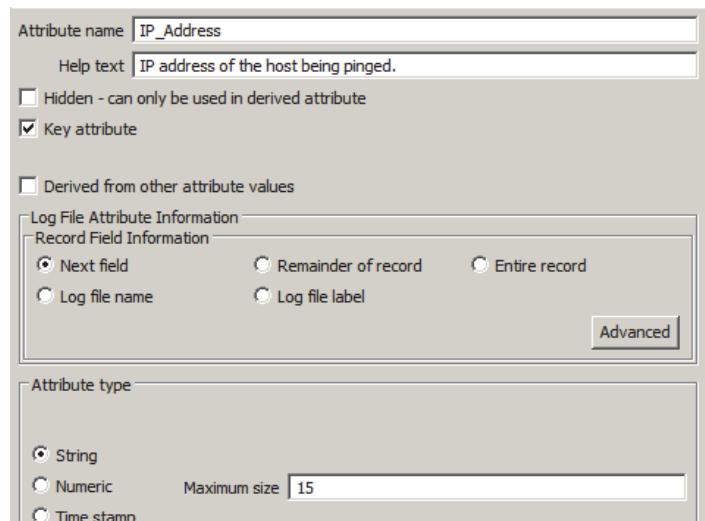
19. Define Attribute_1 so that it is defined correctly for your target data (IP address), is a key attribute, and retrieves the IP address.

- Stop the agent.
- Right-click a cell in the **Attribute_1** column and select **Edit Attribute**.

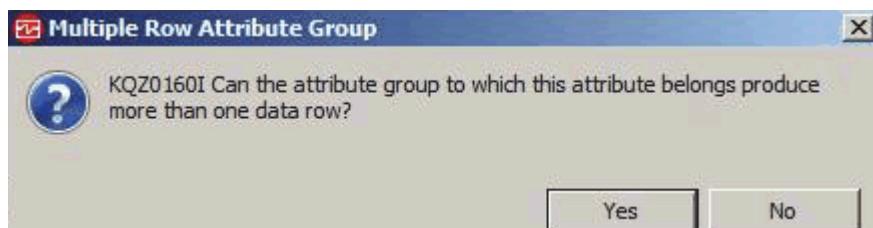
The Attribute Information window opens.



- Modify the following properties:
 - ◆ Attribute name: **IP_Address**
 - ◆ Help text: **IP address of the host being pinged.**
 - ◆ Key attribute: **Selected**
 - ◆ String Maximum size: **15**



- If prompted, select **Yes** the attribute group can produce more than one row.



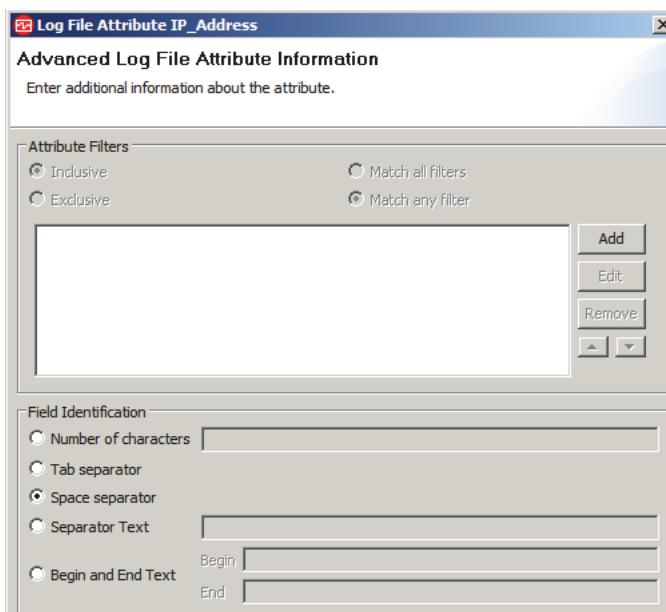
The Record Field Information is set to **Next field** and the attribute type is **String**. Because the global delimiter is a space, this definition does not skip over the first fields to retrieve the

IP address. You need a unique filter. Specify the information that comes before and after the text you want.

```
Reply from 192.168.1.13: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
```

e. Click **Advanced**.

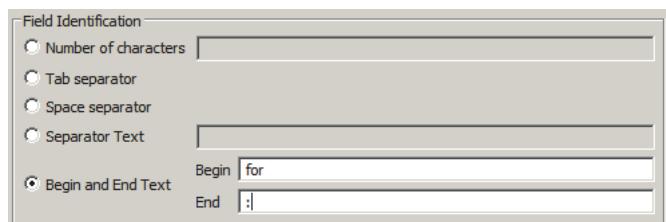
The Advanced Log File Attribute Information window opens.



The Field Identification pane defines how to distinguish the data for this attribute. This setting overrides the global delimiter, which is a space for your agent. In this case, you configure the Field Identification items to get whatever lies between the word **for** and a colon (:).

f. Select **Begin and End Text**.

g. Enter **for** in the **Begin** field and a colon (**:**) in the **End** field.



h. Click **OK** to save the Advance Log File Attribute Information.

i. Click **OK** to save your new IP_Address attribute.

- j. If prompted, click **No** to any prompts to change the data type after redefining an attribute.
 The Parse Log window opens with IP_Address as the first column header.
20. Click **Start Agent** and **Collect Data** to reparse the ping.txt file based on your new attribute definition.
21. If prompted, click **No** to change the data type after redefining an attribute.

Results						
<input checked="" type="checkbox"/> Show hidden attributes						
IP_Address	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7
192.168.1.107	Packets:	Sent	=	4,	Received	=
192.168.1.103	Packets:	Sent	=	4,	Received	=
192.168.1.107	Packets:	Sent	=	4,	Received	=

You now see all rows that start with the IP address and no rows that start with words you do not want.

22. Edit the other attributes so that they select for their specific values and that they are configured to the data they hold. Modify Attributes 2, 3, and 4 to match the information in the following table:

	Attribute_2	Attribute_3	Attribute_4
Attribute name	Sent	Received	Lost
Help Text	Number of packets sent.	Number of packets received.	Number of packets lost.
Key	No	No	No
Attribute Type	Numeric > Gauge	Numeric > Gauge	Numeric > Gauge
Begin delimiter	Sent =	Received =	Lost =
End delimiter	,	,	(



Note: For the Begin delimiter, include a space between *Sent*, *Received*, *Lost*, and the equal sign.

- ◆ Answer **No** to any prompts to change the data type after redefining an attribute.
- ◆ You can click **Start Agent** and **Collect Data** as you finish each attribute definition to ensure that it parses correctly.
- ◆ Click **Stop Agent** before editing the next attribute.

After you define all four attributes, your results look like the following example.

Results									
<input checked="" type="checkbox"/> Show hidden attributes									
IP_Address	Sent	Received	Lost	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	
192.168.1.107	Packets:	Sent	=	4,	Received	=	4,	Lost	

After you parse the log file again, your output looks like the following example.

Results									
<input checked="" type="checkbox"/> Show hidden attributes									
IP_Address	Sent	Received	Lost	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	
192.168.1.107	4	4	0	0%	loss),	Approximate	round	trip	tir

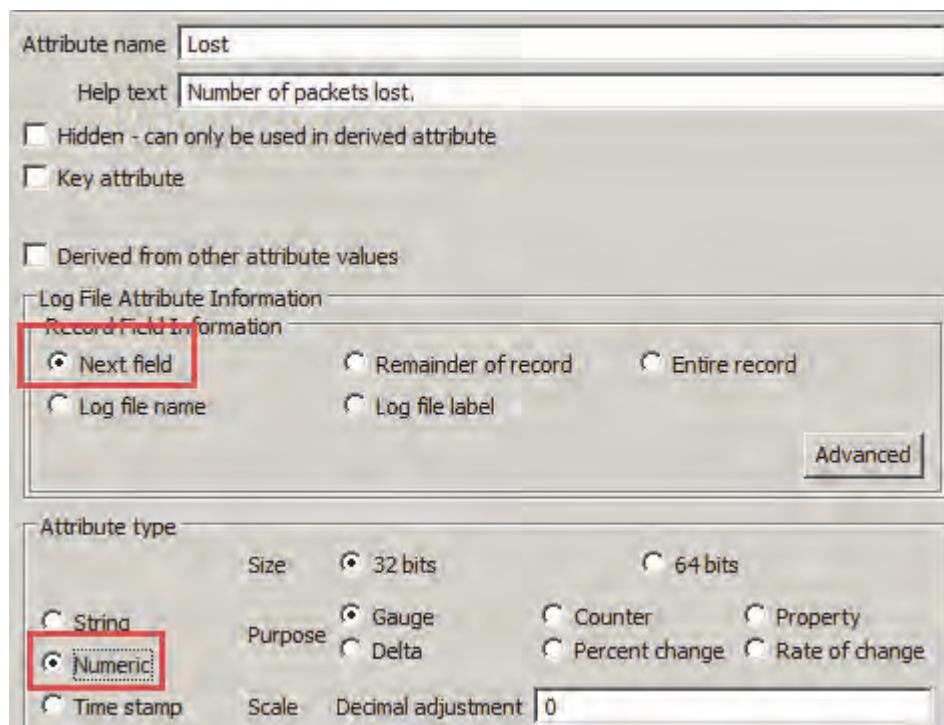
In some instances, you might have blank lines in your Parse Log results. There are several ways to remove blank links. Attribute filters are a common way to remove unwanted rows (records). You might add a filter to IP_Address, or you might set a filter on another attribute. In this instance, you modify the Sent attribute to exclude rows where Sent is equal to zero.

23. Right-click the **Lost** cell for a data row and select **Remove Subsequent Attributes**.

Results									
<input checked="" type="checkbox"/> Show hidden attributes									
IP_Address	Sent	Received	Lost	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	
192.168.1.107	4	4	0	0%	loss),	Approximate	round	trip	tir

Edit Attribute
 Hide Attribute
 Insert Attribute Before
 Insert Attribute After
 Remove
Remove Subsequent Attributes
 Remove All

The final attribute is redefined to hold the remainder of the record. Therefore, it gets changed to String instead of what Lost was, Numeric

24. Reset **Lost** to the **Next field** and **Numeric**.25. Click **OK** to close the Parse Log window and save these attribute definitions.

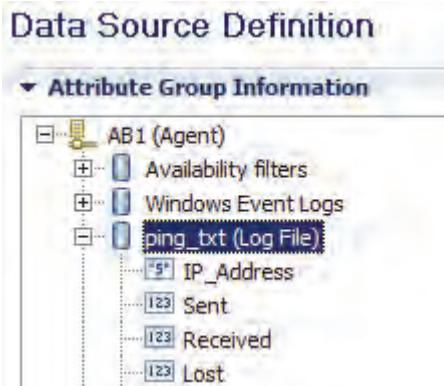
Note: Clicking **OK** to exit is important; it saves your attribute definitions. If you click **Cancel**, you lose the attribute definitions and must return to the Parse Log utility to redo them.

The **Next** and **Finish** buttons are active. You completed enough steps to finish this agent. If you wanted to continue working in the wizard to modify this agent, such as adding more data sources, you would click **Next**.

26. Click **Finish** to exit the Agent wizard and save your changes.

You are returned to the **Data Sources** tab.

27. Expand **ping_txt (Log File)**.



You see your four defined attributes.

28. If you still have unwanted attributes, you can remove them here.

29. Replace the attribute group name, **ping_txt**, with **Host_Availability**.

a. Click **ping_txt (Log File)**.

b. Change **Attribute group name** to **Host_Availability**.

The screenshot shows the 'Attribute Group Information' section of the Data Source Definition tool. The 'ping_txt (Log File)' group has been renamed to 'Host_Availability (Log File)'. The 'Log File Attribute Group Information' panel at the bottom displays the 'Attribute group name' field set to 'Host_Availability'.

c. Press **Enter** to activate your change.

30. Save your agent project.

Add derived attributes

In this section, you add derived attributes to Host_Availability that do the following tasks:

- Give a status of Good or Failure to a pinged server based on the number of received packets.
- Calculate the average number of lost packets for the last 10 pings.



Note: The addition and testing of these derived attributes might be done at the same time you defined the original four attributes in the Parse Log utility. It is done here to show that you can define new attributes and test your log file from the **Data Source** tab in the Agent Editor.

31. Add a derived attribute to Host_Availability according to the following specifications:

- Attribute name: **Status**
- Help text: **Displays status of host based on the number of received pings.**
- Derived Attribute Formula: **Received**
- Attribute Type: **Numeric > Gauge**
- Enumerations:

Name	Value	Severity
Good	4	Normal
Poor	3	Warning
Poor	2	Warning
Poor	1	Warning
Failure	0	Critical

a. Right-click **Host_Availability (Log File)** and click **Add Attribute**.

The Attribute information window opens.

b. Enter the **Attribute name** and **Help text** values.

Attribute name: Status
Help text: Displays status of host based on the number of received pings.
 Hidden - can only be used in derived attribute
 Key attribute

c. Set the attribute type to **Numeric > Gauge** to match the Received attribute.

d. Select **Derived from other attribute values**.

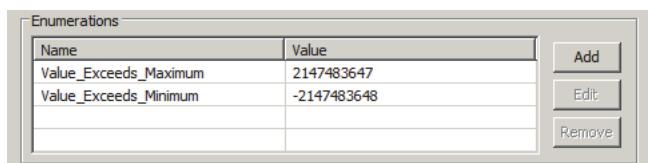
Derived from other attribute values
Derived Attribute Information
Formula: Edit
 Interval specific calculations

- e. Enter **Received** in the **Formula** field.

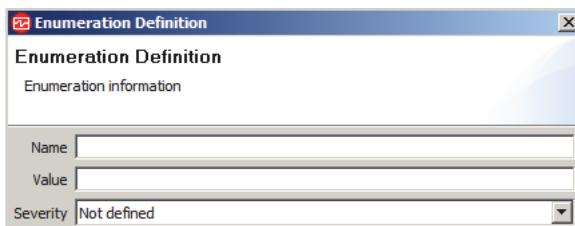


This action sets the new attribute to the value of the Received attribute.

- f. In the Enumerations section, click **Add**.

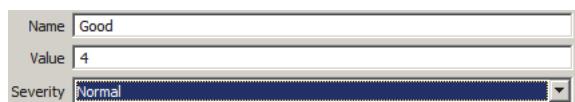


The Enumeration Definition window opens.



- g. Enter the following values:

- ◆ Name: **Good**
- ◆ Value: **4**
- ◆ Severity: **Normal**

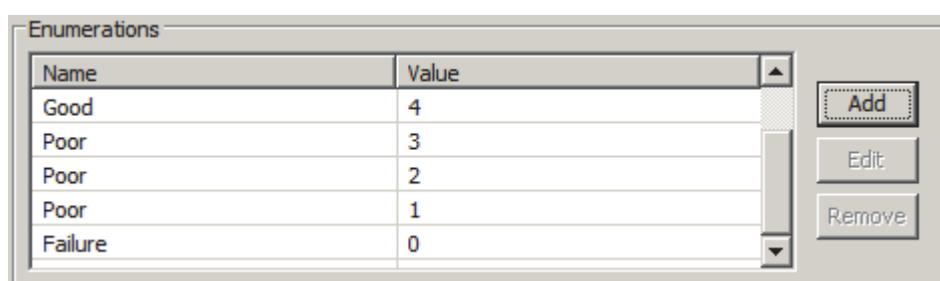


If Received is 4, Status is Good.

- h. Click **OK** to save the enumeration.

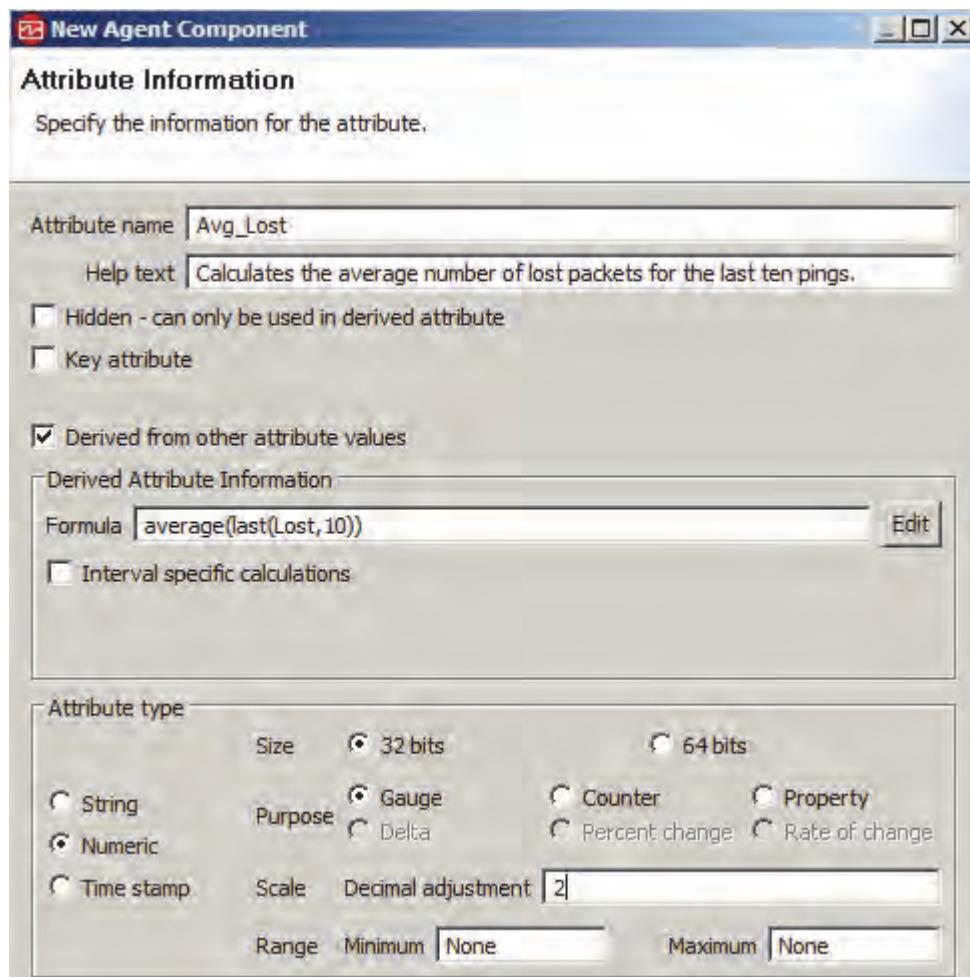
- i. Repeat the previous three steps for the other enumerations.

Your final Enumerations pane looks like the following example (order is not important).



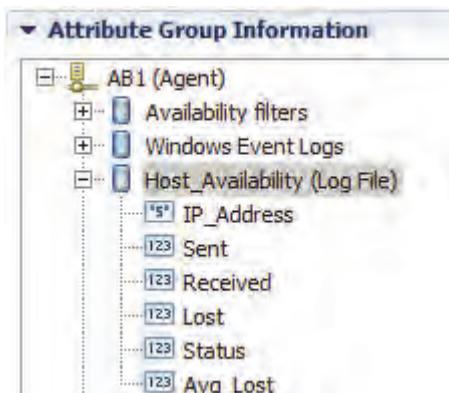
- j. Click **Finish** to close the Attribute Information window and save the **Status** attribute.
32. Add a derived attribute to Host_Availability according to the following specifications:
 - Attribute name: **Avg_Lost**
 - Help text: **Calculates the average number of lost packets for the last ten pings.**
 - Derived Attribute Formula:
`average(last(Lost,10))`
 - Attribute Type: **Numeric, Gauge, 32-bit, Decimal adjustment: 2**
- a. Right-click **Host_Availability (Log File)** and click **Add Attribute**.
The Attribute information window opens.
- b. Enter the **Attribute name** and **Help text** values.
- c. Set the attribute type to **Numeric > Gauge**.
- d. Set **Decimal adjustment** to **2**.
- e. Select **Derived from other attribute values**.
- f. Enter the following formula in the **Formula** field.
`average(last(Lost,10))`

The Attribute Information window looks like the following screen capture.



33. Click **Finish** to close the Attribute Information window.

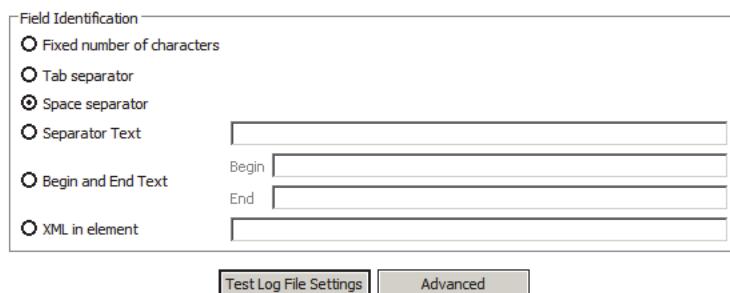
The new attributes are displayed on the **Data Source Definition** tab.



34. Save your agent project.

35. Test your log file parsing.

- Click **Host_Availability (Log File)**.
- Scroll down and click **Test Log File Settings** in the **Log File Attribute Group Information** section.



The Parse Log window opens.

- Click **Specify a sample log file**.
- Browse and locate the **ping.txt** file.
- Start the agent and collect data.

Your parsed log file looks like the following example.

Results						
<input checked="" type="checkbox"/> Show hidden attributes						
IP_Address	Sent	Received	Lost	Status	Avg_Lost	Attribute_7 (Remainder of record)
192.168.1.107	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
192.168.1.103	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
192.168.1.107	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
192.168.1.103	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms

Status shows Good for any rows with a Received value of 4 and Failure for any rows with a Received of 0. For rows with a Received value of 1, 2, or 3, the actual value is shown.

- Confirm that the derived values are shown.
- Scroll to the right as far as you can.

Attribute_7 (Remainder of record)
0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms

Notice the new attribute that holds the remainder of the record. You do not need this attribute. You might delete it now. Because you already defined all of your attributes before you came into the Parse Log utility, another solution is to click **Cancel** and exit without saving the new attribute.

- Click **Cancel** to close the **Parse Log** window and not save the new attribute.

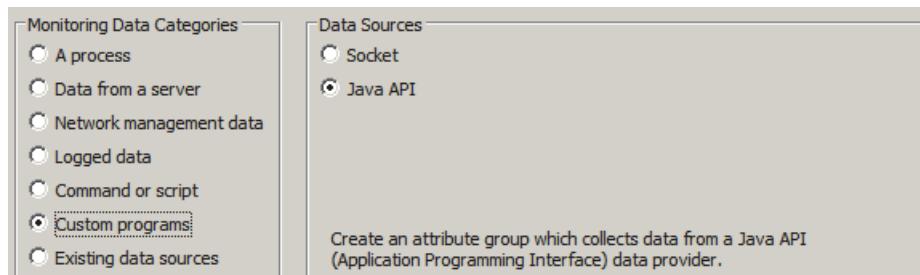
You created an agent that monitors a log file for specific data and derives new monitored data.

Add a Java API data source

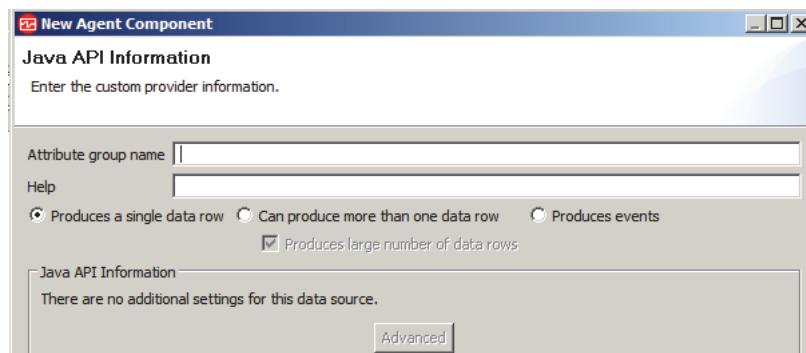
In this section, you add a Java API data source that gathers four data points from a custom Java application. The custom Java code that creates the monitored data is provided and you do not need to create it. An example of the data it generates is shown here.

Name	Integer Value	String Value	Float Value
One_SSN_arg1_One	14	One_SSN_arg1_One	11.12
One_SSN_arg2_One	15	One_SSN_arg2_One	12.12
One_SSN_arg3_One	16	One_SSN_arg3_One	13.12

36. Add a Java API data source that gathers one text string attribute called Name.
- In the **Data Sources** tab, right-click **AB1 (Agent)** and click **Add Data Source**.
The Data Source Location window opens.
 - Click **Custom Programs** under **Java API**.



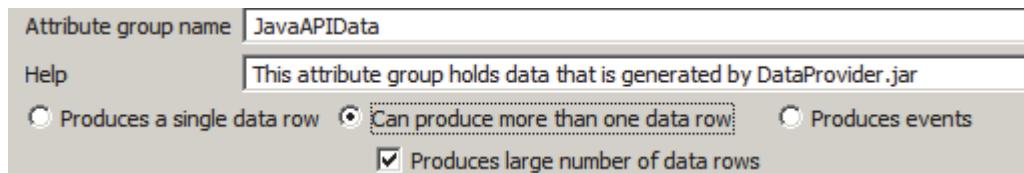
- Click **Next**.
The Java API Information window opens.



Here, you define an attribute group for data that is gathered by the custom Java application, **DataProvider.jar** in this instance.

- Enter the following values:
 - ◆ Attribute group name: **JavaAPIData**
 - ◆ Help: **This attribute group holds data that is generated by DataProvider.jar**

- e. Select **Can produce more than one data row**.



- f. Click **Next**.

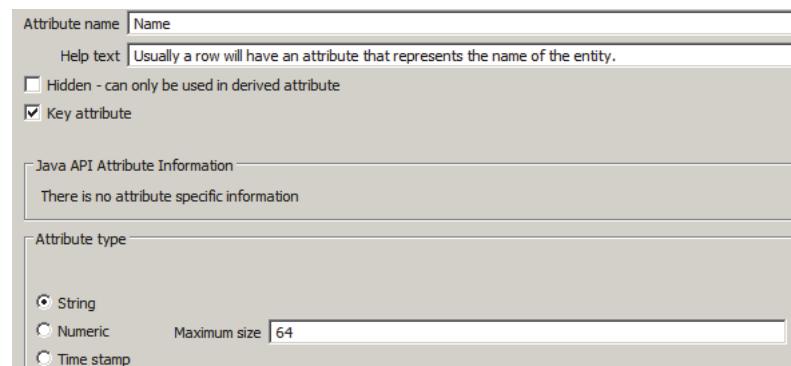
The Attribute Information window opens.



You must define one attribute while in the Agent wizard. Other attributes can be created when you leave the Agent wizard.

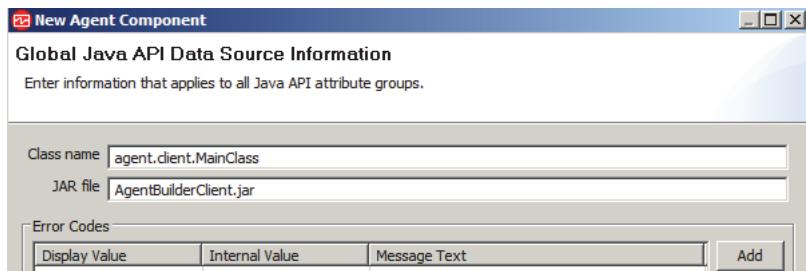
- g. Set the following values:

- ◆ Attribute name: **Name**
- ◆ Help text: **Usually a row will have an attribute that represents the name of the entity.**
- ◆ Key attribute: **Selected**
- ◆ Attribute type: **String, Maximum size: 64**



- h. Click **Next**.

The Global Java API Data Source Information window opens.



These files are provided with Agent Builder.

- i. Click **Finish**.

You are returned to the **Data Source Definition** tab.

37. Expand **JavaAPIData** and confirm the one attribute, **Name**, was created.



You add more attributes later in this exercise.

38. Save your agent project.

39. Test the JavaAPIData data source and confirm that it returns data the expected sample data.

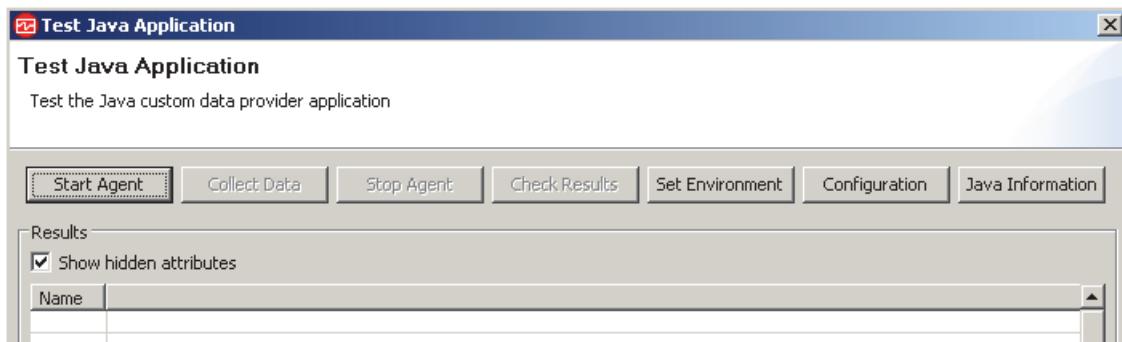
- a. Click **JavaAPIData (Java API)** in the **Data Source Definition** tab.
- b. Click **Test**.



Hint: If necessary, scroll down to see the **Test** button.

Attribute group name	JavaAPIData	
Help	Help: This attribute group holds data that is generated by DataProvider.jar	
<input type="radio"/> Produces a single data row	<input checked="" type="radio"/> Can produce more than one data row	<input type="radio"/> Produces events
<input checked="" type="checkbox"/> Produces large number of data rows		
<input checked="" type="checkbox"/> Add this attribute group to a reporting category		Performance
<input checked="" type="checkbox"/> Add this attribute group to the warehouse		
Refresh interval	minutes	
Java API Information		
There are no additional settings for this data source.		
<input type="button" value="Edit Java Source"/> <input type="button" value="Global Settings"/> <input type="button" value="Test"/> <input type="button" value="Advanced"/>		

The Test Java Application window opens.



The single attribute that you added to this data source, Name, is shown.

- c. Click **Start Agent** and **Collect Data**.

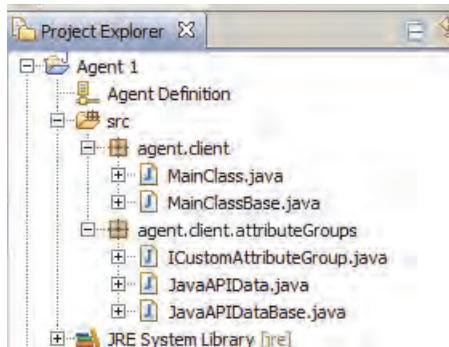
Results	
<input checked="" type="checkbox"/> Show hidden attributes	
Name	
arg1	
arg2	
arg3	

Confirm that three rows of data are returned. The JavaAPIData.java class that is created by Agent Builder generates this sample data. To enable this data source to gather real data, you must edit or replace this Java class.

- d. Click **OK** to close the Test Java Application window.

40. Identify the code that generated the sample data.

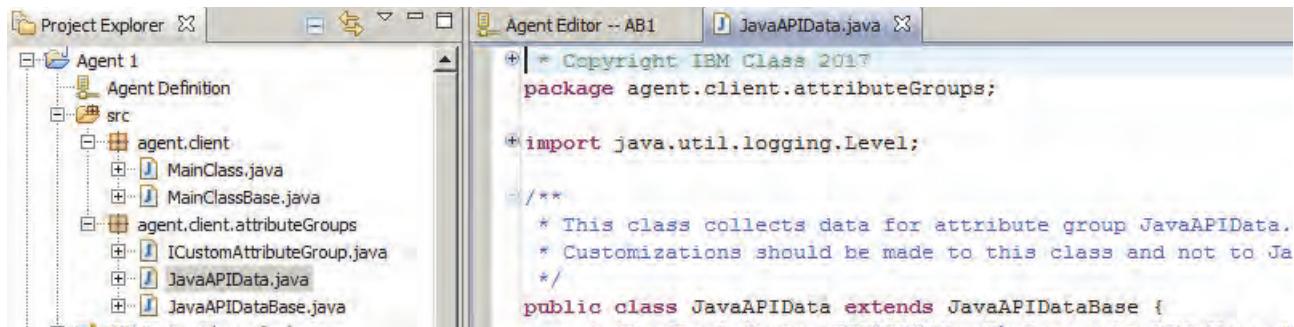
- In the Project Explorer, expand **Agent 1 > src**.
- Expand both **agent.client** and **agent.client.attributeGroups**.



The **agent.client** package contains the main code that interacts with the agent: a base class (**MainClassBase.java**) and a subclass that you can modify. The **agent.client.attributeGroups** package contains two classes for each attribute group. Again, the package contains a base class (<attr group name>**Base.java**) and a subclass that you can modify.

The Agent Builder always regenerates the base classes. The Agent Builder does not regenerate the subclasses unless they are not present.

- In the Project Explorer, double-click **JavaAPIData.java** to open the class in an editor.



- d. Locate the **collectData** section and the lines that generate the sample attributes for the Name attribute.

```

| public void collectData(Request request) throws Exception {
|     final String METHOD = "collectData";
|     logger.entering(CLASS, METHOD, request);
|     AgentConnection agentConnection = mainClient.getAgentConnection();
|     Attributes row;
|
|     // This example sends 3 data rows with arbitrary values.
|     // Replace the rows with data from the monitored resource.
|     for (int rowNum = 1; rowNum <= 3; ++rowNum) {
|         row = new Attributes(
|             "arg" + rowNum
|         );
|         logger.logp(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
|         row.setAttributeValues(request.getID());
|     }
}

```

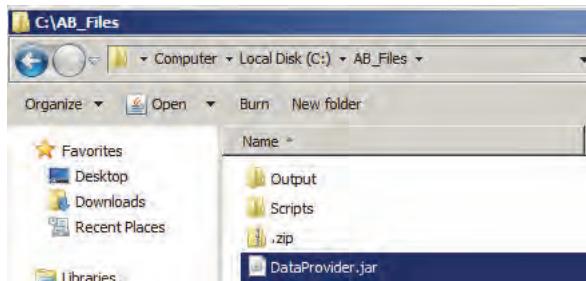
Agent Builder creates Java class similar to JavaAPIData for each Java API attribute group that you define and configures it to create sample data for each attribute that you initially add to the data source. You must modify this class to either generate the actual data you want collected and monitored or modify it to collect that data from custom Java code that generates the data.

You add a custom .jar file to this agent that generates the monitored data. You must modify the **JavaAPIData.java** class to gather the data from this custom .jar file.

- e. Close the JavaAPIData.java editor.

41. Locate the following file:

C:\AB_Files\DataProvider.jar



The DataProvider.jar file is a custom Java application that is created to generate custom data that this agent monitors. It represents custom code that you must create to generate the data that you want to gather and monitor. **DataProvider.jar** specifically generates four attributes that are called Name, Integer Value, String Value, and Float Value.

Name	Integer Value	String Value	Float Value
One_SSN_arg1_One	14	One_SSN_arg1_One	11.12
One_SSN_arg2_One	15	One_SSN_arg2_One	12.12
One_SSN_arg3_One	16	One_SSN_arg3_One	13.12

To enable your Agent Builder agent, you must do the following tasks:

- Add **DataProvider.jar** to your agent so that the agent can use it to generate the monitored data
- Add the three new attributes that are created by **DataProvider.jar**
- Re-create the **JavaAPIData.java** class to enable it to gather the new attributes
- Modify the **JavaAPIData.jar** class to gather the four attributes from **DataProvider.jar**

42. Add **DataProvider.jar to your agent.**

- a. In Agent Builder, go to the **Data Sources** tab of your agent.
- b. Click **JavaAPIData (Java API)**.
- c. Scroll down in the attribute group information pane and select **Global Settings**.



- d. Click **Add** to the right of **Supplemental Files**.
- e. Locate and open the **DataProvider.jar** file in the **C:\AB_Files** directory.



- f. Save your agent project.

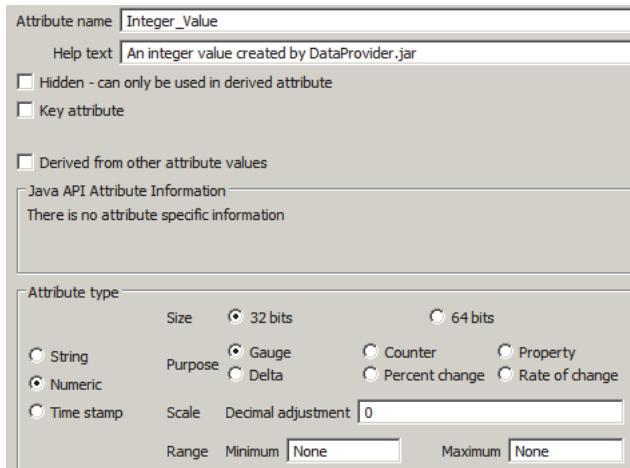
43. Add the three new attributes.

- a. In the **Data Sources** tab, ensure **JavaAPIData (Java API)** is selected and then click **Add to Selected** on the right.

The Attribute Information window opens.

- b. Set the following attribute properties:
 - ◆ Attribute name: **Integer_Value**
 - ◆ Help text: **An integer value created by DataProvider.jar**
 - ◆ Attribute type: **Numeric, 32-bit, Gauge**

The attribute information window looks like this screen capture.

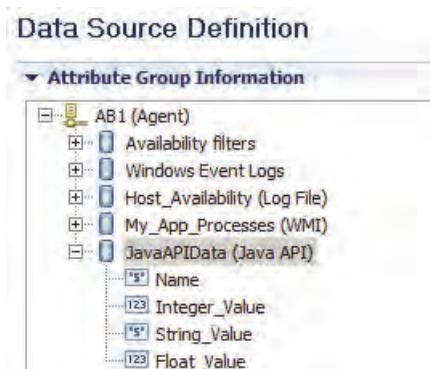


- c. Select **Add additional attributes** and click **Next**.
- d. Repeat this step for the remaining two new attributes according to this table.

Attribute name	Help text	Attribute type
String_Value	A string value that is created by DataProvider.jar	String with maximum size of 64
Float_Value	A floating-point value that is created by DataProvider.jar	Numeric, 64-bits, Counter, Decimal adjustment of 2

- e. Click **Finish** when you are done with the **Float_Value** attribute.

The **Data Source Definition** tab looks like this screen capture.



- f. Save your agent project.
44. Re-create the **JavaAPIData.java** class to enable it to gather the new attributes.
- a. If it is still open, close the JavaAPIData.java editor.
 - b. In the Project Explorer, double-click the **JavaAPIData.java** class to reopen it.

- c. Locate the **collectData** section as you did earlier in this lab.

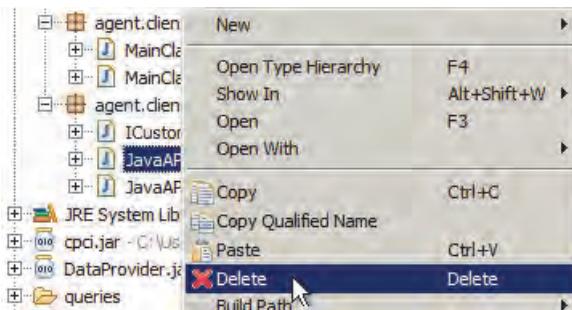
```
| public void collectData(Request request) throws Exception {
    final String METHOD = "collectData";
    logger.entering(CLASS, METHOD, request);
    AgentConnection agentConnection = mainClient.getAgentConnection();
    Attributes row;

    // This example sends 3 data rows with arbitrary values.
    // Replace the rows with data from the monitored resource.
    for (int rowNum = 1; rowNum <= 3; ++rowNum) {
        row = new Attributes(
            "arg" + rowNum
        );
        logger.logp(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
        row.setAttributeValues(request.getID());
    }
}
```

It still generates only data for one attribute. This behavior is expected. The Attribute Group class is created when the attribute group is initially defined and is not modified as you add more attributed. You must re-create this class.

To re-create this class with the new attributes delete the existing class, modify the agent in some manner and save those changes.

- d. Close the JavaAPIData.java editor.
e. Right-click the **JavaAPIData.java** class in the Project Explorer and click **Delete**.

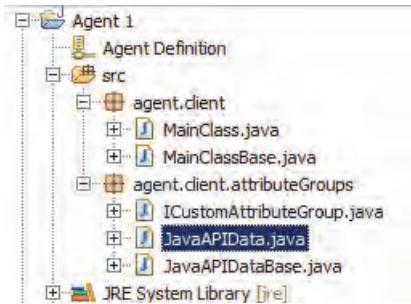


- f. Click **OK** to confirm the deletion.

Notice that you cannot save your agent project, which you need to do to re-create the JavaAPIData class. The next step changes the agent so that you can save your agent project. It then reverses the change because the change made was not needed.

- g. On the **Agent Information** tab, change the product code to **K02** and then change it back to **K00**.
You can now save your agent project.
h. Save your agent project.

The **JavaAPIData.java** class is re-created.



- i. Test the **JavaAPIData** data source and confirm that it returns values to all four attributes.

Results				
<input checked="" type="checkbox"/> Show hidden attributes				
Name	Integer_Value	String_Value	Float_Value	
arg1	11	arg1	11.12	
arg2	12	arg2	12.12	
arg3	13	arg3	13.12	

Sample values are again created by the **JavaAPIData.java** class. Next you modify that class to gather data from your custom Java application, **DataProvider.jar**.

45. Modify the **JavaAPIData.java** class to gather the four attributes from **DataProvider.jar**.

- a. Double-click the **JavaAPIData.java** class in the Project Explorer to open it in an editor.
- b. Locate the **collectData** section.

```
public void collectData(Request request) throws Exception {
    final String METHOD = "collectData";
    logger.entering(CLASS, METHOD, request);
    AgentConnection agentConnection = mainClient.getAgentConnection();
    Attributes row;

    // This example sends 3 data rows with arbitrary values.
    // Replace the rows with data from the monitored resource.
    for (int rowNum = 1; rowNum <= 3; ++rowNum) {
        row = new Attributes(
            "arg" + rowNum,
            10 + rowNum,
            "arg" + rowNum,
            10.12 + rowNum
        );
        logger.logp(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
        row.setAttributeValues(request.getID());
    }
}
```

- c. Replace the row arguments ("arg" through 10.12) with the following text:

```
DataProvider.getStringValue("Name"),
DataProvider.getIntValue("Integer_Value"),
DataProvider.getStringValue("String_Value"),
DataProvider.getFloatValue("Float_Value")
```



Hint: Type "DataProvider." and a menu appears. Double-click the correct **get** option from the menu. Type the value name with quotations.

With this code, the JavaAPIData.java class can gather the four attributes from **DataProvider.jar**.

The row section looks like this screen capture:

```
public void collectData(Request request) throws Exception {
    final String METHOD = "collectData";
    logger.entering(CLASS, METHOD, request);
    AgentConnection agentConnection = mainClient.getAgentConnection();
    Attributes row;

    // This example sends 3 data rows with arbitrary values.
    // Replace the rows with data from the monitored resource.
    for (int rowNum = 1; rowNum <= 3; ++rowNum) {
        row = new Attributes(
            DataProvider.getStringValue("Name"),
            DataProvider.getIntValue("Integer_Value"),
            DataProvider.getStringValue("String_Value"),
            DataProvider.getFloatValue("Float_Value")
        );
        logger.log(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
        row.setAttributeValues(request.getID());
    }
}
```

- d. Save **JavaAPIData.java**.

- e. Return to the AB1 agent editor and test the JavaAPIData data source. Confirm it returns values to all four attributes.

Results				
<input checked="" type="checkbox"/> Show hidden attributes				
Name	Integer_Value	String_Value	Float_Value	
value of Name0	0	value of String_Value1	1.12	
value of Name2	2	value of String_Value3	3.12	
value of Name4	4	value of String_Value5	5.12	

Notice that values returned by the **DataProvider.jar** are different from those values that are originally created by the **JavaAPIData.java** class.

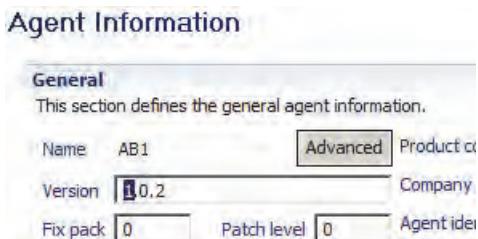
You successfully added a Java API data source and enabled it to gather and monitor data from a custom Java application.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

46. Select to the **Agent Information** tab.

47. Change the version for 1.0.1 to **1.0.2**.



Important: If your initial version number is not 1.0.1, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

48. Save your agent project.

Exercise 4 Install and confirm the updated AB1 agent in an IBM Tivoli Monitoring environment

In this exercise, you test your updated AB1 agent and confirm that script and log file data is being gathered in an IBM Tivoli Monitoring environment.

Create the updated AB1 agent installation scripts

1. In Agent Builder on WIN1, generate the AB1 agent installation scripts.
2. Delete the contents of the **C:\share\K00** directory.
3. Copy the new **C:\Users\Administrator\smai-ab1-01.02.00.00.zip** file into **C:\share\K00** directory.
4. Extract the files from **smai-ab1-01.02.00.00.zip** into **C:\share\K00** with **7-Zip**.

Install the updated AB1 application support on ITM

5. Install the IBM Tivoli Monitoring application support on ITM.
 - Use the drive map to **\WIN1\share**.
 - Run the following commands:
`installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00
installIraAgentTEPS.bat C:\IBM\ITM`

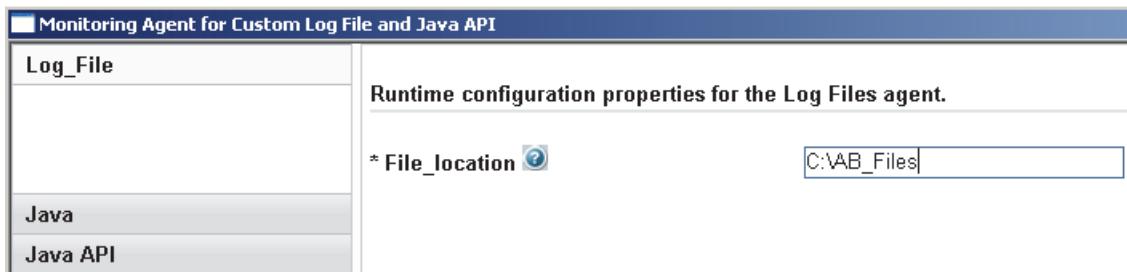


Hint: Before installing the Tivoli Enterprise Portal Server support, do not forget to close the TEP client.

Reinstall and configure the updated AB1 agent on WIN1

6. Install the agent on WIN1 in any way you choose.
7. In MTMS, reconfigure your **Monitoring Agent for AB1**.
 - a. Start the agent configuration.
You are prompted to configure the TEMS connection.
 - b. Ensure that **ITM** is the target TEMS and click **OK**.

The Agent Configuration window opens.



Notice that the file location is the incorrect directory for this host.

- c. Replace **C:\AB_Files** with **C:\AB_Files\Output** in the **File_location** field.



- d. Click **Next**.

You are prompted for Java configuration parameters. The default location of Java (Java 50) is not valid on this host.

- e. Browse to and change the Java home path to the following directory:

C:\Program Files (x86)\ibm\Java70



- f. Click **Next**.



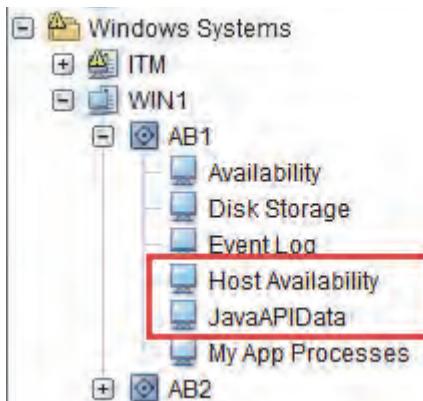
Do not set the Java API class path to external .jar files.

- g. Click **OK** to close the configuration editor and save your changes.

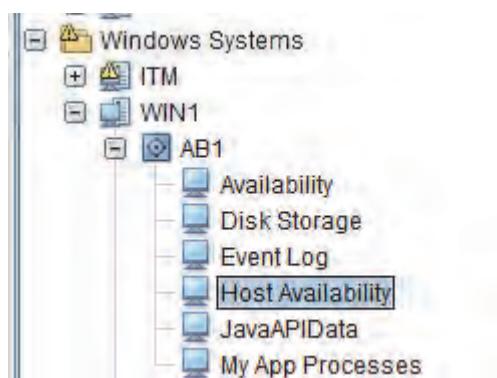
- h. Click **Yes** to restart your **Monitoring Agent for AB1**.

Confirm the log file data source

8. On ITM, start the Tivoli Enterprise Portal Client and confirm that your **AB1 Host Availability** and **JavaAPIData** attribute groups are displayed below WIN1.



9. After you confirm that the attribute groups are visible, generate the log file data on WIN1.
 - a. On WIN1, go to the following directory and delete the current **ping.txt** file.
C:\AB_Files\Output
 - b. Open a command prompt and change to the following directory:
C:\AB_Files\Scripts
 - c. Run the following command:
pinginterval.bat
10. Return to the Tivoli Enterprise Portal and click **Host_Availability** under **AB1**.



11. Periodically, press **F5** to refresh the workspace until you have three or more rows of data.

Node	Timestamp	IP Address	Sent	Received	Lost	Status	Avg Lost
WIN1:00	03/08/17 21:49:22	192.168.1.107	0	0	0	Failure	0.00
WIN1:00	03/08/17 21:49:14	192.168.1.103	4	4	0	Good	0.00
WIN1:00	03/08/17 21:49:06	192.168.1.107	4	4	0	Good	0.00
WIN1:00	03/08/17 21:48:58	192.168.1.103	4	4	0	Good	0.00

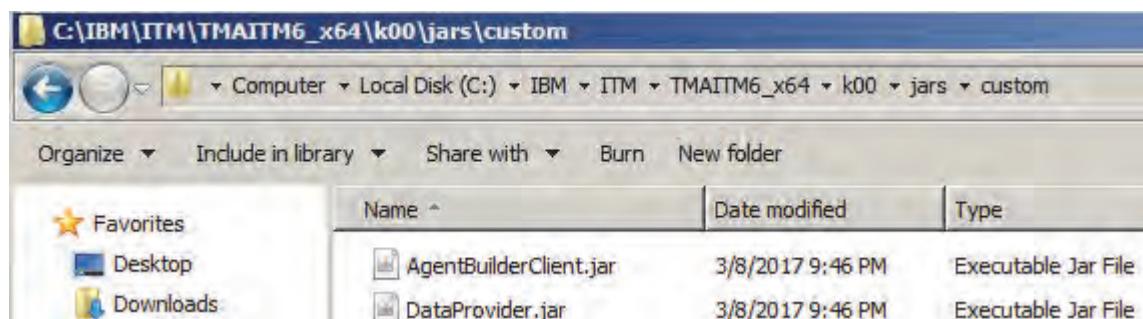
12. Confirm that the Report table view displays only the data that you wanted from the **ping.txt** file.

13. Stop the **pinginternal.bat** batch file.

Confirm the Java API data source

14. On the WIN1 server, open File Explorer and go to this directory:

C:\IBM\ITM\TMAITM6_x64\k00\jars\custom



The **AgentBuilderClient.jar** file is the Java API client. It includes the Java client class files that were under **src** in Agent Builder. The **DataProvider.jar** file is the custom .jar file that you added to the agent that generates the data that is gathered by the client.

15. In the Tivoli Enterprise Portal on ITM, click the **JavaAPIData** node under AB1 and confirm that valid data is returned.

The screenshot shows the IBM Tivoli Enterprise Portal interface. The Navigator pane on the left displays a tree view of system nodes:

- Enterprise
- Linux Systems
- Windows Systems
 - ITM
 - WIN1
 - AB1
 - Availability
 - Disk Storage
 - Event Log
 - Host Availability
 - JavaAPIData** (highlighted)
 - My App Processes

The Report pane on the right displays a table of data for the JavaAPIData node:

Node	Timestamp	Name	Integer Value	String Value	Float Value
WIN1:00	03/08/17 21:53:15	value of Name0	0	value of String_Value1	1.12
WIN1:00	03/08/17 21:53:15	value of Name2	2	value of String_Value3	3.12
WIN1:00	03/08/17 21:53:15	value of Name4	4	value of String_Value5	5.12

Unit 7 Monitoring remote and optional resources exercises

In this unit, you modify your agents to monitor remote resources and contain optional data sources.

In the first exercise, you modify the AB1 agent to monitor multiple hosts at the same time. It pings a list of hosts to confirm their availability. It tests multiple URLs from multiple HTTP servers to confirm their availability. It uses subnodes to gather the same network data from multiple servers.

In the second exercise, you modify the AB2 agent by placing optional sets of monitored data sources into their own subnode. With this setup, you choose which subnodes to configure when you configure the agent.

Exercise 1 Remotely monitor many resources

Your company has several resources that you must monitor where you are not allowed to or cannot install an agent.

In this exercise, you modify the AB1 agent to monitor the following resources:

- SNMP data with several SNMP data sources within a subnode. On the subnode, you can configure a single agent to monitor multiple hosts.
- HTTP server URLs and their embedded objects
- The network availability of the application servers

Create an agent to monitor multiple hosts from one agent

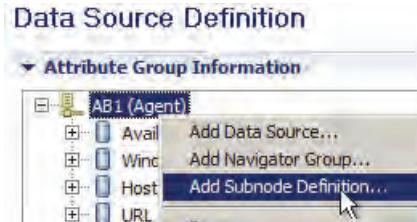
In this section, you modify the AB1 agent to gather process performance information from Simple Network Management Protocol (SNMP) from all three systems in class.

- System
- ifTable
- ipAddrTable

You must complete this exercise from the WIN1 server.

Complete the following steps:

1. If it is not running, start Agent Builder.
2. If the **AB1** agent is not already open, expand **Agent 1** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
3. Click the **Data Sources** tab.
4. Right-click **AB1 (Agent)** under **Attribute Group Information** and click **Add Subnode Definition**.



The Subnode Information window opens.

Subnode Information	
Name	<input type="text"/>
Type	<input type="text"/>
Description	<input type="text"/>
Company identifier	<input type="text"/> IBM_E
Subnode identifier	<input type="text"/> K00_
<input checked="" type="checkbox"/> Show nodes attribute group for this type of subnode	

5. Enter the following information:
 - Name: **SNMP**
 - Type: **SNP**
 - Description: **Subnode to gather SNMP information from multiple hosts**

Name	<input type="text"/> SNMP
Type	<input type="text"/> SNP
Description	<input type="text"/> Subnode to gather SNMP information from multiple hosts
Company identifier	<input type="text"/> IBM_E
Subnode identifier	<input type="text"/> K04SNP
<input checked="" type="checkbox"/> Show nodes attribute group for this type of subnode	

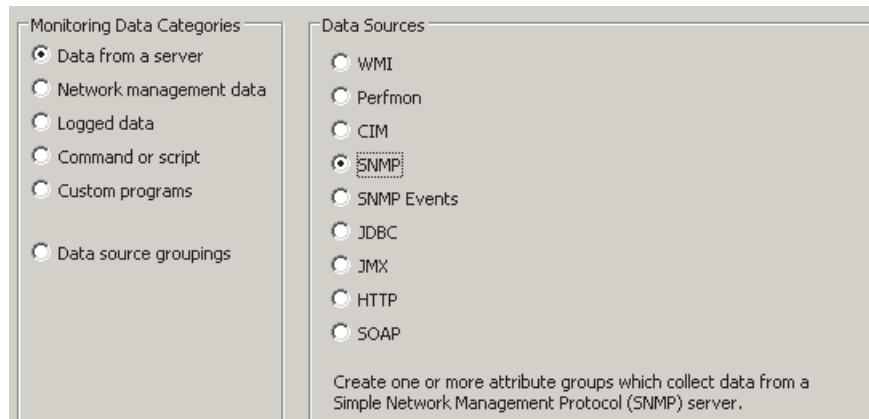


Note: Type is a unique three-character code that identifies this subnode.

6. Click Next.

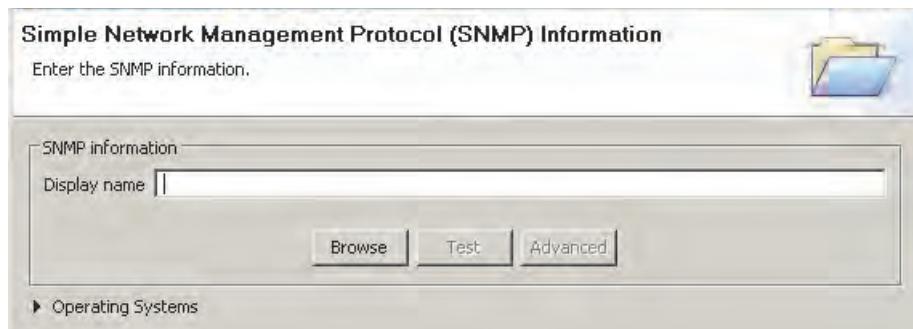
The Agent Initial Data Source window opens.

7. Define the data sources for this subnode. Click **Data from a server under **Monitoring Data Categories** and **SNMP** under **Data Sources**.**



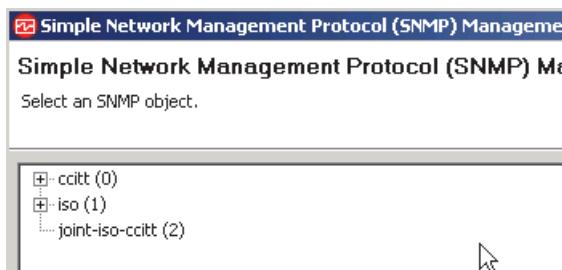
8. Click Next.

The Simple Network Management Protocol (SNMP) Information window opens.



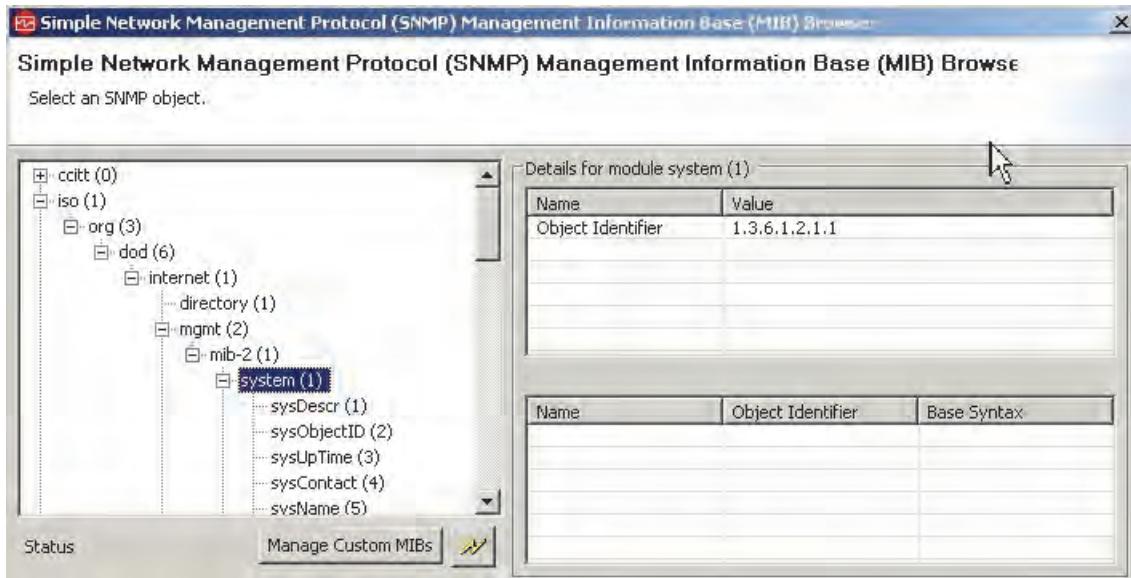
9. Click **Browse in the **SNMP information** pane.**

The SNMP MIB Browser opens.



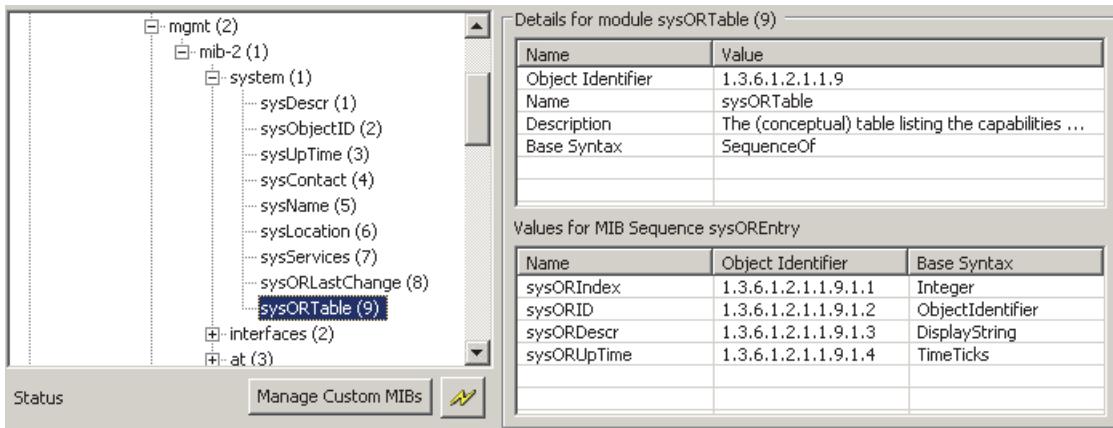
10. Locate and expand the following SNMP object:

iso/org/dod/internet/mgmt/mib-2/system



Notice the properties that are listed below **system**. This data can be gathered from this SNMP object. Agent Builder creates an attribute group for system and an attribute for each of these properties.

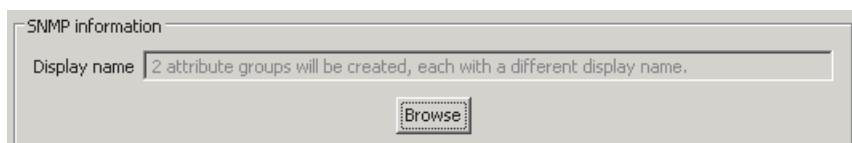
11. Scroll down the SNMP object lists and click **sysORTable**.



Notice the values for MIB Sequence sysOREEntry pane. The sysORTable object contains subattributes, and it can be selected as its own data source. By selecting **system** in your agent, the child object sysORTable generates its own attribute group with its own set of attributes.

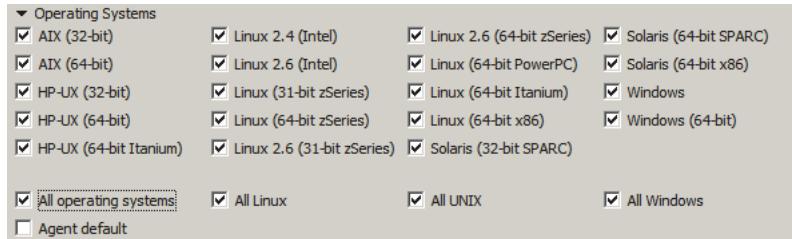
12. Reselect **system** and click **OK**.

The SNMP information window is displayed.



System is not listed in the **Display name** field. The sysORTable object causes your agent to generate two attribute groups, one for system and another for sysORTable. In subsequent steps, as you select other SNMP objects, you see the object name that is listed here.

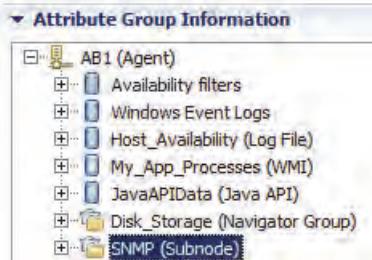
13. Expand **Operating Systems** and select **All operating systems**.



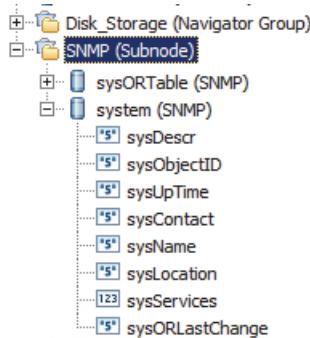
14. Click **Finish**.

You are returned to the **Data Source Definition** tab.

Data Source Definition



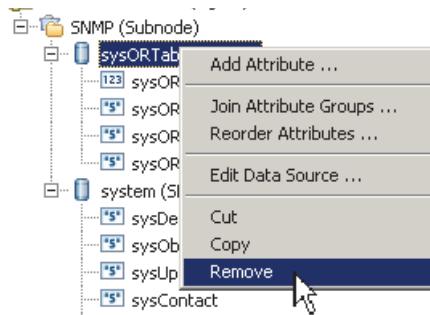
15. Expand **SNMP (Subnode)** and **system**.



The attribute group and attribute names are based on the SNMP objects you saw in the SNMP MIB browser.

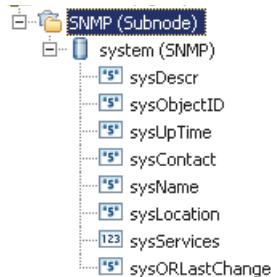
16. Remove the **sysORTable** attribute group.

- Right-click **sysORTable (SNMP)** and click **Remove**.



- Click **Yes** to the prompt to remove one attribute group and four attributes.

Your SNMP subnode looks like the following image.

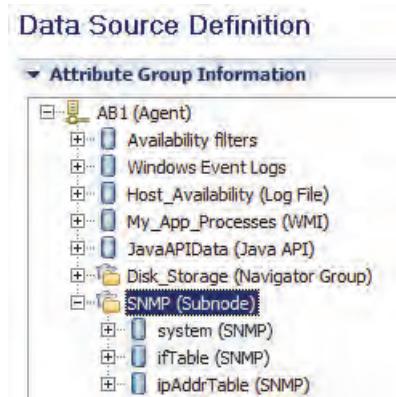


17. Add the following extra SNMP objects by selecting **SNMP (Subnode)**, clicking **Add to Selected** and repeating [Step 7](#) on page 7-3 through [Step 14](#) on page 7-5.

iso/org/dod/internet/mgmt/mib-2/interfaces/ifTable

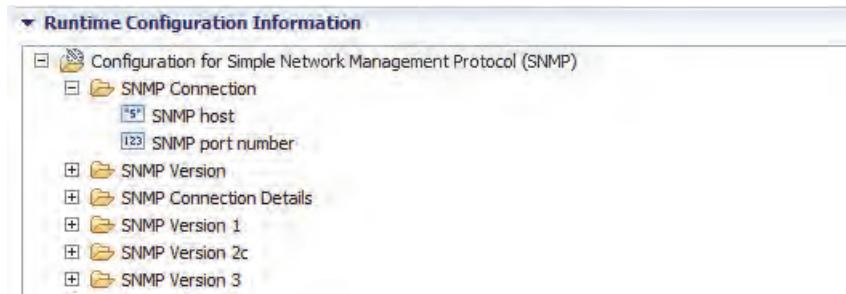
iso/org/dod/internet/mgmt/mib-2/ip/ipAddrTable

Your final Data Source Definition window looks like the following example.



18. Click the **Runtime Configuration Information** tab.

19. Expand Configuration for Simple Network Management Protocol (SNMP).

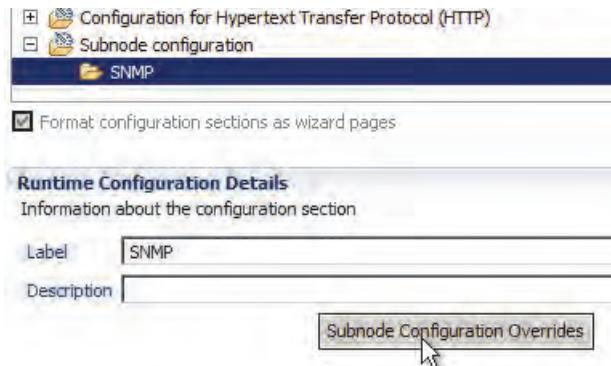


The Configuration for Simple Network Management Protocol object contains global runtime configuration parameters. The Subnode configuration object contains those parameters that must be set for each subnode.

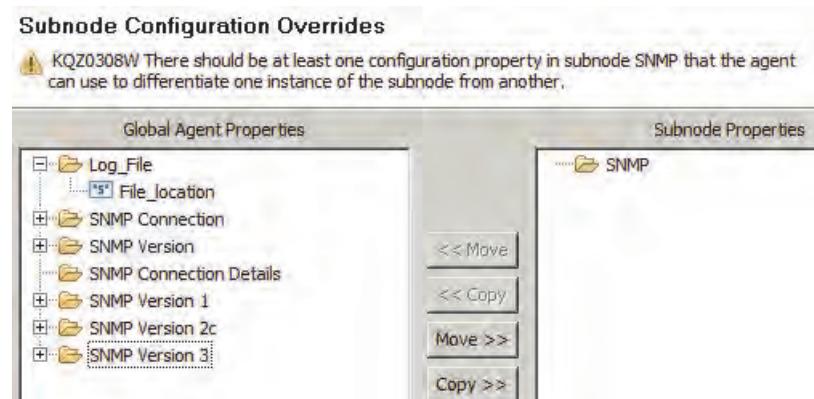
These SNMP properties are set when the agent is configured and establish the connection into the database server, whether it is local or remote. By moving these configuration parameters into the subnode, they can be set differently for each subnode.

20. Set the SNMP Runtime Configuration properties to be configured per subnode.

- Close Configuration for Simple Network Management Protocol (SNMP).
- Expand Subnode Configuration and click SNMP.
- Click Subnode Configuration Overrides.



The Subnode Configuration Overrides window opens.



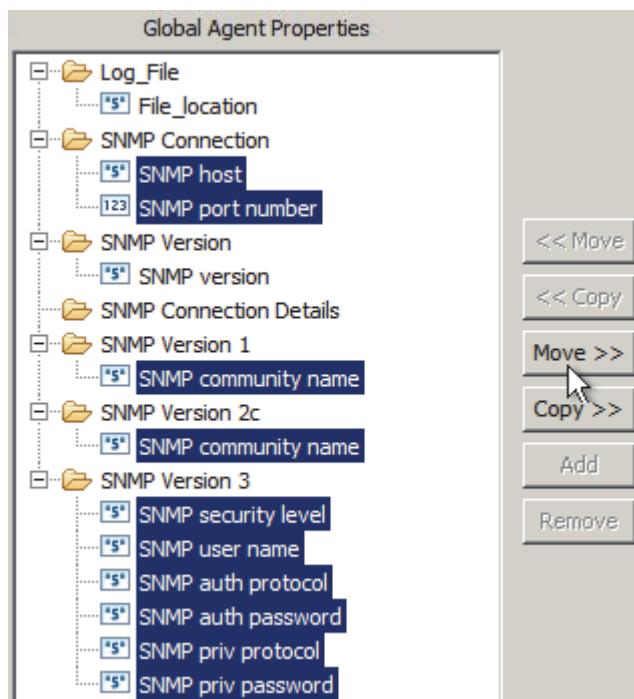
In this window, you can determine which configuration data, which you set during installation of the agent, is configurable for each subnode. Currently, all properties are global. At the time of the installation, you enter them once and they apply to all subnodes. Because you are monitoring multiple remote SNMP servers, this result is not acceptable.

At a minimum, you set the SNMP host name as a subnode property so that you can identify the three host systems. Other information that you might set per subnode include the port that is used to connect to SNMP, the SNMP version, the SNMP community name, and the various SNMP version 3 authentication properties. In this exercise, set all SNMP possible properties configurable for each subnode.

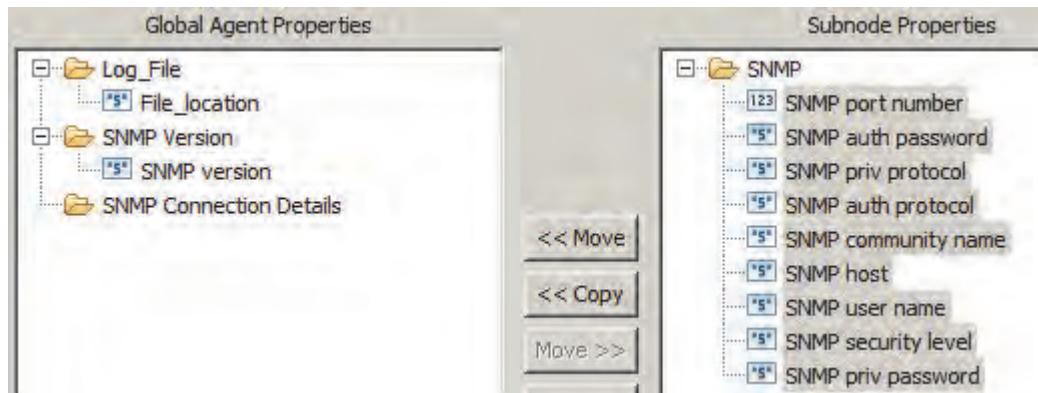
d. Select the following properties:

- ◆ SNMP host
- ◆ SNMP port number
- ◆ SNMP Version 1> SNMP community name
- ◆ SNMP Version 2c > SNMP community name
- ◆ SNMP security level
- ◆ SNMP user name
- ◆ SNMP auth protocol
- ◆ SNMP auth password
- ◆ SNMP priv protocol
- ◆ SNMP priv password

e. Click **Move>>**.



The subnode configuration overrides looks like the following screen capture when you are done.



Now the configuration properties are set when defining subnodes during agent configuration.

21. Click **OK** to save your changes and exit the Subnode Configuration Overrides window.
 You are returned to the **Data Source Definition** tab.
22. Set the default SNMP port to 162.
 - a. Under **Subnode configuration > SNMP**, select **SNMP port number**.
 SNMP port number does not have a default value.
 - b. Enter **161** in **Default Value**.

The screenshot shows the 'Subnode configuration' section with 'SNMP' selected. Under 'SNMP', 'SNMP port number' is highlighted with a red arrow. Below this, there is a checkbox for 'Format configuration sections as wizard pages' which is checked.

Runtime Configuration Details	
Information about the configuration property	
Label	SNMP port number
Environment variable	SNMP_PORT
Description	The port number of the SNMP server.
Type	Numeric
Default value	161

23. Save your agent project.

Add a data source to monitor HTTP pages

In this section, you add an HTTP data source to monitor select pages from the lab web servers.

24. On WIN1, locate and open the following text file:

C:\AB_Files\urls.txt

```
urls.txt - Notepad
File Edit Format View Help
http://win1/
http://win1/AnyBank.htm
http://win1/bad.html
http://win1/AnySiteMap.html
http://lin4
http://lin4/AnyBank.htm
http://lin4/AnyInsurance.htm
http://lin4/AnyInvestment.htm
http://lin4/AnySiteMap.html
http://itm/bad.html
```

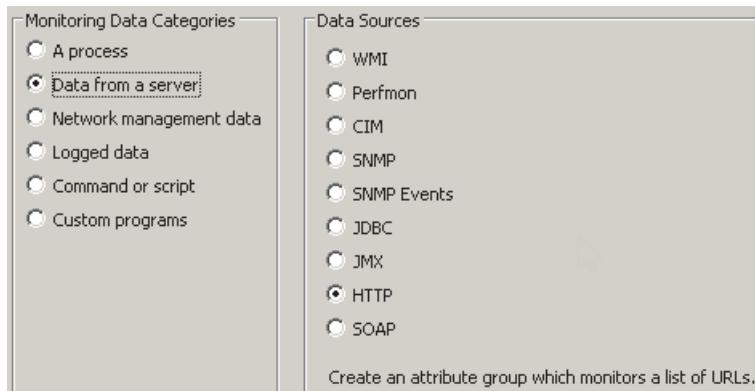
The HTTP data source uses a local text file that lists the URLs to be monitored. For this lab, you use the urls.txt file.

25. In Agent Builder, click the **Data Sources** tab in the AB1 agent editor.

26. Right-click **AB1 (Agent)** and click **Add Data Source**.

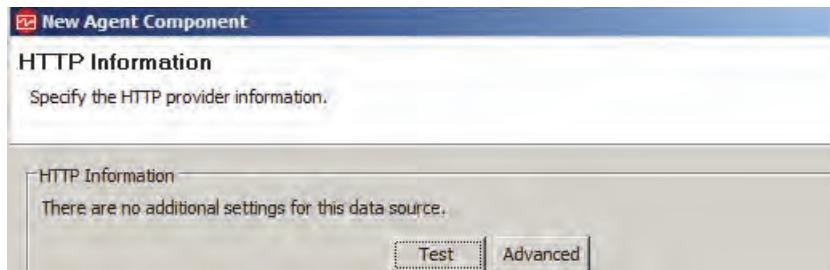
The Data Source Location window opens.

27. Click **HTTP** under **Data from a server**.



28. Click **Next**.

The HTTP Information window opens.

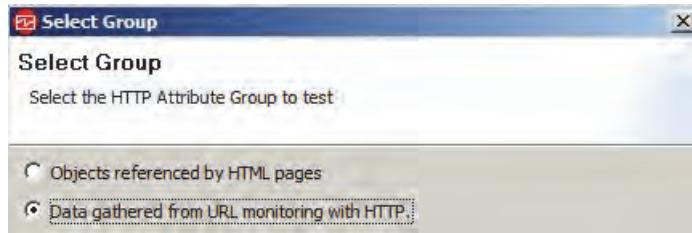


An HTTP data source requires no extra configuration. The files that contain the URLs to monitor are identified when the agent is installed and configured.

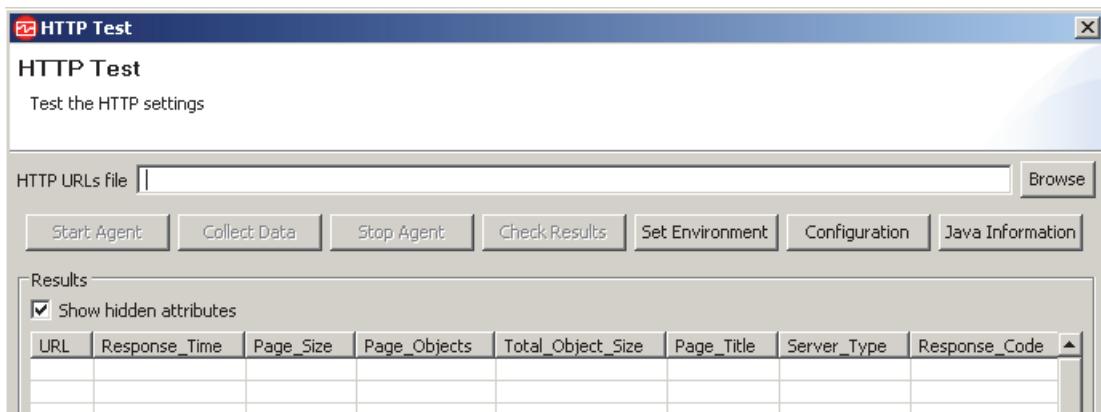
29. Click **Test**.

The Select Group window opens.

30. Click **Data gathered from URL monitoring with HTTP** and click **OK**.

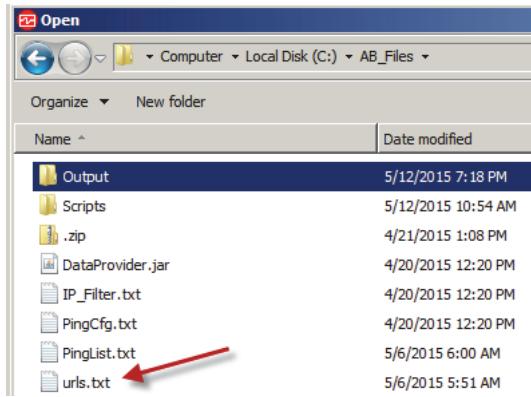


The HTTP Test window opens.

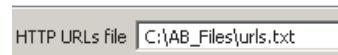


31. Click **Browse**.

32. Locate and open the **C:\AB_Files\urls.txt** file.



You are returned to the HTTP Test window with the **urls.txt** file in the **HTTP URLs file** field.



33. Click **Start Agent** and **Collect Data**.

Results						
URL	Response_Time	Page_Size	Page_Objects	Total_Object_Size	Page_Title	
http://win1/	31	1501	0	NOT_COLLECTED	Any Bank Home page	
http://win1/AnyBank.htm	0	861	0	NOT_COLLECTED	Any Bank Banking page	
http://win1/bad.html	0	206	0	NOT_COLLECTED	404 Not Found	
http://win1/AnySiteMap.html	31	848	0	NOT_COLLECTED	Any Bank Site Map	
http://lin4/	16	1501	0	NOT_COLLECTED	Any Bank Home page	
http://lin4/AnyBank.htm	16	861	0	NOT_COLLECTED	Any Bank Banking page	
http://lin4/AnyInsurance.htm	16	868	0	NOT_COLLECTED	Any Bank Insurance page	

34. Confirm that agent successfully gathered data on the monitored URLs.

35. Click **OK** to close the HTTP Test window.

36. Click **Finish** to close the HTTP Information window.

You are returned to the **Data Source Definition** tab with two new data sources: URL_Objects and Managed_URLs.



37. View the configuration properties that you would manage for this data source.

a. Click the **Runtime Configuration** tab.

b. Expand the **Configuration for Hypertext Transfer Proxy > Proxy Server Configuration** nodes.

Runtime Configuration Details

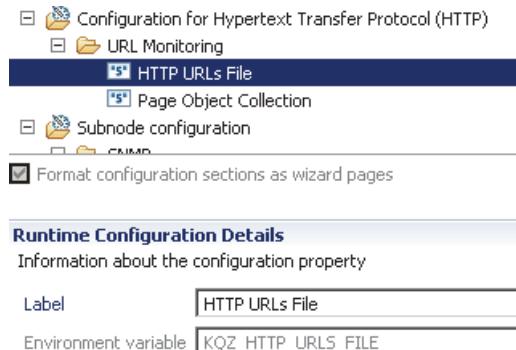
Information about the configuration property

Label: HTTP Data Provider Proxy Information

Environment variable: KQZ_HTTP_PROXY_TITLE

You must configure these properties if the URLs that the agent monitors go through a proxy server.

- c. Click each property and review the values to confirm that they are configured the way that you want. For example, do you want to require the property, or do you want to set a default value?
- d. Expand the Configuration for Hypertext Transfer Protocol (HTTP) > URL Monitoring nodes.



The **HTTP URLs file** property prompts the user for the name and location of the HTTP URLs file that tells the agent which URL to monitor. The **Page Object Collection** property allows the user to configure the agent to monitor or not monitor embedded objects.

- e. Click each property and review the values to confirm that they are configured the way that you want.

38. Save your agent project.

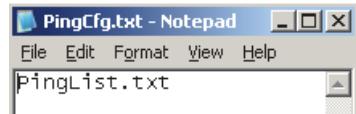
You successfully added an HTTP data source to this agent.

Add a data source to monitor network availability

In this section, you add a Ping (ICMP) data source to the agent that enables it to monitor the network availability of the lab servers.

39. Locate and open the following text file:

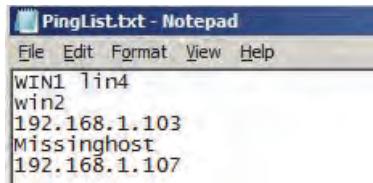
C:\AB_Files\PingCfg.txt



The Ping data source uses several local text files to determine which systems to monitor. The ping configuration file is a text file that must be specified as a runtime configuration parameter to the agent. The contents of each line in this file must be the location of a device list file. Each entry in the ping configuration file causes the agent to start a separate thread for monitoring the devices in the provided device list file.

40. Locate and open the following text file:

C:\AB_Files\PingList.txt



PingList.txt - Notepad
File Edit Format View Help
WIN1 Tint4
win2
192.168.1.103
Missinghost
192.168.1.107

A device list file is a text file that contains a list of device and host names. The list of device and host names can be separated with spaces and line breaks.

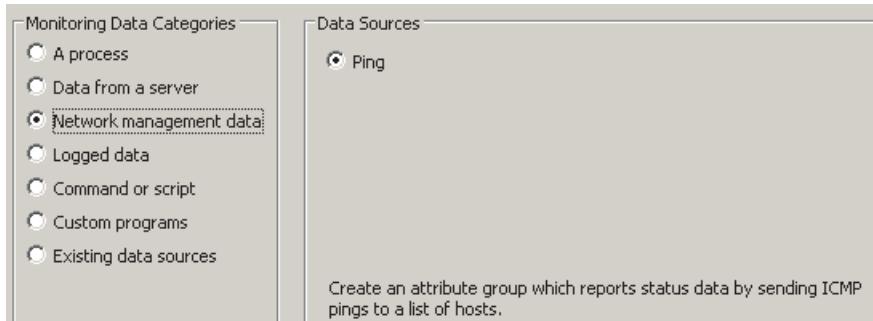
You use these files to identify which hosts to monitor.

41. In Agent Builder, click the **Data Sources** tab in the AB1 agent editor.

42. Right-click **AB1 (Agent)** and click **Add Data Source**.

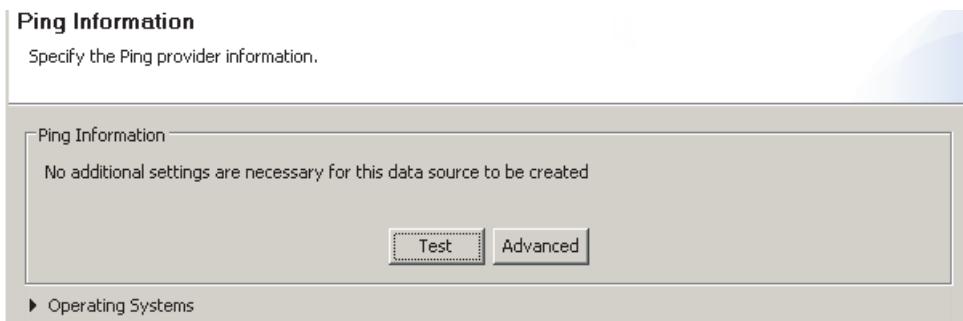
The Data Source Location window opens.

43. Click **Ping** under **Network management data**.



44. Click **Next**.

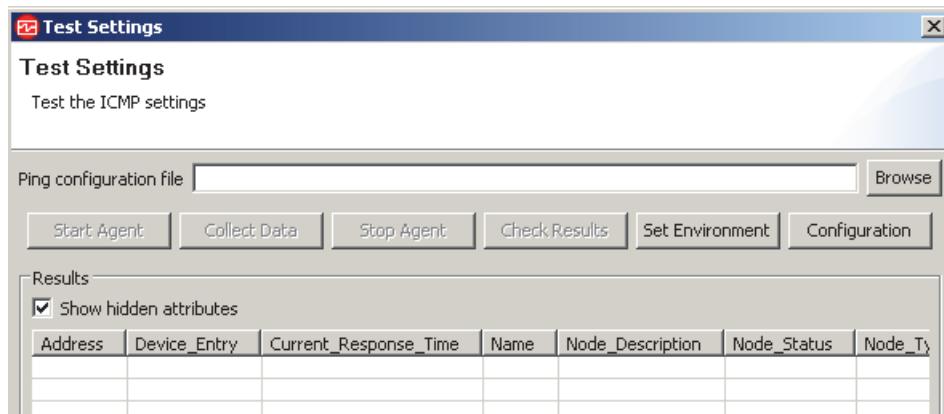
The Ping Information window opens.



A Ping data source requires no extra configuration. The file that contains the host names or IP addresses to monitor is identified when the agent is installed and configured.

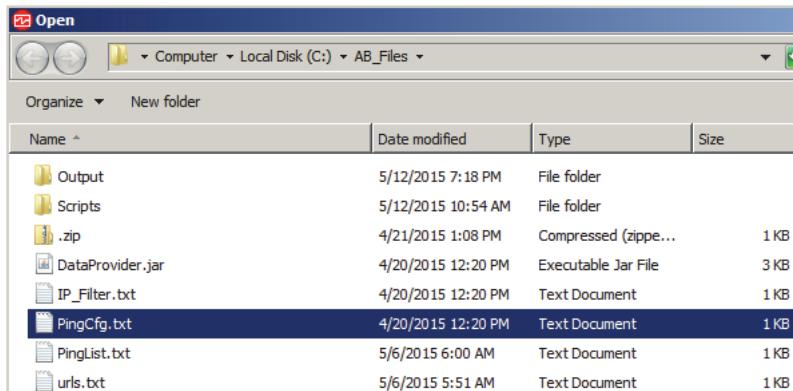
45. Click **Test**.

The Test Settings window opens.



46. Click **Browse**.

47. Locate and open the **C:\AB_Files\PingCfg.txt** file.



You are returned to the Test Settings window with the **PingCFG.txt** file in the **Ping configuration file** field.

Ping configuration file C:\AB_Files\PingCfg.txt

48. Click **Start Agent** and **Collect Data**.

Results							
<input checked="" type="checkbox"/> Show hidden attributes							
Address	Device_Entry	Current_Response_Time	Name	Node_Description	Node_Status	Node_Type	Status_Timestamp
192.168.1.103	192.168.1.103	0	win1.ibm.edu	PingList	Unknown	IP Node	
192.168.1.107	192.168.1.107	0	lin4.ibm.edu	PingList	Unknown	IP Node	
UNKNOWN_ADDRESS	Missinghost	0	UNKNOWN_HOSTNAME	PingList	Unknown	IP Node	
192.168.1.103	win1	0	win1.ibm.edu	PingList	Unknown	IP Node	
192.168.1.109	win2	0	win2.ibm.edu	PingList	Unknown	IP Node	
192.168.1.107	lin4	0	lin4.ibm.edu	PingList	Unknown	IP Node	

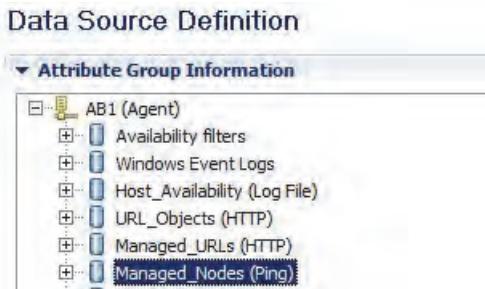
The device **Missinghost** is a name that cannot be resolved and therefore is not pinged.

49. Confirm that agent successfully gathered data on the monitored systems.

50. Click **OK** to close the Test Settings window.

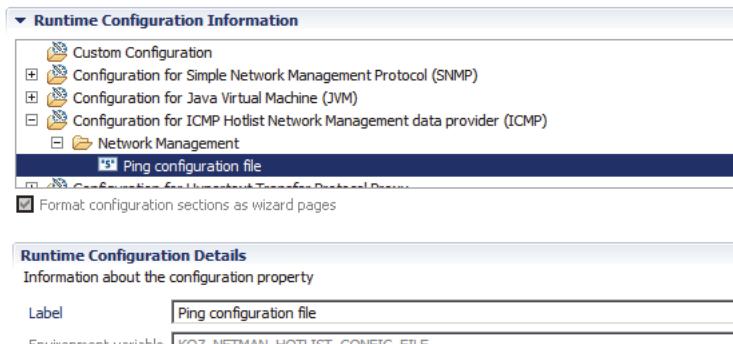
51. Click **Finish** to close the Ping Information window.

You are returned to the **Data Source Definition** tab with two new data sources: URL_Objects and Managed_URLs.



52. View the configuration properties that you would manage for this data source.

- Click the **Runtime Configuration** tab.
- Expand the **Configuration for ICMP Hotlist Network Management data provider (ICMP) > Network Management** nodes.



The only runtime configuration property for this data source is the ping configuration file, which leads the agent to the list or lists of devices it monitors.

53. Save your agent project.

You successfully added a Ping (ICMP) data source to this agent.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

54. Select to the **Agent Information** tab.

55. Change the version for 1.0.2 to **1.0.3**.

Agent Information

General
This section defines the general agent information.

Name	AB1	Advanced	Product o
Version	1.0.3	Company	
Fix pack	0	Patch level	0
Agent ide			



Important: If your initial version number is not 1.0.2, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

56. Save your agent project.

Exercise 2 Install and confirm the updated AB1 agent in an IBM Tivoli Monitoring environment

In this exercise, you test your updated AB1 agent and confirm that HTTP URL, ping, and SNMP data is being gathered in an IBM Tivoli Monitoring environment.

Create the updated AB1 agent installation scripts

1. In Agent Builder on WIN1, generate the AB1 agent installation scripts.
2. Delete the contents of the **C:\share\K00** directory.
3. Copy the new **smai-ab1-01.03.00.00.zip** file into **C:\share\K00** directory.
4. Extract the files from **smai-ab1-01.03.00.00.zip** into **C:\share\K00** with **7-Zip**.



Important: Your agent version numbers are different if you updated your agent more times than the labs.

Install the updated AB1 application support on ITM



Hint: Before installing the Tivoli Enterprise Portal Server support, do not forget to close the TEP client.

5. Install the IBM Tivoli Monitoring application support on ITM.

- Use the drive map to **\WIN1\share**.
- Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00  
installIraAgentTEPS.bat C:\IBM\ITM
```

Install, configure, and start the updated AB1 agent

6. Reinstall the AB1 agent on WIN1 in any manner you would like.

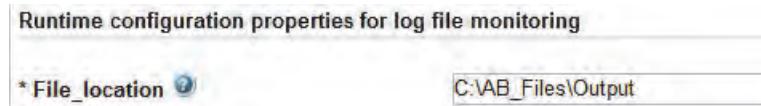
7. Configure your **Monitoring Agent for AB1**.

- a. In the MTMS utility, right-click the agent and select **Reconfigure** to start the configuration.

You are prompted to configure the TEMS connection.

- b. Ensure that **ITM** is the target TEMS and click **OK**.

The Agent Configuration window opens.



You are prompted for the log file location.

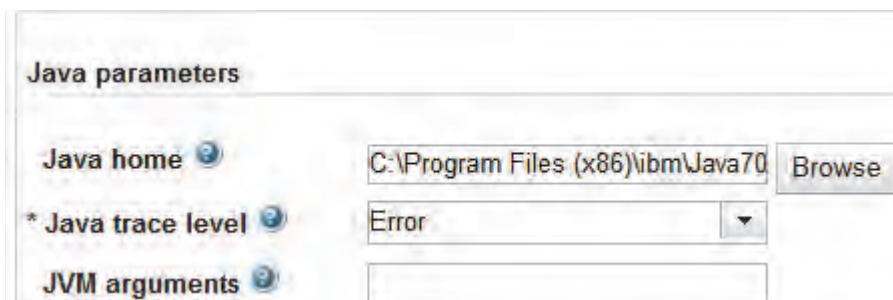
- c. Keep the current value and click **Next**.



From this pane, you specify the version of SNMP of the monitored system. Selecting a version of SNMP brings forward the configuration parameters of that version in the next pane and you can set the default values. Because this property was defined only as global, and not per subnode, you cannot change this property per subnode.

- d. Keep the default **SNMP version** and click **Next**.

You are prompted to Java parameters.



You are prompted to Java parameters.

- e. Keep the current value and click **Next**.

You are prompted for Java API client configuration properties.



- f. Do not set any Java API properties and click **Next**.

Network management properties

Ping configuration file 

Here you identify the ping configuration file that identifies which hosts to monitoring.

- g. Click **Browse** and open the C:\AB_Files\PingCfg.txt file.

Ping configuration file 

- h. Click **Next**.

Configuration for a proxy server used by HTTP providers

HTTP Data Provider Proxy Information

Proxy Hostname 	<input type="text"/>
Proxy Port 	80
Proxy User Name 	<input type="text"/>
Proxy Password 	<input type="text"/>
Confirm Proxy Password	<input type="text"/>

Here you configure the connection through a proxy server for HTTP monitoring. This agent does not use a proxy server.

- i. Click **Next**.

Configuration for monitoring a list of URLs

* HTTP URLs File 

* Page Object Collection 

Here you define which HTTP URLs to monitor and whether to monitor embedded objects.

- j. Click **Browse** and open the C:\AB_Files\urls.txt file.

* HTTP URLs File 

* Page Object Collection 

- k. Keep the default **Yes** in Page Object Collection and click **Next**.

You are prompted for SNMP configuration properties and to manage subnodes.

SNMP	
SNMP user name	New...
SNMP security level	noAuthNoPriv
SNMP community name	
Confirm SNMP community name	
SNMP auth password	
Confirm SNMP auth password	
SNMP auth protocol	MD5
SNMP priv password	
Confirm SNMP priv password	
SNMP port number	161
SNMP priv protocol	DES
SNMP host	

Here you define and configure a subnode for each remote system you want to monitor with the SNMP subnode. You can set default values for SNMP host and port number. In this lab, you monitor all three lab systems with the SNMP subnode.

8. Enter the following field to be set for each subnode:

- SNMP community name: **ibm**
- Confirm SNMP community name: **ibm**
- SNMP host: **192.168.1**.

SNMP auth protocol	MD5
SNMP community name	***
Confirm SNMP community name	***
SNMP host	192.168.1
SNMP user name	



Hint: The fields are case-sensitive.

9. Add the following three hosts as SNMP nodes:

- **WIN1, 192.168.1.103**
- **WIN2, 192.168.1.109**
- **LIN4, 192.168.1.107**

a. Click **New**.

The subnode is created.



Hint: You might need to scroll down to see the new subnode definition.

<input type="button" value="Delete"/>	* SNMP	<input type="text"/>
	SNMP port number	161
	SNMP auth password	<input type="password"/>
	Confirm SNMP auth password	<input type="password"/>

Notice that values that you entered are repeated for each subnode. SNMP is the name that is shown in the Tivoli Enterprise Portal. SNMP host is the host name or IP address of the remote system to be monitored.

b. Enter the host name in the **SNMP** field.

- c. Enter the last numbers of the IP address in the **SNMP host** fields.

<input type="button" value="Delete"/>	
* SNMP	
SNMP user name ?	WIN1
SNMP security level ?	noAuthNoPriv
SNMP community name ?	***
Confirm SNMP community name	***
SNMP auth password ?	
Confirm SNMP auth password	
SNMP auth protocol ?	MD5
SNMP priv password ?	
Confirm SNMP priv password	
SNMP port number ?	161
SNMP priv protocol ?	DES
SNMP host ?	192.168.1.103

- d. Repeat these steps for the remaining two hosts by clicking **New** and entering the **SNMP** and **SNMP host** values for the other hosts.

<input type="button" value="Delete"/>	
* SNMP	
SNMP user name ?	WIN2
SNMP security level ?	noAuthNoPriv
SNMP community name ?	***
Confirm SNMP community name	***
SNMP auth password ?	
Confirm SNMP auth password	
SNMP auth protocol ?	MD5
SNMP priv password ?	
Confirm SNMP priv password	
SNMP port number ?	161
SNMP priv protocol ?	DES
SNMP host ?	192.168.1.109

<input type="button" value="Delete"/>	
* SNMP	LIN4
SNMP user name ?	
SNMP security level ?	noAuthNoPriv
SNMP community name ?	***
Confirm SNMP community name	***
SNMP auth password ?	
Confirm SNMP auth password	
SNMP auth protocol ?	MD5
SNMP priv password ?	
Confirm SNMP priv password	
SNMP port number ?	161
SNMP priv protocol ?	DES
SNMP host ?	192.168.1.107

e. Click **OK** to save the agent configuration.

f. Click **Yes** to restart your agent.

The MTMS utility main window is displayed.

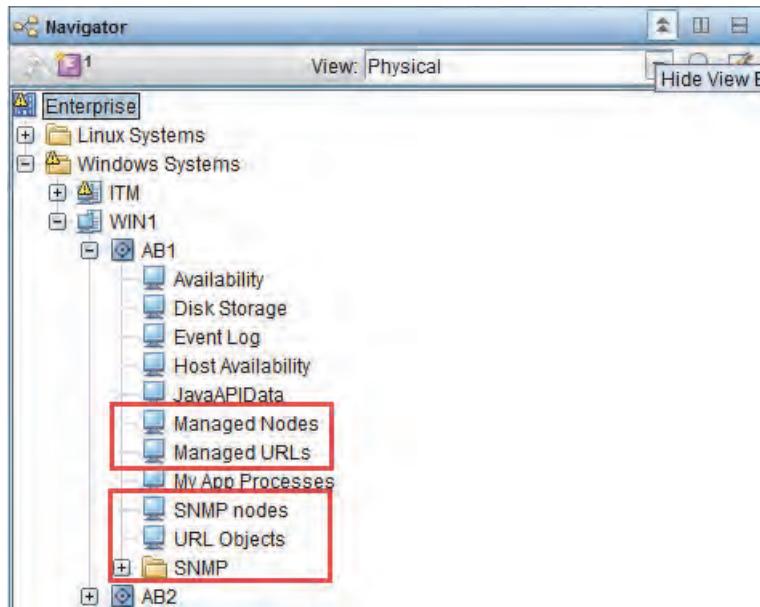
10. Restart your AB1 agent if you were not prompted to restart it.

You completed the installation, configuration, and starting of the updated AB1 agent.

Confirm the agent in the Tivoli Enterprise Portal

11. Start the TEP client and expand **AB1** under **WIN1**.

12. Confirm your new navigator nodes.



13. Investigate each new Navigator node. Confirm that each default workspace retrieves data.

Confirm the following items:

- Managed Nodes, Managed URLs, and URL Objects have valid data.
- You have three subnodes under SNMP, each displaying their version of the same data sources. Additionally, each subnode has an SNP Performance Object Status node, which provides the status information about each monitor that runs on the subnode.



Note: If your subnode names do not display correctly with the host names included, you might have to restart the TEP client or Tivoli Enterprise Portal Server.

Exercise 3 Allow data sources to be optional

Your company has many servers with similar and dissimilar resources that need to be monitored. In this lab, you build an agent to monitor these resources:

- Queue status that is gathered directly from the application server with the Java Management Extension (JMX) data source
 - Processor information that is gathered from the application servers with the Common Information Model (CIM) data source
- Both JMX and CIM are multiplatform and can provide the data that is needed from both Windows and Linux.
- Software code status in a database

Modify the AB2 agent so that the data sources that are monitored are determined when the agent is installed and configured on a specific server. Enable this ability by putting each optional data source in its own subnode.

Add monitoring of Common Information Management data sources

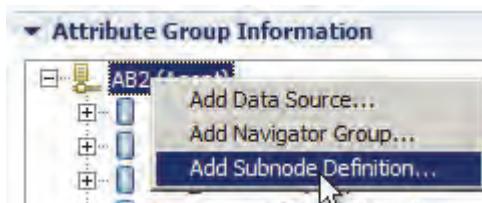
In this section, you enable the agent to monitor processors that run on a system that contains a Common Information Management data source.

1. If it is not running, start Agent Builder.
2. Close all open Agent Editor windows.
3. If the **AB2** agent is not already open, expand **Agent 2** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
4. Click the **Data Sources** tab.



Note: If you deployed the AB2 agent into the IBM Performance Management environment, you also see the Availability_Filtered (Filter) data source.

- Right-click **AB2 (Agent)** and select **Add Subnode Definition...**.



The Subnode Information window opens.

- Enter the following information:
 - Name: **AB_Processor**
 - Type: **CIM**
 - Description: **Subnode to gather processor data through CIM**

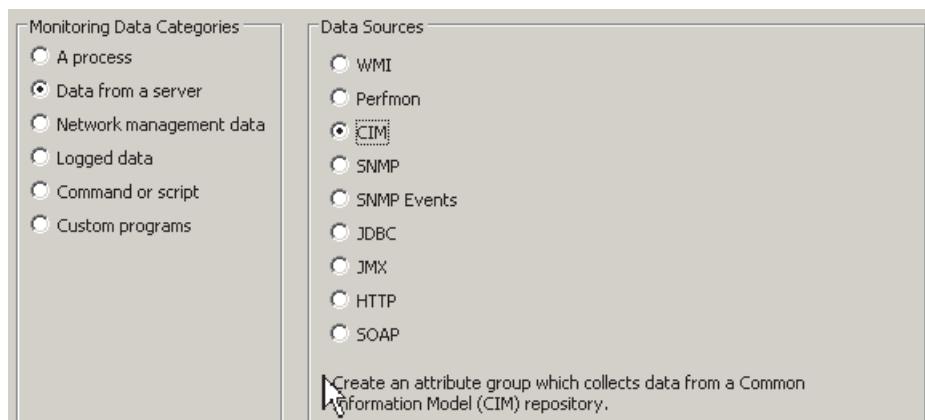
Subnode Information	
Name	<input type="text" value="AB_Processor"/>
Type	<input type="text" value="CIM"/>
Description	<input type="text" value="Subnode to gather processor data through CIM "/>
Company identifier	<input type="text" value="IBM_E"/>
Subnode identifier	<input type="text" value="K01CIM"/>
<input checked="" type="checkbox"/> Show nodes attribute group for this type of subnode	

Note: Type is a unique three-character code identifying this subnode.

- Click **Next**.

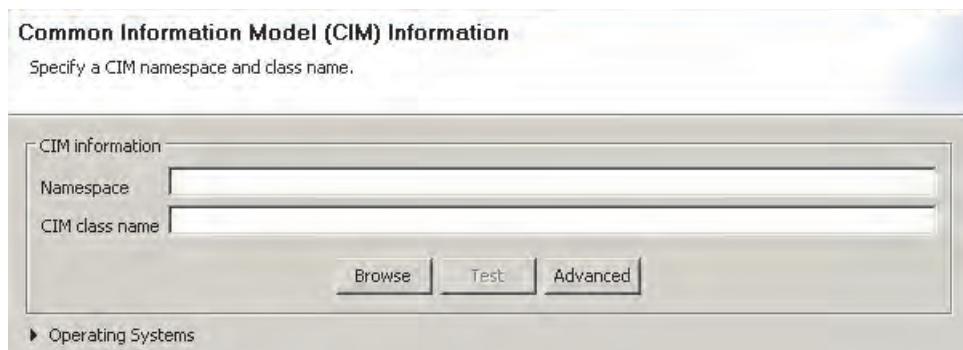
The Initial Subnode Data Source window is displayed.

- Click Data from a server in the Monitoring Data Categories section and CIM in the Data Sources section.



- Click Next.

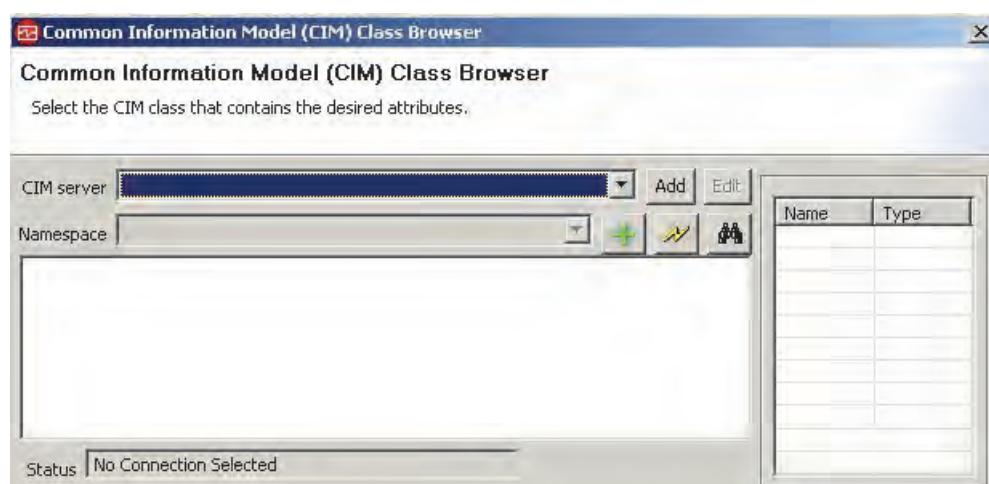
The Common Information Model (CIM) Information window opens.



This data source works similarly to WMI except that it is multiplatform.

- Click Browse.

The Common Information Model (CIM) Browser opens.



- Click Add.

12. Select **CIM server** and click **Next**.



Both WIN1 and WIN2 run a CIM managed object server.

13. Enter the following values:

- **Host name:** WIN2
- **User name:** Administrator
- **Password:** object00
- Save the password: **Selected**



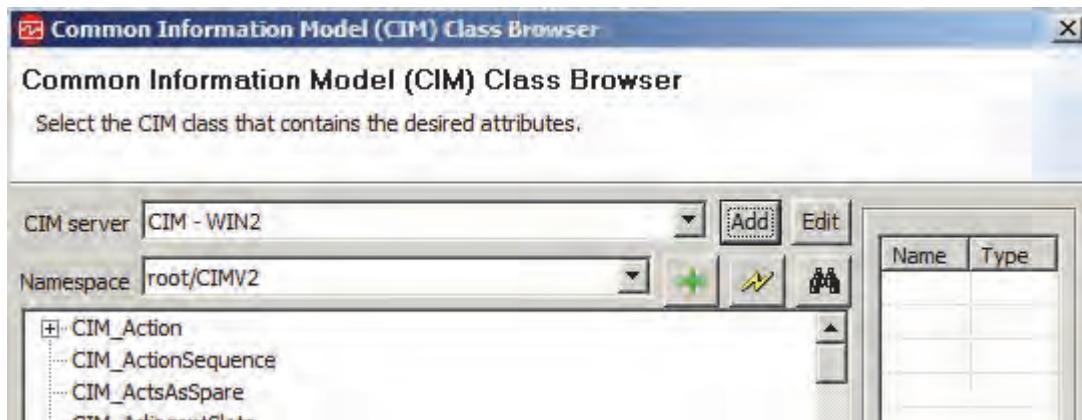
14. Click **Test Connection**.

You receive an indication that the connection was successful.



15. When you have a successful test connection, click **Finish**.

The browser is displayed with the root **Namespace** selected.



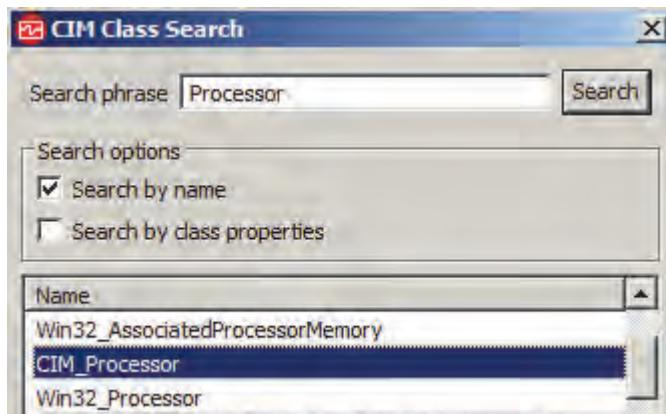
16. Browse the namespaces by clicking the **Namespace** menu.

You see a list of all the CIM namespaces on WIN2. The list of namespaces is similar to the WMI namespaces from [Unit 4, Exercise 1](#) on page 4-3. For example, notice the CIMV2. It is the same namespace from which you monitored the HTTP Server and DB2 process information and the disk space information. Because WMI namespaces are built on the CIM model, the CIM agent you are using, IBM Director, presents all WMI namespaces as CIM namespaces.

17. Select the **root/CIMV2** namespace.

18. Using the search tool, search for class names with the word **Processor** in them.

19. Select **CIM_Processor** and click **OK**.

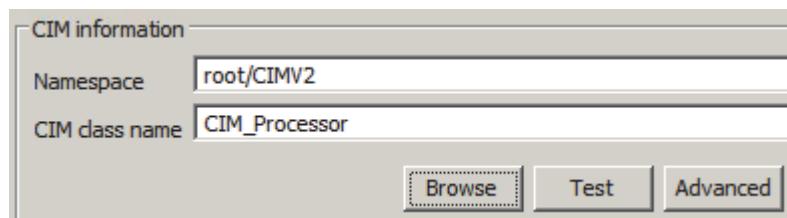


The CIM browser is displayed.

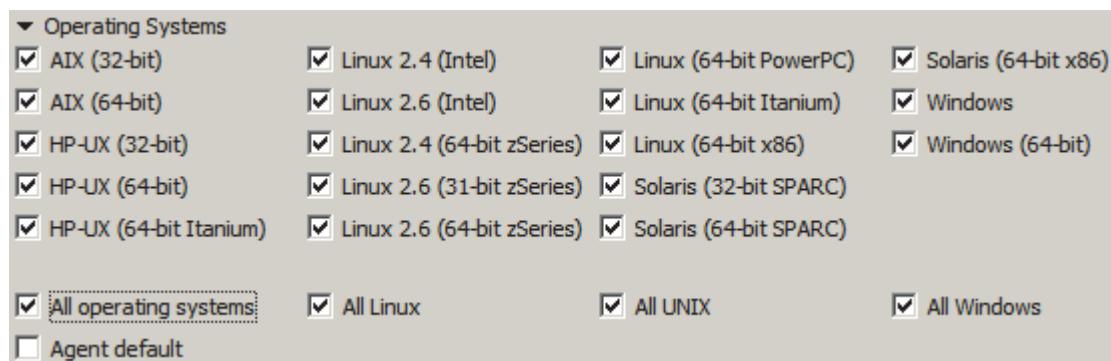
CIM_Processor properties	
Name	Type
AddressWidth	uint16
Availability	uint16
Caption	string
ConfigMan...	uint32
ConfigMan...	boolean

20. Click **OK**.

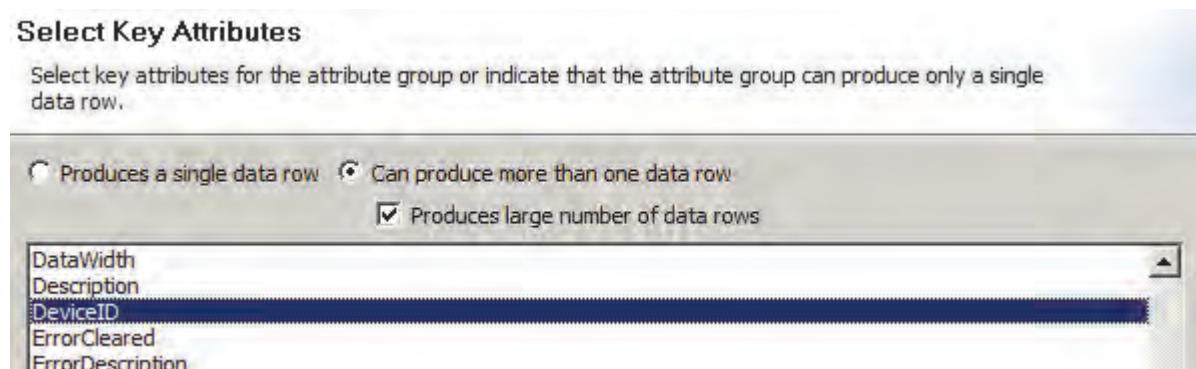
Your CIM information is displayed in the Common Information Model (CIM) Information window.



Because CIM is platform independent, unlike WMI, this monitor can collect CIM_Memory from any host with a CIM agent.

21. Expand Operating Systems and select **All operating systems** in the **Operating Systems** pane.22. Click **Next**.

The Select key attributes window opens.

23. Keep **Can produce more than one data row** selected and click **DeviceID** from the list.24. Click **Finish**.

You return to the **Data Source Definition** tab and the new AB_Processor subnode is displayed.

Data Source Definition

Attribute Group Information

- AB2 (Agent)
 - + Availability filters
 - + Processor_Utilization (Script)
 - + AppDataSocket (Socket)
 - AB_Processor (Subnode)
 - + CIM_Processor (CIM)

25. Expand **AB_Processor (Subnode)**.

The new CIM data source is displayed.

Data Source Definition

Attribute Group Information

- AB2 (Agent)
 - + Availability filters
 - + Processor_Utilization (Script)
 - + AppDataSocket (Socket)
 - AB_Processor (Subnode)
 - + CIM_Processor (CIM)

26. Browse the attributes of the **CIM_Processor** attribute group.

27. Click the **Runtime Configuration** tab.

A list of new CIM runtime configuration objects is displayed.

Runtime Configuration Information

Runtime Configuration Information

- Custom Configuration
- Configuration for Common Information Model (CIM)**

28. Expand **Configuration for Common Information Model (CIM)** and **CIM Connection Configuration**.

Runtime Configuration Information

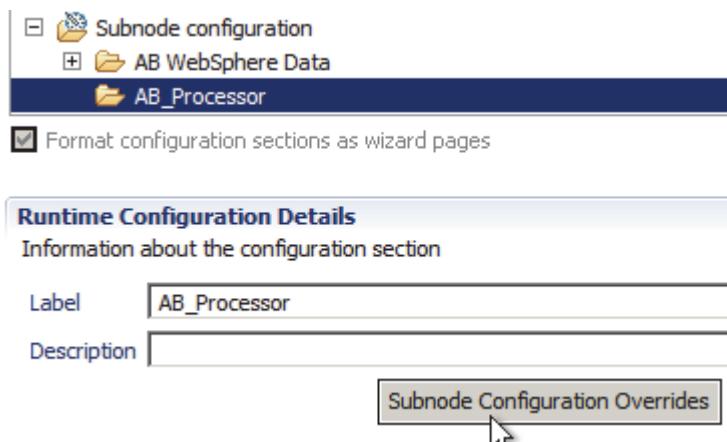
Runtime Configuration Information

- Configuration for Common Information Model (CIM)
 - CIM Connection Configuration
 - + CIM Local or Remote
 - + CIM user ID
 - + CIM password
 - + CIM host name
 - + CIM over SSL

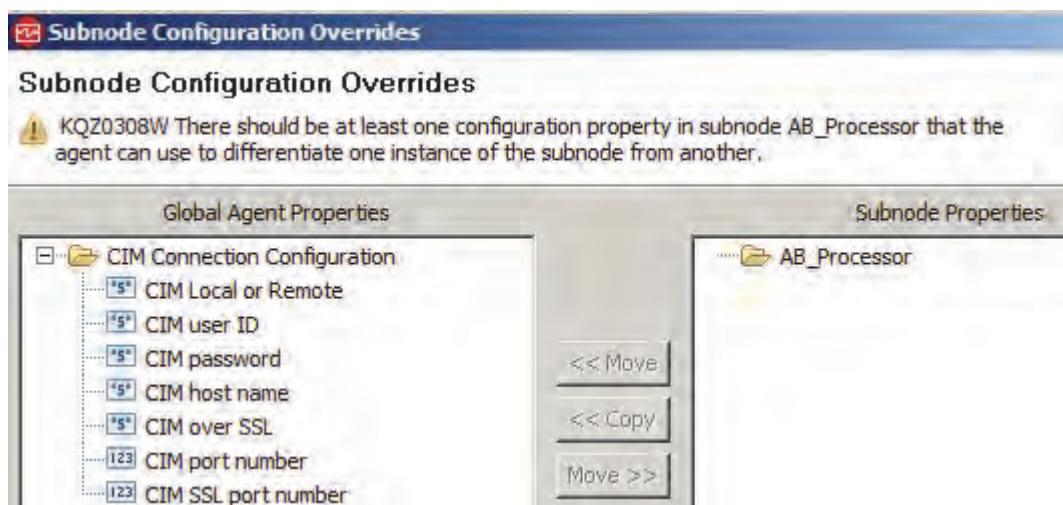
Format configuration sections as wizard pages

These CIM parameters are set when the agent is configured. You can require values, set default values, and configure if properties are prompted for when configuring the agent. By moving these configuration parameters into the subnode, you can set them differently for each subnode.

29. Set the Configuration for Common Information Model properties to be configured per subnode.
 - a. Close **Configuration for Common Information Model (CIM)**.
 - b. Expand **Subnode Configuration** and click **AB_Processor**.
 - c. Click **Subnode Configuration Overrides**.



The Subnode Configuration Overrides window opens.



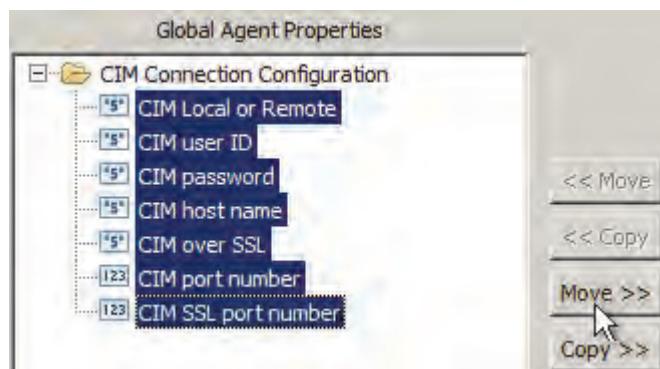
In this window, you can determine which configuration data, which you set during installation of the agent, is configurable for each subnode. Currently, all properties are global. At the time of the installation, you enter them once and they apply to all subnodes. Because the agent might monitor multiple remote CIM servers, this result is not acceptable.

At a minimum, you set the CIM host name as a subnode property so that you can identify the host systems. Other information that you might set per subnode includes the user ID, password, and ports to be used. In this exercise, set all CIM possible properties configurable for each subnode.

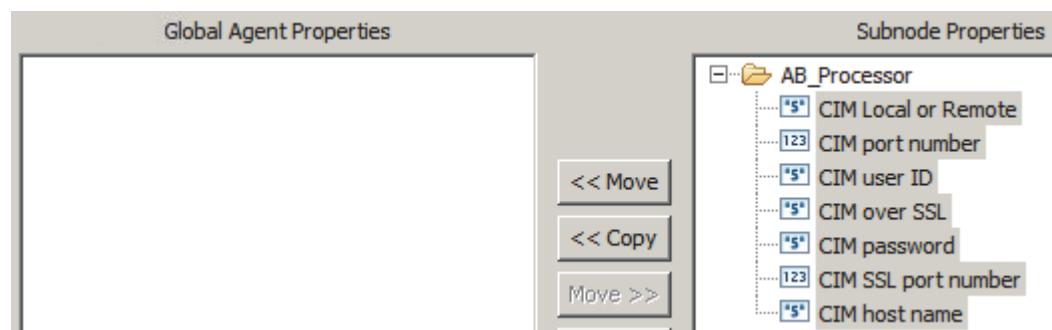
d. Select the following properties

- ◆ CIM Local or Remote
- ◆ CIM user ID
- ◆ CIM password
- ◆ CIM host name
- ◆ CIM over SSL
- ◆ CIM port number
- ◆ CIM SSL port number

e. Click **Move>>**.



The subnode configuration overrides looks like the following screen capture when you are done.



You set the configuration properties when you define subnodes.

30. Click **OK** to save your changes and exit the Subnode Configuration Overrides window.

You are returned to the Data Source Definition tab.

31. Save your agent project.

Add monitoring of remote WebSphere Application Server Java Management Extensions

In this section, you enable the agent to gather and monitor Active MQ queues within a WebSphere server.

Because this data source cannot be used on all servers where the agent is installed, you use a subnode to define the data to be retrieved. At run time, if you want to monitor this JMX data source, configure the agent to create a subnode.

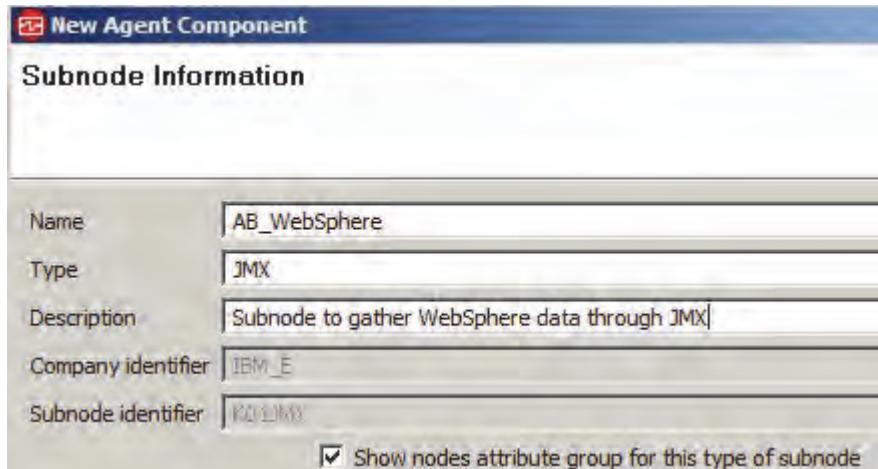
Complete the following steps:

32. In the **Data Sources** tab, right-click **AB2 (Agent)** and select **Add Subnode Definition**.

The Subnode Information window opens.

33. Enter the following information:

- Name: **AB_WebSphere**
- Type: **JMX**
- Description: **Subnode to gather WebSphere data through JMX**

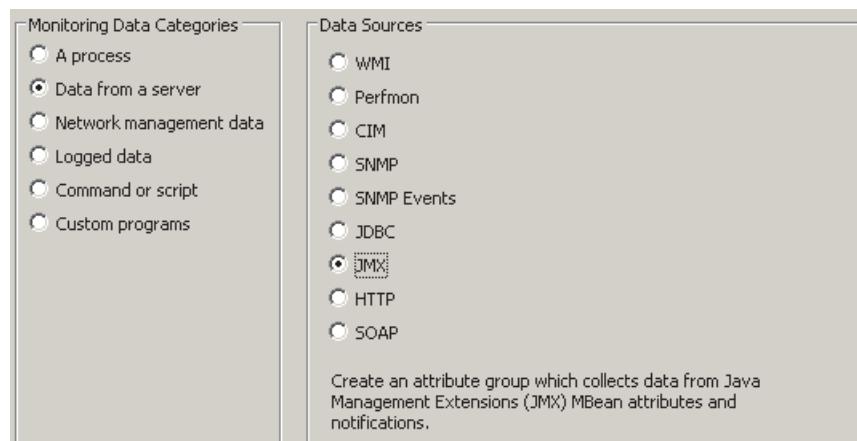


Note: Type is a unique three-character code identifying this subnode.

34. Click **Next**.

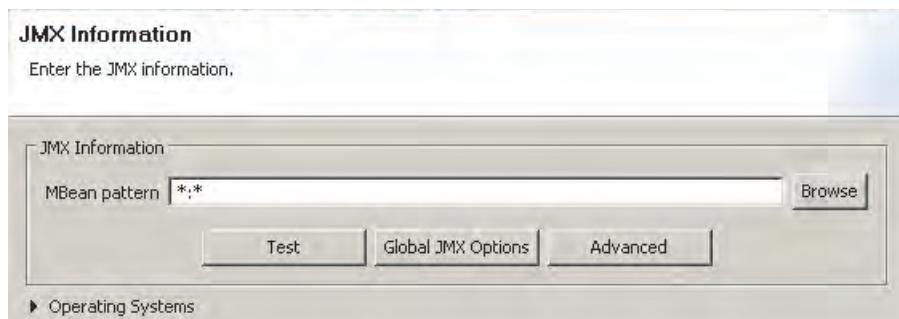
The Agent Initial Data Source window is displayed.

35. Click Data from a server under Monitoring Data Categories and JMX under Data Sources.



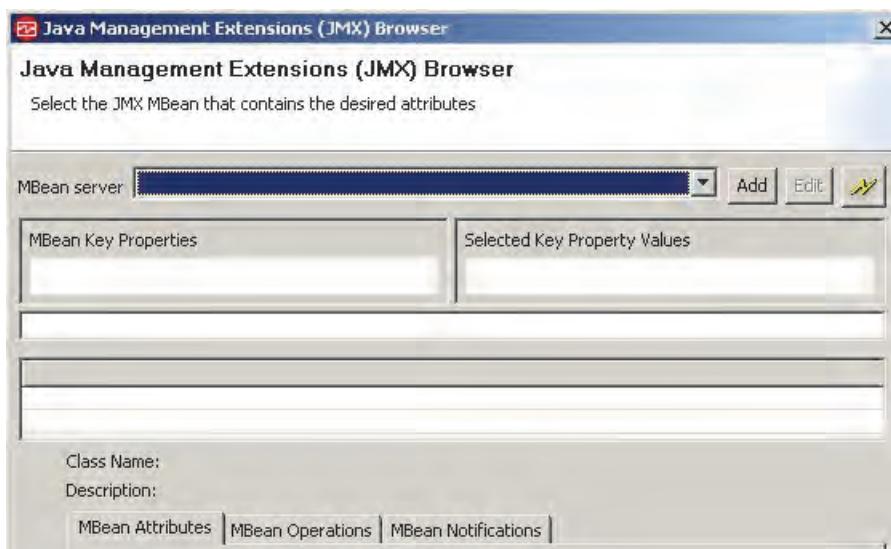
36. Click Next.

The JMX Information window opens.



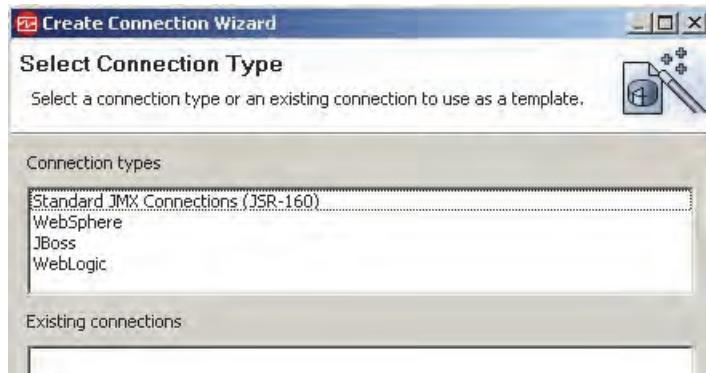
37. Click Browse.

The Java Management Extensions (JMX) Browser opens.

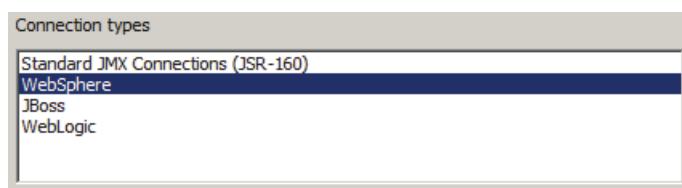


38. Click Add.

The JMX Connection Selection window opens.

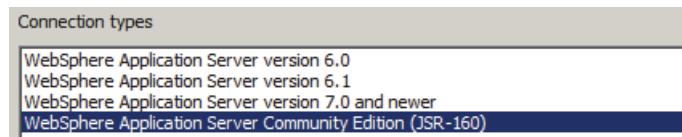


39. Click **WebSphere** in the **Connection Types** pane and click **Next**.

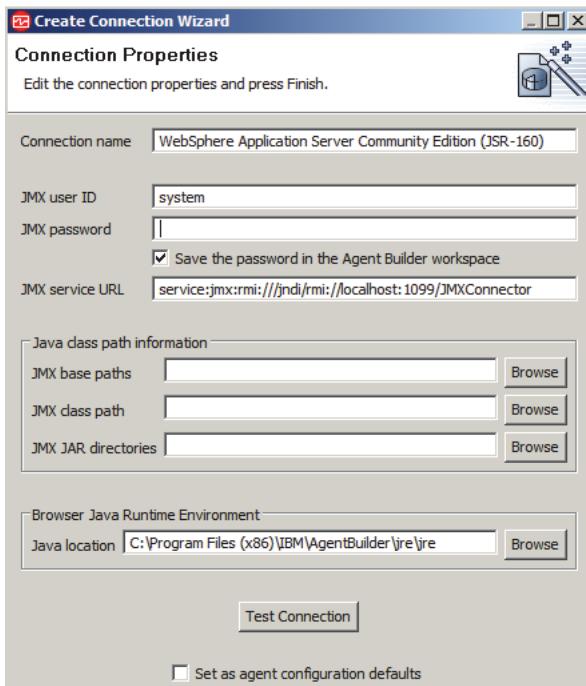


The connections templates are displayed in the window.

40. Click **WebSphere Application Server Community Edition** and click **Next**.



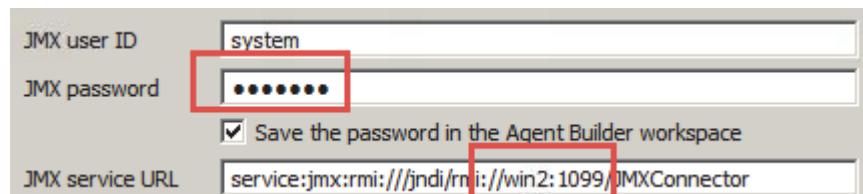
The Connection Properties window opens.



41. Enter **manager** in JMX password.

Because WIN2 is the only host with WebSphere Application Server Community Edition installed, it is the target for browsing.

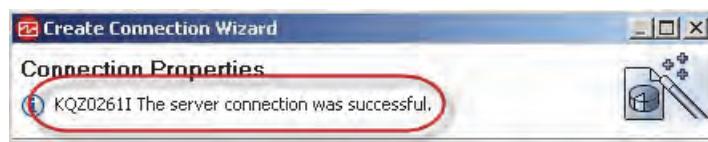
42. Replace **localhost** in JMX service URL with **WIN2**.



43. Click **Test Connection**.



You receive an indication that the connection was successful.

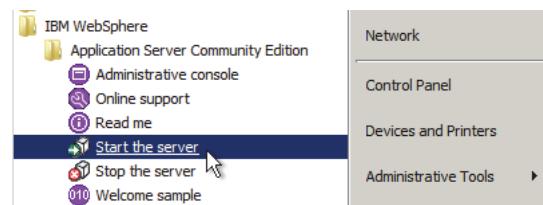


If your connection was not successful, do the following tasks:

- Confirm that WIN2 is running, its network card is enabled, and that the WIN1 server can be pinged.
- Confirm that WAS Community Edition is running.



- If it is not running, you must start it.



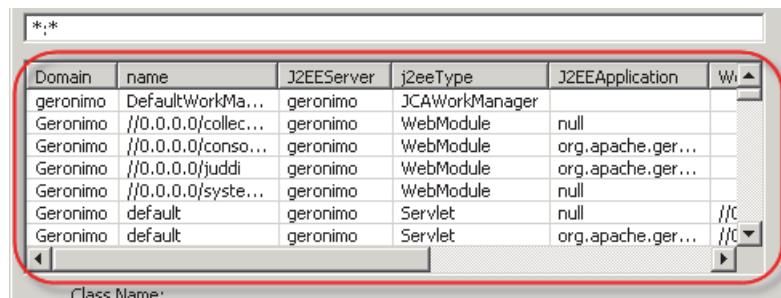
44. When you have a successful test connection, click **Finish**.

The Java Management Extensions (JMX) Browser is displayed with data now shown in the various windows.

The screenshot shows the JMX browser interface. At the top, it says "MBean server WebSphere Application Server Community Edition (JSR-160)". Below that are two panes: "MBean Key Properties" on the left and "Selected Key Property Values" on the right. The properties pane lists several checkboxes: [Domain], name, J2EEServer, j2eeType, J2EEApplication, WebModule, and type. The values pane is currently empty. At the bottom, there is a table with columns: Domain, name, J2EEServer, j2eeType, J2EEApplication, WebModule, and Other Key. The table contains three rows of data:

Domain	name	J2EEServer	j2eeType	J2EEApplication	WebModule	Other Key
Geronimo	//0.0.0.0/	io	geronimo	WebModule	null	
Geronimo	//0.0.0.0/activ...	geronimo		WebModule	null	
Geronimo	//0.0.0.0/CAHe...	geronimo		WebModule	null	

45. Browse the MBeans to familiarize yourself with the tool. The current pattern (*.*) retrieves all MBeans.
- Before selecting any MBean key properties, scroll through the MBean list to see the MBeans from which you can pull data.



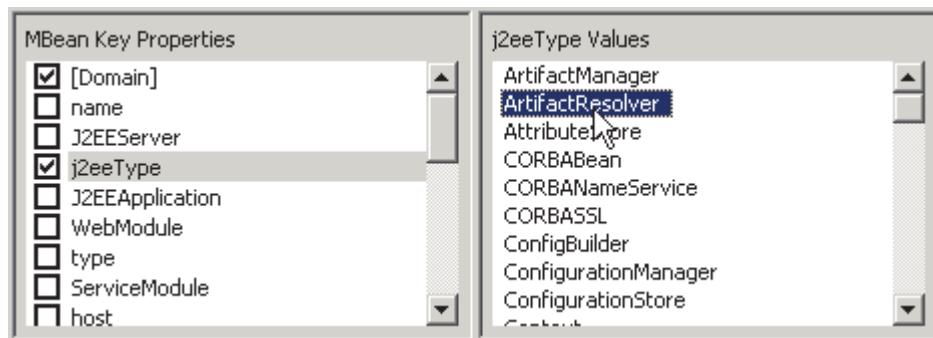
The screenshot shows a table titled 'MBeans' with a red box highlighting the first seven columns: Domain, name, J2EEServer, j2eeType, J2EEApplication, and WebModule. The table lists various Geronimo MBeans, such as DefaultWorkManager, JCAWorkManager, and several WebModules and Servlets.

Domain	name	J2EEServer	j2eeType	J2EEApplication	WebModule
geronimo	DefaultWorkMa...	geronimo	JCAWorkManager		
Geronimo	//0.0.0.0/collect...	geronimo	WebModule	null	
Geronimo	//0.0.0.0/consol...	geronimo	WebModule	org.apache.ger...	
Geronimo	//0.0.0.0/juddi	geronimo	WebModule	org.apache.ger...	
Geronimo	//0.0.0.0/system...	geronimo	WebModule	null	
Geronimo	default	geronimo	Servlet	null	//0.0.0.0/default
Geronimo	default	geronimo	Servlet	org.apache.ger...	//0.0.0.0/default

- Scroll to the right and review the column names.

Notice that the first set of column names matches the first names in the MBean Key Properties list. The final column, Other Key Properties, shows all the other MBean Key Properties.

- Select a single **MBean Key Property**. If **Key Property Values** are shown, click them one at a time.

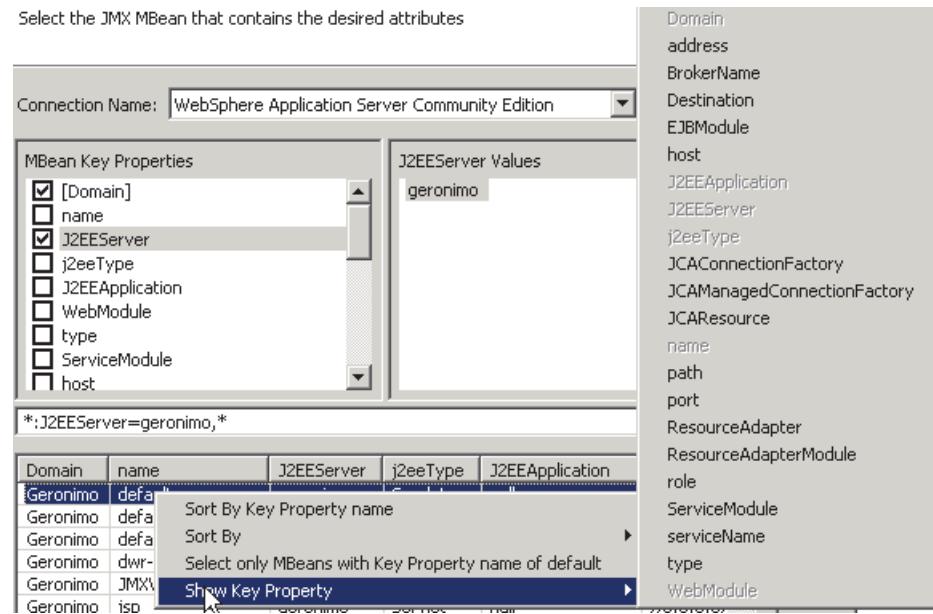


When you select an item from the Key Properties list, the item modifies the Object Name pattern. In the following table, the item shows the pattern all of the MBeans that match the new pattern. Selecting a key property also displays the list of values of that key property in the list on the right. Selecting a key property value further modifies the Object Name pattern and further filters the list of matching MBeans.

- Clear the **MBean Key Property** that you selected, select another property, and select its values one at a time. Repeat this process for several key properties.

You can select one or more key properties, but you can select only one value for each key property. You can also sort each column.

- e. Right-click any cell in the list of MBeans.



A context-sensitive menu is displayed with other options that can help you locate the MBean that you want.

The goal is to create an MBean pattern that filters all unwanted MBeans and then to select a specific MBean from the list that contains the attributes that you want to monitor. You can select only one MBean from the list, regardless of the object pattern.

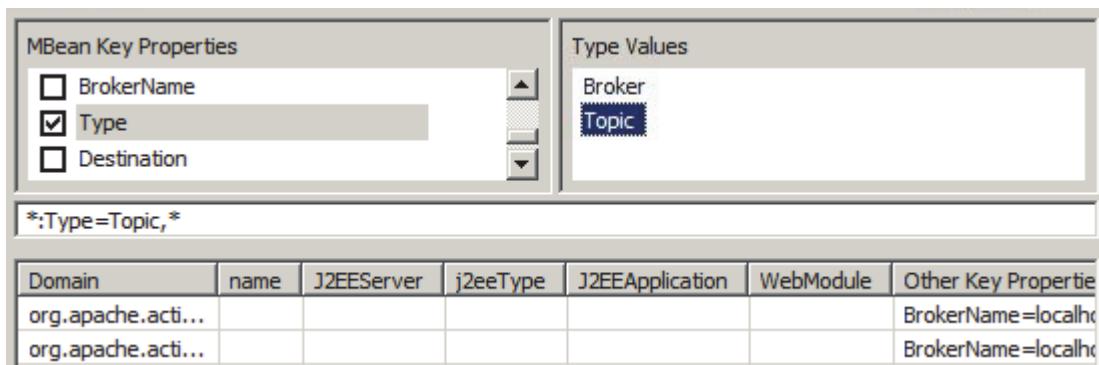
Use the Object Name pattern, the sorting feature, and the menu to locate the MBean you want.

- f. When you located a specific MBean of interest, select it in the list.

On the **MBean Attributes** tab is a list of attributes that are collected for the MBean attribute group you selected. Now it is time to select a specific MBean to monitor.

46. Clear any **MBean Key Properties** selections you made, but keep **Domain** selected.

47. Select **Type** in **MBean Key Properties** and **Topic** in **Type Values**.





Important: A type and Type MBean Key Property exist. Select **Type**.

48. Click the first MBean in the MBean list.

The screenshot shows the JMX Information window with the following details:

- MBean List:** The search bar shows "*:Type=Topic,*". The table lists two MBeans:

Domain	name	J2EEServer	j2eeType	J2EEApplication	WebModule	Other Key Properties
org.apache.acti...						BrokerName=localho
org.apache.acti...						BrokerName=localho
- MBean Details:**
 - Class Name: org.apache.activemq.broker.jmx.TopicView
 - Description: Information on the management interface of the MBean
 - Attributes Tab (selected): Shows the following table:

Name	Description	Type	Read/Write
Name	Attribute exposed for ma...	java.lang.String	Read Only
EnqueueCount	Attribute exposed for ma...	long	Read Only
DispatchCount	Attribute exposed for ma...	long	Read Only
DequeueCount	Attribute exposed for ma...	long	Read Only
ConsumerCount	Attribute exposed for ma...	long	Read Only
QueueSize	Attribute exposed for ma...	long	Read Only
MemoryPercentageUsed	Attribute exposed for ma...	int	Read Only
MemoryLimit	Attribute exposed for ma...	long	Read/Write

The attributes that are listed in the **MBean Attributes** tab are the ones to be monitored.

49. Click **OK** to save your selection.

The JMX Information window is displayed with the MBean pattern completed.

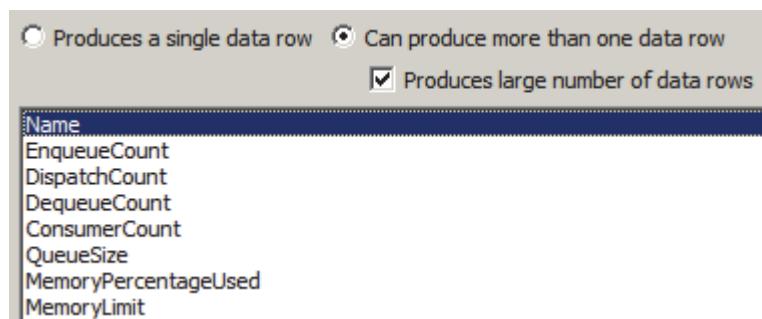
The screenshot shows the JMX Information window with the following details:

- MBean pattern:** *:Type=Topic,*
- Buttons:** Test, Global JMX Options, Advanced

50. Click **Next**.

The Select Key Attributes window opens and lists specific attributes for the MBean that you selected.

51. Keep **Can produce more than one data row** selected and click **Name** from the list.



52. Click **Next**.

The JMX Agent-Wide Options window opens.



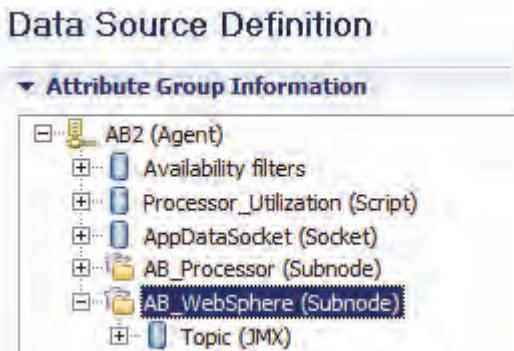
Even though you browsed MBeans from a WebSphere Community Edition application server, this MBean might be used by several types of application servers. With this window, you can identify all the types of application servers that you can monitor for this MBean.

53. Expand **WebSphere** and select **WebSphere Application Server Community Edition**.



54. Click **Finish**.

You are returned to the **Data Source Definition** tab.



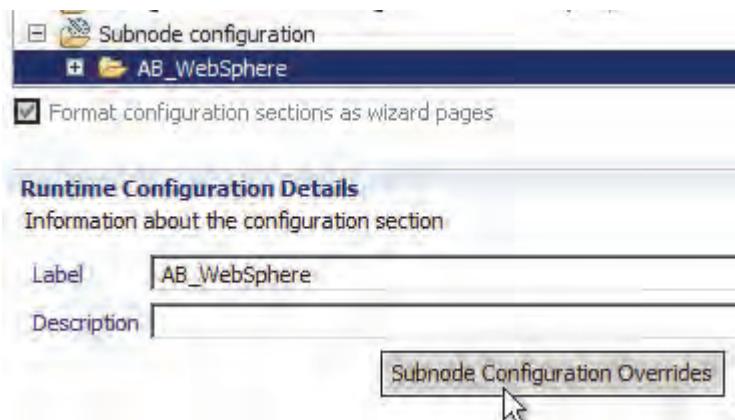
55. Browse the attributes of each new JMX attribute group.
56. Move the JMX Runtime Configuration properties to the subnode.
 - a. Click the **Runtime Configuration Information** tab.
 - b. Expand **Configuration for Java Management Extensions** and **WebSphere Application Server Community Edition**.



You set these WebSphere Application Server Community Edition properties when you configure the agent and establish the connection into the JMX server, whether it is local or remote. By moving these configuration parameters into the subnode, you can set them differently for each subnode.

- c. Close **Configuration for Java Management Extensions**.
- d. Expand **Subnode Configuration** and click **AB_WebSphere**.

e. Click **Subnode Configuration Overrides**.



f. Select the following properties and click **Move>>**.

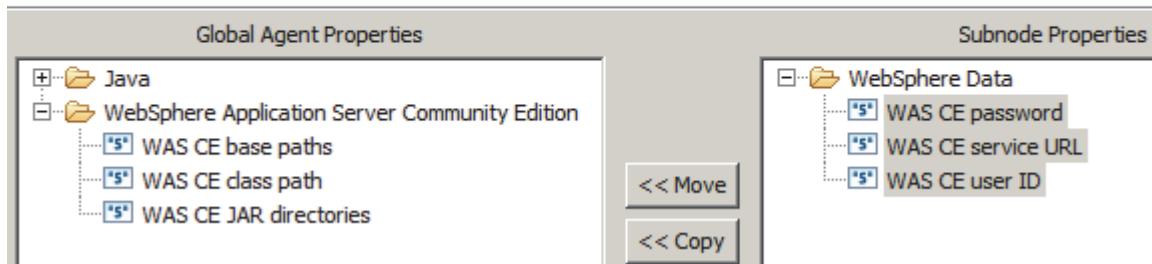
- ◆ WAS CD user ID
- ◆ WAS CE password
- ◆ WAS CE service URL



The subnode configuration overrides looks like the following screen capture when you are done.

Subnode Configuration Overrides

Add configuration properties to subnode WebSphere Data which will differentiate one instance of the subnode from another when the agent runs.



- g. Click **OK** to save your changes and exit the window.
 57. Save your agent project.

Add a JDBC data source to the agent

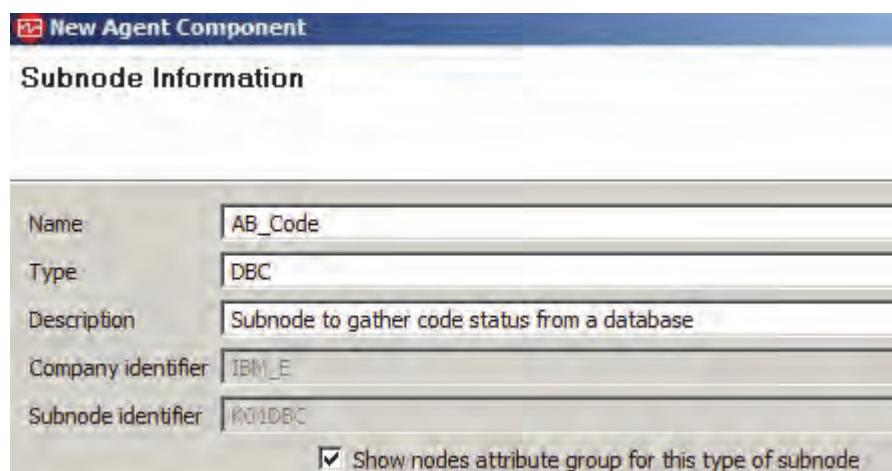
In this section, you enable the agent to gather user data from a database with JDBC.

58. In the **Data Sources** tab, right-click **AB2 (Agent)** and select **Add Subnode Definition**.

The Subnode Information window opens.

59. Enter the following information:

- Name: **AB_Code**
- Type: **DBC**
- Description: **Subnode to gather code status from a database**

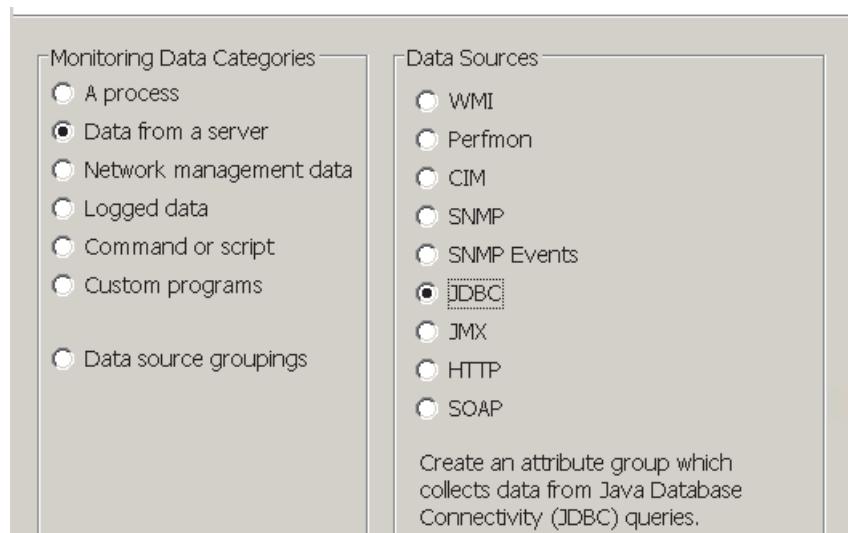


Note: **Type** is a unique three-character code that identifies this subnode.

60. Click Next.

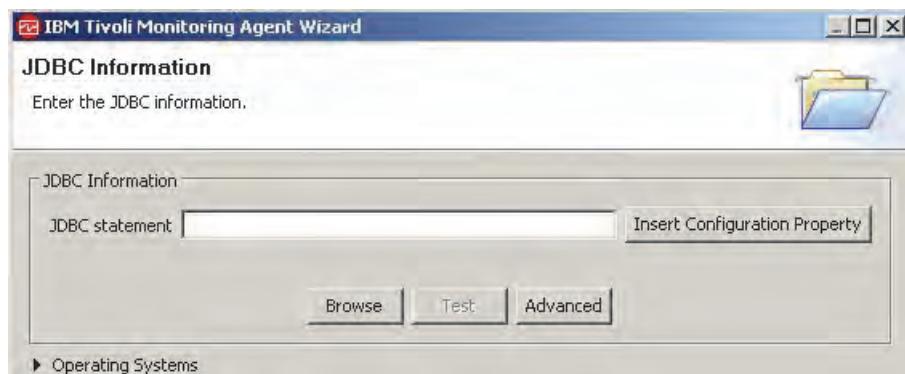
The Data Source Location window opens. Set your agent to monitor a database through JDBC.

61. Select **Data from a server from **Monitoring Data Categories** and **JDBC** from **Data Sources**.**



62. Click Next.

The JDBC Information window opens.

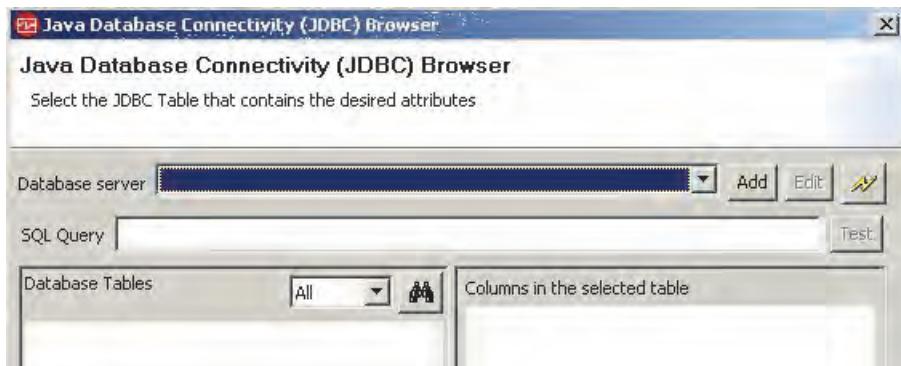


From this window, you can enter the JDBC statement to retrieve data from a database.

Typically, this statement is either an SQL command or a command that runs a stored procedure in the database. While you can enter the command manually, it is best to use the browser, which completes several tasks at once.

63. Click **Browse.**

The Java Database Connectivity (JDBC) Browser window opens.



To use the JDBC browser, you must first establish a JDBC connection with a local or remote database.

64. Click **Add**.



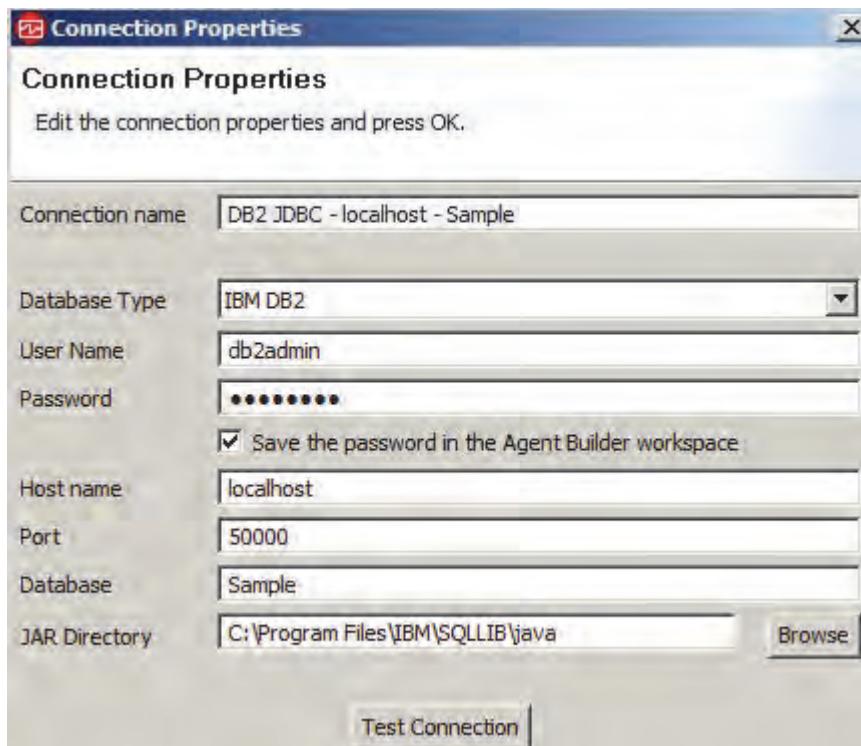
The Create Connection Wizard window opens.



65. Click **JDBC Connection** in the Connection types pane and click **Next**.

66. Keep the **Connection name** value as is and enter the following values:

- Password: **object00**
- Database: **Sample**



Notice that you can connect to a database on a remote host by entering the appropriate host name or IP address in the **Host name** field.

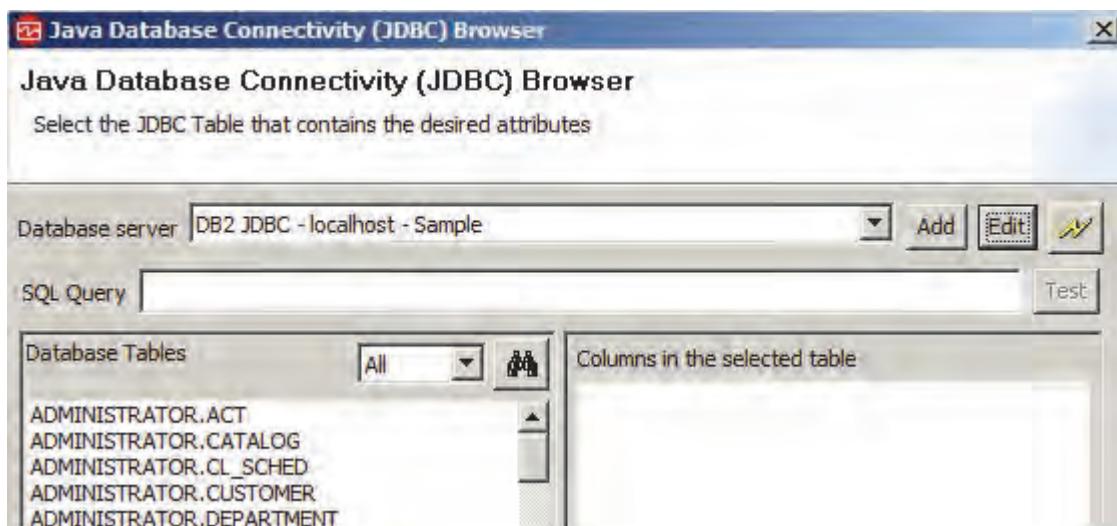
67. Click **Test Connection**.

A message that indicates a successful connection is displayed at the top of the window.

Connection Properties
(i) KQZ0261I The server connection was successful.

68. Click **Finish** to close the Connection Properties window.

You are returned to the JDBC browser, and the tables in the database are listed.



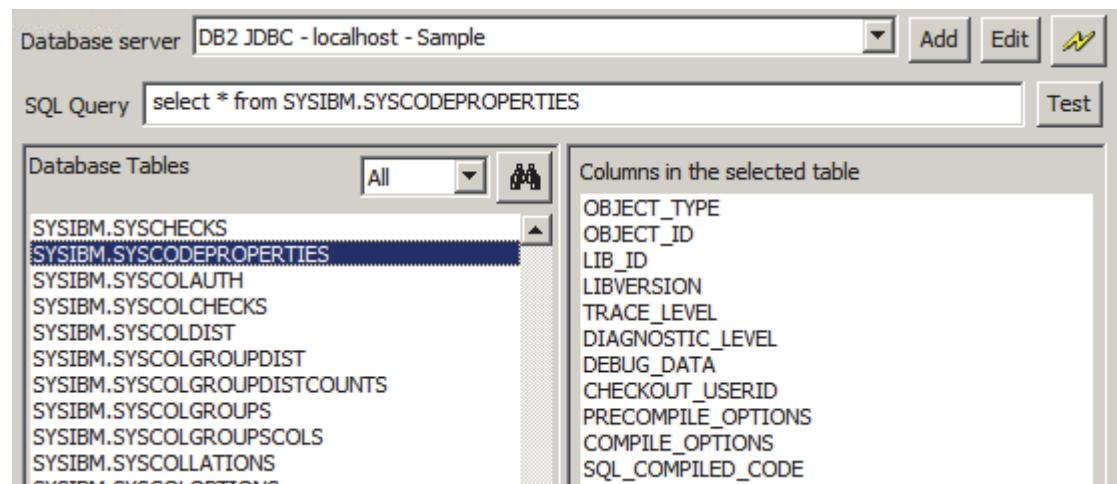
A browser does several important tasks:

- It confirms that your agent can connect to a target data source.
- It defines the exact data that you want gathered.
- It confirms that the data can be gathered.
- Additionally, the browser retrieves information about the data and the metadata from the data source. Using this metadata, Agent Builder can define attribute groups and attributes to hold the data, which is a critical task when defining an agent.

The purpose of browsing is not to identify a target host to monitor. The target host is identified when the agent is installed.

If you click a table in the Database Tables pane, the table column names are displayed in the Columns in the Select Table pane. An SQL SELECT statement is shown in the **SQL Query** field.

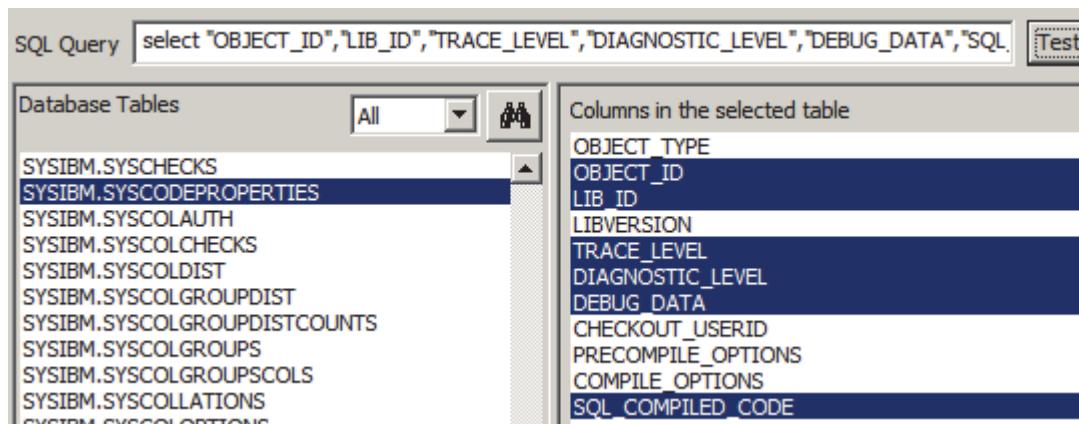
69. Find and select the **SYSIBM.SYSCODEPROPERTIES** table.



Notice that the SELECT statement includes an asterisk (*), which indicates that the command retrieves all the columns. You can manually edit this command to suit your needs. You can also select columns and modify the select statement.

70. Holding the CTRL key, click the following column names to select them. This action modifies the select statement to retrieve only data from the columns.

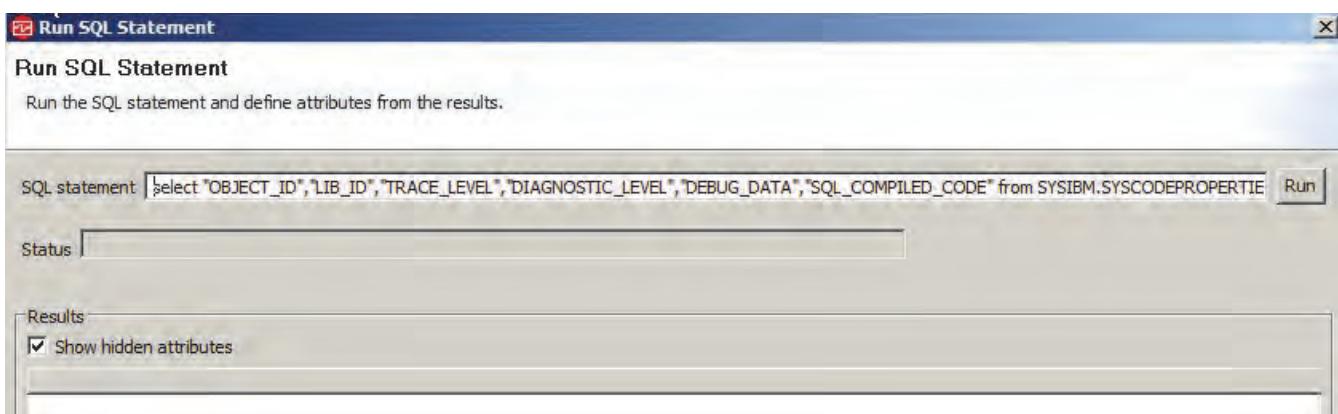
- OBJECT_ID
- LIB_ID
- TRACE_LEVEL
- DIAGNOSTIC_LEVEL
- DEBUG_DATA
- SQL_COMPILED_CODE



Notice that the SQL statement changed to retrieve only the columns that you selected.

71. Click **Test** to open the test utility.

The Test JDBC Statement window opens.



Notice that the SQL command you built is displayed. You can edit it manually here, if you want to test a different command.

72. Click **Run**.

73. If prompted, click **Yes** to view any suggested data types, and click **OK** to accept those data types.

The screenshot shows a utility window with a status bar at the top stating "Status Displaying the first 50 rows of 240 total rows." Below this is a section titled "Results" with a checked checkbox for "Show hidden attributes". A table displays six rows of data with columns: OBJECT_ID, LIB_ID, TRACE_LEVEL, DIAGNOSTIC_LEVEL, DEBUG_DATA, and SQL_COMPILED_CODE. The data is as follows:

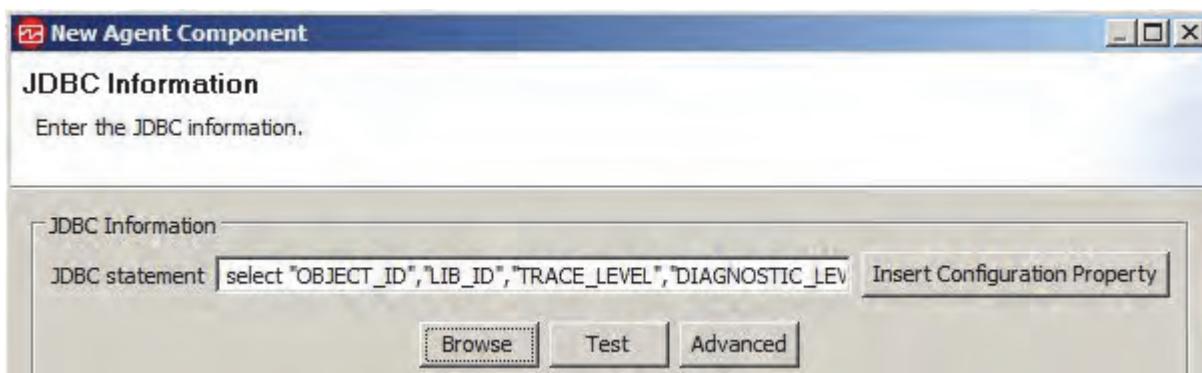
OBJECT_ID	LIB_ID	TRACE_LEVEL	DIAGNOSTIC_LEVEL	DEBUG_DATA	SQL_COMPILED_CODE
66124	1756752525	0	0	0	1356
66135	1854528969	0	0	0	1088
66139	1589140265	0	0	0	1008
66145	1864981170	0	0	0	1672

The utility displays the data that is retrieved. You can manually edit the statement and test different commands.

If you click **Cancel**, you return to the JDBC browser and the metadata that is gathered by this test is not kept. This action can result in an attribute group not being created or your changes not being saved. If you click **OK**, you exit the test utility and the JDBC browser, and the column definition information you viewed is used to create an attribute group and its attributes in your agent.

74. Click **OK** to close the test window.

The JDBC Information window opens.



Your SQL statement is shown in the **JDBC statement** field.

Next, you identify the operating systems on which this query is used.

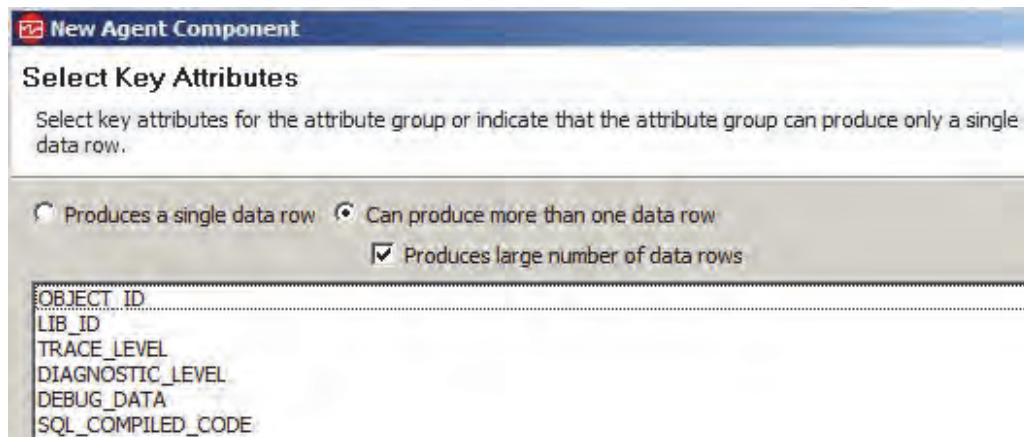
75. Expand **Operating Systems**.

As you can see, Agent Builder uses the default operating systems set for the agent, but because this agent monitors a database through a JDBC connection, this data source can be platform-independent.

76. Select **All operating systems**.

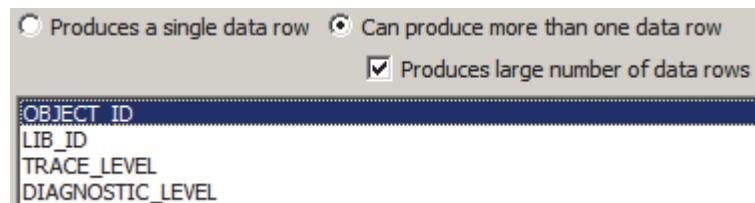
77. Click Next.

The Select Key Attributes window opens.



If the retrieved data consists of more than one row of data, you must define key attributes. Then, the agent can identify unique records and manage the data. For example, this SQL command retrieves information about Tivoli Enterprise Portal users. The agent needs to know which attribute uniquely identifies each user.

78. Select **OBJECT_ID.**

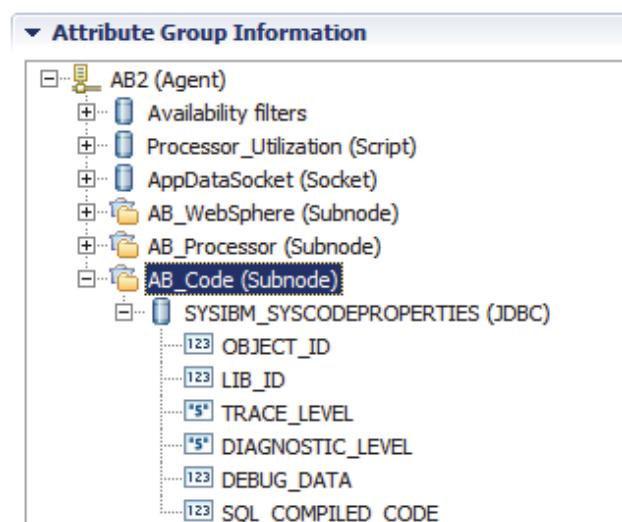


79. Click Finish.

You are returned to the **Data Source Definition** tab with the new subnode and attribute group.

80. Expand **AB_Code (Subnode).**

81. Expand **SYSIBM_SYSCODEPROPERTIES (JDBC).**



Notice that the attribute group name matches the target table name in the database. The attribute list is the subset of columns that you identified in your SQL command, not the full list of columns in the database table.

82. Move the JDBC Runtime Configuration properties to the subnode.

- Click the **Runtime Configuration Information** tab.

- Expand **Subnode Configuration** and click **AB_Code**.

You set these JDBC properties when you configure the agent and establish the connection into the database server, whether it is local or remote. By moving these configuration parameters into the subnode, you can set them differently for each subnode.

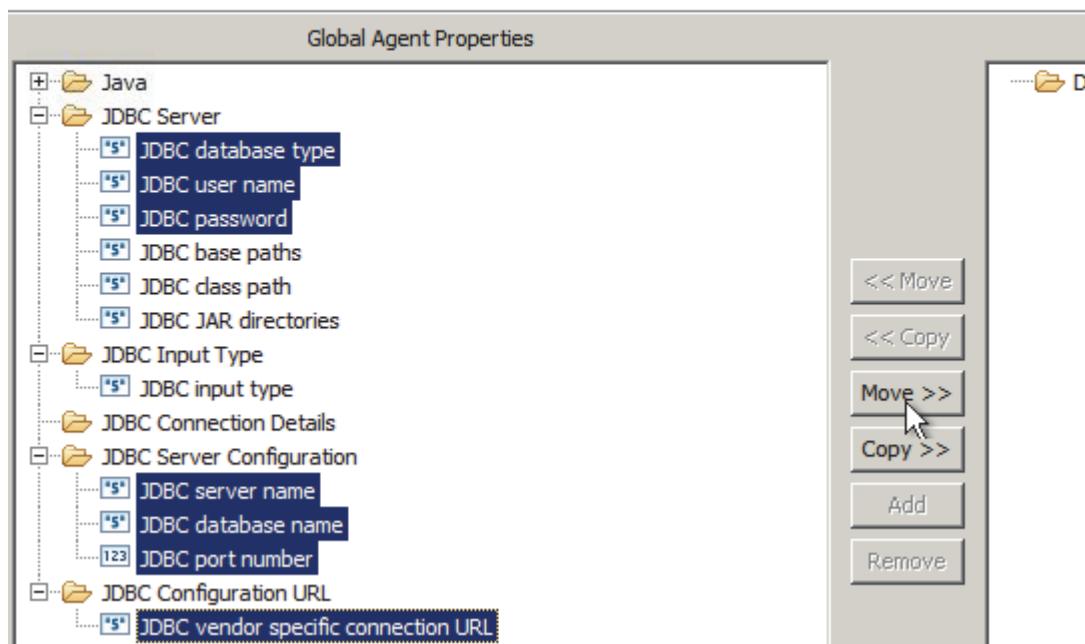
- Click **Subnode Configuration Overrides**.

- Select the following properties and click **Move>>**.

- ◆ JDBC database type
- ◆ JDBC user name
- ◆ JDBC password
- ◆ JDBC server name
- ◆ JDBC database name
- ◆ JDBC port number
- ◆ JDBC vendor specific connection URL

Subnode Configuration Overrides

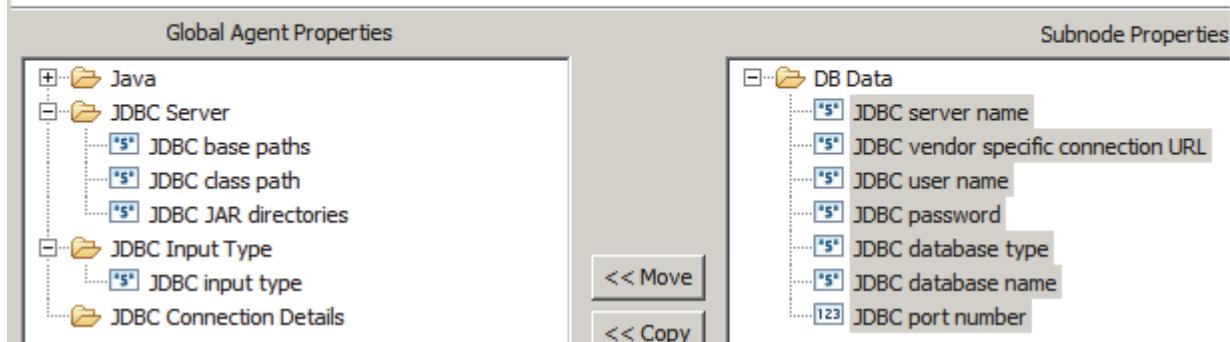
 KQZ0308W There should be at least one configuration property in subnode DB Data that the agent can use



The subnode configuration overrides looks like the following screen capture when you are done.

Subnode Configuration Overrides

Add configuration properties to subnode DB Data which will differentiate one instance of the subnode from another when the



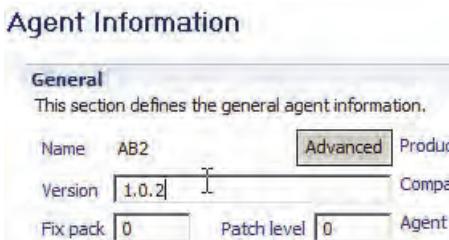
- e. Click **OK** to the window.
83. Save your agent project.

You successfully added a JDBC data source to this agent.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

84. Select to the **Agent Information** tab.
85. Change the version for 1.0.1 to **1.0.2**.



Important: If your initial version number is not 1.0.1, increase the current value by 1 to ensure that your monitoring server identifies the changes.

86. Save your agent project.

Exercise 4 Install and confirm the updated AB2 agent in an IBM Tivoli Monitoring environment

In this exercise, you test your updated AB1 agent and confirm that JMX, JDBC, and CIM data are being gathered in an IBM Tivoli Monitoring environment.

Create the updated AB2 agent installation scripts

1. In Agent Builder on WIN1, generate the AB2 agent installation scripts.
2. Delete the contents of the **C:\share\K00** directory.
3. Copy the new **smai-ab2-01.02.00.00.zip** file into **C:\share\K00** directory.
4. Extract the files from **smai-ab2-01.02.00.00.zip** into **C:\share\K00** with **7-Zip**.



Important: Your agent version numbers are different if you updated your agent more times than the labs.

Install the updated AB2 application support on ITM

5. Install the IBM Tivoli Monitoring application support on ITM.
 - Use the drive map to **\WIN1\share**.
 - Run the following commands:

```
installIraAgentTEMS.bat C:\IBM\ITM -h ITM -u sysadmin -p object00  
installIraAgentTEPS.bat C:\IBM\ITM
```

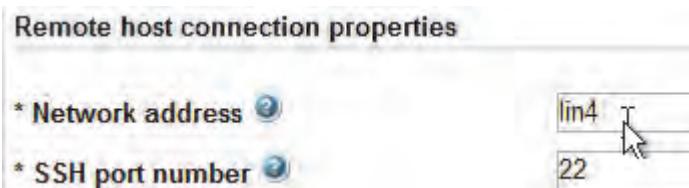


Hint: Before installing the Tivoli Enterprise Portal Server support, do not forget to close the TEP client.

Reinstall and configure the updated AB2 agent on WIN1

In this section, do the following actions:

- Reinstall the AB2 agent on WIN1
 - Create an AB_WebSphere subnode to monitor WebSphere on WIN2
 - Create an AB_Code subnode to monitor code data in the Sample database that uses the JDBC data source
 - Do not create an AB_Processor subnode to monitor WIN2
6. Reinstall the AB2 agent on WIN1 in any manner you would like.
7. Configure your **Monitoring Agent for AB2** in the MTMS utility.
- a. Locate the AB2 agent in the MTMS utility.
 - b. Right-click the agent and select **Reconfigure**.
You are prompted to configure the TEMS connection.
 - c. Ensure that **ITM** is the target TEMS and click **OK**.
The Agent Configuration window opens. You are prompted for the remote host to monitor with the script data source.



- d. Keep the current values and click **Next**.

You are prompted for the user ID and password of the remote host to monitor with the script data source.

- e. Keep the current values and click **Next**.

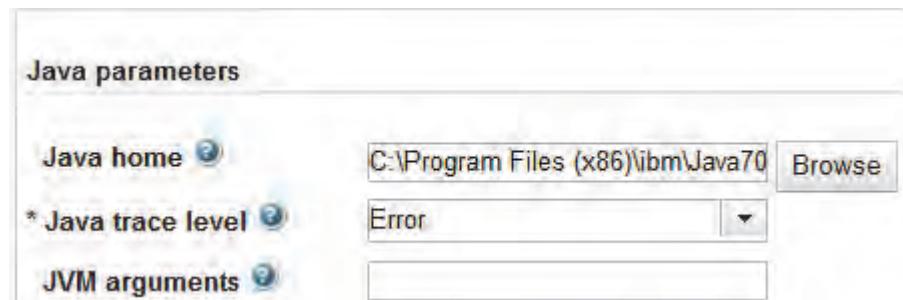
You are prompted for the socket to use to monitor for data from the socket data source.

- f. Keep the current values and click **Next**.

You are prompted for Java configuration parameters. The default location of Java (Java 50) is not valid on this host.

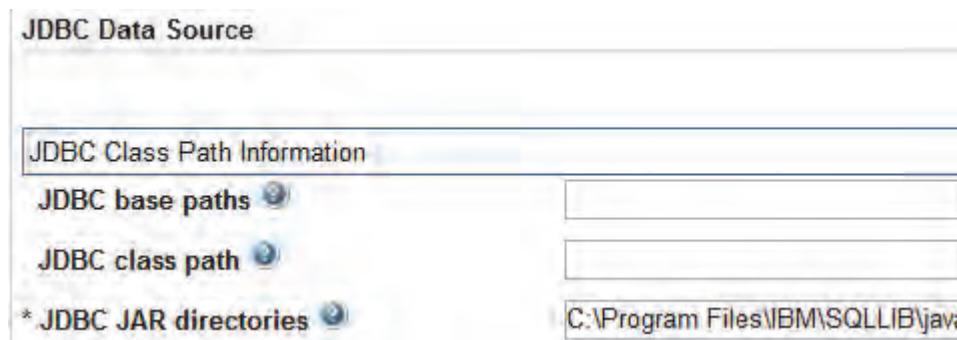
- g. Browse to and change the Java home path to the following directory:

C:\Program Files (x86)\ibm\Java70



- h. Click **Next**.

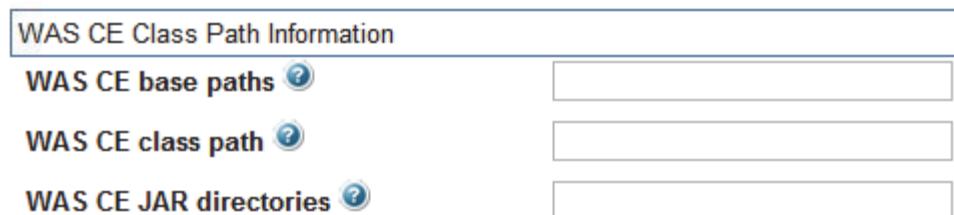
- i. Enter **C:\Program Files\IBM\SQLLIB\java** into the JDBC JAR directories field.



- j. Click **Next**.



- k. Keep the **JDBC Input Type** at **Specify basic JDBC properties** and click **Next**.



Here you configure the location of extra software for connecting to the WebSphere Community Edition application server the agent monitors. None are needed.

- I. Do not set any parameters and click **Next**.

You are prompted to manage AB_Processor subnodes.

AB_Processor	
CIM Local or Remote	<input type="button" value="New..."/>
CIM port number	<input type="text" value="Local"/>
CIM user ID	<input type="text"/>
CIM over SSL	<input type="button" value="Yes"/>
CIM password	<input type="text"/>
Confirm CIM password	<input type="text"/>
CIM SSL port number	<input type="text"/>
CIM host name	<input type="text"/>

This agent does monitor processors through the CIM data source, so you do not need to configure any AB_Processor subnodes.

- m. Click **Next**.

You are prompted to manage AB_WebSphere subnodes.

WebSphere Data	
WAS CE password	<input type="button" value="New..."/>
Confirm WAS CE password	<input type="text"/>
* WAS CE service URL	<input type="text" value="service:jmx:rmi:///jndi/rmi://localhost"/>
WAS CE user ID	<input type="text" value="system"/>

Here you create a subnode to monitor your JMX (WebSphere) data source. Subnodes can contain optional monitoring that you can enable or not, depending on your needs.

- n. Click **New** to create an AB_WebSphere subnode.

The screenshot shows a configuration dialog for creating a subnode. At the top left is a 'Delete' button. Below it, the subnode name is set to '* AB_WebSphere'. The 'WAS CE service URL' field contains 'service:jmx:rmi://jndi/rmi://localhost:099/JMXCo'. The 'WAS CE user ID' field is set to 'system'. The 'WAS CE password' and 'Confirm WAS CE password' fields both contain masked text (*****). The entire dialog is enclosed in a light gray border.

A subnode definition is created with global properties that you set at the top of the page.

- o. In **WAS CE service URL**, replace **localhost** with **WIN2**.
p. Enter the following values:
 - ◆ AB_WebSphere: **WIN2**
 - ◆ WAS CE Password: **manager**
 - ◆ Confirm WAS CE Password: **manager**

This screenshot shows the same configuration dialog as above, but with specific fields highlighted by red boxes. The 'WAS CE service URL' field has 'WIN2' highlighted. The 'WAS CE password' and 'Confirm WAS CE password' fields both have masked text (*****), with the second one also having a red box around its input field. The rest of the dialog remains the same.

- q. Click **Next**.

You are prompted to manage AB_Code subnodes.

AB_Code	
New...	
JDBC port number ?	
JDBC vendor specific connection URL ?	
* JDBC password ?	
* Confirm JDBC password	
* JDBC user name ?	
JDBC database name ?	
* JDBC database type ?	IBM DB2 Database
JDBC server name ?	

This agent monitors code data that is found in the Sample database on WIN1.

- r. Enter the following global properties:
 - ◆ JDBC port number: **50000**
 - ◆ JDBC password: **object00**
 - ◆ Confirm JDBC password: **object00**
 - ◆ JDBC user name: **db2admin**

AB_Code	
New...	
JDBC port number ?	50000
JDBC vendor specific connection URL ?	
* JDBC password ?	*****
* Confirm JDBC password	*****
* JDBC user name ?	db2admin
JDBC database name ?	
* JDBC database type ?	IBM DB2 Database
JDBC server name ?	

- s. Click **New** to create an AB_Code subnode.

The screenshot shows the configuration window for creating a new subnode. The 'Delete' button is at the top left. The form fields are as follows:

* AB_Code	<input type="text"/>
JDBC port number ?	50,000
JDBC vendor specific connection URL ?	<input type="text"/>
* JDBC password ?	<input type="password"/> (displayed as dots)
* Confirm JDBC password	<input type="password"/> (displayed as dots)
* JDBC user name ?	db2admin
JDBC database name ?	<input type="text"/>
* JDBC database type ?	IBM DB2 Database
JDBC server name ?	<input type="text"/>

- t. Enter the following CIM server connection properties.

- ◆ AB_Code: **WIN1**
- ◆ JDBC database name: **Sample**
- ◆ JDBC server name: **WIN1**

The screenshot shows the configuration window with the following values entered:

* AB_Code	WIN1
JDBC port number ?	50000
JDBC vendor specific connection URL ?	<input type="text"/>
* JDBC password ?	<input type="password"/> (displayed as dots)
* Confirm JDBC password	<input type="password"/> (displayed as dots)
* JDBC user name ?	db2admin
JDBC database name ?	Sample
* JDBC database type ?	IBM DB2 Database
JDBC server name ?	WIN1

- u. Click **OK** to close the configuration window and save your changes.

The MTMS utility main window is displayed.

- v. Click **Yes** to restart your agent.

8. If you are not prompted to restart your agent, start it now.

Reinstall and configure the AB2 agent on LIN4

In this section, do the following actions:

- Reinstall the AB2 agent on LIN4
 - Create an AB_Processor subnode to monitor processors on WIN2 with the CIM data source
 - Do not create an AB_WebSphere subnode to enable JMX monitoring
 - Do not create an AB_Code subnode to enable JDBC monitoring
9. Install the agent on LIN4.

If the K01 directory is not visible in /mnt/share, run the mount command:

```
mount /mnt/share
```

10. Start configuring the AB2 agent.

- a. Open a terminal window and run the following command:

```
/opt/IBM/ITM/bin/itmcmd config -A 01
```

You are first prompted if you want to configure the agent.

- b. Press Enter to select the default, **Yes**.

```
Lin4:/mnt/share/K01 # /opt/IBM/ITM/bin/itmcmd config -A 01
Agent configuration started...
Edit "Monitoring Agent for AB2" settings? [ 1=Yes, 2=No ] (default is: 1):
```

11. Accept the previously configured script data source properties.

- a. Press Enter to monitor remotely.

You are prompted for the host name or IP address of the system where the monitoring script (**script1.sh**) is run.

- b. Press Enter to accept **lin4** as the target host for the script data source.

```
SSH Remote Connection :
Remote host connection properties

Host name or IP address of remote system
Network address (default is: lin4): █
```

- c. Press Enter to accept port **22**.

```
The port number used for SSH communication.
SSH port number (default is: 22): █
```

- d. Press Enter to set a password.

```
The type of authentication to use to make the connection
Authentication Type [ 1=Password, 2=Public Key ] (default is: ): 1
```

- e. Press Enter to select **No** to disconnecting after each collection interval.

Disconnect from the remote system after each data collection interval. By default, the SSH communication socket will establish an authenticated session at the first connection and will leave the connection active until the agent is stopped. Disconnect from the remote system after each collection interval [1=Yes, 2=No] (default is: 2):

- f. Press Enter to select **No** to deleting the script after each collection interval.

Delete the script(s) from the remote system after each data collection interval. By default, the script(s) will be uploaded to the remote system at the first connection and will only be refreshed if the local copy changes. The script(s) will be removed from the remote system immediately before the agent is stopped. Remove script(s) from the remote system after each collection interval [1=Yes, 2=No] (default is: 2):

- g. Press Enter to accept **root** at the user name.

Password :

The credentials used to make the connection

Username for the remote system

Username (default is: root): █

- h. Press Enter twice to accept the current password.

Password used for the remote system.

Enter Password (default is: *****):

Re-type : Password (default is: *****):

12. Press Enter to accept the previously configured socket of **0**.

Socket :

Socket Data Source

The port that the agent will use to listen on for data from socket clients. A value of 0 indicates an ephemeral port will be used.

Port Number (default is: 0):

13. Set global Java properties for a possible JDBC data source. This agent does not use the JDBC data source.

- a. Press Enter to accept the default Java home directory.

```
Java :  
Java parameters  
  
The path to where java is installed  
Java home (default is: /opt/ibm/java): █
```

- b. Press Enter to accept the default Java trace level.
c. Press Enter to not enter JVM arguments.

This parameter allows you to specify the trace level used by the Java providers
Java trace level [1=Off, 2=Error, 3=Warning, 4=Information, 5=Minimum Debug, 6=Med
ug, 7=Maximum Debug, 8=All] (default is: 2):

This parameter allows you to specify an optional list of arguments to the java virt
hine
JVM arguments (default is:):

The next several fields are used to set alternative paths to files used by the JDBC data
source.

- d. Press Enter to select the default JDBC base path.
e. Press Enter to select the default JDBC class path.
f. Enter a forward slash (/) and press Enter to set the JDBC JAR directories.

- g. Press Enter to select the default JDBC input type.

```
JDBC Server :  
JDBC Data Source  
-----  
JDBC Class Path Information  
Directories used to locate any Java class path entry or JAR file directory which  
is not fully-qualified  
JDBC base paths (default is: ):  
  
JAR files which Java will look in to find a needed class or resource  
JDBC class path (default is: ):  
  
Directories which contain JAR files which Java will look in to find a needed class  
or resource. All JAR files in each of these directories will be looked in.  
JDBC JAR directories (default is: ): /  
  
JDBC Input Type :  
Select the type of configuration necessary to connect to the server.  
  
The type of input properties you want to use to connect to the database  
JDBC input type [ 1=Specify basic JDBC properties, 2=Specify the JDBC URL ] (default is: 1):  
  
The next several fields are used to locate extra software that is needed to make the JMX  
connection to WebSphere Community Edition. None is needed in this instance.  
  
14. Set global WebSphere properties for a possible JMX data source. This agent does not use the  
JMX data source.  
a. Press Enter to select the default WAS CE base path.  
b. Press Enter to select the default WAS CE class path.
```

- c. Press Enter to select the default WAS CE JAR directories.

WebSphere Application Server Community Edition :

WAS CE Class Path Information

Directories used to locate any Java class path entry or JAR file directory which is not fully-qualified

WAS CE base paths (default is:):

JAR files which Java will look in to find a needed class or resource

WAS CE class path (default is:):

Directories which contain JAR files which Java will look in to find a needed class or resource. All JAR files in each of these directories will be looked in.

WAS CE JAR directories (default is:):

You are prompted to manage AB_Processor subnodes.

15. Configure subnodes.

Because this agent monitors processor data through the CIM data source, create an AB_Processor subnode.

- a. Type **1** and press Enter to create an AB_Processor subnode.

AB_Processor :

No 'AB_Processor' settings available.

Edit 'AB_Processor' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit] (default is : 5): 1

b. Enter the following values:

- ◆ AB_Processor: **WIN2**
- ◆ CIM Local or Remote: **2 (Remote)**
- ◆ CIM port number: **5988**
- ◆ CIM user ID: **Administrator**
- ◆ CIM of SSL: **2 (No)**
- ◆ CIM password: **object00**
- ◆ CIM SSL port: **5989**
- ◆ CIM host name: **WIN2**

```
AB_Processor (default is: ): WIN2
Local or remote authentication to the CIM server
CIM Local or Remote [ 1=Local, 2=Remote ] (default is: ): 2

The port number used for communication that is not secure
CIM port number (default is: ): 5988

The user ID used to access the CIM server
CIM user ID (default is: ): Administrator

Use SSL for communication with the CIM server
CIM over SSL [ 1=Yes, 2=No ] (default is: ): 2

The password to access the CIM server
Enter CIM password (default is: ):

Re-type : CIM password (default is: )

The port number used for secure communication
CIM SSL port number (default is: ): 5989

The host name to be accessed for CIM data
CIM host name (default is: ): WIN2
```

You are prompted again to manage AB_Processor subnodes.

c. Type **5** and press Enter to exit managing AB_Processor subnodes.

```
AB_Processor settings: AB_Processor=WIN2
Edit 'AB_Processor' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit] (default is
: 5): 5
```

You are prompted to manage AB_WebSphere subnodes. You can create, edit, and delete subnodes. Because this agent does not monitor WebSphere, do not create any AB_WebSphere subnodes.

- a. Type **5** and press Enter to exit managing AB_WebSphere subnodes.

```
AB_WebSphere :  
  
No 'AB_WebSphere' settings available.  
Edit 'AB_WebSphere' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit]  
(default is: 5): 5
```

You are prompted to manage AB_Code subnodes. This agent does not monitor code data in a database. Do not create any AB_Processor subnodes.

- b. Type **5** and press Enter to exit managing AB_Code subnodes.

```
AB_Code :  
  
No 'AB_Code' settings available.  
Edit 'AB_Code' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit] (default is: 5): 5
```

You are asked if the agent connects to a TEMS.

- c. Enter **1** and press Enter to answer Yes.

```
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1): 1
```

d. Press Enter to accept each of the following default values:

- ◆ ip.pipe as the Network Protocol
- ◆ Not to install a Network Protocol 2
- ◆ ITM is the TEMS Host Name
- ◆ Port of 1918
- ◆ Null as the KDC_PARTITION name
- ◆ No to configuring a connection to a secondary TEMS
- ◆ 0 for an Optional Primary Network Name
- ◆ Disable HTTP

```
Network Protocol [ip, sna, ip.pipe, ip.spipe, ip6, ip6.pipe or ip6.spipe] (Default is: ip.pipe):  
Now choose the next protocol from one of these:  
- ip  
- sna  
- ip.spipe  
- ip6  
- ip6.pipe  
- ip6.spipe  
- 0 for none  
Network Protocol 2 (Default is: 0):  
TEMS Host Name for IPv4 (Default is: itm):  
IP.PIPE Port Number (Default is: 1918):  
Enter name of KDC_PARTITION (Default is: null):  
Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):  
Enter Optional Primary Network Name or 0 for "none" (Default is: 0):  
Disable HTTP? [1=YES, 2=NO] (Default is: 2):  
Agent configuration completed...
```

You receive a prompt that indicates the agent configuration is complete and prompted to restart the agent.

16. If prompted, press Enter to restart the agent.

```
Would you like to restart the component to allow new configuration to take effect? [1=Yes, 2=No] (Default is: 1):  
Stopping Monitoring Agent for AB2  
Monitoring Agent for AB2 stopped  
Starting Monitoring Agent for AB2  
Monitoring Agent for AB2 started  
lin4:/opt/IBM/ITM/bin #
```

17. If you are not prompted to restart the agent, start it now.

18. Start the MTMS utility and confirm the agent status by running the following command:

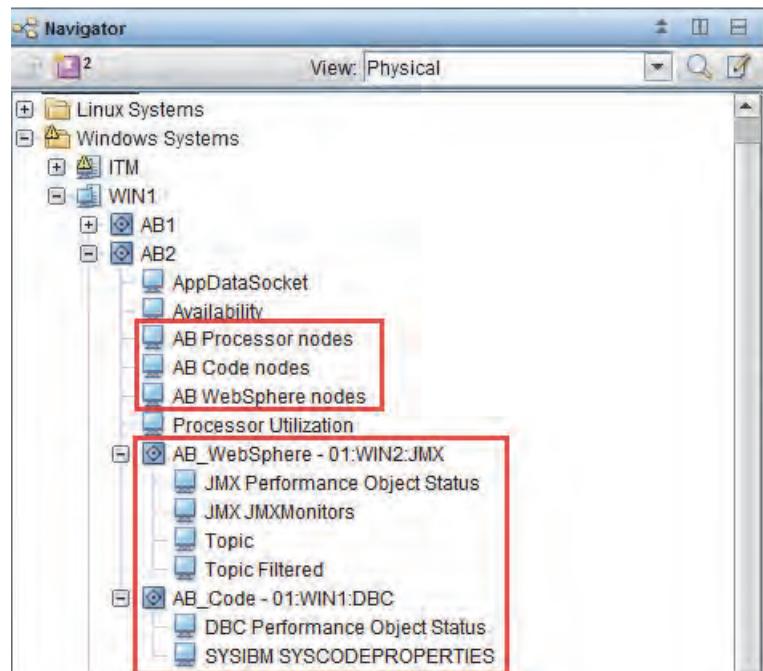
```
/opt/IBM/ITM/bin/itmcmd manage &
```

You successfully installed the agent on LIN4.

Confirm the updated AB2 agents in the Tivoli Enterprise Portal

19. On ITM, start the **TEP** client and expand **AB2** under **WIN1**.

20. Confirm your new navigator nodes.



Note: If your subnode names do not display correctly with the host names included, you might have to restart the TEP client or Tivoli Enterprise Portal Server.

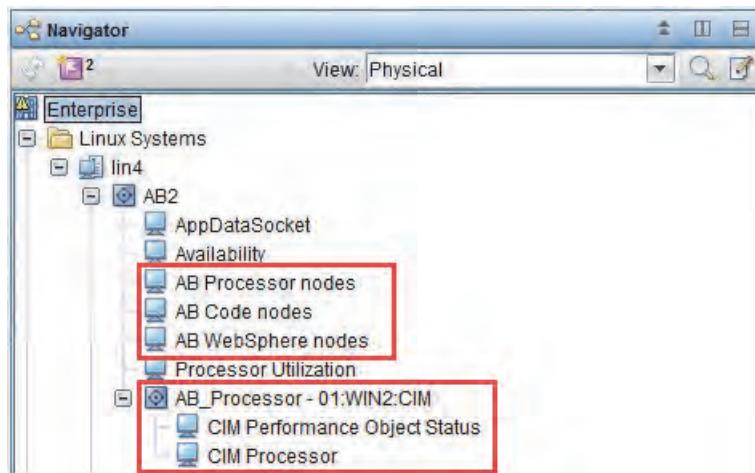
21. Investigate each subnode and new Navigator node.

Notice the following items:

- The three nodes report on the defined subnodes of each type and **AB Processor nodes** is empty because you did not define one on WIN1.
- Topic and SYSIBM SYSCODEPROPERTIES return the expected monitoring data.
- JMX JMXMonitors has no data because you did not enable any JMX monitors through the JMX Take Actions.
- Each subnode had a Performance Object Status node that reports on the status of that data collector.
- No subnode for CIM processor data were created.

22. Expand **AB2** under **LIN4**.

23. Confirm your new navigator nodes.



Note: If your subnode names do not display correctly with the host names included, you might have to restart the TEP client or Tivoli Enterprise Portal Server.

24. Investigate each subnode and new Navigator node and confirm that they perform as expected.



IBM Training



© Copyright IBM Corporation 2017. All Rights Reserved.

Course Exercises

Agent Builder 6.3.4 Labs for an IBM Application Performance Management Environment

Course code TV384 ERC 1.0

Click for [Agent Builder 6.3.4 Labs for an IBM Tivoli Monitoring Environment](#)



June 2017 edition

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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.

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.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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.

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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2017.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

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

Contents

Unit 1 Introduction to IBM Agent Builder exercises.....	1-1
This unit has no student exercises.	
Unit 2 Agent creation basics exercises.....	2-1
Exercise 1 Create an agent to monitor specific Windows services	2-1
Create the agent	2-1
Browse Windows services	2-6
Troubleshoot an agent definition problem	2-13
Troubleshoot Agent Builder	2-15
Unit 3 Customizing agents for IBM Application Performance Management exercises	3-1
Exercise 1 Modify and install the AB1 agent into an IBM Performance Management environment	3-1
Confirm the Performance Management server	3-1
Create data points for the summary dashboard	3-3
Create dashboards and resource definitions	3-8
Configure the agent for quick identification within the Performance Management Console	3-13
Install the AB1 agent and create the installation scripts	3-14
Unit 4 Monitoring Windows resources exercises.....	4-1
Exercise 1 Monitor Windows resources	4-1
Prepare to edit your agent	4-1
Add WMI process information to the agent	4-2
Add combined disk information to the agent	4-10
Add DB2 Windows log event information to the agent	4-18
Test the full agent in Agent Builder	4-22
Change the agent version	4-30
Exercise 2 Install and confirm the updated AB1 agent in an IBM Performance Management environment	4-31
Update the dashboard and resource definitions	4-31
Create the installation files	4-32
Reinstall the agent on WIN1 with the installation scripts	4-36
Confirm the updated AB1 agent in the Performance Management Console	4-37
Unit 5 Monitoring processes and command return codes exercises.....	5-1
Exercise 1 Monitor processes and command return codes	5-1
Create an agent to monitor a process on Windows and Linux	5-2
Create the HTTP server monitor for Windows	5-3
Add the Linux HTTP server process	5-6
Create an agent to monitor a functionality test	5-9
Exercise 2 Install and confirm the AB2 agent in an IBM Performance Management environment	5-19

Create data points for the summary dashboard	5-19
Create dashboards and resource definitions	5-22
Configure the agent for quick identification within the Performance Management Console	5-26
Create the installation files	5-26
Install the agent on WIN1	5-27
Install the agent on LIN4	5-28
Confirm the agents in the Performance Management Console	5-29
Unit 6 Monitoring custom data sources exercises.....	6-1
Exercise 1 Create an agent to monitor local and remote systems with scripts and socket connections	6-1
Confirm the script	6-2
Modify AB2 to monitor the script	6-3
Add derived attributes	6-13
Enable the agent to monitor local and remote sockets	6-17
Change the agent version	6-25
Exercise 2 Install and confirm the updated AB2 agent in an IBM Performance Management environment	6-26
Add the script and socket data sources to the AB2 detailed dashboard	6-26
Create the updated AB2 agent installation scripts	6-27
Reinstall the AB2 agent on WIN1	6-28
Reinstall the agent on LIN4	6-30
Confirm the agent changes in the Performance Management Console	6-32
Exercise 3 Create an agent to monitor a log file and a custom Java data source	6-36
Determine the log file data for the agent to gather	6-36
Modify the AB1 agent to monitor the ping.txt log file	6-38
Add derived attributes	6-52
Add a Java API data source	6-58
Change the agent version	6-69
Exercise 4 Install and confirm the updated AB1 agent in an IBM Performance Management environment	6-70
Add the log file and Java API data sources to the AB1 agent detailed dashboard	6-70
Create dashboards and resource definitions	6-70
Reinstall the AB1 agent on WIN1	6-71
Confirm the AB1 modifications in the Performance Management Console	6-72
Unit 7 Monitoring remote and optional resources exercises	7-1
Exercise 1 Remotely monitor many resources	7-1
Create an agent to monitor multiple hosts from one agent	7-1
Add a data source to monitor HTTP pages	7-10
Add a data source to monitor network availability	7-13
Change the agent version	7-16
Exercise 2 Install and confirm the updated AB1 agent in an IBM Performance Management environment	7-18
Create dashboards and resource definitions	7-18
Reinstall the AB1 agent on WIN1	7-23
Confirm the AB1 modifications in the Performance Management Console	7-28
Exercise 3 Allow data sources to be optional	7-34

Add monitoring of Common Information Management data sources	7-34
Add monitoring of remote WebSphere Application Server Java Management Extensions	7-43
Add a JDBC data source to the agent	7-54
Change the agent version	7-63
Exercise 4 Install and confirm the updated AB2 agent in an IBM Performance Management environment	7-64
Create Summary Dashboard data points for the AB_WebSphere subnode	7-64
Create dashboards and resource definitions	7-68
Reinstall and configure the agent on WIN1	7-77
Reinstall and configure the AB2 agent on LIN4	7-83
Confirm the agents in the Performance Management Console	7-89

Unit 1 Introduction to IBM Agent Builder exercises

This unit has no student exercises.

Unit 2 Agent creation basics exercises

In this unit, you create an agent that is named AB1. It monitors the HTTP server and DB2® services of any target Windows host.

Exercise 1 Create an agent to monitor specific Windows services

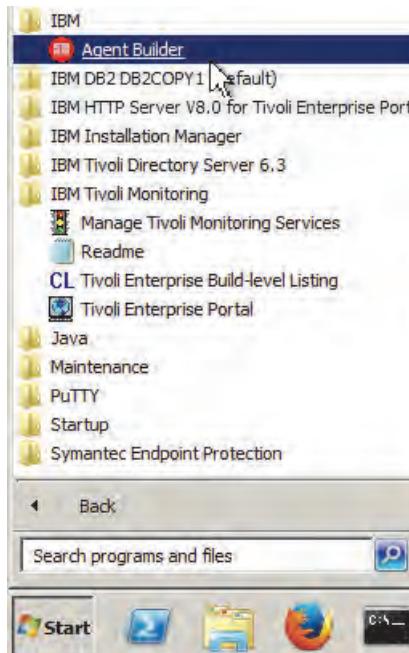
In this exercise, you create a monitoring agent that gathers availability information for Windows DB2 and HTTP services. You must complete this exercise from the WIN1 server.

Create the agent

1. Log in to WIN1 as **Administrator** with password **object00**.
2. Start Agent Builder by using either of the following methods:
 - Double-click the **Agent Builder** icon on the desktop.



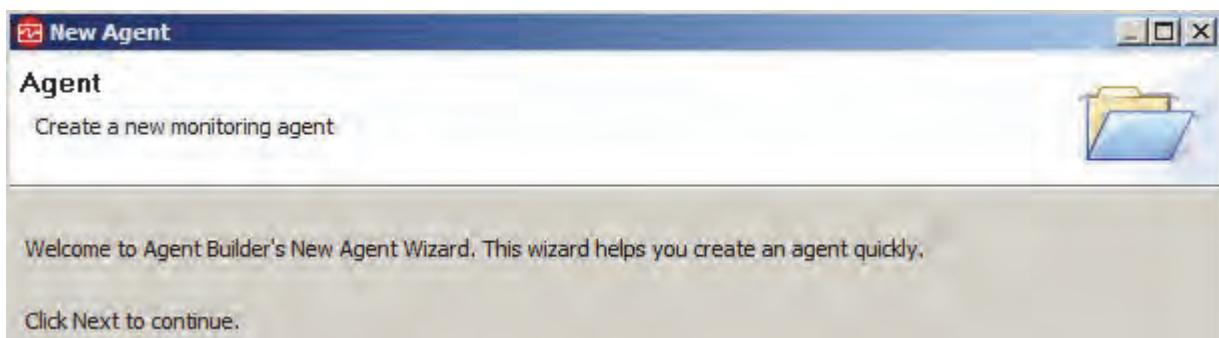
- Select **Start > All Programs > IBM > Agent Builder.**



3. Maximize the Agent Builder application window.
4. Select **File > New > Agent.**

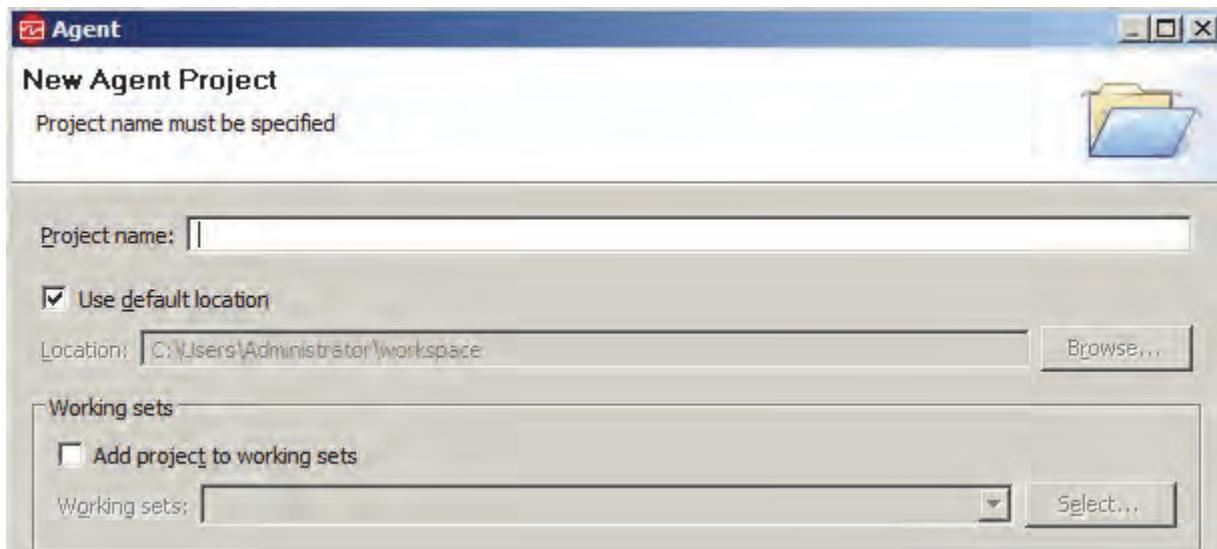


The New Agent wizard opens.

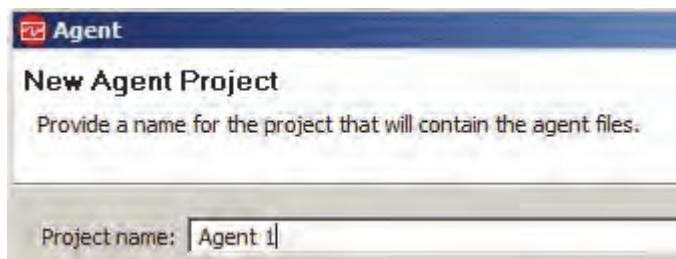


5. Click **Next** to continue with the wizard.

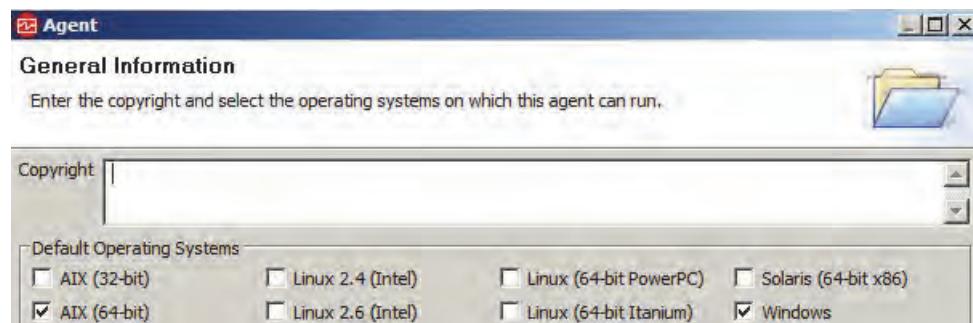
The New Agent Project window opens.



6. Enter **Agent 1** in the **Project name** field and click **Next**.

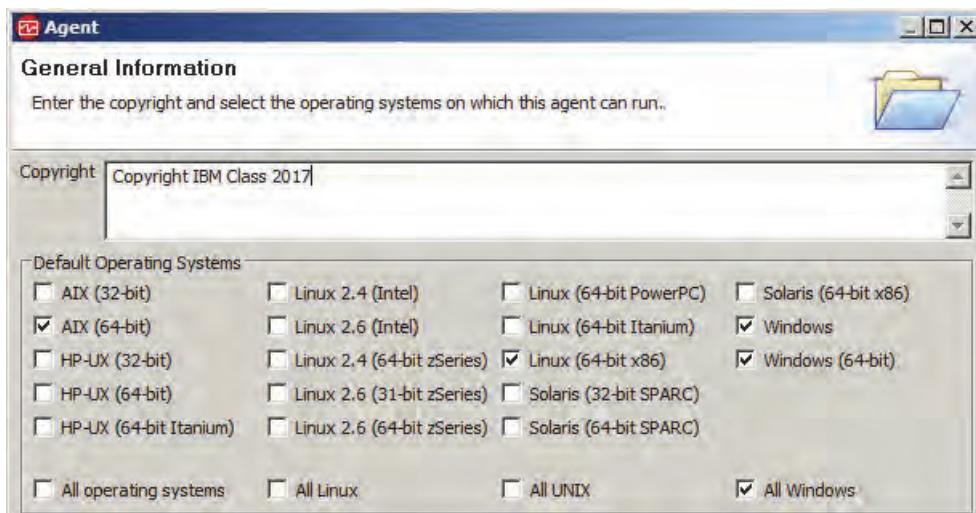


The General Information window opens.



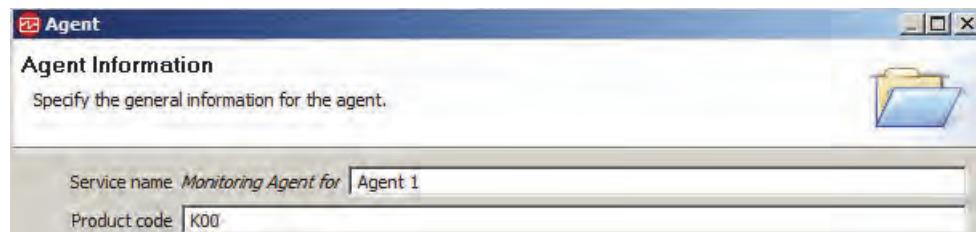
From this window, you can set several general property values for this agent. Enter a copyright statement and identify the specific operating systems on which this agent can be installed. Select the default operating systems on which this agent can be installed.

7. Complete the General Information window.
 - a. Enter **Copyright IBM Class 2017** into the **Copyright** field.
 - b. Keep the default operating systems.



8. Click **Next**.

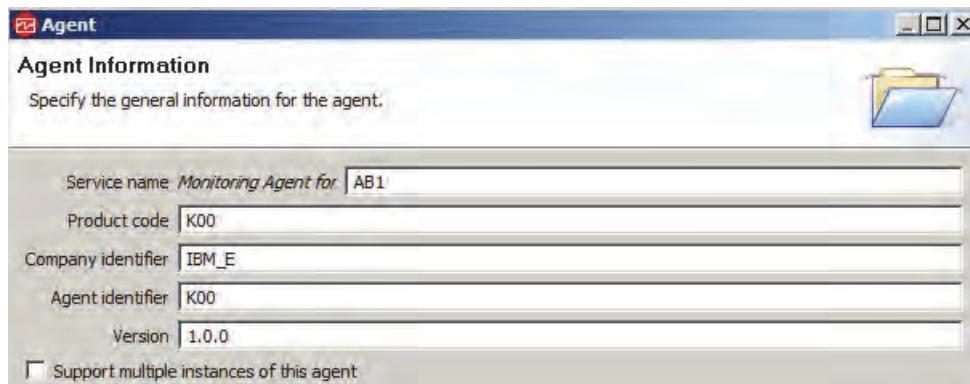
The Agent Information window opens.



Notice that the default **Service name** is the same as the **Project name**, but you can change it. The service name is the name of the service that is created on Windows. It is also the name that is displayed in the IBM Performance Management and Manage Tivoli Monitoring Services utilities for managing installed agents.

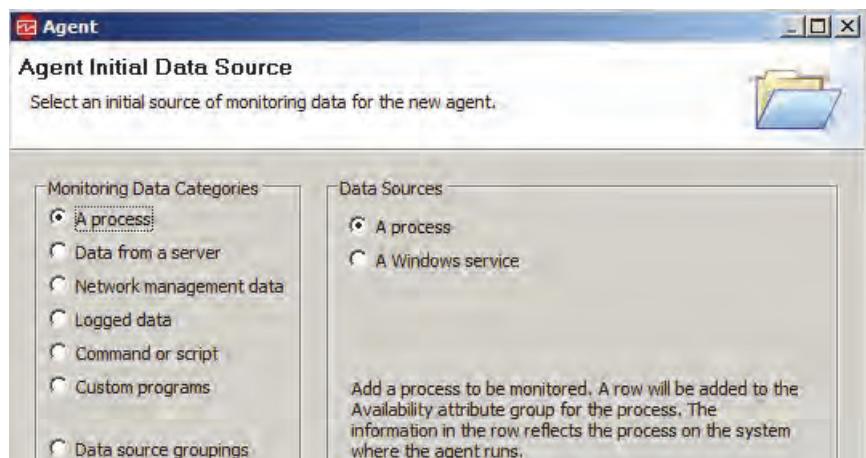
9. Change the Service name to **AB1**.
10. Change the Company identifier to **IBM_E**.

The Agent Information window looks like this screen capture.



11. Click Next.

The Agent Initial Data Source window opens.

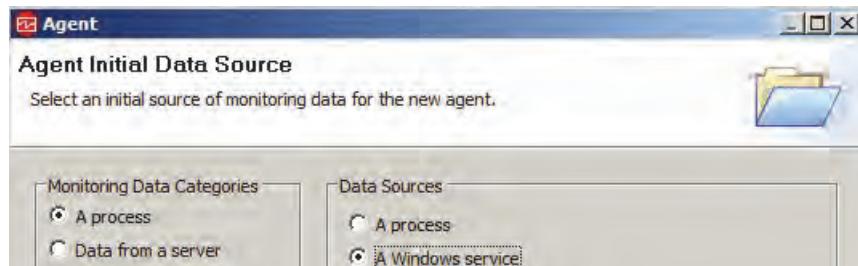


12. Click the different options under **Monitoring Data Categories and browse the different types of data and data groupings you can monitor.**

How each data source is configured can differ greatly, but some data sources work in similar ways. You work with each data source and see their similarities and dissimilarities in later exercises.

For this exercise, you want your agent to monitor several Windows services.

13. Select **A process from **Monitoring Data Categories** and **A Windows service** from **Data Sources**.**



14. Click Next.

The Service Monitor window opens.



From this window, you can enter the display name and the service name of the service you want to monitor. You can also browse for a list of services and select one or more services from the list.



Note: If you plan to browse, do not enter a display name because it is overwritten.

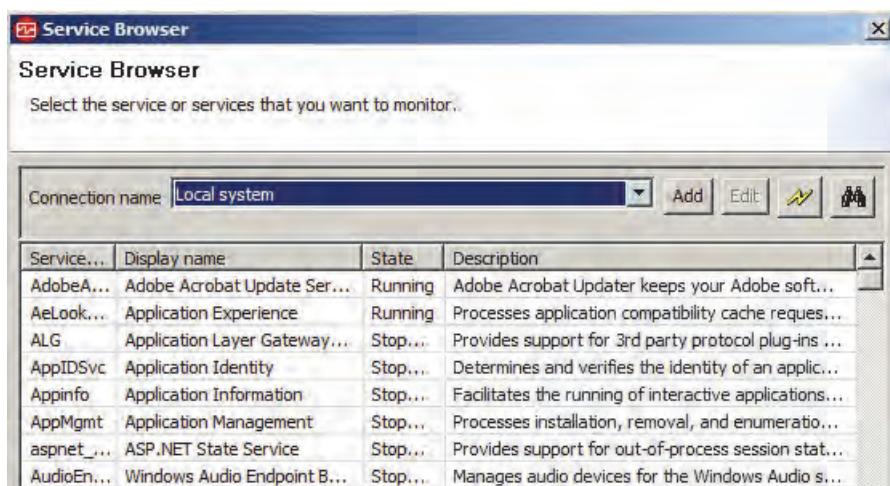
Browse Windows services

You select a single service first.

15. Configure the agent to monitor the DB2 service.

- In the Agent wizard, click **Browse**.

The Service Browser opens.



From this browser, you see all of the services that are installed on this system.

- b. Locate and click the **DB2** service and click **OK**.

Service Name	Display name	State	Description
CryptSvc	Cryptographic Services	Running	Provides four management services: Catalog
DB2	DB2 - DB2COPY1 - DB2	Running	Allows applications to create, update, control
DB2DAS00	DB2DAS - DB2DAS00	Running	Supports local and remote database administrati
DB2GOVERNO...	DB2 Governor (DB2COPY1)	Stop...	Collects statistics for applications connected t

The Service Monitor window is displayed.



Both the display name and the service name are entered. You can also connect to and list the services that are installed on remote hosts.

16. Browse processes on a remote system.
- Click **Browse** again to return to the **Service Browser**.
 - Click **Add**.

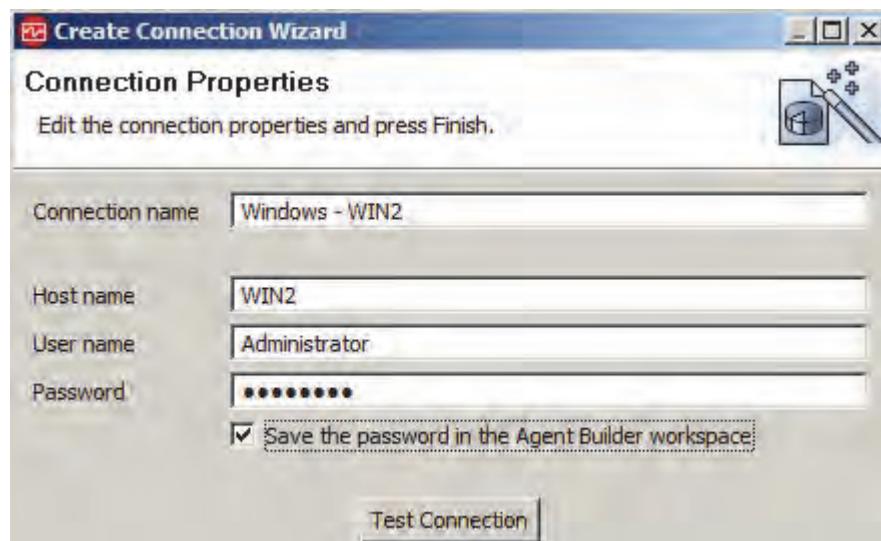


- Select **Windows systems** and click **Next**.



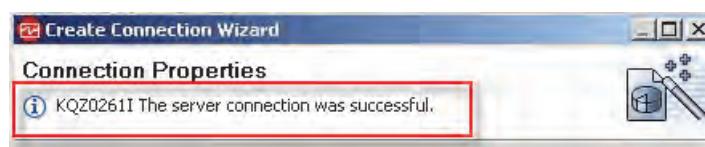
d. Enter the following values:

- ◆ Connection name: **Windows - WIN2**
- ◆ Host name: **WIN2**
- ◆ User name: **Administrator**
- ◆ Password: **object00**
- ◆ Save the password: **Checked**



e. Click **Test Connection**.

f. Confirm that the connection was successful.

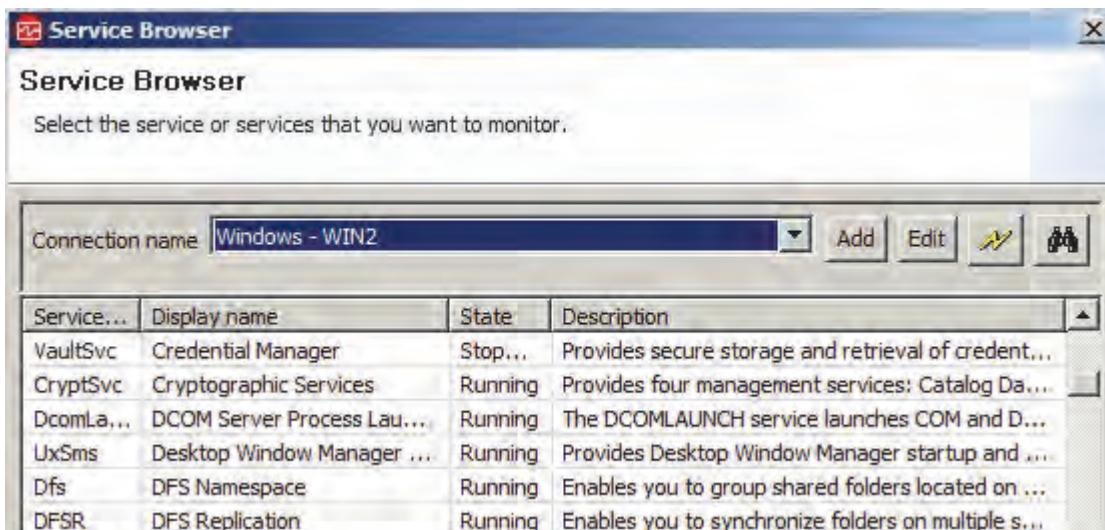


g. Click **Finish**.

You are returned to the Service Browser and the services currently running on WIN2 are displayed.

h. Click the **Display Name** column header to sort the list of services by their display name.

- If you scroll down the list, you find that DB2 is not installed on WIN2.



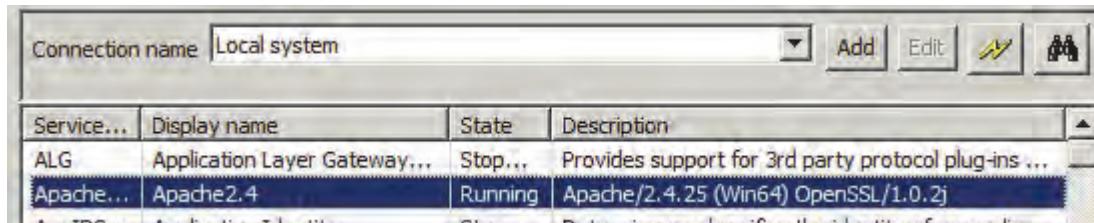
Note: The purpose of browsing a remote data source is not to identify a target host to monitor. Instead, the purpose is to identify data that you want to monitor that is not found on the local host. You don't browse WIN2 to identify WIN2 as a target to monitor, but to select the services that run on it. In this instance, browsing to WIN2 was for demonstration purposes only, and the services you want to monitor are not on it. You must select these services on WIN1.

17. Select the HTTP and DB2 services to monitor:

- Select **local system** (WIN1) from the **Connection name** menu to return to services that run on the local host.
- Select all services that start with DB2 by clicking the first DB2 service, holding down the Shift key, and clicking the last DB2 service.



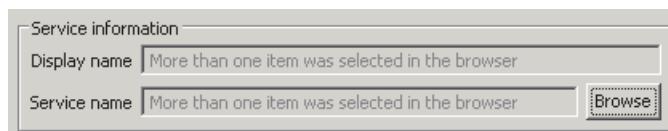
- c. Scroll up and select the **Apache2.4** service by holding down the CTRL key and clicking the service.



- d. Scroll down and confirm all the DB2 services are still selected.

- e. Click **OK**.

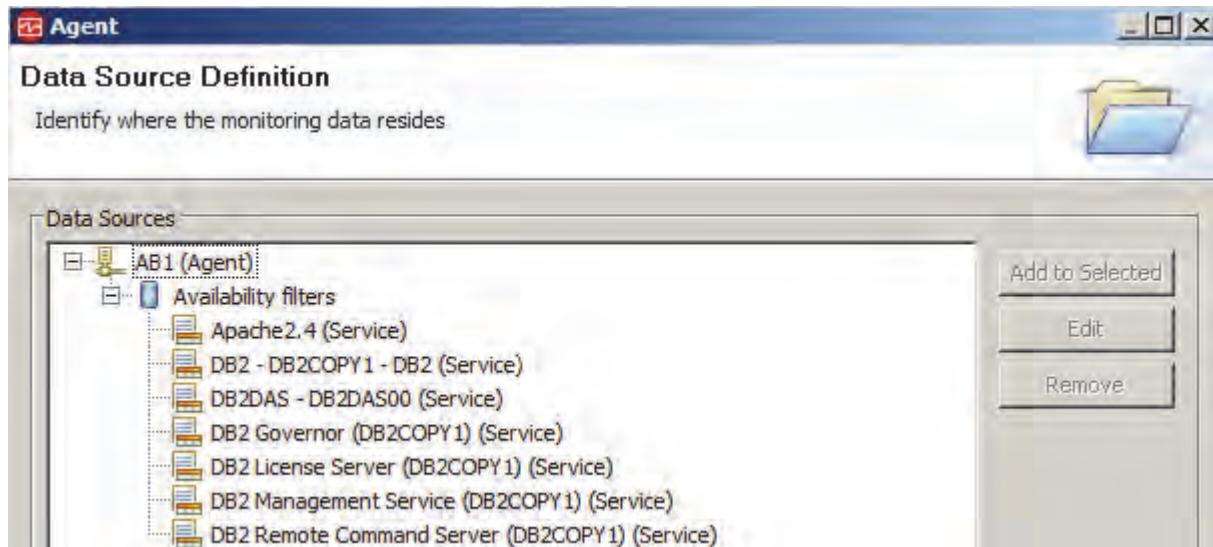
The Service Monitor window is displayed. The **Display name** and **Service name** fields contain a phrase that indicates that you selected more than one service to monitor.



18. Click **Next**.

The Data Source Definition window opens. Your Tivoli Enterprise Portal display name of your agent is displayed, and one attribute group named **Availability filters** is shown.

19. If it is not expanded, click the plus sign (+) beside **Availability filters** to expand the availability filters.



Each of the services that is monitored is listed here.

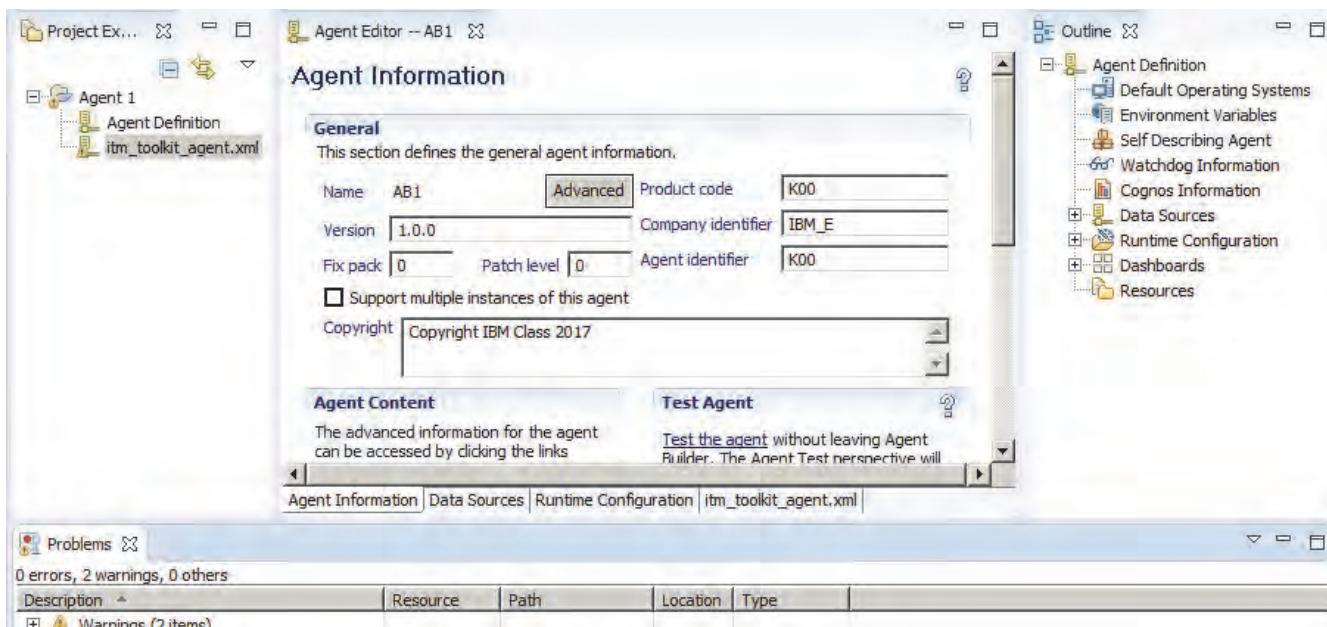
From this window, you can continue to modify the data sources this agent monitors. For example, you can perform the following tasks:

- Add data sources to monitor with this agent. (**Hint:** Highlight **AB1** to add any type of data source, or highlight **Availability filters** to limit the new data source to availability filters.)
- Remove any availability filter or the entire **Availability filters** attribute group.
- Edit an availability filter.

As defined by this scenario, the basic agent definition is complete.

20. Click **Finish**.

You exit the Agent wizard, and the Agent Builder workspace is displayed.

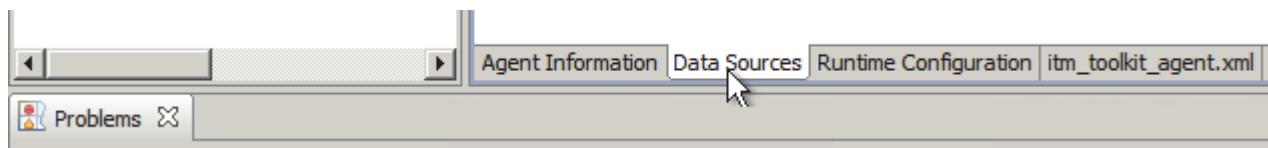


Using the Project Explorer on the left, you can browse your agent projects and their components. The **itm_toolkit_agent.xml** component, which holds your basic agent definition, can be opened in the Agent Editor on the right.

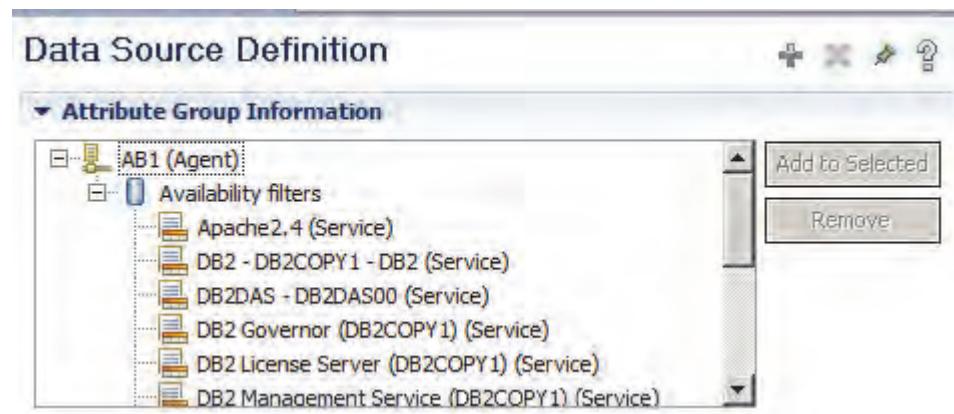
You can use the Agent Editor to modify your agent without restarting the Agent wizard.

21. Browse the other agent definition tabs.

- Click the **Data Sources** tab at the bottom of the Agent Editor window.

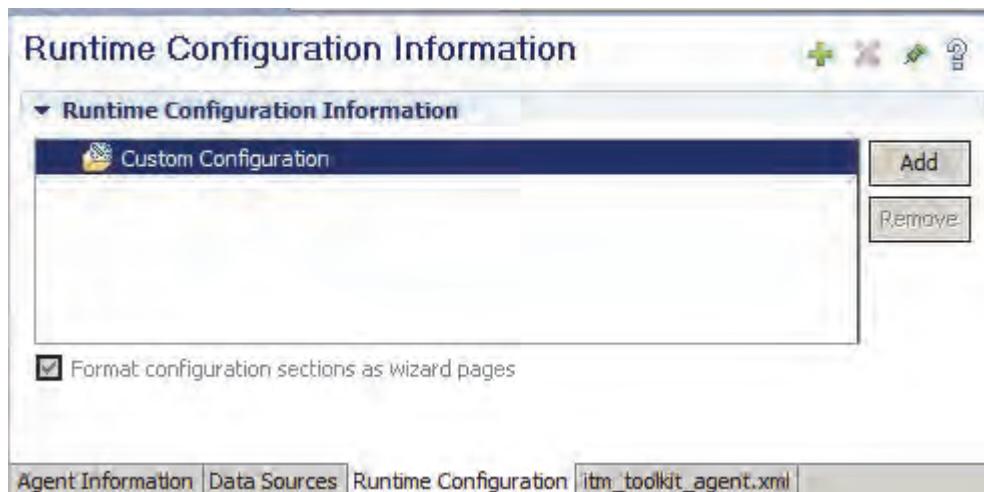


- b. Expand the **Availability filters** attribute group, and click one of the services.



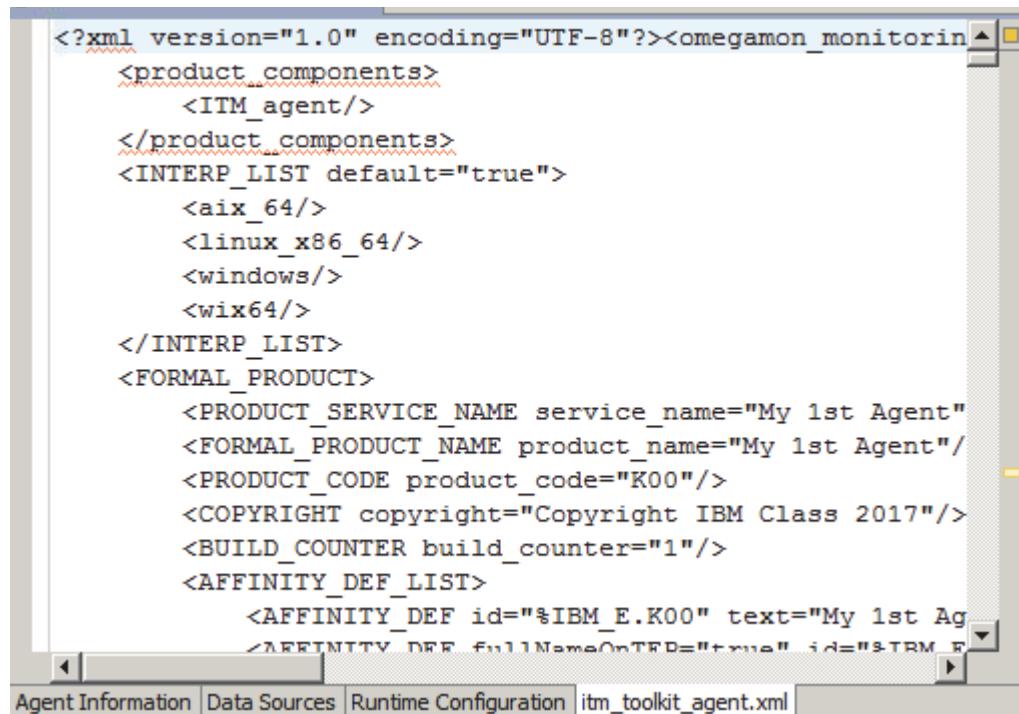
This tab contains all of the data source information that you entered with the Agent wizard. As with the wizard, you can use this tab to edit, remove, and add attributes and attribute groups. You can also edit the availability filter metadata, such as the display name.

- c. Click the **Runtime Configuration** tab at the bottom of the Agent Editor window.



From this tab, you can configure the agent to prompt for data when the agent is configured, such as a local host or directory name. For this exercise, you do not work with runtime configuration.

- d. Click the **itm_toolkit_agent.xml** tab at the bottom of the Agent Editor window.



```

<?xml version="1.0" encoding="UTF-8"?><omegamon_monitorin
<product_components>
    <ITM_agent/>
</product_components>
<INTERP_LIST default="true">
    <aix_64/>
    <linux_x86_64/>
    <windows/>
    <wix64/>
</INTERP_LIST>
<FORMAL_PRODUCT>
    <PRODUCT_SERVICE_NAME service_name="My 1st Agent"
    <FORMAL_PRODUCT_NAME product_name="My 1st Agent"/>
    <PRODUCT_CODE product_code="K00"/>
    <COPYRIGHT copyright="Copyright IBM Class 2017"/>
    <BUILD_COUNTER build_counter="1"/>
    <AFFINITY_DEF_LIST>
        <AFFINITY_DEF id="%IBM_E.K00" text="My 1st Ag
    <AFFINITY_DEF id="STRM_E
</AFFINITY_DEF_LIST>
</FORMAL_PRODUCT>

```

The XML code defines the agent's configuration, including supported platforms (aix_64, linux_x86_64, windows, wix64), the product name (My 1st Agent), product code (K00), copyright information (Copyright IBM Class 2017), and a build counter (1). It also includes an affinity definition list with an entry for the company identifier (IBM_E.K00) and another entry for STRM_E.

The agent information that is displayed in the other tabs is written to this .xml file, which is stored in your Agent Builder workspace directory.



Note: Editing the data in the .xml file edits the data in the tabbed windows, but it is not a recommended or supported practice.

- e. Browse the .xml file and see whether you can identify any of the agent data you entered in this exercise.

Troubleshoot an agent definition problem

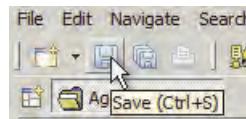
In this section, you create a problem within your agent definition and then use the **Problems** pane to troubleshoot the issue.

22. Click the **Agent Information** tab in the **Agent Editor**.

23. Change the **Company identifier** from **IBM_E** to **IBM_Education**.

Company identifier	IBM_Education
--------------------	---------------

24. Click the **Save** icon to save your changes.



Notice the error in the **Problems** pane at the bottom of the Agent Builder application.

Problems					
1 error, 2 warnings, 0 others					
Description	Resource	Path	Location	Type	
+ Errors (1 item)					
+ Warnings (2 items)					

Hint: If the Problems pane is not visible, drag the resize bar just above the **Problems** tab.

25. Click the plus (+) next to the error to expand the display and see the error.

Problems					
1 error, 2 warnings, 0 others					
Description	Resource	Path	Location	Type	
- Errors (1 item)					
KQZ0197E The combined length of company identifier and ag	itm_toolkit_...	/Agent 1	line 21	Agent Build...	

The window lists the error description, the resource where the problem occurs, and the location where the error exists.

26. Place your cursor on the **Description** field to display the full description.

Description	Resource	Path	Location	Type
- Errors (1 item)				
KQZ0197E The combined length of company identifier and agent identifier cannot exceed 11 characters.				Agent Build...
+ Warnings (2 items)				

Changing the Company identifier made the property too long. Double-clicking the error in the **Problems** tab takes you to an **Agent Editor** tab where you can locate and fix the error.



Important: Ignore the warnings in the Problems pane for now.

27. Click the **Data Sources** tab to take you away from the **Agent Information** tab.

28. Double-click the error in the **Problems** tab.

You are returned to the **Agent Information** tab where the problem can be fixed.

29. Change the **Company identifier** back to **IBM_E** and save your agent project.

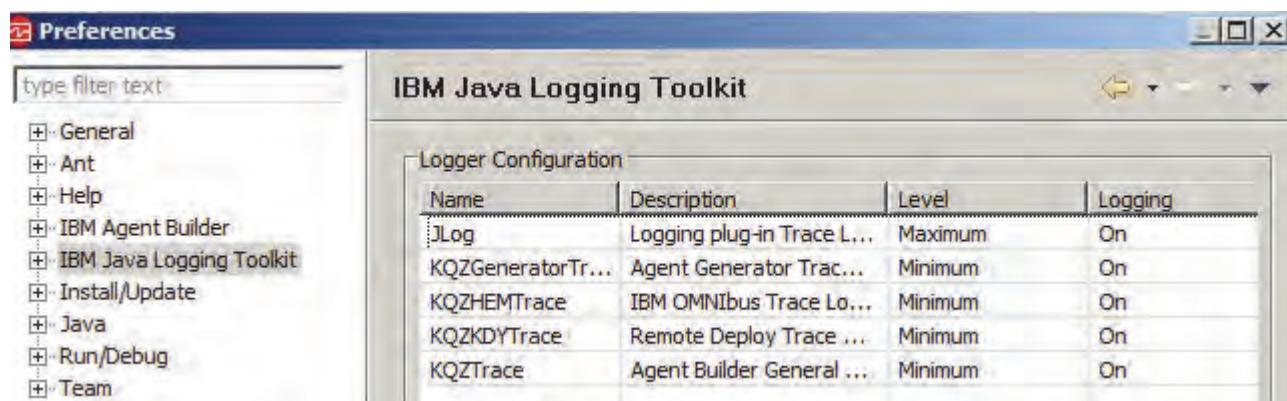


30. Notice that the error in the Problems pane closes.

Troubleshoot Agent Builder

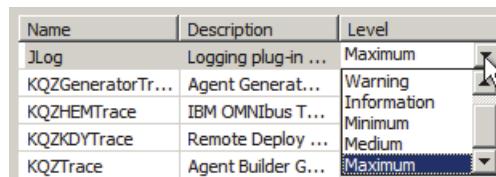
In this section, you see where you can configure and view logging for the Agent Builder application if you must troubleshoot the Agent Builder application.

31. View and configure logging in the Preferences window.
- In Agent Builder, select **Window > Preferences**.
 - Click **IBM Java Logging Toolkit**.



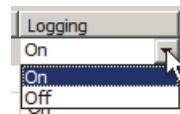
Notice the current logger configurations for Agent Builder.

- Click **Maximum** in the **Level** field for **JLog**.
 - Click the menu button to the right of Maximum.
- A list is displayed. You can use this list to change the log level.



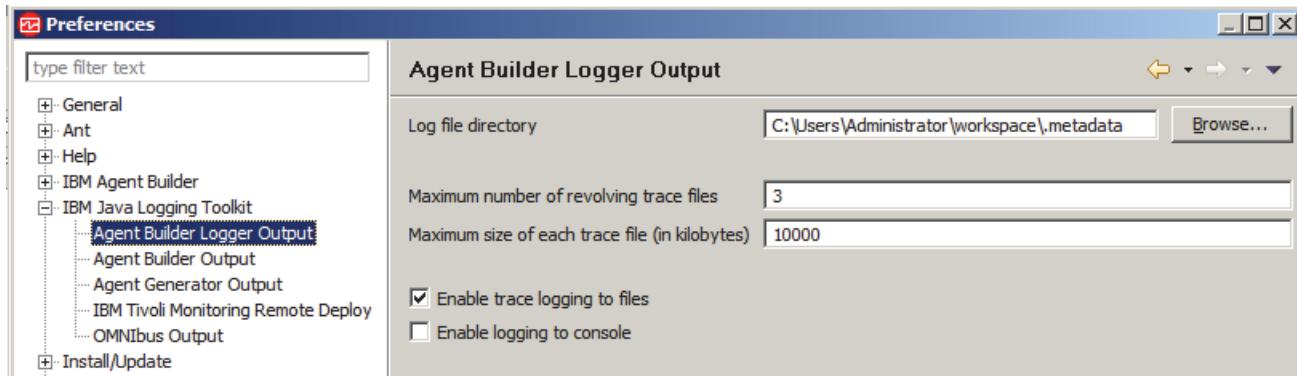
- Click in the **Logging** field for **JLog** and its menu button.

Another menu is displayed.



You can use this menu to turn logging on and off. To see detailed information for each log, you must expand the **IBM Java Logging Toolkit** entry on the left.

- f. Expand **IBM Java Logging Toolkit** and click each subcomponent's logging configuration.



You can now select the individual logs and adjust settings for each one. You can set the logging directory, the number of revolving trace files, and the maximum size of the trace file. You can also enable or disable logging to file or the console.

Notice the default location of each log file.

- g. Click **OK** to close the Preferences window.

32. Browse the log files.

- a. Using Windows Explorer, access the default log files directory:
C:\Users\Administrator\workspace\.metadata
- b. Browse the contents of any log file and look for any errors or warnings.

You successfully created an agent in Agent Builder. In the next units, you learn how to modify this agent for and install into an IBM Performance Management and an IBM Tivoli Monitoring environment.

Unit 3 Customizing agents for IBM Application Performance Management exercises

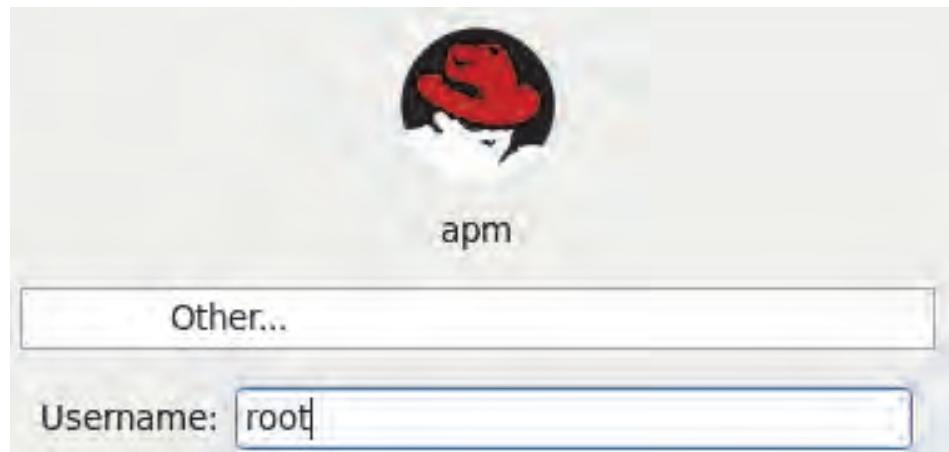
In the exercise in this unit, you modify the AB1 agent in preparation for installing it into an IBM Performance Management environment.

Exercise 1 Modify and install the AB1 agent into an IBM Performance Management environment

In this exercise, you modify the AB1 agent in preparation for installing it into an IBM Performance Management environment. You define the Summary dashboard and the Open Services for Lifecycle Collaboration (OSLC) properties.

Confirm the Performance Management server

1. Log in to the APM image as **root** with password **object00**.



2. Compare the times that are shown on the WIN1 and APM desktops and confirm that they are synchronized. If they are different, do the following tasks.

- a. On WIN1, confirm that the Windows Time Service is running by opening a command prompt and running this command:

```
net start w32time
```

The service starts or you are notified it is already running.

- b. On APM, confirm that the NTP client is running on NTP by opening a terminal window and running this command:

```
service ntpd start
```

The service starts or you are notified it is already running.

- c. On APM, run this command to force a time update:

```
ntpdate -u win1
```

A synchronization message is displayed and the time on APM now matches the time on WIN1.

Time should now be synchronized between WIN1 and APM.



Important: Time synchronization is critical between the APM server and its agent hosts. Confirm that time is synchronized across all servers throughout this class and fix as needed.

3. On APM, open a terminal window and run the following command to confirm that all services for the Performance Management server have started.

```
apm status
```

```
[root@apm Desktop]# apm status
Service db2 is started (PID 2781).
Service ksy is started (PID 17373).
Service kafka is started (PID 19305).
Service mongodb is started (PID 19555).
Service scr is started (PID 19766).
Service oidc is started (PID 19939).
Service server1 is started (PID 20311).
Service min is started (PID 23762).
Service apmui is started (PID 27323).
Service oslc is started (PID 4747).
[root@apm Desktop]# █
```

- a. If any services are not started, either give the server more time to start completely or run the following command to start a service that is not running:

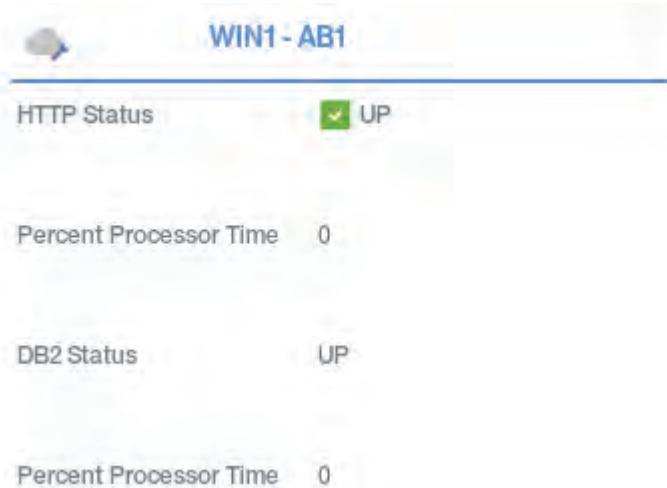
```
apm start service_name
```

where *service_name* is the name of the service that is not running.

The APM server can take 15 - 30 minutes to start completely.

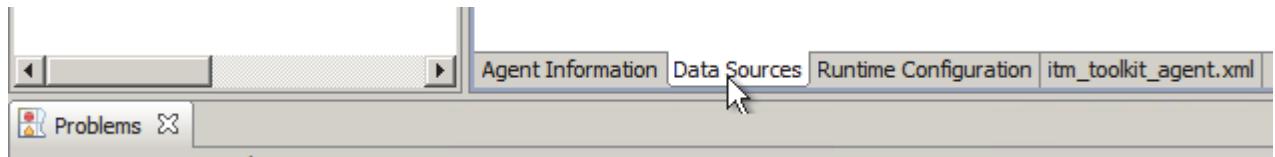
Create data points for the summary dashboard

In this section, you create single row data source that gathers the data for the summary dashboard. Your goal is to create a summary dashboard like this screen capture.

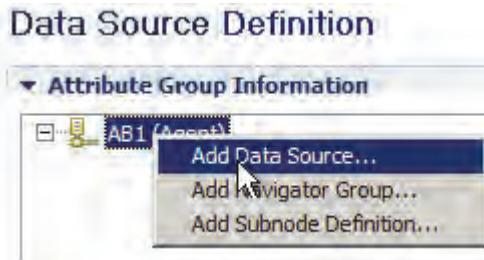


While your agent is already gathering this data, it is part of a multi-row data set that reports the status of multiple entities. A Summary dashboard can pull data only from a single row data set. In this section, you create a custom data set that gathers only a single row of data.

4. Create a filter that reduces your availability data source to a single row, giving the status of the Apache HTTP server service.
 - a. In Agent Builder on WIN1, click the **Data Sources** tab at the bottom of the Agent Editor.

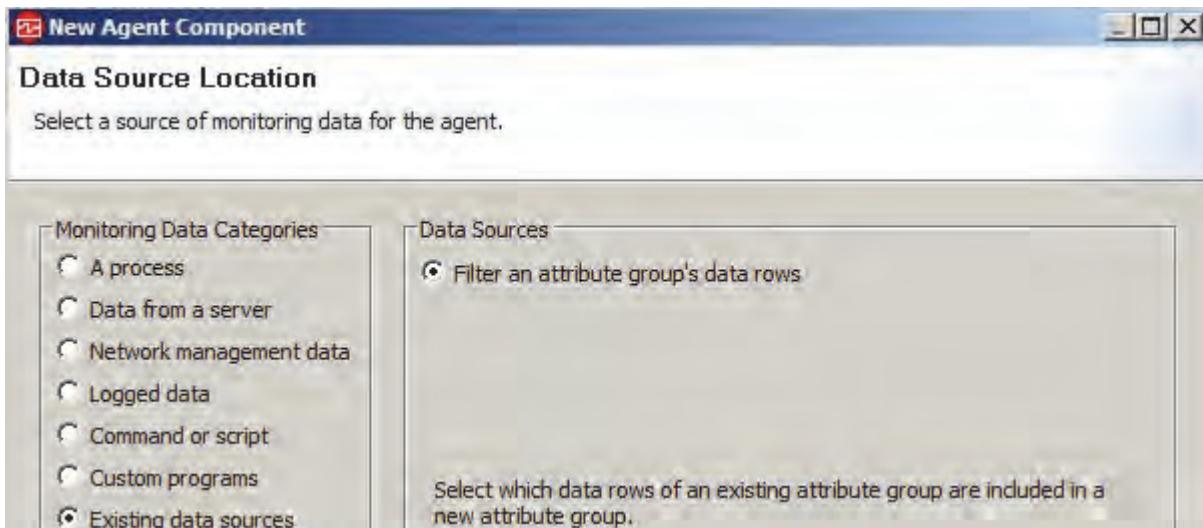


- b. Right-click **AB1 (Agent)** and select **Add Data Source...**.

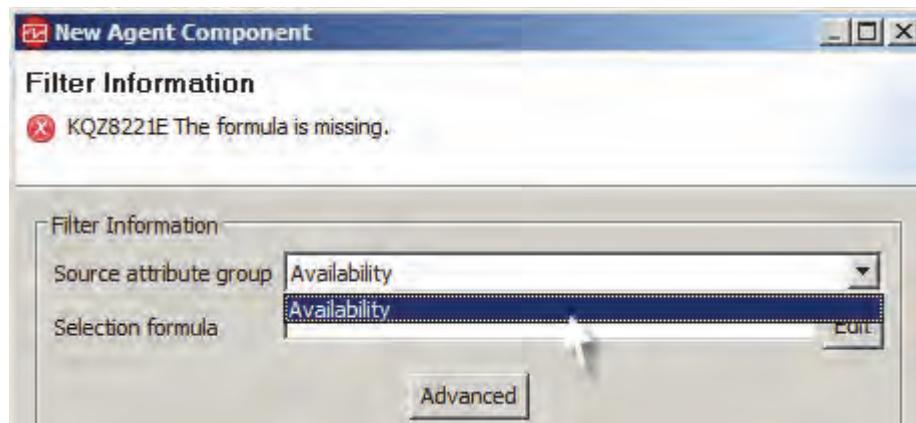


The Data Source Location window opens.

- c. Select **Existing data sources** > **Filter an attribute groups data rows**.

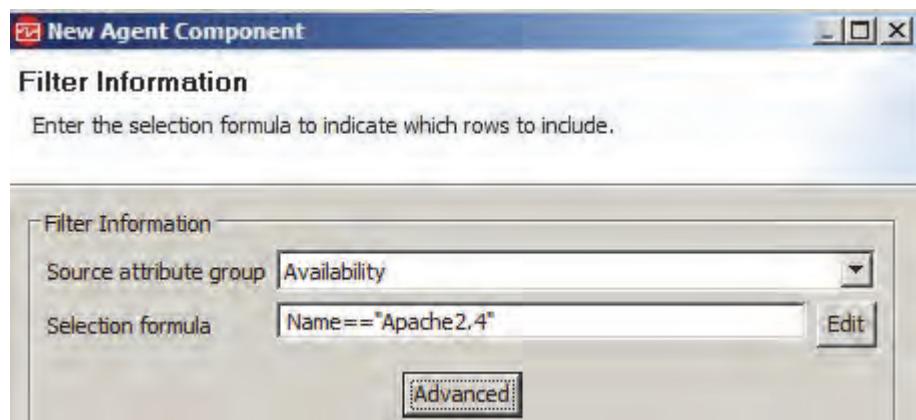


- d. Click **Next**.
e. Select **Availability** from the **Source attribute group** menu.



- f. Enter the following text in the **Selection formula** row.

Name=="Apache2.4"



This command gathers availability data for only the Apache2.4 service.



Hint: Look to the top of the window to see whether the editor identifies errors in your formula.

- g. Click **Finish**.

You are returned to the **Data Source Definition** tab.

5. Edit the filtered data source name and set it to produce only a single row of data.
 - a. Click **Availability_Filtered (Filter)**.
 - b. Delete the Attribute group name **Availability_Filtered** and replace with **ApacheStatus**.
 - c. Select **Produces a single data row** in the Filtered Attribute Group Information pane.

Data Source Definition

Attribute Group Information

- AB1 (Agent)
 - + Availability filters
 - + ApacheStatus (Filter) **(Selected)**

Filtered Attribute Group Information

Attribute group name: ApacheStatus

Help: Filtered data rows from attribute group Availability.

Produces a single data row Can produce more than one data row Produces large number of data rows

6. Ensure ApacheStatus (Filter) has no keyed attributes.
 - a. Expand **ApacheStatus (Filter)**.
 - b. Select **Application_Component** and clear the **Key attribute** check box.

Attribute Group Information

- ApacheStatus (Filter)
 - + Application_Component **(Selected)**
 - + Name
 - + HTTP_Status

Filtered Attribute Information

Attribute name: Application_Component

Help: The descriptive name of a part of the application.

Hidden - can only be used in derived attribute

Key attribute

7. Under **ApacheStatus (Filter)**, change the **Status** attribute name to **HTTP_Status**.
8. Set the following **Severity** criteria for the **HTTP_Status** attribute.

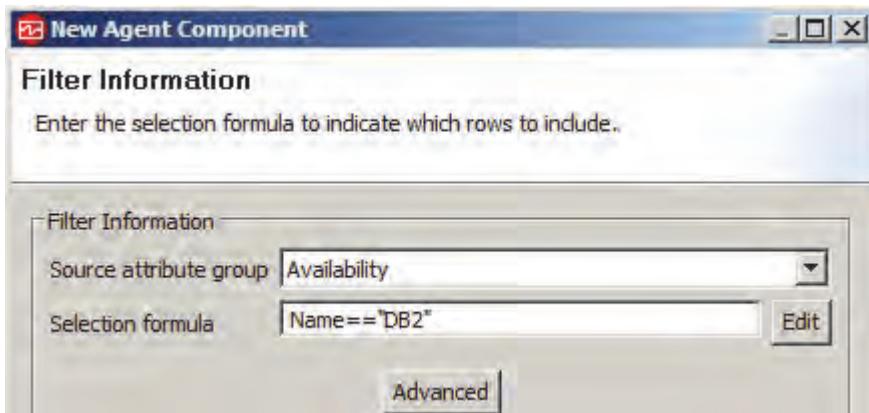
The screenshot shows the 'Attribute Group Information' window with the 'ApacheStatus (Filter)' group selected. Under 'Attribute type', the 'Severity' tab is active, displaying a table with four rows: 'Normal' (Value: 1), 'Warning' (Value: 0), 'Critical' (Value: 0), and 'Not defined'. The first two rows ('Normal' and 'Warning') are highlighted with red boxes.

Severity	Value
Normal	1
Warning	0
Critical	0
Not defined	

- a. With **HTTP_Status** selected in Attribute Group Information, scroll down in the Attribute Information window and select the **Severity** tab.
- b. Select the Severity status row and click **Edit**.
- c. Select **Single number** and enter the number in **Value**.
- d. Click **OK** to save the severity.
9. Select **File > Save** to save your agent project.
10. Create a filter that reduces your availability data source to a single row which gives the status of the DB2 server service.
 - a. From the **Data Sources** tab, right-click **AB1 (Agent)** and click **Add Data Source**. The Navigator Group Data Source window opens.
 - b. Select **Existing data sources > Filter an attribute group's data rows**.
 - c. Click **Next**.
 - d. Select **Availability** from the **Source attribute group** menu.

- e. Enter the following text in the **Selection formula** row.

Name=="DB2"



This command gathers availability data for only the DB2 service.



Hint: Look to the top of the window to see whether the editor identifies errors in your formula.

- f. Click **Finish**.

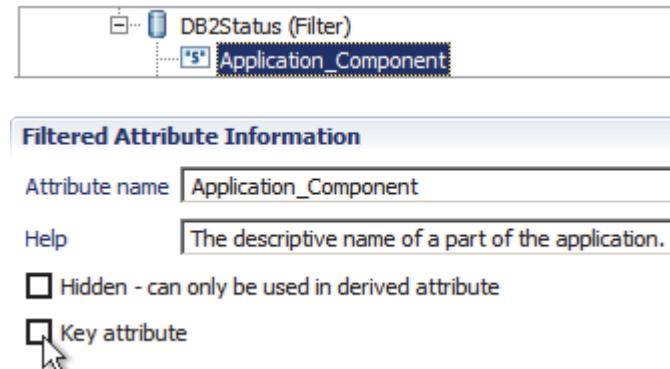
You are returned to the **Data Source Definition** tab.

11. Edit the filtered data source name and set it to produce only a single row of data.
 - a. Click **Availability_Filtered (Filter)** and change the name to **DB2Status**.
 - b. Select **Produces a single data row** in the Filtered Attribute Group Information pane.

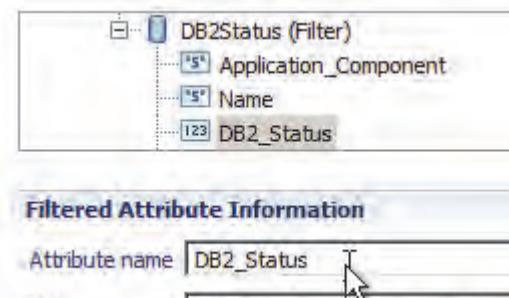
Data Source Definition

The screenshot shows the "Data Source Definition" pane. In the "Attribute Group Information" section, there is a tree view with "AB1 (Agent)" expanded, showing "Availability filters", "AppStatus (Filter)", and "DB2Status (Filter)". In the "Filtered Attribute Group Information" section, the "Attribute group name" is set to "DB2Status". Underneath, there is a note: "Filtered data rows from attribute group Availability." Below this, there are three radio buttons: "Produces a single data row" (selected), "Can produce more than one data row", and "Produces large number of data rows".

12. Verify that DB2Status (Filter) has no keyed attributes.
 - a. Expand **DB2Status (Filter)**.
 - b. Select **Application_Component** and clear the **Key attribute** check box.



13. Under **DB2Status (Filter)**, change the **Status** attribute name to **DB2_Status**.



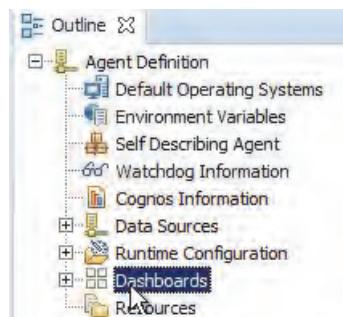
14. Save your agent project.

You successfully created metrics that can be displayed in the Summary dashboard.

Create dashboards and resource definitions

Use the Agent Builder Dashboard Setup wizard to define summary dashboard, details dashboards, and OSLC resources for your agent. These resources determine how the agent is displayed the Application Performance Dashboard and the values that uniquely identify this new type of agent.

15. Select **Dashboards** from the Agent Builder Outline view.

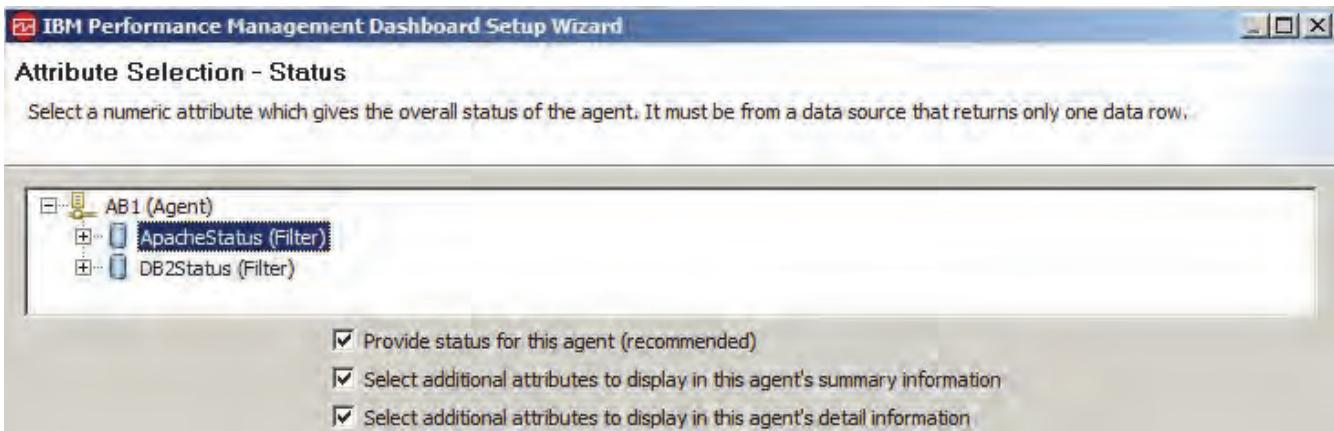


The Dashboards Overview opens.

16. Click the **Dashboard Setup wizard** link in the Overview.

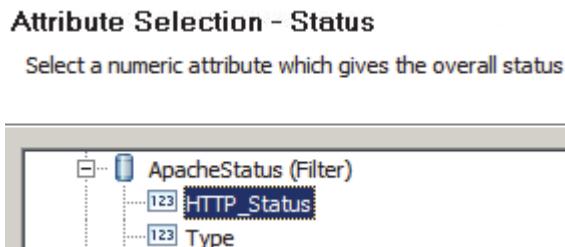


The IBM Performance Management Dashboard Setup wizard opens.



You are prompted to select a numeric attribute that gives the overall status of the agent. Only data sources that generate a single row of data are shown. The attribute must also define values for status severity. If the dashboard is for a subnode, the attribute must be from a data source that belongs to this subnode.

17. Under **ApacheStatus (Filter)**, select the **HTTP_Status** attribute.



You are also prompted to identify the parts of the IPM agent configuration you want to configure.

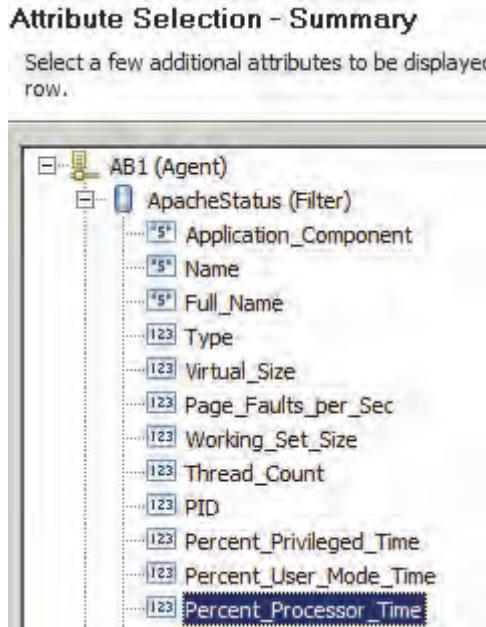
18. Ensure that all three check boxes are selected.

<input checked="" type="checkbox"/> Provide status for this agent (recommended)
<input checked="" type="checkbox"/> Select additional attributes to display in this agent's summary information
<input checked="" type="checkbox"/> Select additional attributes to display in this agent's detail information

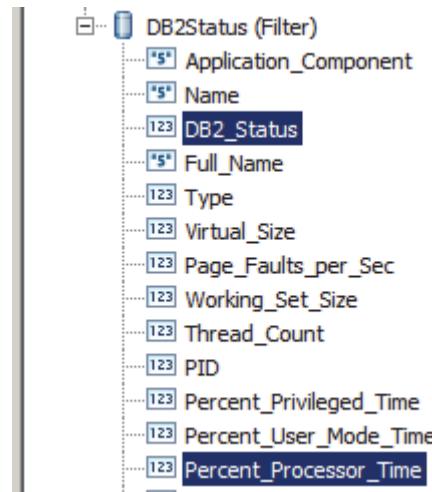
19. Click **Next**.

The Attribute Selection - Summary view opens. You are prompted to select more attributes for the Summary dashboard.

20. Under **ApacheStatus (Filter)**, select **Percent_Processor_Time**.



21. Pressing Ctrl while you click under **DB2Status (Filter)**, select **DB2_Status** and **Percent_Processor_Time**.



Up to five attributes can be displayed in the Summary dashboard.

22. Click **Next**.

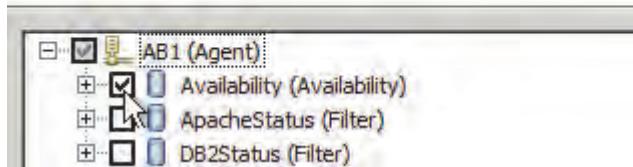
The Attribute Selection - Details view opens.

You are prompted to identify data groups and items to display in the details dashboards.

23. Select the **Availability (Availability)** data group:

Attribute Selection - Details

Select attributes to be displayed in the agent detail page.



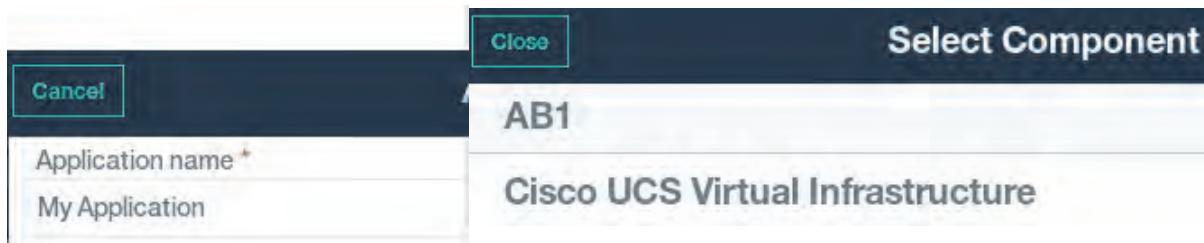
24. Click **Next**.

Resource Type

Enter a name indicating the type of software server being monitored by this agent.

String to identify this type of software server

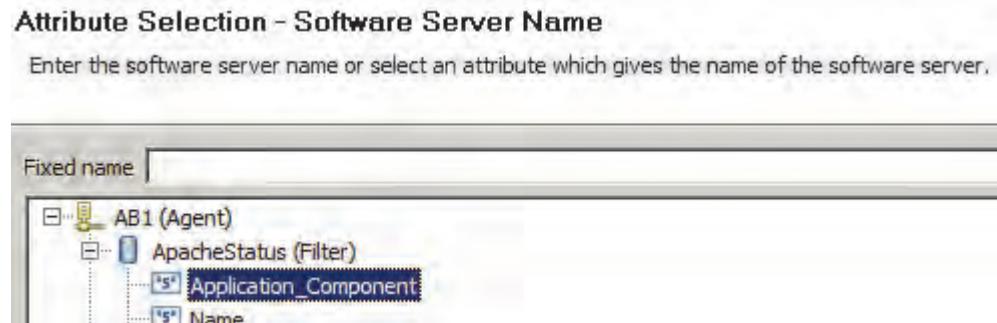
You are prompted for the Resource Type. Enter the server type that you are monitoring, for example, Email server or SampleCo Database Server. This name shows in the Performance Management Console when you add an application component to an agent definition as shown here.



25. Keep **AB1** in the resource type name field and click **Next**.

The Attribute Selection - Software Server Name view opens.

26. Select the **Application_Component** attribute under **ApacheStatus (Filter)**.



This name is displayed to the user for this particular monitored instance, for example, the name of the Apache application server instance. In this agent, the Software Server Name is being created by an attribute generated by the agent, **Application_Component**.

27. Click **Next**.

The Attribute Selection - IP Address view opens.

28. Check the top check box that selects the agent's IP address.

Attribute Selection - IP Address

Select the source of the IP address of the software server: either from an attribute or from the agent's IP address.



Hint: If the agent has an attribute that provides the host name, you can create a derived attribute that uses the nameToIPAddress function to determine the IP address.

29. Click **Next**.

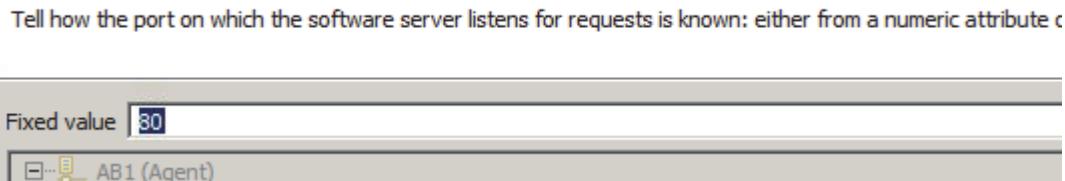
The Attribute Selection - Port view opens.

You use Port to further distinguish the specific software server when the software server name and IP address is not sufficient. For example, if you have multiple agent instances on a single host, use their listening port or some other derived attribute is needed to distinguish the instances.

If you do not need this value, any number is acceptable.

30. Enter **80** in the **Fixed value** field.

Attribute Selection - Port



31. Click **Finish** to complete the wizard.

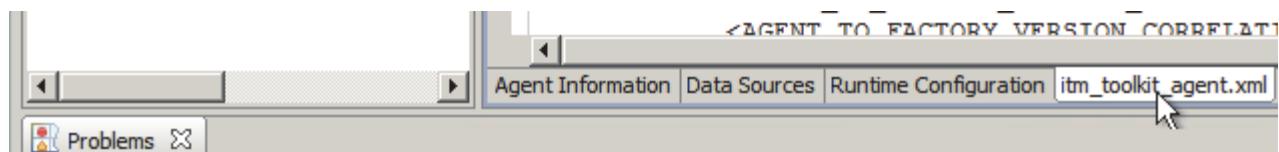
32. Save your agent changes.

You successfully defined the summary dashboard detailed dashboard, and the monitored resource data within OSLC.

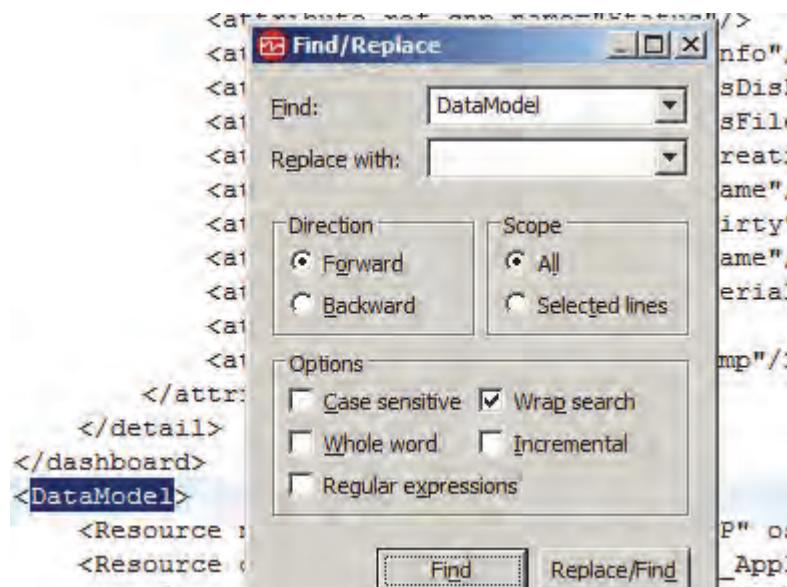
Configure the agent for quick identification within the Performance Management Console

This change ensures that the new agents show promptly in the Performance Management Console so that you can confirm your agent.

33. Click the **itm_toolkit_agent.xml** tab at the bottom of the Agent Editor.



34. Press CTRL+F to locate the **<DataModel>** tag.



35. Add the following to the DataModel tag:

```
refreshInterval="5"  
  
</dashboard>  
<DataModel refreshInterval="5">  
  <Resource name="My_Windows_Application_IP" oslcType="crtv:IPAddress"/>
```

The default refresh interval is 60 minutes.



Important: In a production environment, a refresh interval of 5 minutes is probably too frequent.

36. Save your agent project.

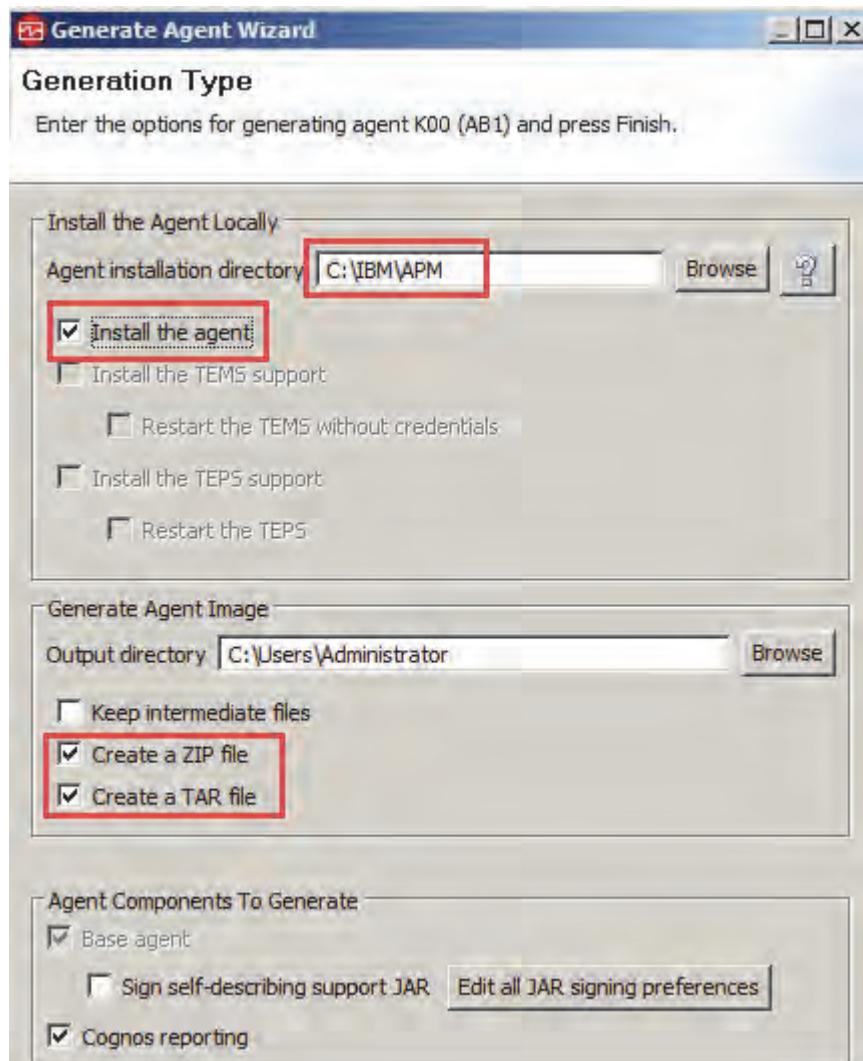
Install the AB1 agent and create the installation scripts

37. Select **Agent Editor > Generate Agent** from the main menu.

38. Set the **Agent installation directory** to **C:\IBM\APM**.

39. Select **Install the agent**.

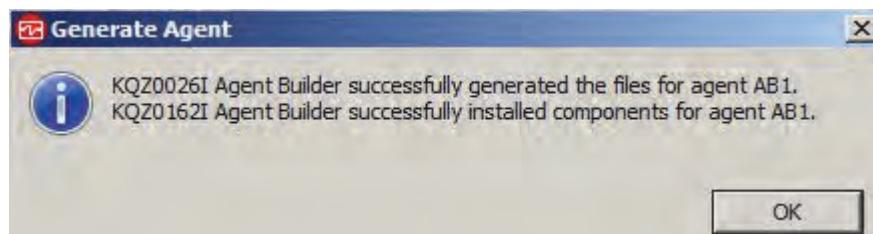
40. Ensure that **Create a ZIP file** and **Create a TAR file** are checked.



Notice the directory where the files are created. You go to this directory to access the generated files. Keep the default output directory for this exercise.

41. Click **Finish**.

A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated and the agent is installed.



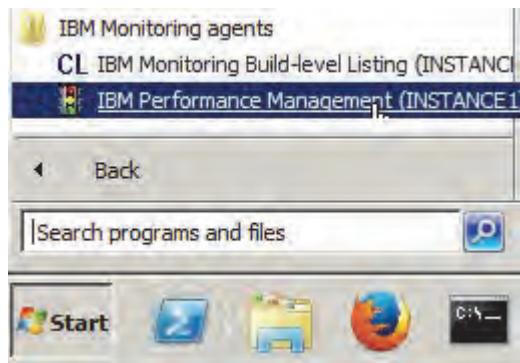
42. Click **OK** to close the confirmation window.

43. Open a command prompt or Explorer window and go to **C:\Users\Administrator** and confirm that the **smai-ab1-01.00.00.00.tgz** and **smai-ab1-01.00.00.00.zip** files are created.

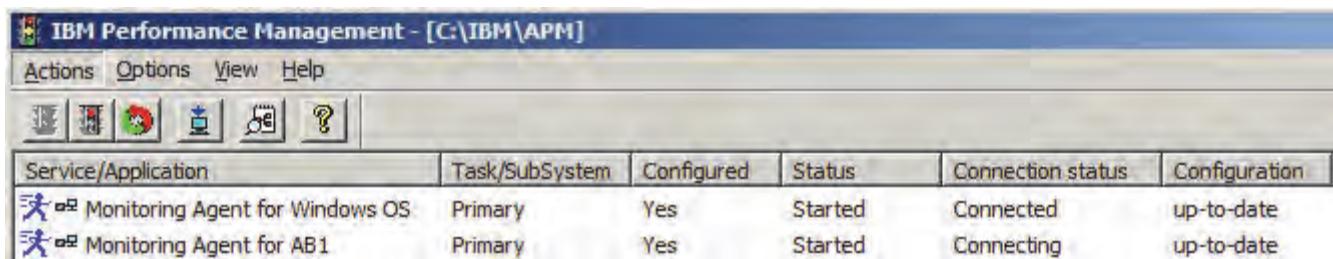
The **smai-ab1-01.00.00.00.tgz** and **smai-ab1-01.00.00.00.zip** files contain the same files that you can use to install the agent and the application support for all operating systems, including Windows, UNIX, and Linux. Because of the archive formats, the files in the .zip file are often used for installing components on Windows and the .tgz file are often used to install components on UNIX and Linux.

You install the agent with the installation scripts in a later exercise.

44. Confirm your agent installation in the IBM Performance Management utility.
a. Select **Start > All Programs > IBM Monitoring agents > IBM Performance Management (Instance1)**.



The IPM utility opens.



- b. Notice the following items:
- ◆ The name of your agent in the MTMS utility is **Monitoring Agent for AB1**.
 - ◆ The agent is started.
 - ◆ The agent version is **1.00.00**.

Confirm the agent in the Performance Management Console

45. Log in to the **APM** server as user **root** with password **object00**.

46. Open a browser and go the following URL:

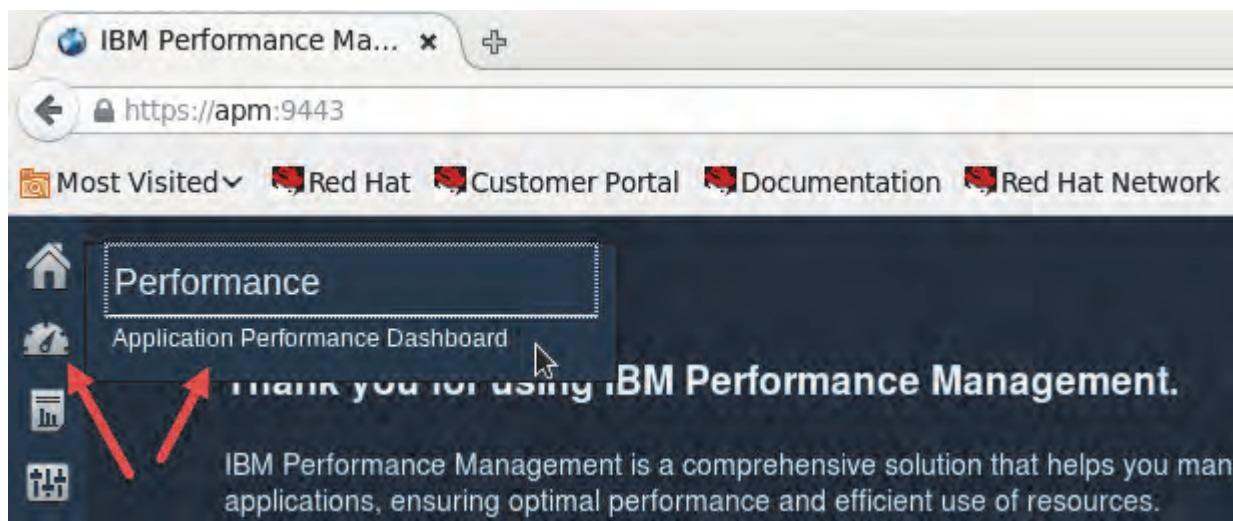
<https://apm:9443>



Hint: A bookmark to this URL is in the browser.

47. Log in to the PM console as user **apmadmin** with password **object00**.

48. Open the Application Performance dashboard.



49. Click the **Add Application** button to create a new application.



The Add Application windows opens.

50. Enter **My Application** in the **Application Name** field.

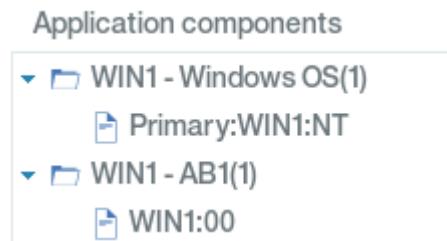
Cancel **Add Application** Save

Application name *
My Application

Description

Read...

51. Add the following components:

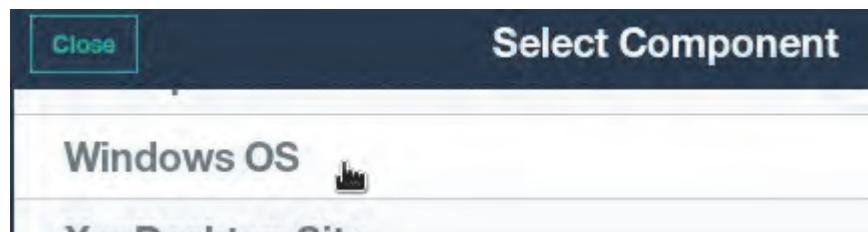


- Click the **Add components** button next to the Application components pane.



The Select Component window opens.

- Scrolling down, locate and select **Windows OS**.

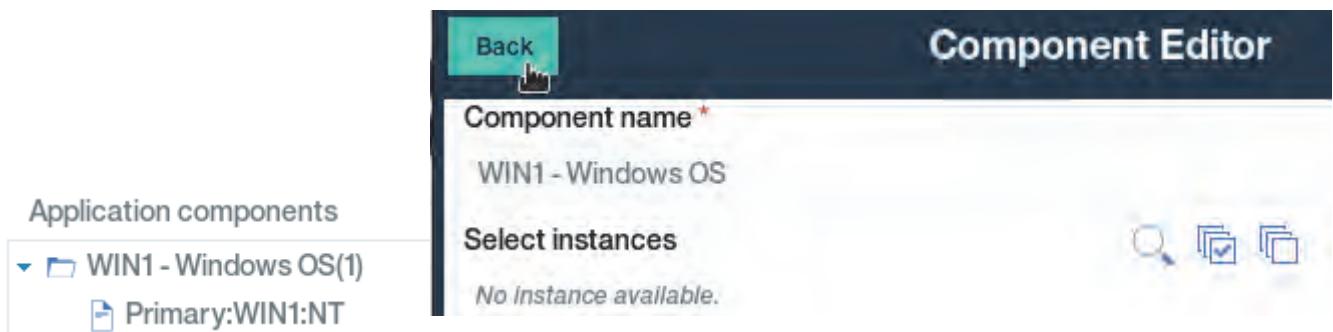


The Component Editor windows opens.

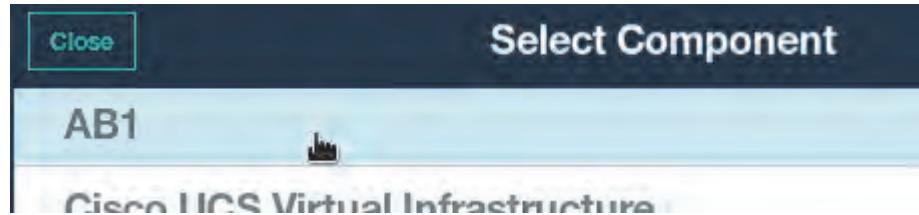
- Select **WIN1:NT** and click **Add**.



The agent is removed from the Component Editor window added to your application definition.

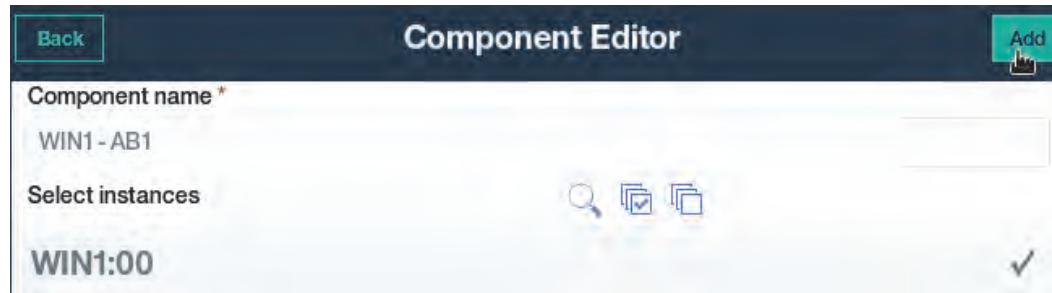


- d. Click **Back** to return to the Select Component window.
- e. Scrolling up, locate and select **AB1**.



The Component Editor windows opens.

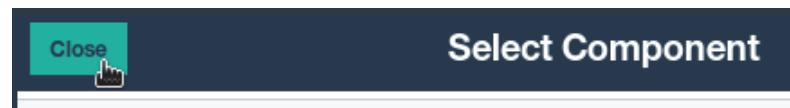
- f. Select **WIN1:00** agent and click **Add**.



Important: It can take 10 TO 15 minutes from when the agent is installed until it appears in the Component Editor window. If the agent is not listed, click Back and AB1 until it is shown. Also, ensure the date and time on APM and WIN1 hosts is the same.

The screenshot shows the 'Component Editor' window. On the left, under 'Application components', there are two collapsed sections: 'WIN1 - Windows OS(1)' which contains 'Primary:WIN1:NT', and 'WIN1 - AB1(1)' which contains 'WIN1:00'. In the center, the 'Component name' field is populated with 'WIN1 - AB1'. Below it, the 'Select instances' section displays the message 'No instance available.' with three icons: a magnifying glass, a checkmark, and a square.

- g. Click **Back** to return to the Select Component window.
- h. Click **Close** to close the Select Component window.

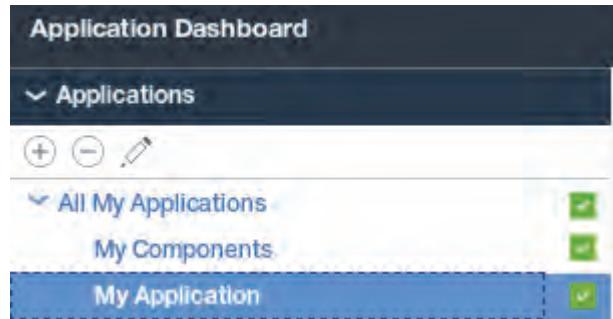


You are returned to the Add Application window.

The screenshot shows the 'Add Application' window. At the top, there are 'Cancel' and 'Save' buttons. The main area includes fields for 'Application name' (containing 'My Application') and 'Description'. Under 'Application read from', there is a 'Read...' button. The 'Template' section shows 'Custom Application' selected. The 'Application components' section lists the same components as the previous window: 'WIN1 - Windows OS(1)' (with 'Primary:WIN1:NT') and 'WIN1 - AB1(1)' (with 'WIN1:00'). To the right of the component list are three icons: a plus sign inside a dotted square, a minus sign inside a dotted square, and a circular arrow.

52. Click **Save** and **OK** to save your application definition and exit the Add Application utility.

You are returned to the Application Dashboard. After a few minutes, your My Application shows in the Applications explorer.

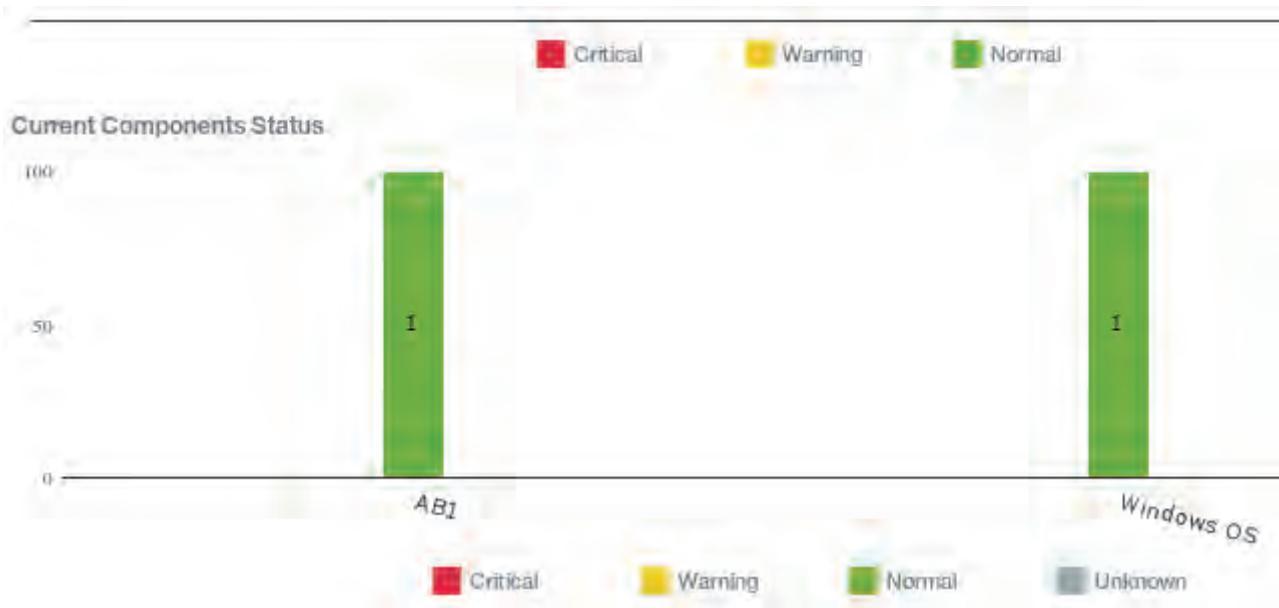


53. Click **My Application** in the Applications explorer.

54. Refresh this view until the AB1 component turns green.



Hint: It can take up to 15 minutes for the Summary dashboard and initial data to be displayed.



55. Click the **AB1** component bar in Current Components Status.

The AB1 Status Overview window opens with a Summary dashboard for the component.

The screenshot shows the 'Status Overview' tab selected in the navigation bar. The main title is 'WIN1-AB1'. Below it, there are two rows of status information:

HTTP Status	UP
Percent Processor Time	0

Below these, there are two more rows of status information:

DB2 Status	UP
Percent Processor Time	0

56. Click the **Summary** dashboard widget to access the detailed dashboards.

The screenshot shows the 'Status Overview' tab selected in the navigation bar. The main title is 'WIN1:00'. Below it, there is a table titled 'Availability' with columns: Application Component, Name, Status, Full Name, and Type. The table lists several application components:

Application Component	Name	Status	Full Name	Type
Apache2.4	Apache2.4	UP	Apache2.4	SERVICE
DB2 - DB2COPY1 - DB2	DB2	UP	DB2 - DB2COPY1 - DB2	SERVICE
DB2DAS - DB2DAS00	DB2DAS00	UP	DB2DAS - DB2DAS00	SERVICE
DB2 Governor (DB2COPY1)	DB2GOVERNOR_DB2COPY1	DOWN	DB2 Governor (DB2COPY1)	SERVICE
DB2 License Server (DB2COPY1)	DB2LICD_DB2COPY1	DOWN	DB2 License Server (DB2COPY1)	SERVICE
DB2 Management Service (DB2...	DB2MGMTSVC_DB2COPY1	UP	DB2 Management Service (DB2...	SERVICE
DB2 Remote Command Server (...	DB2REMOTECMD_DB2COPY1	UP	DB2 Remote Command Server (...	SERVICE

Here you can display the attributes that were added to the detailed dashboards.

You successfully confirmed the following about your agent in an APM environment:

- The agent is available to be added to an application.
- The Summary dashboard shows the core metrics that you added.
- A subset of attributes is visible in the Details dashboard.

Unit 4 Monitoring Windows resources exercises

Exercise 1 Monitor Windows resources

Your company sells products online. The application has a web server front end that pulls information from a DB2 database. Every component for this application runs on 15 different Windows servers. You need to build a custom agent that you can install on each server. The agent monitors the specific software and hardware components of this application.

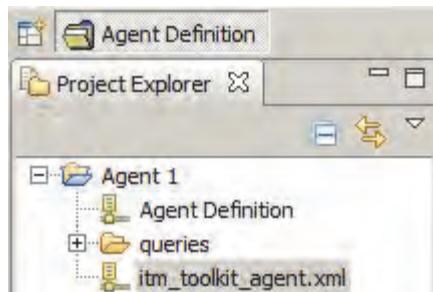
In this exercise, you modify your AB1 agent to monitor the following items:

- The web server that provides the HTML portions of the application
- The DB2 database services and process that provide the database of this application
- The logical disk space where the database is stored
- Windows systems events for events that are related to the HTTP server and DB2

Prepare to edit your agent

In this section, you start Agent Builder and open your AB1 agent.

1. On WIN1, launch Agent Builder if it is not already running.
2. If prompted, keep the default workspace directory and click **OK**.
3. Maximize the Agent Builder application window.
4. If the **AB1** agent is not already open, expand **Agent 1** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.



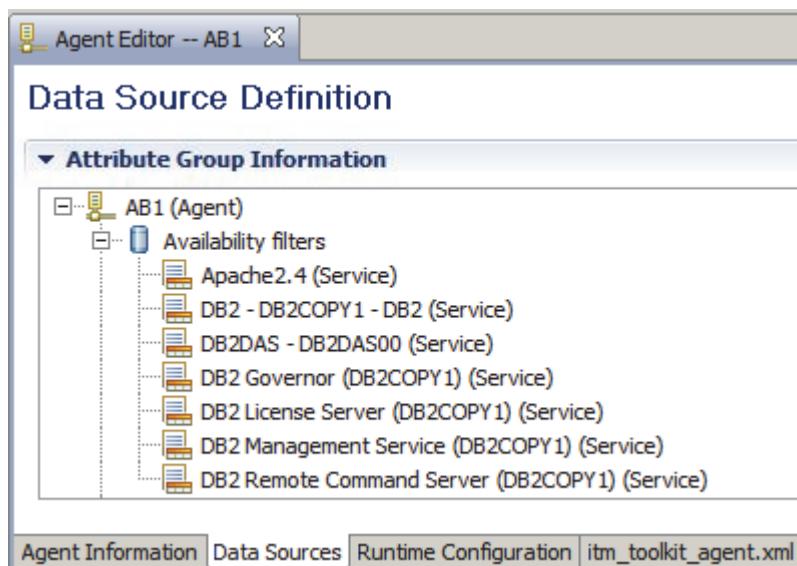
Add WMI process information to the agent

In this section, you do the following tasks:

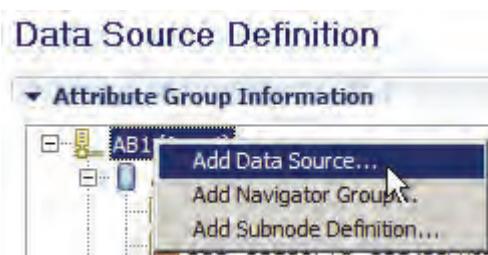
- Add Windows process information from WMI to your agent
- Change the default attribute group name
- Remove several attributes from the group

Complete the following steps:

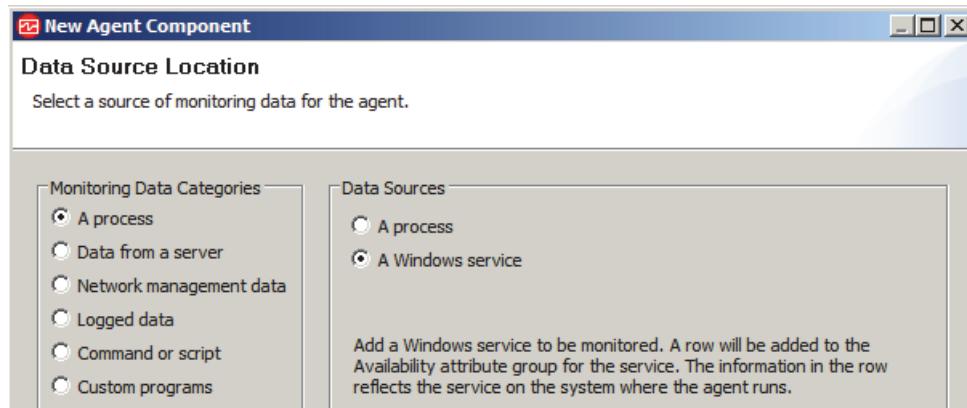
5. Click the **Data Sources** tab.



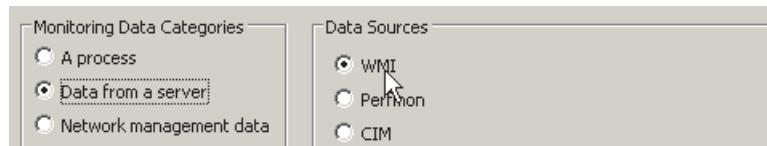
6. Right-click **AB1 (Agent)** under **Attribute Group Information** and click **Add Data Source**.



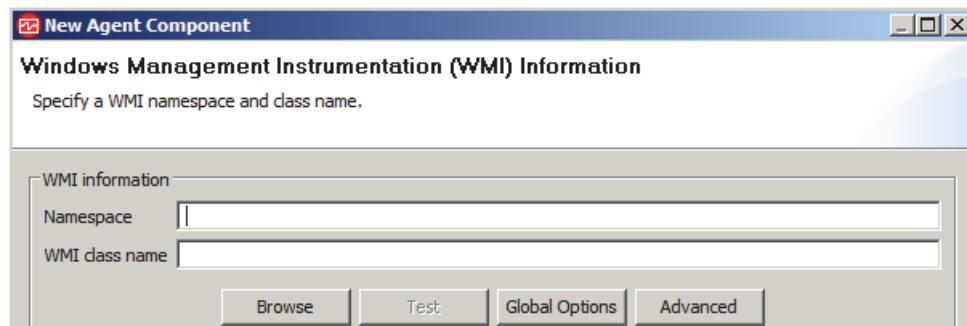
The Data Source Location window opens.



7. Select **Data from a server**.
8. Verify that **WMI** is selected and click **Next**.



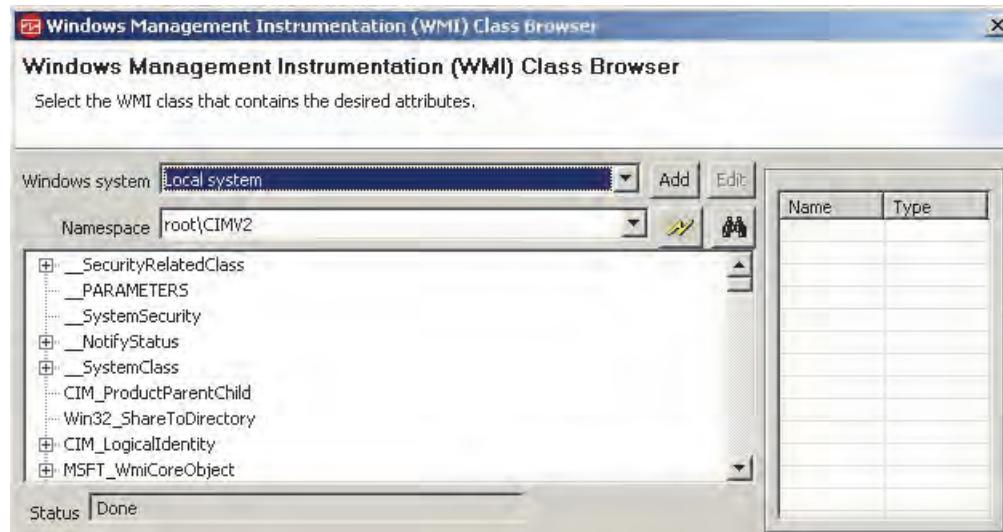
The Windows Management Instrumentation (WMI) Information window opens.



To create a successful agent, you must know the information that needs to be monitored or how to identify the data. For WMI, you must know the namespace and class that provide the data that you want. You can either enter the information manually in the fields that are shown or you can browse a running Windows system for the WMI information.

9. Click **Browse**.

The Windows Management Instrumentation (WMI) Class Browser opens.



10. Verify that **Windows system** is **local system** and **Namespace** is **root/CIMV2**.

11. Click the **Search** icon

12. Enter **Process** in the **Search phrase** field and click **Search**.



13. Locate and select **Win32_Process**.

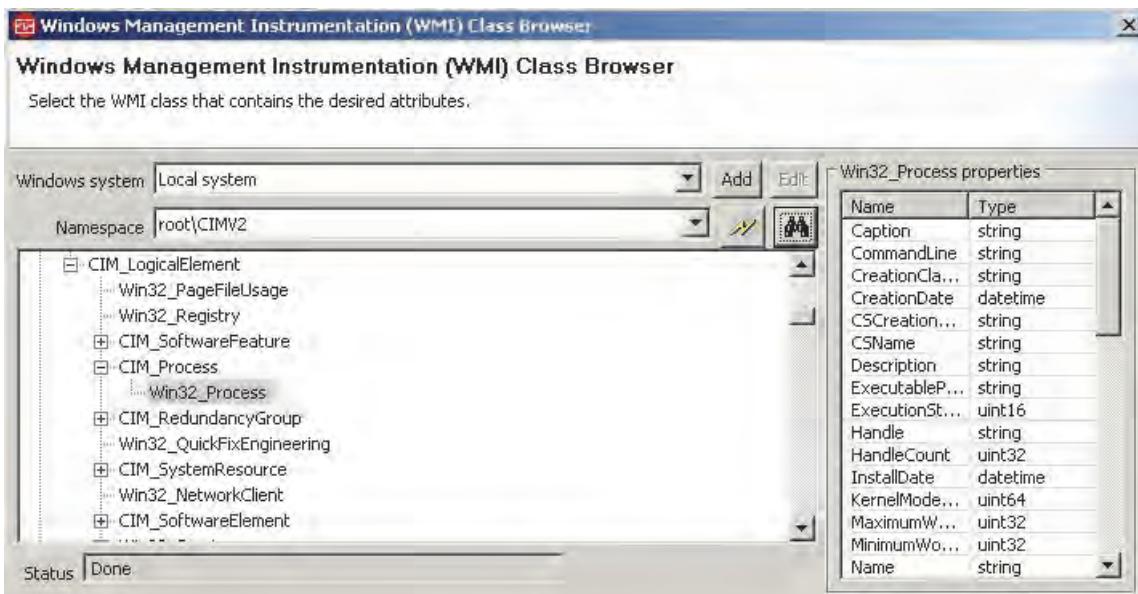
14. Click **OK**.





Important: Be careful when selecting **Win32_Process**. Several items have similar names, such as **Win32_Processor**.

You are returned to the Class Browser window.

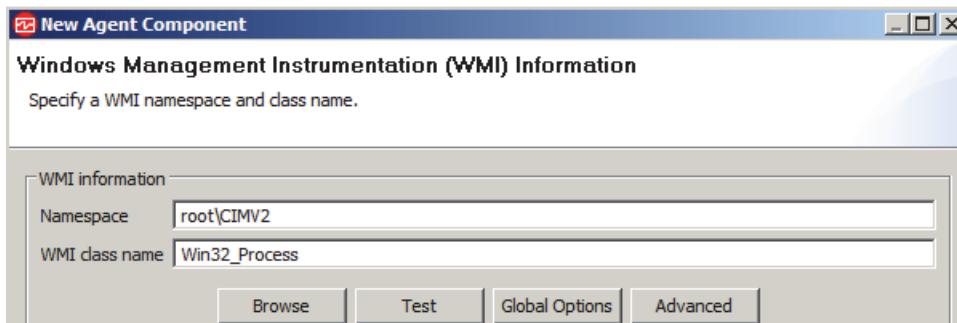


Notice the **Win32_Process** properties on the right. This data can be gathered from this WMI class. Agent Builder creates a table structure, called an **attribute group**, to hold this class and field or column, called an **attribute**, for each of these properties when they are monitored. Later you see where you can edit this attribute list by removing attributes that you do not want the agent to monitor.

15. Click **OK** to select the **Win32_Process** class and all of its properties.

The Class Browser window closes.

The Windows Management Instrumentation (WMI) Information window is displayed with the **Namespace** and **WMI class name** fields completed.

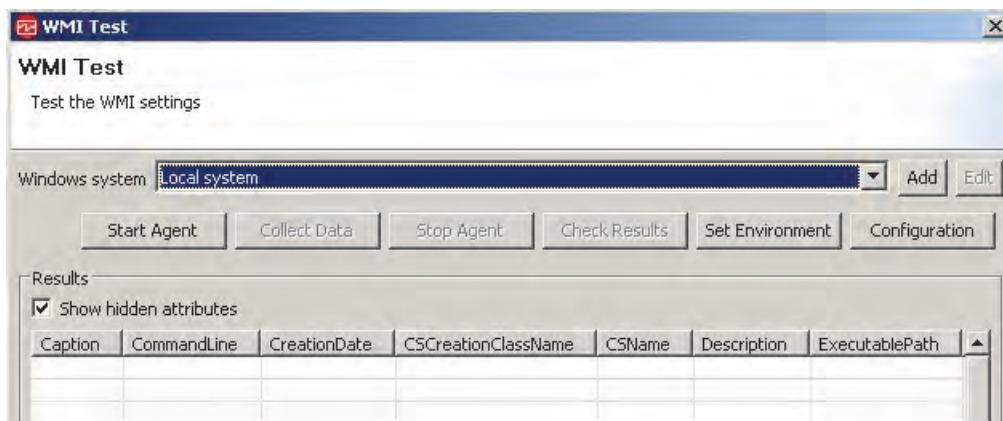


Many data sources have a test feature to confirm that the data source can gather the target data and to configure the attributes.

16. Test this individual attribute group.

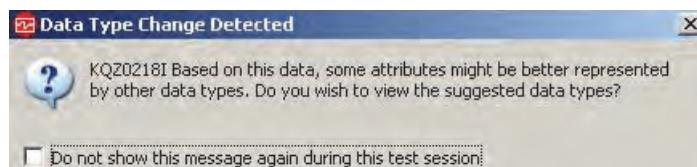
- a. Click **Test**.

The WMI Test window opens.

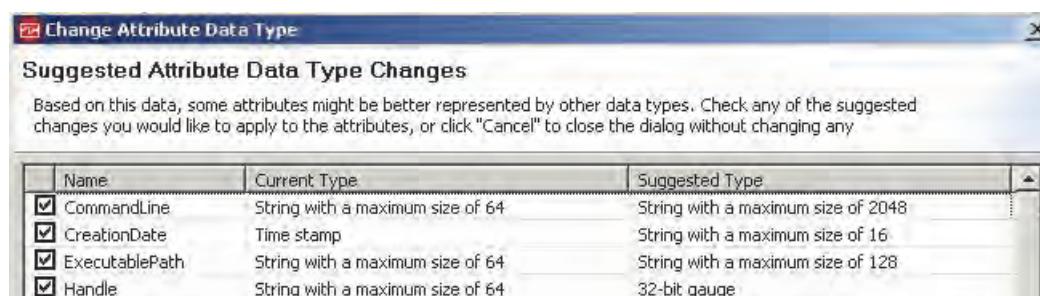


The default of the tester is to gathering data from the local system.

- b. Click **Start Agent** and **Collect Data**.



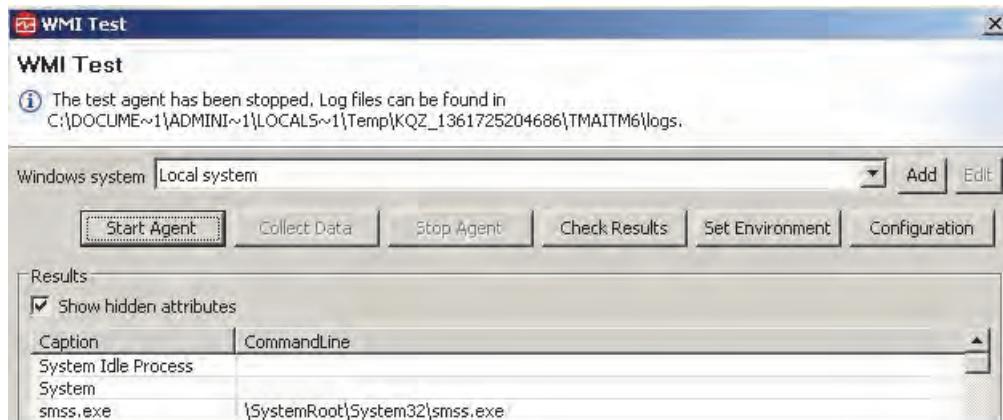
- c. Read the **Data Type Change Detected** window and click **Yes**.



The window suggests several data type changes. Clear individual attributes that you do not want changed. Click **OK** to save all checked changes. Click **Cancel** to make none of the suggested changes.

- d. Click **OK** to apply all of these suggested data type changes.

The WMI Test window reopens with data that is gathered from the local host.



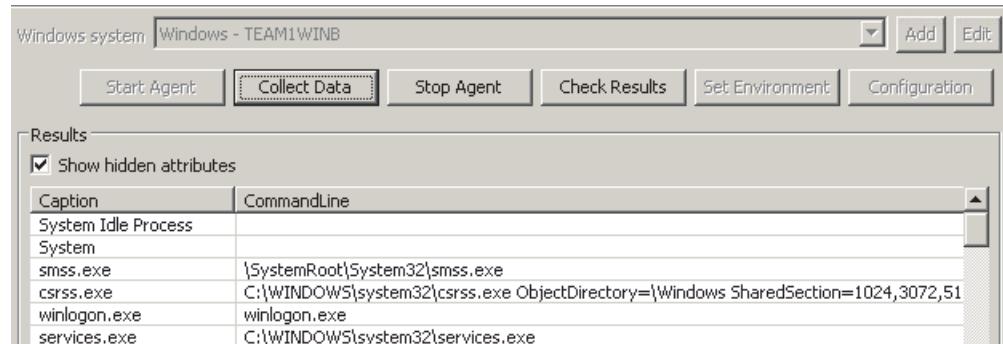
The tester can also gather data from a remote system.

- e. Click **Stop Agent**, if it is available.
- f. Select **Windows - WIN2** from the **Windows systems** menu.



Important: If the WIN2 connection does not exist, return to [Unit 2, Exercise 1, Step 16](#) on page 2-7 for instructions on creating the connection definition to WIN2.

- g. Click **Start Agent** and **Collect Data**.



- h. Confirm that the tester gathers process data from WIN2.

- i. Click **OK** to close the WMI Tester.

You modified your agent to gather the **Win32_Process** data from WMI and tested the individual data source with the tester utility.

17. Click **Finish** to close the **New Agent Component** window.

The **Data Source Definition** tab on the Agent Editor is displayed.

Attribute group information

- AB1 (Agent)
- Availability filters
- Win32_Process (WMI)

WMI Class Information

Attribute group name: Win32_Process

Help: Data gathered from WMI class Win32_Process.

Produces a single data row Can produce more than one data row Produces large number of data rows

Add this attribute group to a reporting category Performance

Agent Information | Data Sources | Runtime Configuration | itm_toolkit_agent.xml

Note: If you modified the agent for IBM Performance Management, you also see the Apache_Status and DB2_Status filter data sources.

With the **Win32_Process** attribute group selected, you can see and edit its WMI class information. The attribute group name acquired the WMI class name. This name is displayed in the Tivoli Enterprise Portal navigator. You can edit it without affecting the data that it retrieves.

18. Change **Win32_Process** to **My_App_Processes**.

- Click **Win32_Process** in the **Attribute Group Information** navigator.
- In the **WMI Class Information** section, replace the attribute group name **Win32_Process** with **My_App_Processes**. Press Enter to activate the change.

Attribute group information

- AB1 (Agent)
- Availability filters
- My_App_Processes (WMI)

WMI Class Information

Attribute group name: My_App_Processes

Help: Data gathered from WMI class Win32_Process.

The attribute group name changes in the upper display also.

19. Click the plus sign (+) next to **My_App_Processes (WMI)** to expand the attribute group and list its attributes.



Notice that the attribute names are based on the WMI names that you saw when you searched for **Process** in the WMI browser. In this window, you can remove unwanted attributes from this attribute group. You can also add attribute groups, remove attribute groups, and edit attributes and attribute groups.

20. Click an attribute, such as **CreationClassName**.

WMI Attribute Information

Attribute name	CreationClassName
Help	Attribute CreationClassName from /
<input type="checkbox"/> Hidden - can only be used in derived attribute	

You can edit the attribute metadata, such as the attribute name, the maximum size, whether the data is hidden, and whether it is a key attribute.

21. Remove the following attributes:

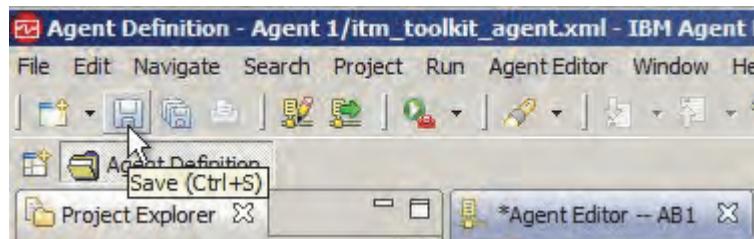
- CreationClassName
- OSCreationClassName
- Quota* (Total of 4)
 - a. Scroll down the list and select one or more attributes.



Hint: Press Ctrl and click to select more than one attribute at a time.

- b. Click **Remove**.
- c. Repeat until you remove all six attributes.

22. Select **File > Save** or click the **Save** icon to save your agent project.



You modified **AB1** to gather Windows process information from WMI. Continue to the next section without closing Agent Builder.

Add combined disk information to the agent

In this section, you do the following tasks:

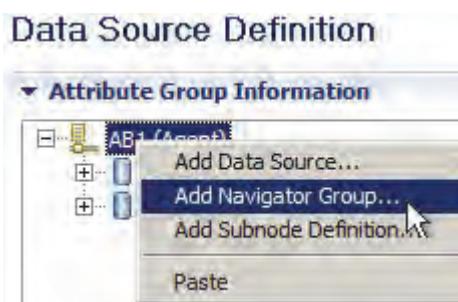
- Add logical disk information from both Perfmon and WMI to your agent
- Create a Navigator node that is called **Disk_Storage** and place both attribute groups under it
- Combine the logical disk attribute groups from Perfmon and WMI into one new attribute group that merges both sets of data

Complete the following steps:

23. From the **Data Source** tab, create a navigator group that is called **Disk_Storage** and add the Perfmon logical disk attributes.
a. Right-click **AB1 (Agent)** under **Data Sources** and click **Add Navigator Group**.



Hint: Collapsing all objects in the Attribute Group Information navigator might help in locating **AB1**.



The Navigator Group Information window opens.



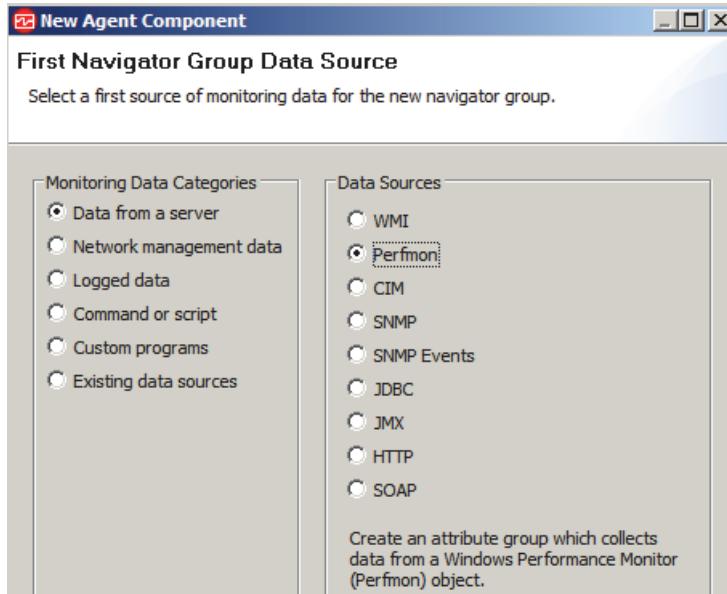
- b. Enter the following information:
- ◆ Navigator group name: **Disk_Storage**
 - ◆ Help: **Logical disk storage information about AB1**

A screenshot of the 'Navigator Group Information' dialog box. The 'Navigator group name' field contains 'Disk_Storage' and the 'Help' field contains 'Logical disk storage information about AB1'.

- c. Click **Next**.

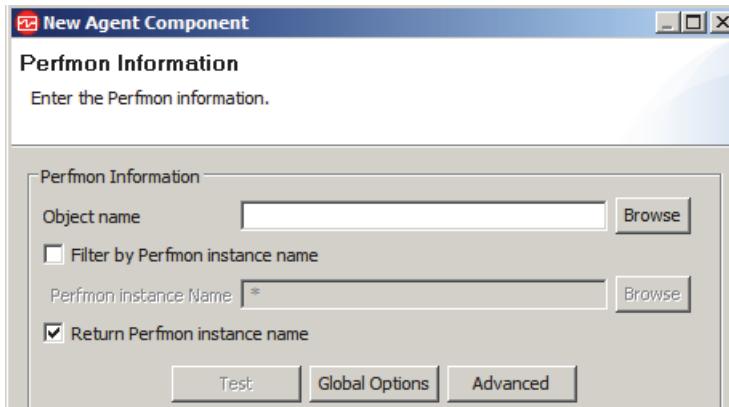
The First Navigator Group Data Source window opens. You now add the Perfmon LogicalDisk attributes.

- d. Select **Data from a server** under **Monitoring Data Categories** and select **Perfmon** under **Data Sources**.



- e. Click **Next**.

The Perfmon Information window opens.



- f. Click **Browse**.

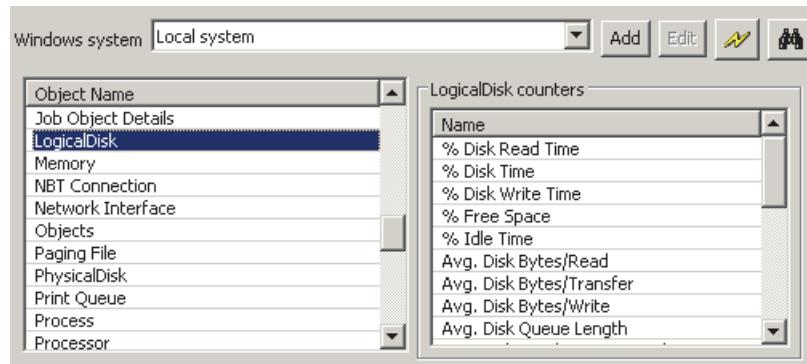
The Performance Monitor (Perfmon) Object Browser opens and displays Perfmon data.



As with the WMI browser, you can connect to remote Windows hosts and browse the Perfmon database. However, you do not browse a remote host in this exercise.

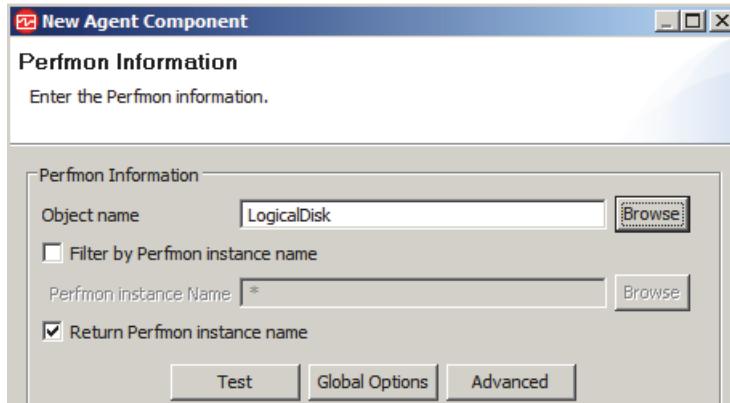
- g. If **Windows - WIN2** is still selected in the **Windows system** field, select **Local system**.

- h. Locate and click **LogicalDisk** from the **Object Name** column.



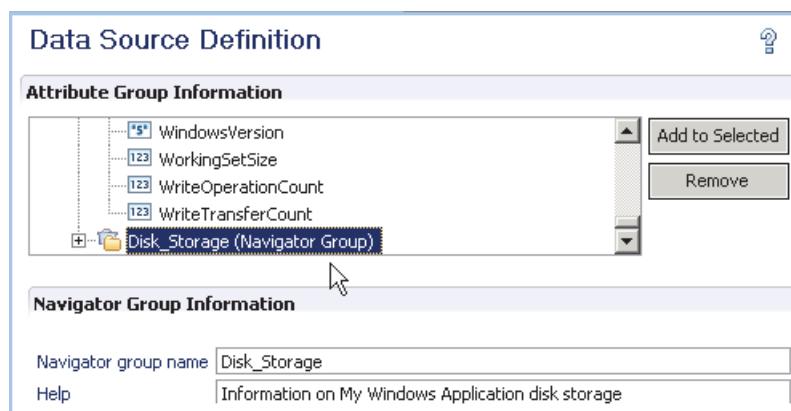
Notice all the counters that are contained in this object.

- i. Click **OK** to identify the LogicalDisk object and all of its counters the agent monitors.
The Perfmon Information window opens.



- j. Click **Finish**.

The Agent Editor **Data Source Definition** tab is displayed.



k. Expand **Disk_Storage (Navigator Group)**.



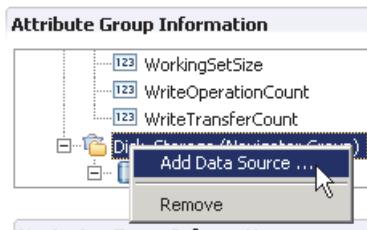
l. Expand **Logical Disk (PerfMon)**.



Now you can see the Navigator Group, the Attribute Group, and the Attributes that the agent gathers and displays in the management console. From here, you can add and remove attributes and attribute groups. You can also edit any of these new components. For this exercise, you keep each of the Perfmon components as they are.

24. Add the **WMI Win32_LogicalDisk** attributes to the **Disk_Storage Navigator Group**.

a. Right-click **Disk_Storage (Navigator Group)** and click **Add Data Source**.



The Navigator Group Data Source window opens.

b. Verify that **Data from a server** under **Monitoring Data Categories** and **WMI** under **Data Sources** are selected.

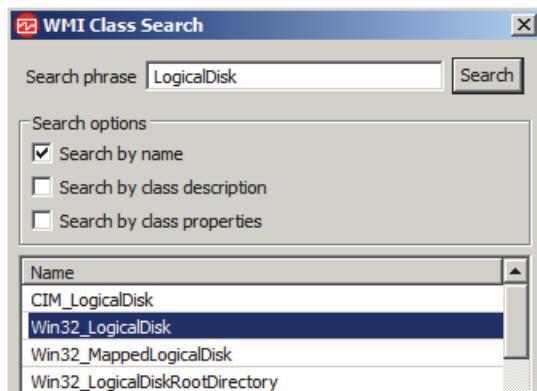


c. Click **Next**.

The Windows Management Instrumentation (WMI) Information window opens.

d. Click **Browse**.

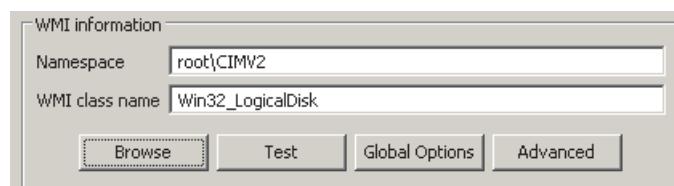
- e. Verify that **Windows system** is **Local system** and **Namespace** is **root\CIMV2**.
- f. Click the **Search** icon (binoculars).
- g. Enter **LogicalDisk** in the **Search phrase** field and click **Search**.
- h. Click **Win32_LogicalDisk** and click **OK**.



The Class Browser window opens.

25. Click **OK** to select the **Win32_LogicalDisk** class and all of its properties to be monitored by this agent.

The Class Browser window closes and the Windows Management Instrumentation (WMI) Information window is displayed with the **Namespace** and **WMI class name** fields completed.



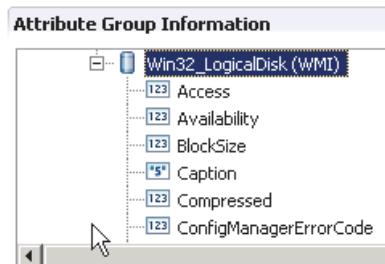
You modified your agent to gather the **Win32_LogicalDisk** data from WMI.

26. Click **Finish**.

The Agent Editor **Data Source Definition** tab is displayed.

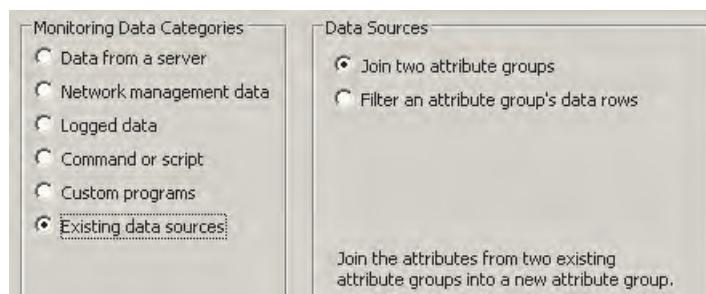
The screenshot shows the 'Data Source Definition' tab in the Agent Editor. The 'Attribute Group Information' section displays a tree view of attributes under 'Disk_Storage (Navigator Group)'. The 'Win32_LogicalDisk (WMI)' node is expanded, showing 'LogicalDisk (PerfMon)' and 'Win32_LogicalDisk (WMI)'. To the right of the tree view are 'Add to Selected' and 'Remove' buttons. The 'WMI Class Information' section shows the 'Attribute group name' as 'Win32_LogicalDisk' and the 'Help' text as 'Data gathered from WMI class Win32_LogicalDisk.'. There are two radio buttons at the bottom: one for 'Produces a single data row' and one for 'Can produce more than one data row'.

27. Expand **Win32_LogicalDisk (WMI)**.



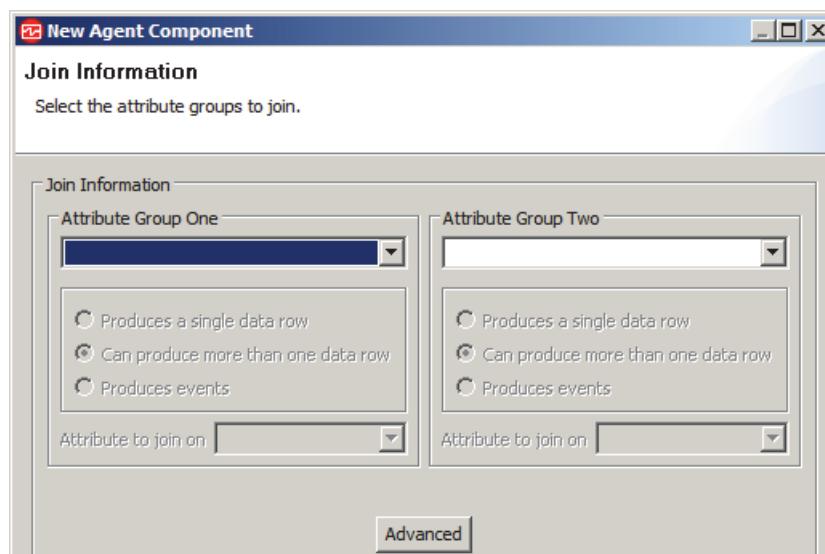
28. Combine the **LogicalDisk** data from Perfmon and the **Win32_LogicalDisk** data from WMI into a new attribute group called **LogicalDiskCombined**.

- Right-click **Disk_Storage (Navigator Group)** and click **Add Data Source**.
- Select **Existing data sources** under **Monitoring Data Categories** and ensure that **Join two attribute groups** is selected.

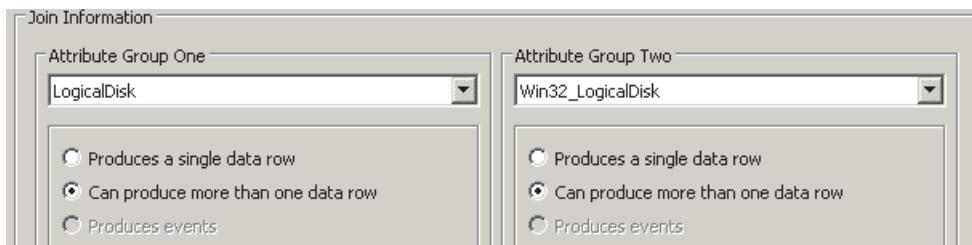


- Click **Next**.

The Join Information window opens.



- d. Select **LogicalDisk** for **Attribute Group One** and **Win32_LogicalDisk** for **Attribute Group Two**.



- e. Browse the attribute names under both **Attribute to join on** menus that look for common attributes. Notice that both have the attribute **Name**.

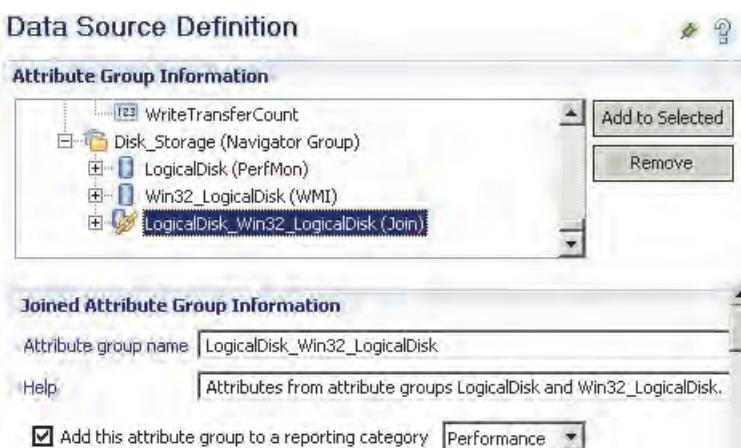
To join two data sources, you must ensure that all attribute names are unique across both data sources. You change the name of one of these attributes later in this exercise.

- f. Select **Name** as the attribute to join on for both attribute groups.



- g. Click **Finish** to create the attribute group.

The **Data Source Definition** tab of the Agent Editor is displayed.



- h. Change the attribute group name **LogicalDisk_Win32_LogicalDisk** to **LogicalDiskCombined**.

Data Source Definition

Attribute Group Information

	<input type="checkbox"/> SupportsDiskQuotas <input type="checkbox"/> SupportsFileBasedCompression <input type="checkbox"/> SystemCreationClassName <input type="checkbox"/> SystemName <input type="checkbox"/> VolumeDirty <input type="checkbox"/> VolumeName <input type="checkbox"/> VolumeSerialNumber <input checked="" type="checkbox"/> LogicalDiskCombined (Join)
--	---

Join Attribute Group Information

Attribute group name:	LogicalDiskCombined
Help:	Attributes from attribute groups LogicalDisk and Win32_LogicalDisk.

- Expand **LogicalDiskCombined** and notice that it has the combined attributes.

Notice that **Name** attribute from the **LogicalDisk** attribute group is changed to **Name_LogicalDisk**.



Agent Builder does this change automatically because an attribute group cannot have two attributes of the same name. This change did not occur because you selected the two attributes for the join, but because the two attributes have the same value: **Name**. This change occurs automatically in joins for all attributes with the same attribute name.

- Save your agent project.

You successfully added the disk storage information from Perfmon and WMI to your agent. To complete this agent, you must add the Windows log events for DB2 and the HTTP Server.

Add DB2 Windows log event information to the agent

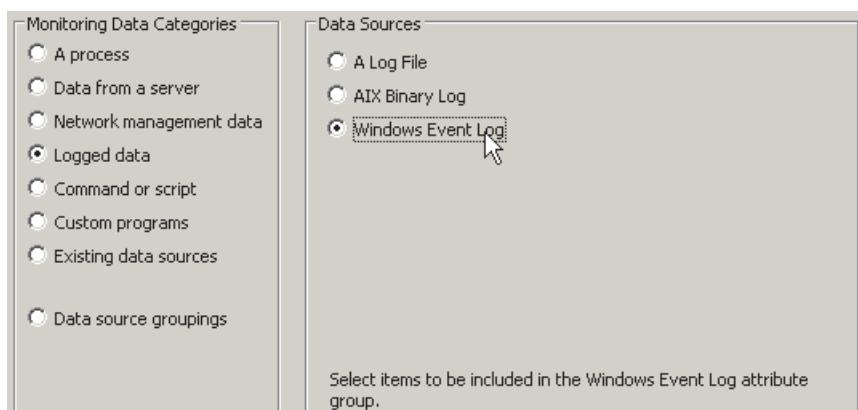
Complete the following steps to add Windows Log Event information that is filtered for DB2.

- Click **AB1 (Agent)** under **Attribute Group Information** and click **Add to Selected**.



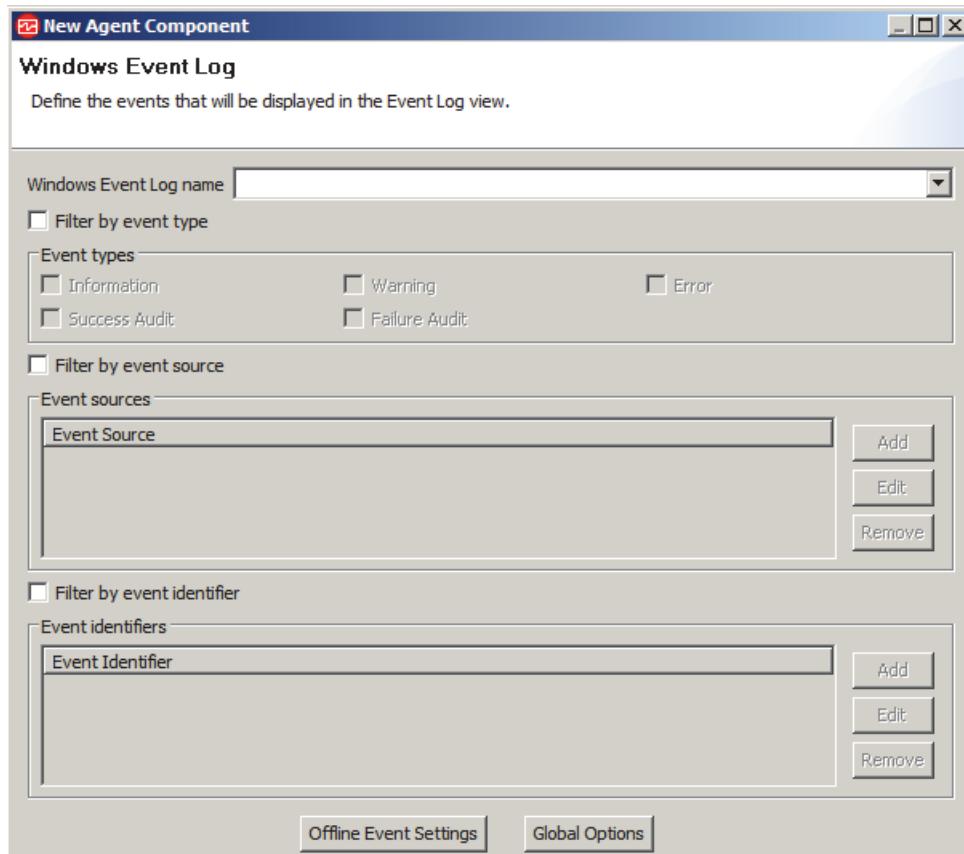
The Data Source Location window opens.

- Click **Logged Data** under **Monitoring Data Categories** and click **Windows Event Log** under **Data Sources**.



32. Click **Next**.

The Windows Event Log window opens.



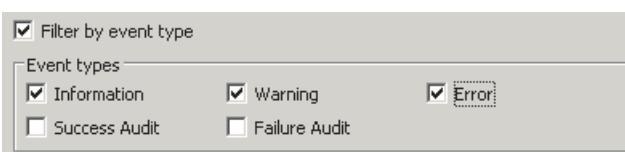
33. Create a filter that reports DB2 Information, Warning, and Error events from the Application log.

- a. Select **Application** from the **Windows Event Log name** list.



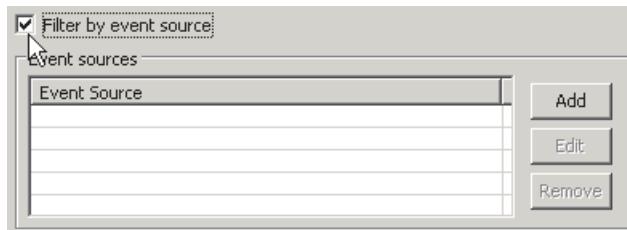
You must specify which event types the agent retrieves. Limit yours to Information, Warning, and Error.

- b. Select **Filter by event type** and select **Information, Warning, and Error**.



Your agent returns all Information, Warning, and Error events from the Application log. Now you need to limit the events to just DB2.

c. Select **Filter by event source**.



From this pane, you can add one or more event sources.

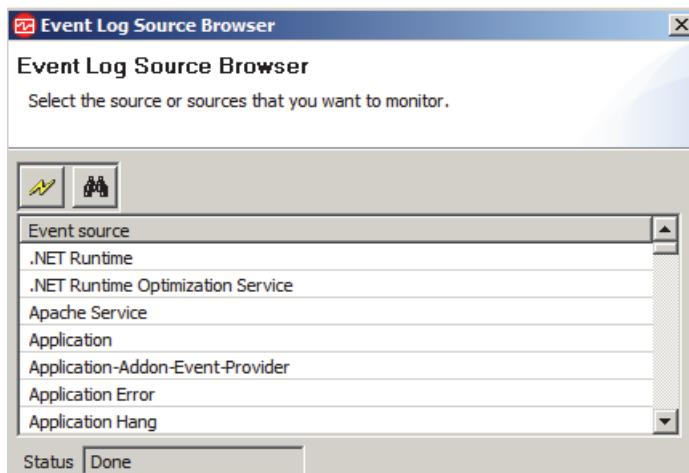
d. Click **Add**.

The Event Source window opens.



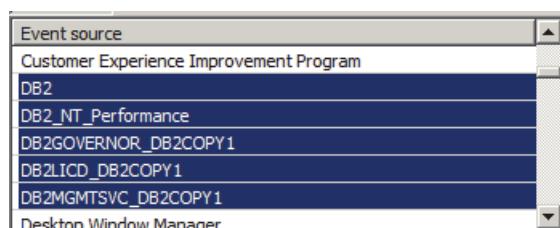
e. Click **Browse**.

The Event Log Source Browser opens with a list of event sources that are found on this host.



There is no connection to remote hosts with this browser. You can select one or more event sources.

f. Select all event sources that start with **DB2**.

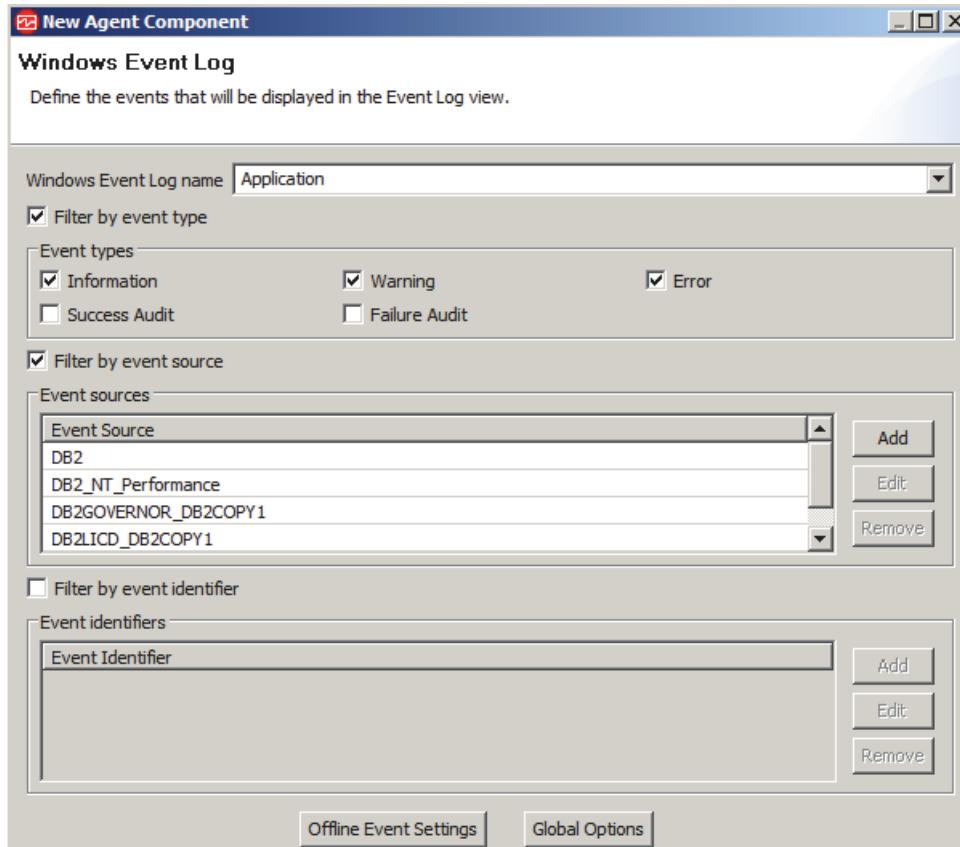




Hint: Press Shift and click to select contiguous items. Press CTRL and click to select noncontiguous items.

- g. Click **OK** to save your event source selections.

Your final configuration should look like the following example.



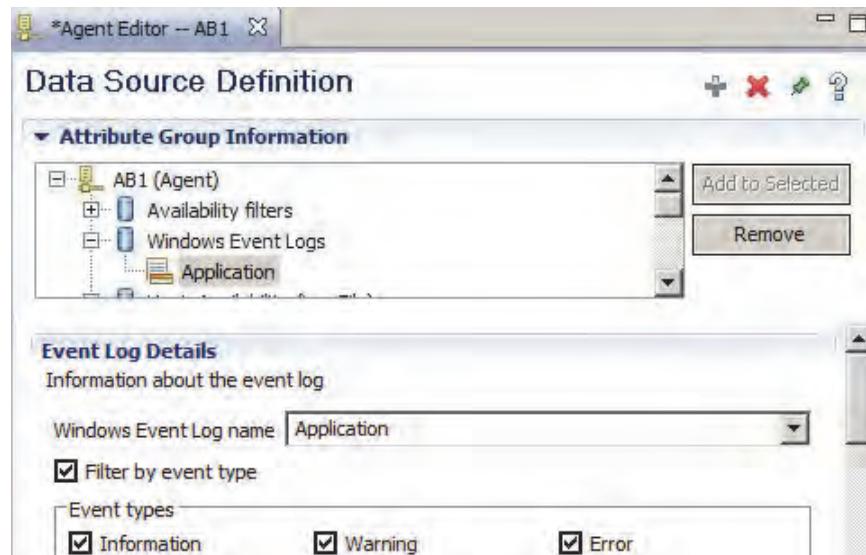
34. Configure the data source to collect offline events.

- a. Click **Offline Event Settings**.



- ◆ **Do not collect any offline events:** Events that are generated while the agent is shut down are not processed. This option is the default option.

- ◆ **Collect all offline objects:** All events that are generated while the agent is shut down are processed.
 - ◆ **Specify custom collection settings:** You can enter a value to throttle the processing of old events that are based on a time value, or number of events, or both. By using this option, you ensure that the management server is not overloaded with events when the agent starts.
- b. Click **Collect all offline events**.
 - c. Click **OK** to save your changes and close the bookmark settings window.
35. Click **Finish** to close the Windows Event Log window.
The Agent Editor **Data Source** tab is displayed.
36. Locate and expand the **Windows Event Logs** attribute group and confirm the single Application attribute.



37. Save your agent project.

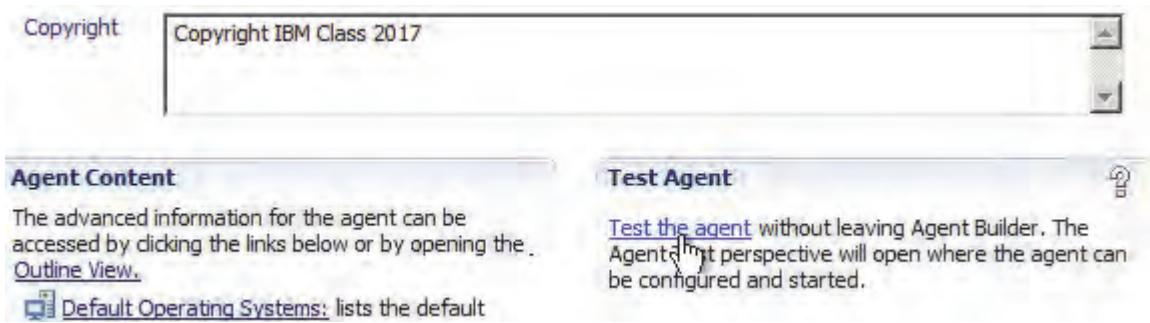
You added Windows Event Log filters for DB2 and HTTP server events to your AB1 agent. You also completed all the agent modifications that are part of this scenario. You now install and test your agent.

Test the full agent in Agent Builder

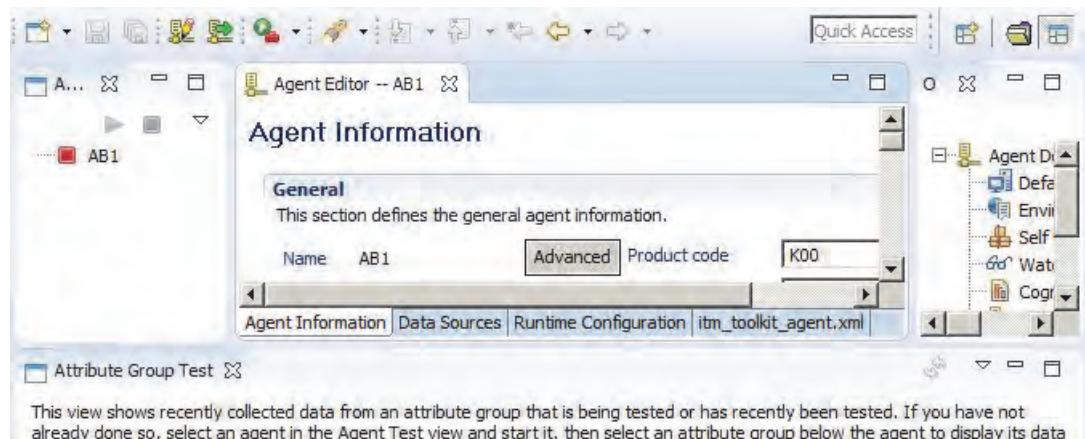
In this section, you test your full agent while in Agent Builder and confirm that all data sources retrieve data.

38. Click the **Agent Information** tab.

39. Click **Test the agent**.



The Agent Test perspective opens.



The Agent Test perspective replaces the Project Explorer window with the Agent Test window and the Problems window with the Attribute Group Test window. It keeps the Agent Editor and Outline windows.

40. Move between the two perspectives.

- Click the **Agent Definition** perspective button on the upper right to return to that perspective.



The Project Explorer and Problems windows replace the Agent Test and Attribute Group Test windows.

- Click the **Agent Test** perspective button on the upper right to return to that perspective.



The Agent Test and Attribute Group Test windows reopen. You can keep the Agent Test perspective open while you edit the agent definition.

- c. With the Agent Test perspective open, click the **Data Sources** tab at the bottom of the **Agent Editor** window.



The Data Sources page opens in the Agent Editor, while the Agent Test and Attribute Group Test windows remain visible. In this way, you can edit the agent definition while keeping the Agent Test and Attribute Group Test windows visible. This perspective can be helpful when defining agent components based on data gathered by the agent.

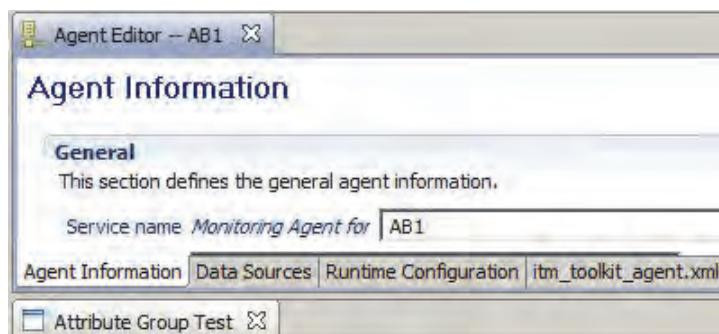
If you modify the Agent Test perspective, such as closing the Agent Test or Attribute Group Test windows, you can restore the perspective to its default layout.

41. Modify and reset the Agent Test perspective.

- a. Click the close icon on the Agent Test window.

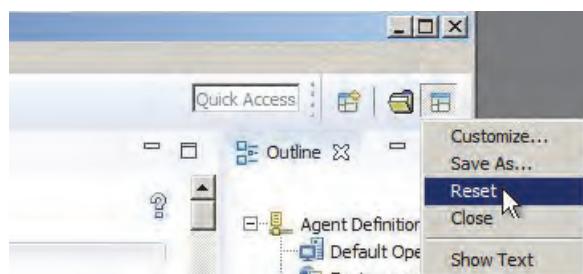


The Agent Test window closes.



In this step, you modified the Agent Test perspective by removing the Agent Test window. Any time that you open this perspective, it will not include the Agent Test window.

- b. Right-click the **Agent Test perspective** button and click **Reset**.

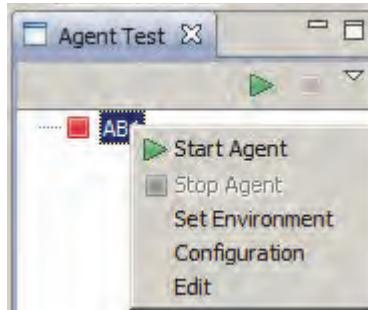


- c. Click **Yes** to confirm resetting the perspective.

The perspective is reset to its default layout and the Agent Test window reopens. Now any time you open this perspective, it will be in its default configuration.

Return to testing your data sources.

42. Right-click **AB1** in the Agent Test window and browse the menu options.



From this menu, you can do the following actions:

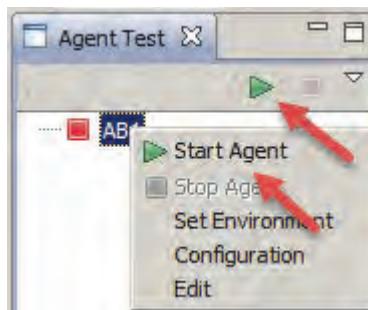
- Set environment variables the agent needs.
- Configure the agent and enter any needed runtime properties.
- Edit the agent.
- Start the agent and test all data sources.



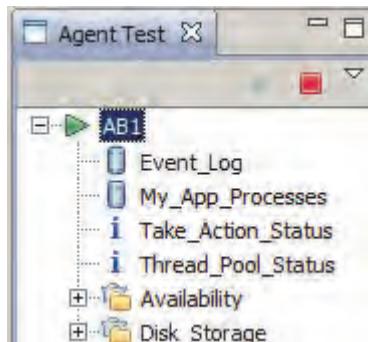
Note: Edits that are made to a running agent are not visible until the agent is restarted.

This agent does not need any environment variables or configuration properties.

43. Start the agent by either clicking **Start Agent** from the menu or the **Start Agent icon** () in the toolbar.



A window indicates that the agent is starting. When the agent starts, its attribute groups are shown as child processes of the agent in the Agent Test view.



The attribute group icon (⌚) the attribute groups.

The status attribute groups that give information about the agent (Performance Object Status, Thread Pool Status, and Take Action Status) are also shown as child processes of the agent in the Agent Test view. The information icon indicates the status attribute groups.

The stop agent icon becomes available when the agent is started.

If your agent has subnodes or navigator groups, they are shown as nodes in the Agent Test view. Subnode definitions are shown under the agent. A subnode instance node is shown under the subnode definition node. Attribute groups and navigator groups are shown under the subnode instance node.



Hint: You can start and run more than one agent at the same time.

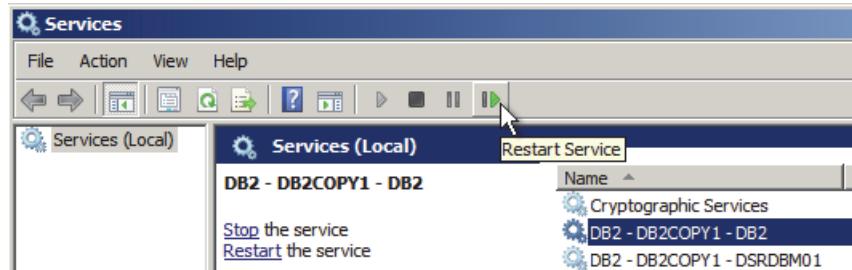
44. Confirm the **Event_Log** data source.

a. Click **Event_Log**.

Log_Name	Event_Source	Event_Type	Event_ID

The initial Attribute Group Test view does not contain data, but you see valid column heading and no error messages.

- To generate some events, restart the **DB2 - DB2COPY1 - DB2** service in the Windows Services utility.



- Refresh the Attribute Group Test view by clicking **AB1** and clicking **Event_Log** again.



Hint: It might take a minute for data to show. Repeat this step until it does.

Attribute Group Test					
7 data rows returned at Jul 11, 2015 10:44:45 AM.					
Log_Name	Event_Source	Event_Type	Event_ID	Event_Category	Message
Application	DB2	Informational	1	None	DB2STOP : SQL
Application	DB2	Warning	5	None	2015-07-11-10
Application	DB2	Informational	1	None	db2start : SQL1

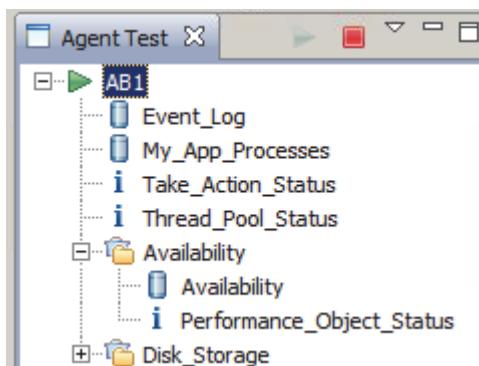
45. Confirm the **My App Processes** data source.

- Click **My App Processes**.

Caption	CommandLine
System Idle Process	
System	
smss.exe	%SystemRoot%\System32\smss.exe
csrss.exe	%SystemRoot%\System32\csrss.exe

The **My App Processes** attribute group pulls process information from WMI, but it is not filtered for the HTTP server and DB2. You might want to edit the view to filter out everything except DB2 and HTTP server process information.

- b. Scroll to the right to see all of the data that is collected by each process. As with all agents that are created with Agent Builder, the default workspace is a single view with all of the data that is displayed in a table.
46. Confirm the Availability data sources.
- a. Expand **Availability** in the Agent Test window.



An Availability attribute group and a Performance_Object_Status node are displayed below the Availability node.

- b. Click the **Availability** attribute group.

Attribute Group Test			
7 data rows returned at Feb 24, 2013 11:23:41 AM.			
Application_Component	Name	Status	Full_Name
DB2 - DB2COPY1 - DB2	DB2	UP	DB2 - DB2COPY1 - DB2
DB2DAS - DB2DAS00	DB2DAS00	UP	DB2DAS - DB2DAS00
DB2 Governor (DB2COPY1)	DB2GOVERNOR_DB2COPY1	DOWN	DB2 Governor (DB2COPY1)
DB2 License Server (DB2COPY1)	DB2LICD_DB2COPY1	DOWN	DB2 License Server (DB2COPY1)
DB2 Management Service (DB2COPY1)	DB2MGMTSVC_DB2COPY1	UP	DB2 Management Service (DB2COPY1)
DB2 Remote Command Server (DB2COPY1)	DB2REMOTECMD_DB2COPY1	UP	DB2 Remote Command Server (DB2COPY1)
HTTP Server	IHSforTivoliEnterprisePortalServer	UP	IHS for Tivoli Enterprise Portal Server

The Attribute Group Test view opens with the Windows Service availability data you added to the agent in a previous exercise.

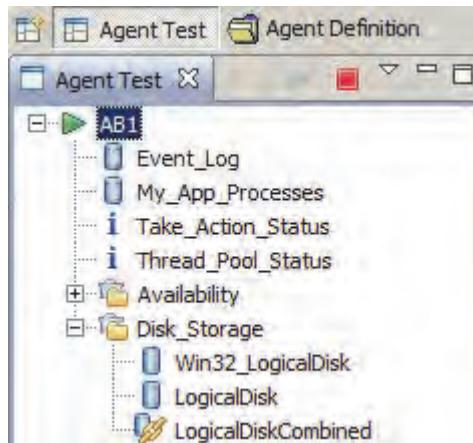
- c. Click **Performance_Object_Status**.

Attribute Group Test				
5 data rows returned at Mar 6, 2017 8:29:11 PM.				
Query_Name	Object_Name	Object_Type	Object_Status	Error
Event_Log	Event Log	NT_EVENT_LOG	ACTIVE	NO_E
My_App_Processes	root\CMIV2:Win32_Process	WMI	ACTIVE	NO_E
Win32_LogicalDisk	root\CMIV2:Win32_LogicalDisk	WMI	ACTIVE	NO_E
LogicalDisk	LogicalDisk	PERFMON	ACTIVE	NO_I

This attribute group monitors the availability of the performance object data sources that you added in this exercise. An entry is added data source for each.

47. Confirm the **Disk_Storage** data sources.

- Expand **Disk_Storage** in the **Agent Test** window.



The two source data sources, **LogicalDisk** and **Win32_LogicalDisk**, and the combined data source, **LogicalDiskCombined**, are shown.

- Click **WIN32_LogicalDisk** and **LogicalDisk** under **Disk_Storage** and confirm that they retrieve data.

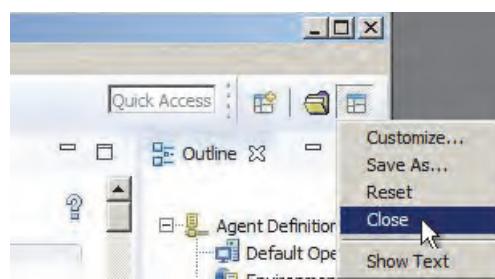


Important: Combined data sources, such as **LogicalDiskCombined**, do not show data in the full agent test utility and might stop the agent.

48. Stop the agent by clicking the **Stop Agent** icon.



49. Close the **Agent Test** perspective by right-clicking the **Agent Test** perspective icon and clicking **Close**.



You successfully tested your full agent in Agent Builder.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

50. Select to the **Agent Information** tab.

51. Change the version for 1.0.0 to **1.0.1**.

Agent Information

General
This section defines the general agent information.

Name	AB1	Advanced	Product c
Version	1.0.1	Company	
Fix pack	0	Patch level	0
		Agent ide	



Important: If your initial version number is not 1.0.0, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

52. Save your agent project.

Exercise 2 Install and confirm the updated AB1 agent in an IBM Performance Management environment

In this section, you reinstall your AB1 agent on the WIN1 server, which is part of an IPM environment. Then you confirm that data is being gathered for the new attribute groups you added.

Update the dashboard and resource definitions

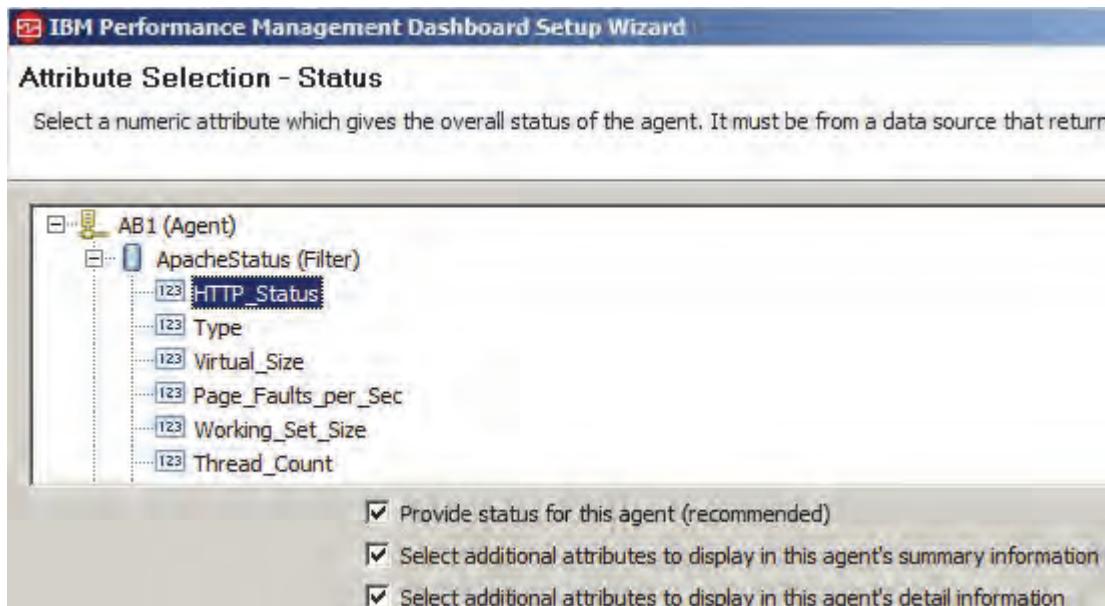
In the previous exercise, you defined your dashboards and OSLC resources. In this section, you review those selections and modify them as needed.

1. Select **Dashboards** from the Agent Builder Outline view.

The Dashboards Overview opens.

2. Click the **Dashboard Setup wizard** link in the Overview.

The IBM Performance Management Dashboard Setup wizard opens.

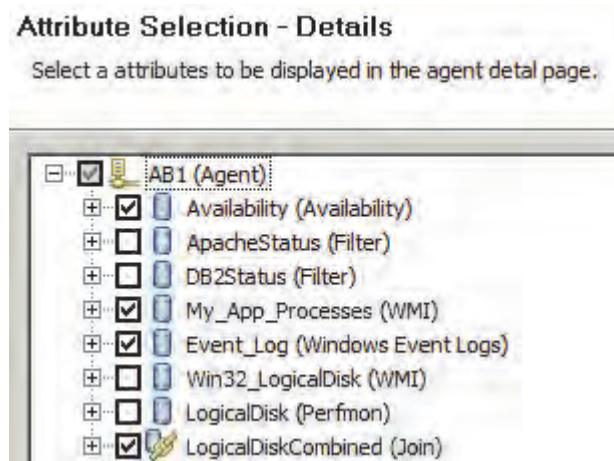


You are again prompted to select a numeric attribute that gives the overall status of the agent with **HTTP_Status** selected from a previous exercise. Although you could change this value, now you will not.

3. Click **Next**.

You are prompted to identify data points to display in the Summary dashboard. Data on the Summary dashboard must come from single-row data sources. The new data sources added to AB1 were multiple-row; so they cannot be displayed in the Summary dashboard.

4. Confirm that the following attributes are still selected for the Summary dashboard:
 - Apache_Status > Percent_Processor_Time
 - DB2_Status > Name
 - DB2_Status > Percent_Processor_Time
5. Click **Next**.
You are prompted to identify data groups and items to display in the details dashboards. Notice that your new data sources are listed.
6. Select the following data groups:
 - My_App_Processes
 - Event_Log
 - LogicalDiskCombined



You do not have to change any other dashboard or resource properties.

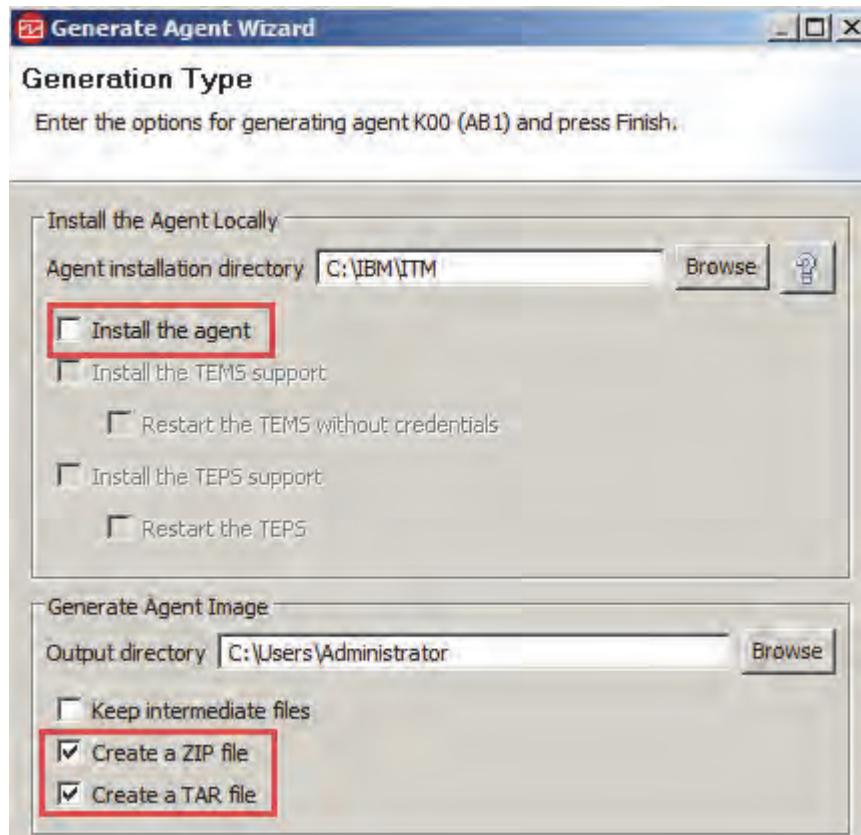
7. Keep the click **Finish** to complete the wizard.
8. Save your agent project.

You successfully defined the summary dashboard, detailed dashboard, and the monitored resource data with OSLC.

Create the installation files

9. Select **Agent Editor > Generate Agent** from the main menu.
10. Verify that **Install the agent** is not selected.

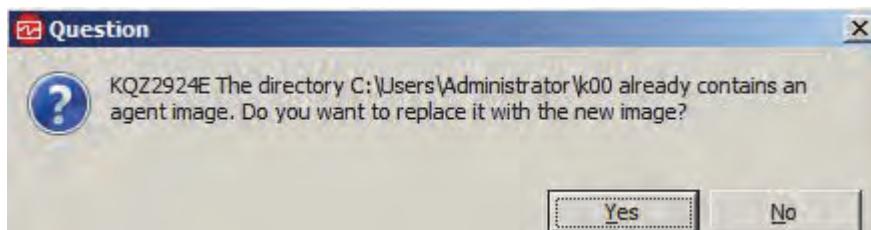
11. Confirm that **Create a ZIP file** and **Create a TAR file** are selected.



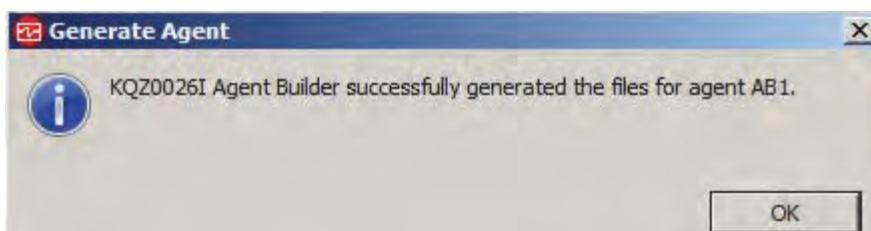
Notice the directory where the files are created. You go to this directory to access the generated files. Keep the default output directory for this exercise.

12. Click **Finish**.

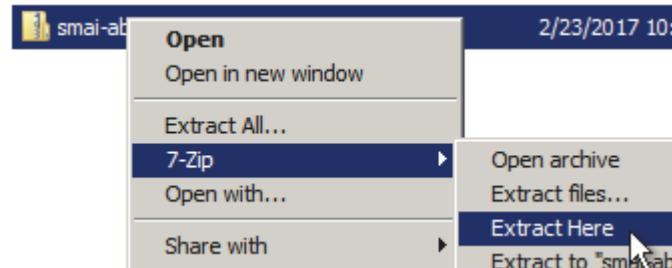
13. Click **Yes** to replace the K00 image.



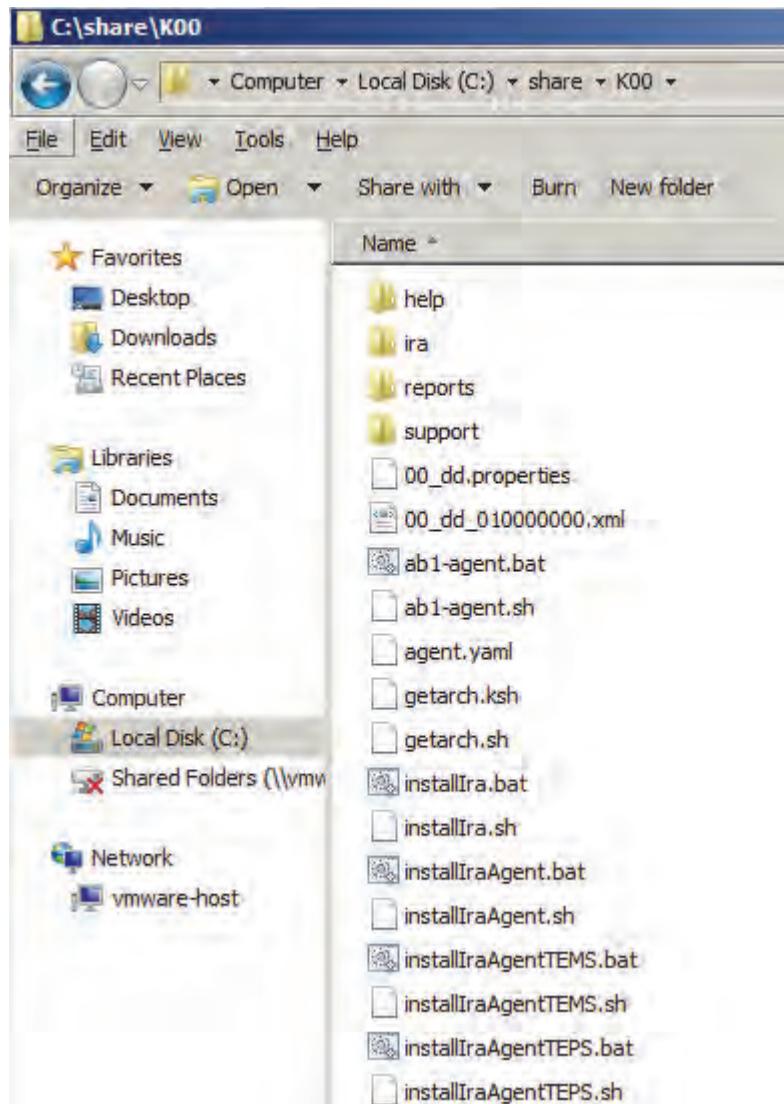
A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated.



14. Click **OK** to close the confirmation window.
15. Open a command prompt or Explorer window and go to **C:\Users\Administrator** to confirm that the **smai-ab1-01.01.00.00.tgz** and **smai-ab1-01.01.00.00.zip** files are created.
16. Delete the contents of the **C:\share\K00** directory.
17. Copy the **smai-ab1-01.01.00.00.zip** file into the **C:\share\K00** directory.
18. From Windows Explorer, right-click the **smai-ab1-01.01.00.00.zip** file and select **7-Zip > Extract Here**.



The agent installation files are extracted.



- **installIraAgent.bat/.sh** installs the agent.
- **installIraAgentTEMS.bat/.sh** installs the Tivoli Enterprise Monitoring Server support.
- **installIraAgentTEPS.bat/.sh** installs the Tivoli Enterprise Portal Server and Tivoli Enterprise Portal support.
- **installIra.bat/.sh** installs everything, including the agent and application support on the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop client.



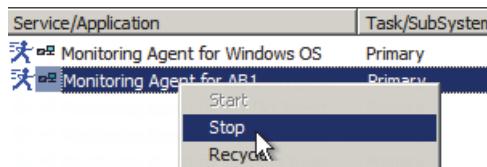
Note: You can run each script without arguments to get a usage statement.

Reinstall the agent on WIN1 with the installation scripts

In this section, you install the agent on WIN1, but first you set up the installation image on the WIN1 server.

19. On WIN1, stop the AB1 agent with *either* of the following actions:

- Open the IPM utility, right-click the agent, and select **Stop**.



- Open a command prompt and run the following command:

```
\IBM\APM\BIN\ab1-agent.bat stop
```

```
C:\Users\Administrator>\IBM\APM\BIN\ab1-agent.bat stop
Stopping Monitoring Agent for AB1 ...
Monitoring Agent for AB1 stopped.

C:\Users\Administrator>
```

20. Open a command prompt and change to the **C:\share\K00** directory.

21. Run the following command:

```
installIraAgent.bat C:\IBM\APM\
```

```
C:\share\K00>installIraAgent.bat C:\IBM\APM
Installing agent into C:\IBM\APM
Installing K00 ....
Install of K00 Agent successful.

C:\share\K00>
```

22. Confirm the agent in the IPM utility.

- a. Select **Start > All Programs > IBM Monitoring agents > IBM Performance Management (Instance1)**.
- b. Select **View > Refresh**.

Service/Application	Task/SubSystem	Configured	Status	Connection status	Configuration
Monitoring Agent for Windows OS	Primary	Yes	Started	Connected	up-to-date
Monitoring Agent for AB1	Primary	Yes	Started	Connecting	up-to-date

Your agent is visible and started.

23. Start your agent if it is not running.

Confirm the updated AB1 agent in the Performance Management Console

24. Log in to the APM server as user **root** with password **object00**.

25. Open a browser and go the following URL:

<https://apm:9443>



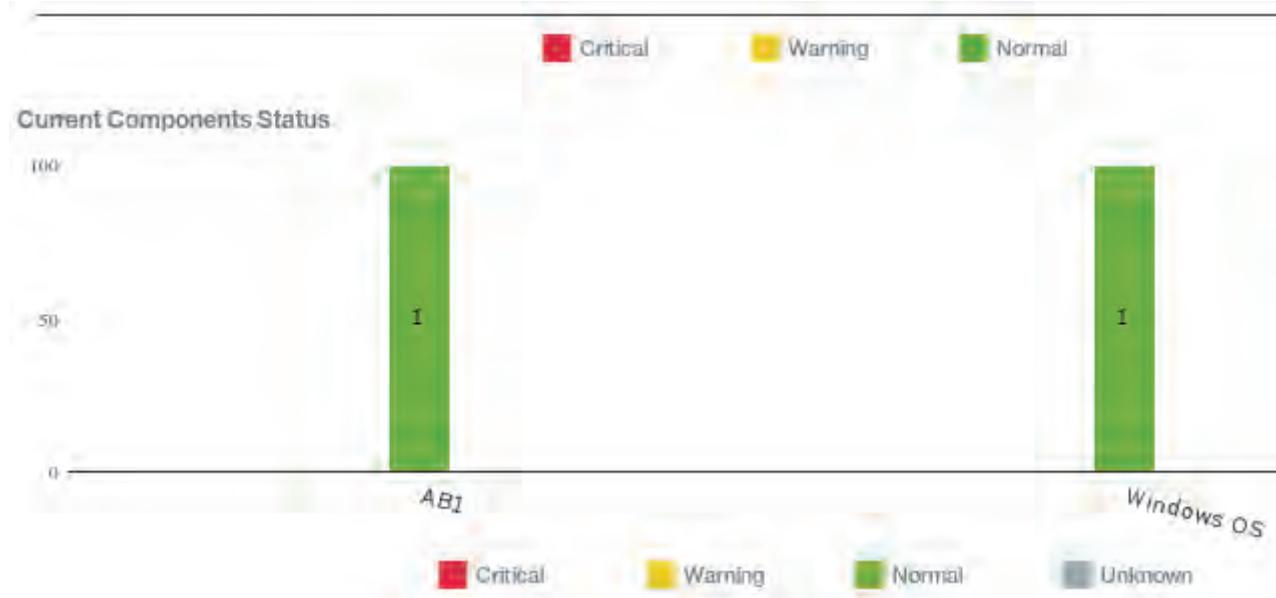
Hint: A bookmark to this URL is in the browser.

26. Log in to the PM console as user **apmadmin** with password **object00**.

27. Open the Application Performance dashboard.

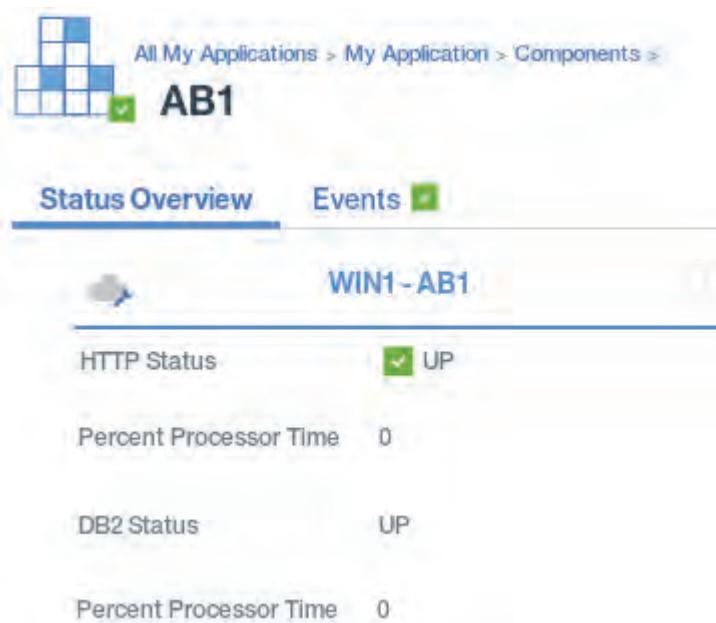
28. Click **My Application** in the Applications explorer.

The My Application Status Overview is displayed with the Current Component Status widget.



29. Click the **AB1** component bar in Current Components Status.

The AB1 Status Overview window opens.



30. Click the **WIN1 - AB1 Summary** dashboard widget to access the detailed dashboards.

31. Scroll down to see the detailed dashboards for your My_App_Processes, Event Log, and LogicalDiskCombined data sources.



Hint: It can take up to 15 minutes for the new detailed dashboards to be displayed.

My_App_Processes									
Caption	CommandLine	CreationDate	CSName						
System Idle Process		Data not available	WIN1						
System		2/23/17, 10:28 PM	WIN1						
emee.exe	\SystemRoot\System32\emee.exe	2/23/17 10:28 PM	WIN1						

LogicalDiskCombined									
Name LogicalDisk	Pct Disk ...	Pct Disk ...	Pct Disk ...	Pct Free ...	Pct Idle ...	Avg Disk ...	Avg Disk ...	Avg Disk ...	Avg Disk ...
C:	0.	0.	0.	11	99	2633	5459	9516	0.

Event_Log					
Log Name	Event Source	Event Ty...	Event ID	Event Category	Message
Application	DB2	Warning	5	None	2017-02-22-20:4
Application	DB2	Informa...	1	None	DB2STOP : SQL
Application	DR0MGMTSVC DR000PV1	Informa...	1	None	monit survmon

Hint: You must restart the DB2 service, as you did in the previous exercise to generate event log data.

You successfully confirmed your agent in the Performance Management Console. The Summary dashboard shows the core metrics that you added. A new subset of attributes is visible in the **Details dashboard**.

Exercise 2 Install and confirm the updated AB1 agent in an IBM Performance Management environment

Unit 5 Monitoring processes and command return codes exercises

In the first exercise in this unit, you create a single, multiplatform agent that can be installed on both Windows and Linux. The agent monitors related but different items on each operating system.

You add availability monitoring of the HTTP server process so that when the agent is installed on Windows, it monitors the Windows HTTP server process. But, when the agent is installed on Linux, it monitors the Linux HTTP server process.

Furthermore, the HTTP server produces a process ID (PID) file whenever it is running. Using availability monitoring of a command return code, you can create platform-specific commands that confirm the existence of this file.

Exercise 1 Monitor processes and command return codes

Your company has a large farm of web servers, which run on both Windows and Linux. You want to create a single agent that can be installed on both operating systems and can return information that indicates the availability of the HTTP server processes.

In this exercise, you complete the following tasks:

- Create an agent to monitor the availability of the following items:
 - The HTTP server process on both Windows and Linux
 - The HTTP server .pid file on Windows and Linux
- Install the agent and application support with installation scripts for both Windows and Linux.

Create an agent to monitor a process on Windows and Linux

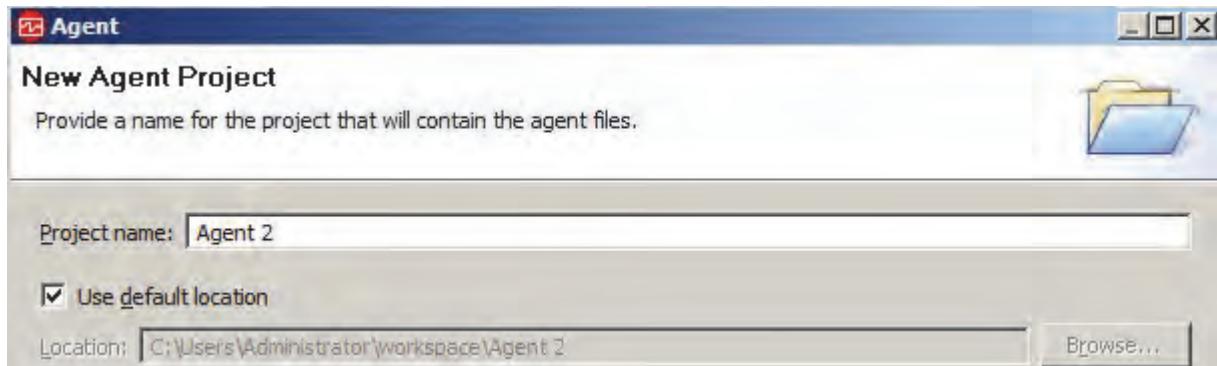
In this exercise, you create an agent to monitor the availability of the HTTP server process on both Windows and Linux.

1. On the WIN1 server, if the **Agent Builder** application is not already started, start it now.
2. Close the **AB1 Agent Editor** window if it is open.
3. Select **File > New > Agent**.

4. Click **Next**.

The New Agent Project window opens.

5. Enter **Agent 2** in the **Project name** field and click **Next**.



The General Information window opens.

6. Confirm that copyright is set to **Copyright IBM Class 2017**.



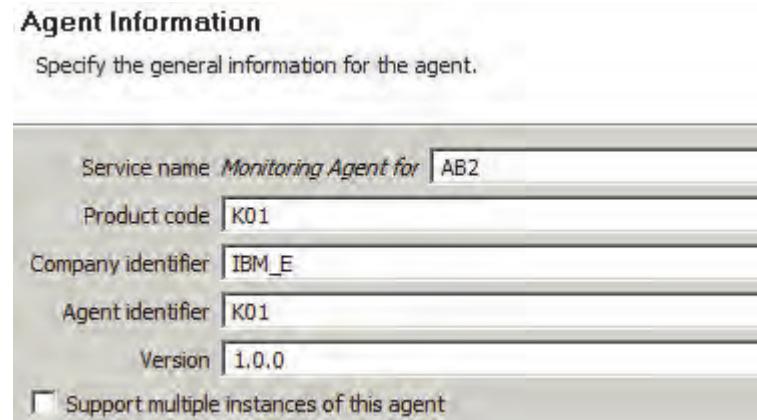
Notice the default operating systems onto which this agent is allowed to install.

7. Click **Next**.

The Agent Information window opens.

8. Change the Service name to **AB2**.

9. Confirm the following agent information:
 - Product code: **K01**
 - Company identifier: **IBM_E**
 - Agent identifier: **K01**
 - Version: **1.0.0**

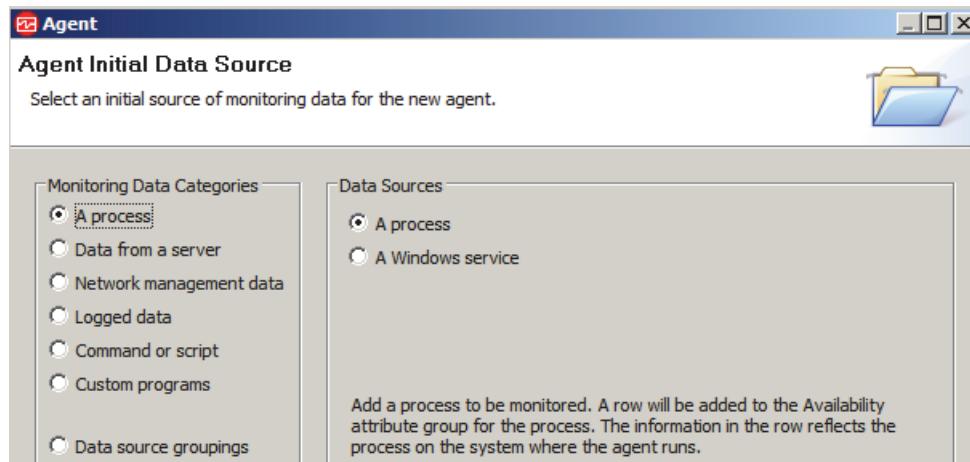


10. Click **Next**.

The Agent Initial Data Source window opens.

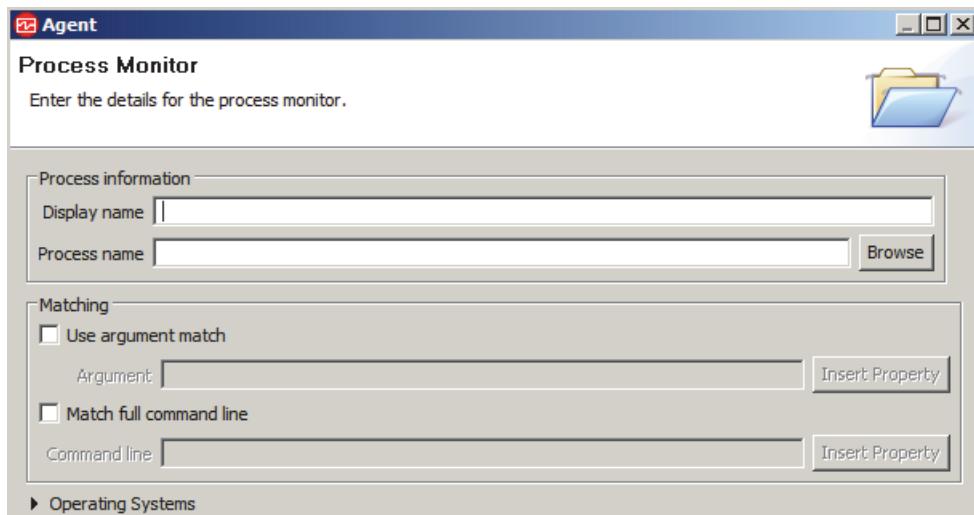
Create the HTTP server monitor for Windows

11. In the Agent wizard, select **A process** under both **Monitoring Data Categories** and **Data Sources**.



12. Click **Next**.

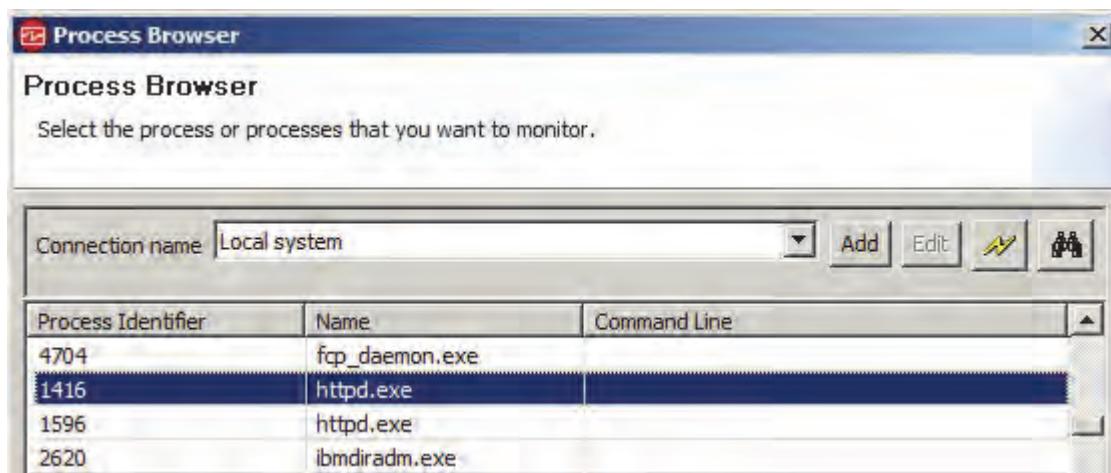
The Process Monitor window opens.



13. Enter **HTTP_Win** in the **Display name** field.



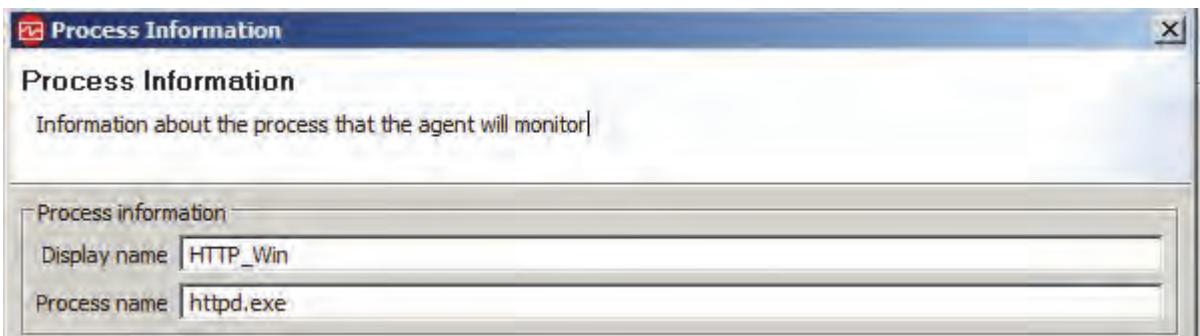
14. Click **Browse** to the right of the **Process name** field.
15. Click the **Name** column to sort by process name.
16. Locate and click one of the **httpd.exe** process rows and click **OK**.



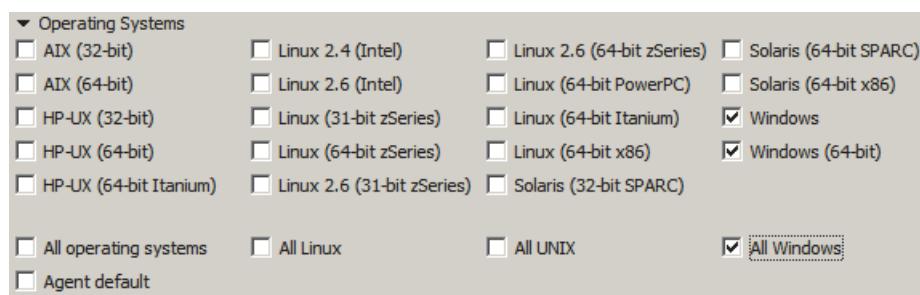
Hint: Click the Name column header to sort by the process name.

You return to the Process Monitor window.

17. Verify that the process name, **httpd.exe**, is entered.



18. Expand **Operating Systems**, clear **Agent default**, and select **All Windows**.

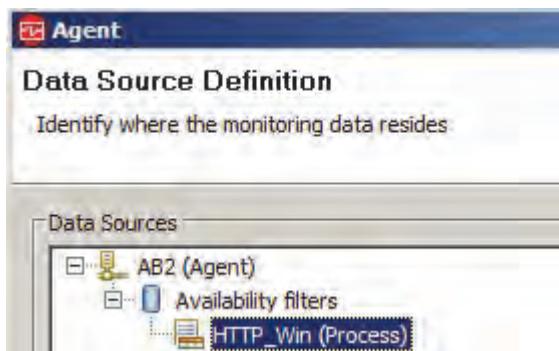


This step causes the agent to monitor only the httpd.exe process when the agent is install on a Windows system.

19. Click **Next**.

The Data Source Definition window opens.

20. If necessary, expand **Availability filters**.



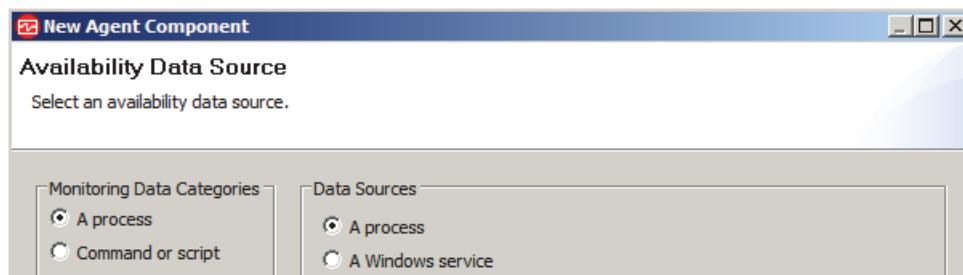
The HTTP_Win availability filter that you created is displayed.

Add the Linux HTTP server process

21. Select **Availability filters** and click **Add to Selected**.



The Availability Data Source window opens.



22. Ensure that **A process** is selected under both **Monitoring Data Categories** and **Data Sources**.

23. Click **Next**.

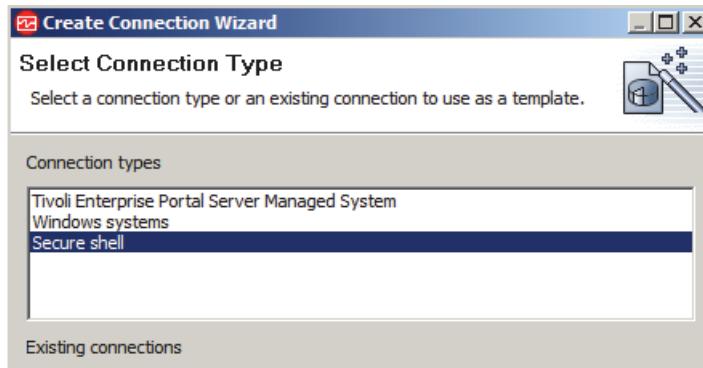
24. Enter **HTTP_Lin** in the **Display name** field.



25. Click **Browse** to the right of the **Process name** field.

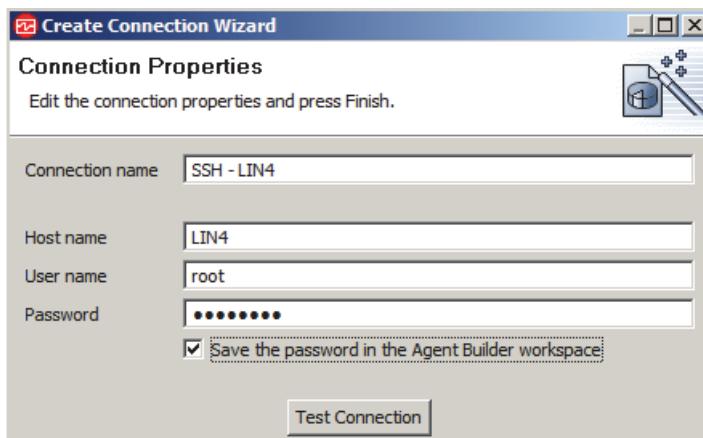
26. Click **Add** to add a connection.

27. Select **Secure Shell** and click **Next**.



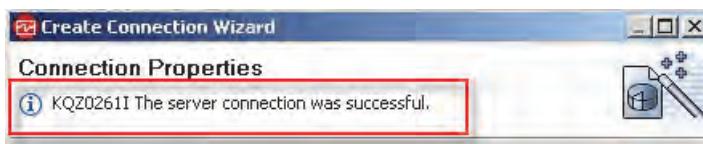
28. Enter the following values:

- Host name: **LIN4**
- User name: **root**
- Password: **object00**
- Save the password: **Checked**



29. Click **Test Connection**.

30. Confirm that the connection was successful.



31. Click **Finish**.

You are returned to the Service Browser. The services currently running on LIN4 are displayed.

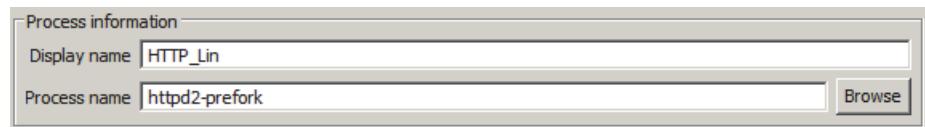
32. Locate and click one of the **httpd2-prefork** (Apache HTTP Server) process rows and click **OK**.

Process Identifier	Name	Command Line
1837	haldd-runner	haldd-runner
5945	haveged	/sbin/haveged -w 1024 -v 1
6718	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...
6719	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...
6720	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...
6721	httpd2-prefork	/usr/sbin/httpd2-prefork -f /etc/apache2/htt...

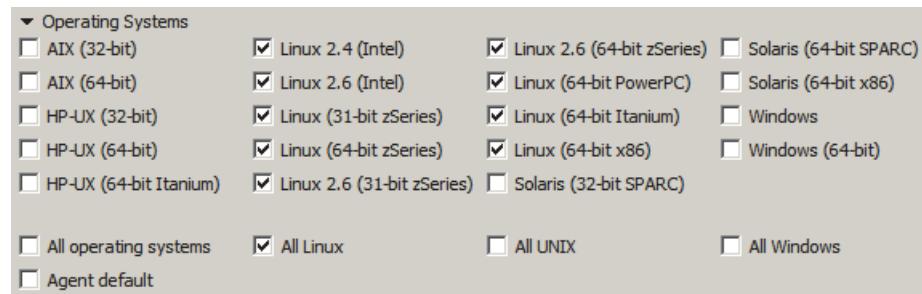


Hint: Click the Name column header to sort by the process name.

The Process Monitor window is displayed. Notice that the process name is listed.



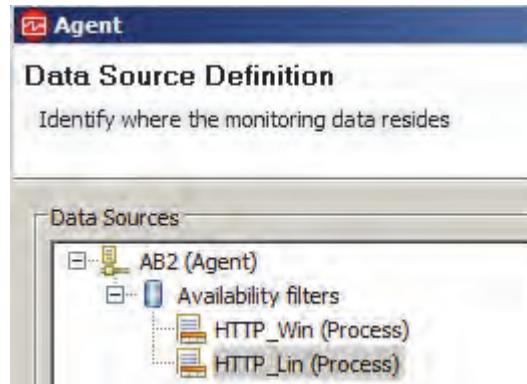
33. Expand **Operating Systems**, clear **Agent default**, and select **All Linux**.



This step causes the agent to monitor only the httpd2-prefork process when the agent is installed on a Linux system.

34. Click **Finish** to close the **Process Monitor** window.

Your two configured processes are displayed in the Data Source Definition window.

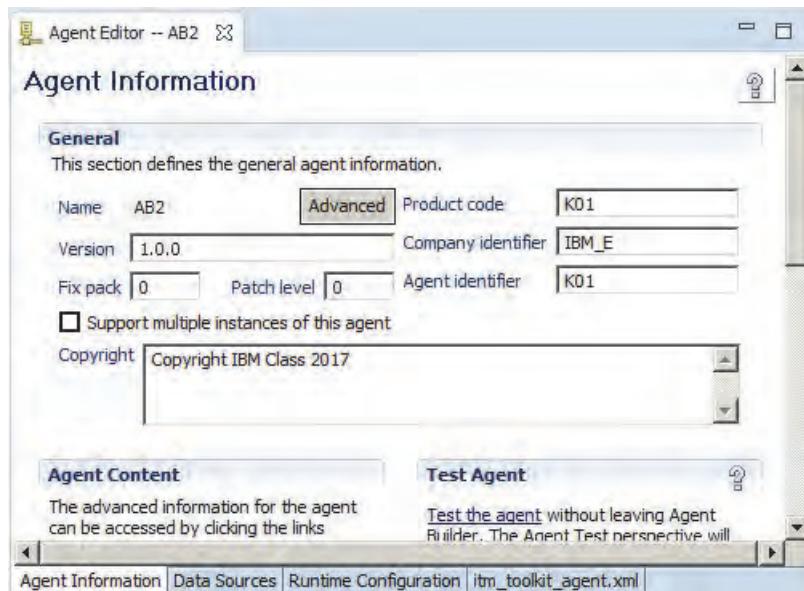


35. Click **Finish** to close the Data Source Definition window and save your changes.



Note: You might continue with adding the next data source without exiting the Agent wizard. Clicking **Finish** is a precaution to save your current work.

Your AB2 agent information is displayed in the Agent Builder interface.



36. Explore the **Agent Information**, **Data Sources**, and **itm_toolkit.xml** tabs.

Create an agent to monitor a functionality test

In this section, you add a command return code functionality test to your agent. You add a command that is run on the agent's host and then monitor the return codes from that command.



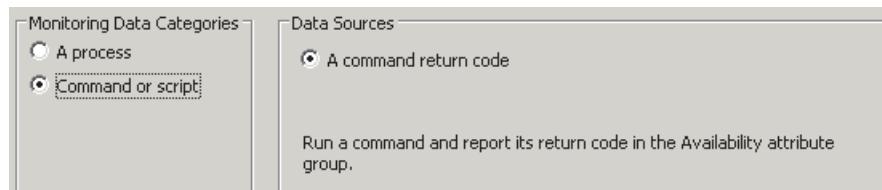
Hint: Save your agent periodically.

37. Click the **Data Sources** tab.

38. Right-click **Availability filters** and click **Add Availability Filter**.

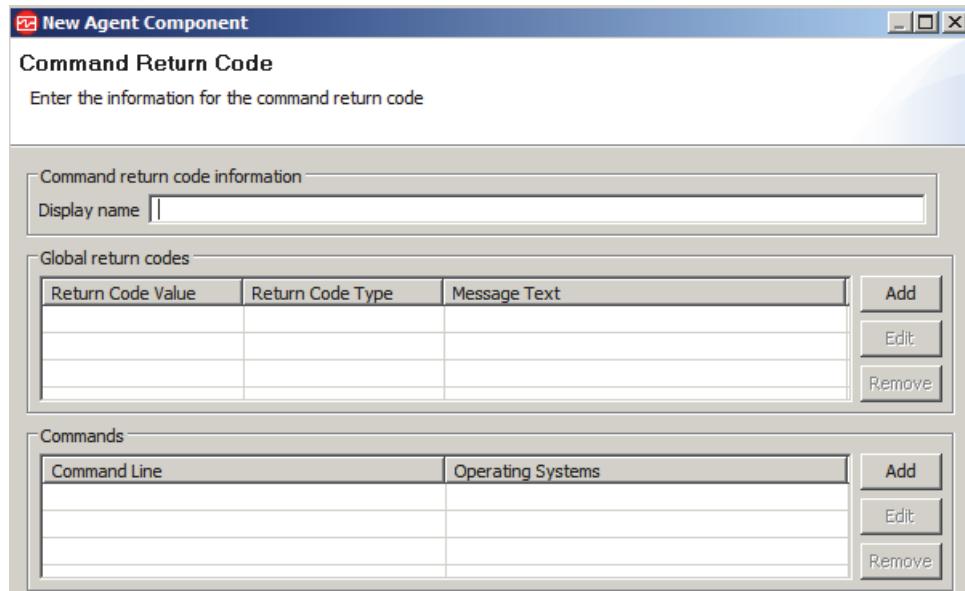
The Data Source Location window opens.

39. Click **Command or script** under **Monitoring Data Categories** and ensure that **A command return code** is selected under **Data Sources**.



40. Click **Next**.

The Command Return Code window opens.



41. Enter **HTTP PID File** in the **Display name** field.

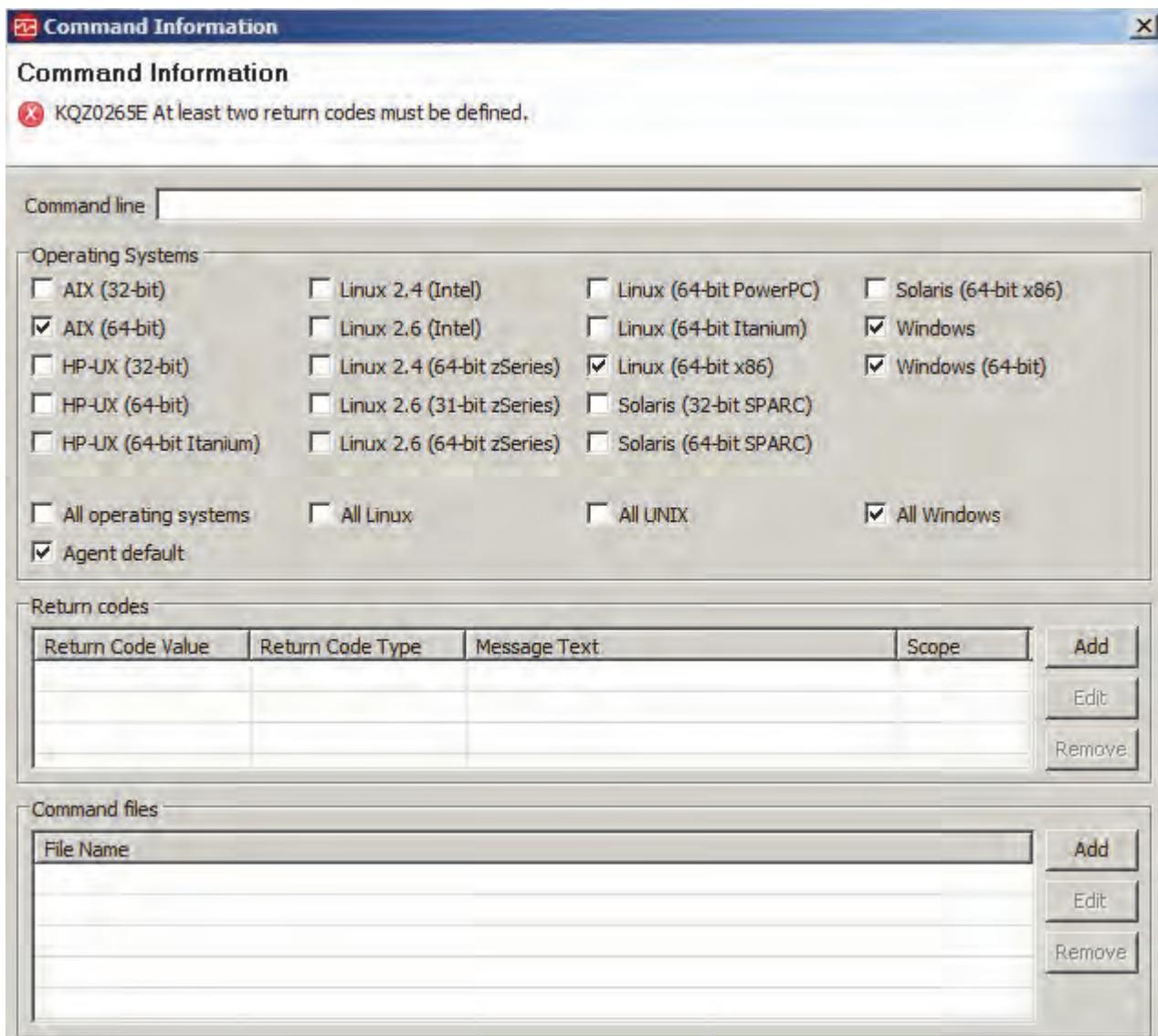
Command return code information	
Display name	HTTP PID File

42. Create the following Windows command.

Commands	
Command Line	Operating Systems
cmd.exe /c dir "C:\Apache24\logs\httpd.pid"	Windows, Windows (64-bit)

a. Click **Add** in the **Commands** sections.

The Command Information window opens.



b. Enter the Windows command in the **Command line** field.

cmd.exe /c dir "C:\Apache24\logs\httpd.pid"

Command line cmd.exe /c dir "C:\Apache24\logs\httpd.pid"



Note: Consider testing the command in a command prompt before adding it to the agent. The command must successfully display the **httpd.pid** file.

```
C:\ Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cmd.exe /c dir "C:\Apache24\logs\httpd.pid"
Volume in drive C has no label.
Volume Serial Number is BC46-ADCF

Directory of C:\Apache24\logs

05/04/2016  04:10 PM                6 httpd.pid
               1 File(s)           6 bytes
               0 Dir(s)   5,986,361,344 bytes free

C:\Users\Administrator>
```

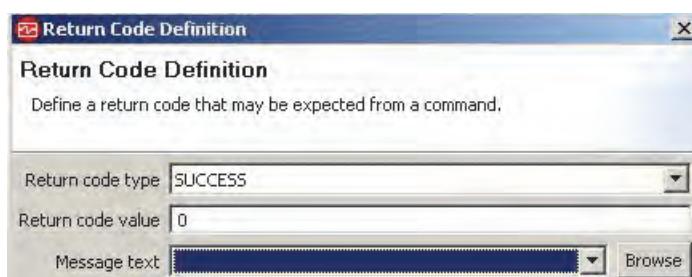
- c. Clear **Agent default**, and select **All Windows**.

43. Create the return codes for this Windows command, as shown in the following image:

Return codes		
Return Code Value	Return Code Type	Message Text
0	OK	PID file exists
1	NOT_RUNNING	PID file does not exist

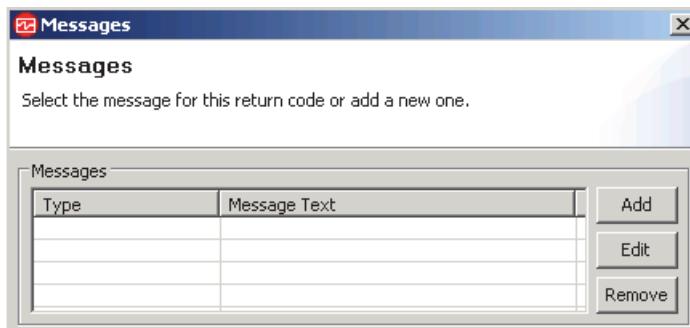
Add | Edit | Remove

- a. Click **Add** next in the Return codes section.
The Return Code Definition window opens.
- b. Select **Success** from the **Return code Type** menu and enter **0** in the **Return code value** field.



- c. Click **Browse** next to the **Message text** menu.

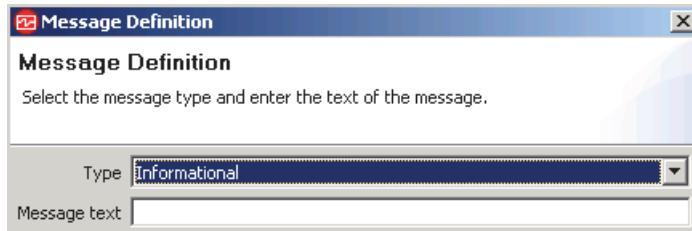
The Messages window opens.



No messages are defined in this agent now. You create two messages: one for when the PID file exists and one for when it does not exist.

- d. Click **Add**.

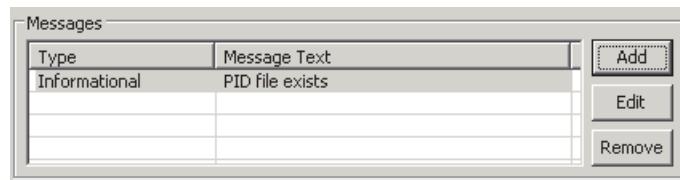
The Message Definition window opens.



- e. Select **Informational** from the **Type** menu and enter **PID file exists** in the **Message text** field. Click **OK**.



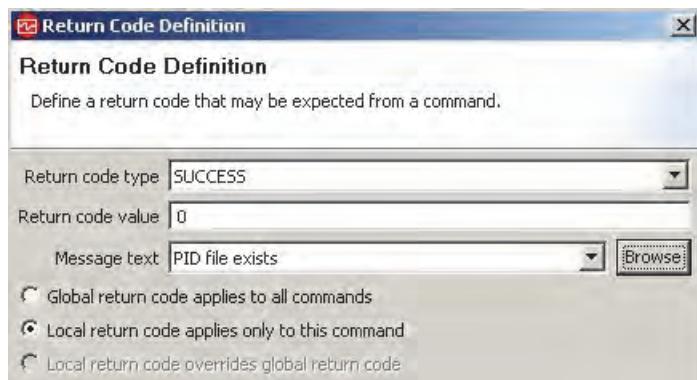
Your new message is displayed in the Messages window.



You can add the other message now, but for this exercise, you create the second message when you create the second return code.

- f. Ensure that the **PID File exists** message is selected and click **OK**.

The Return Code Definition window is displayed.



Notice that this return code is defined locally and applies only to this command. From this menu, you can define the return code globally so that other commands might use it. Because this code also means success because the PID file exists, you can use it for the Linux command when you define it globally.

- g. Click **Global return code applies to all commands**.



Important: When you repeat the previous step for the second command return code, keep that return code defined locally.

- h. Click **OK** to save your return code definition.

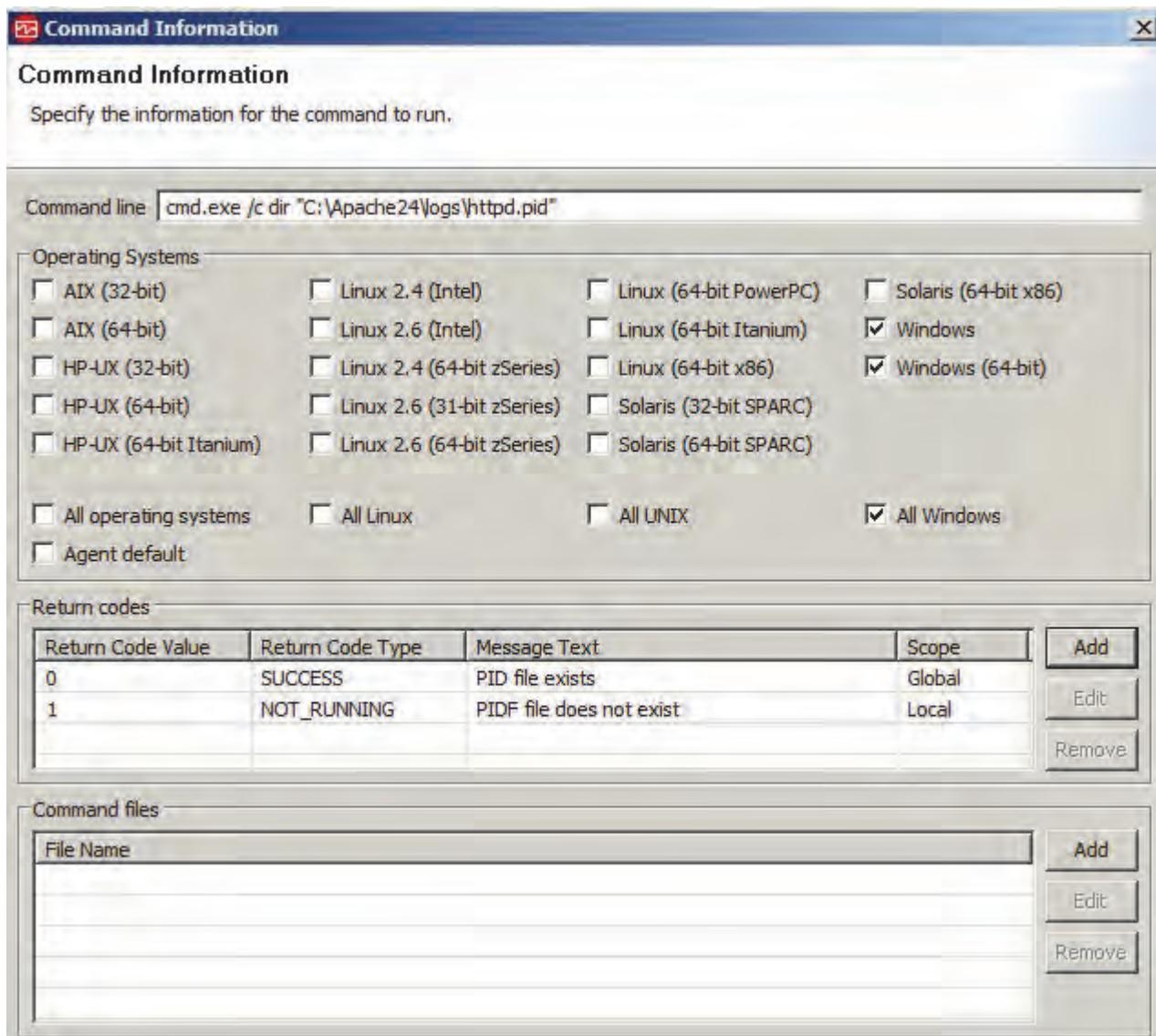
The Command Return Code window is displayed.

Return codes			
Return Code Value	Return Code Type	Message Text	Scope
0	SUCCESS	PID file exists	Global

Add
Edit
Remove

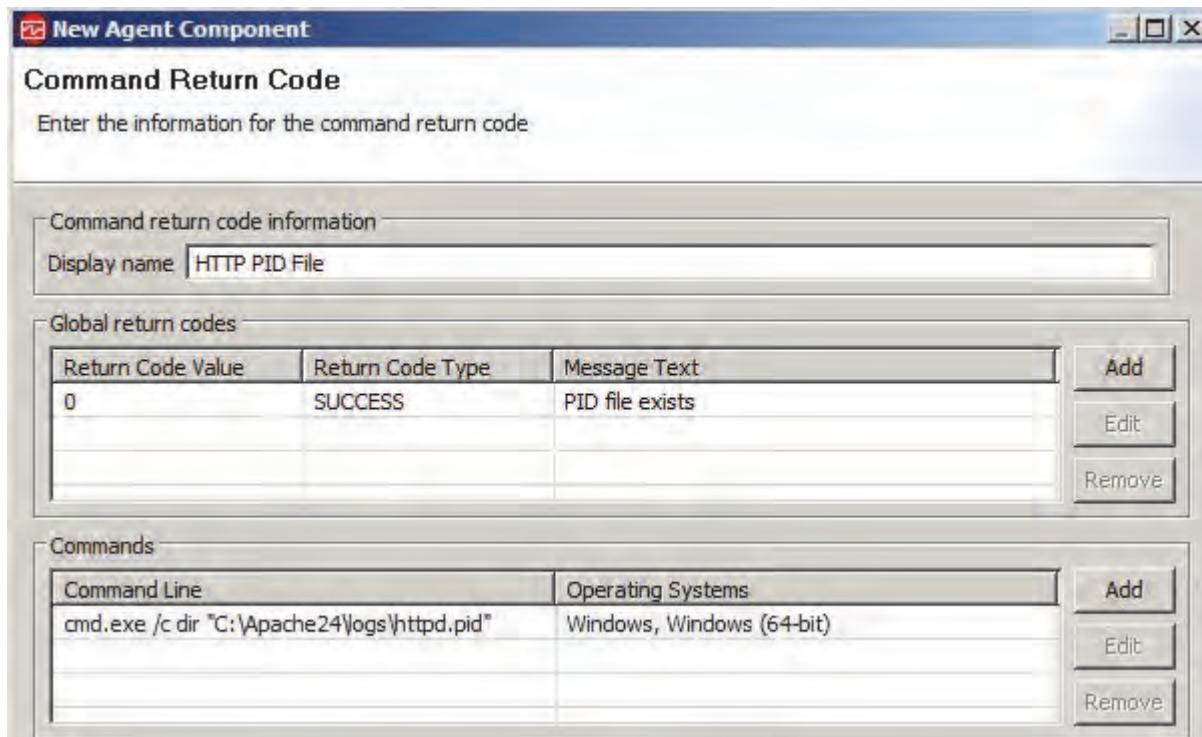
- i. Repeat these steps to create the second return code definition according to the following specifications:
- ◆ Return Code type: **NOT_RUNNING**
 - ◆ Return code: **1**
 - ◆ Message Type: **Error**
 - ◆ Message Text: **PID file does not exist**
 - ◆ Local return code applies only to this command

Your final Command Information window for this command looks like the following example:

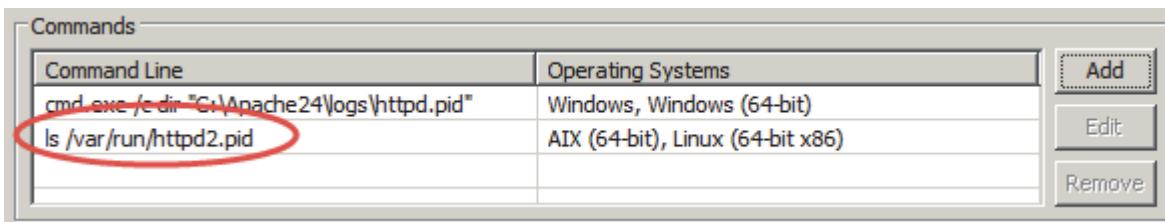


- j. Click **OK** to save the Windows command and close the Command Information window.

The Windows command is displayed in the Command Return Code section and the global return code is shown in the Global return codes section.



44. Create the Linux command that is shown in the following image.



a. Click **Add** in the Commands section to add the Linux command.

b. Enter the Linux command in the **Command line** field.

`ls /var/run/httpd2.pid`

Command line | ls /var/run/httpd2.pid

Again, test the command before adding it to your agent.

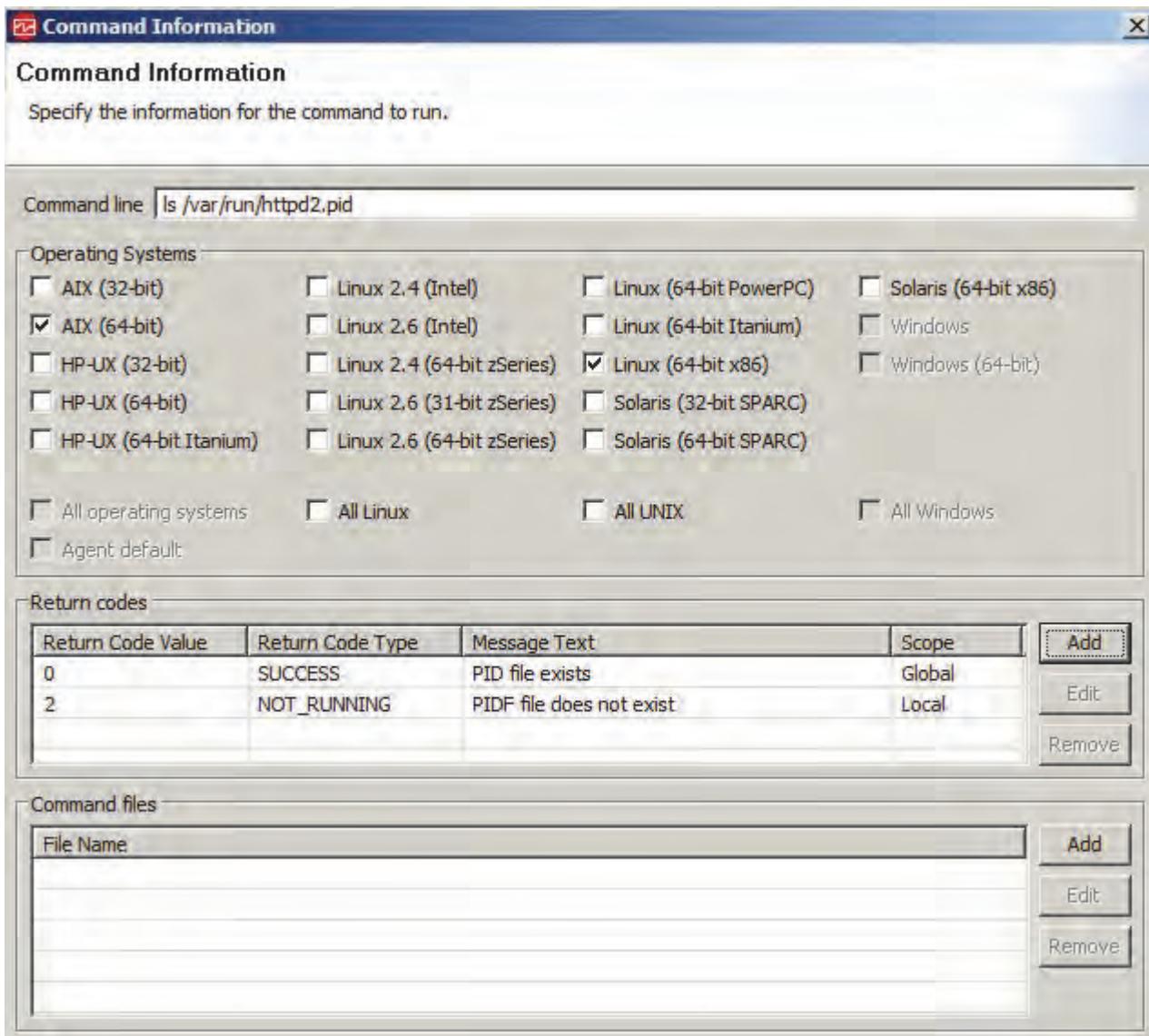
45. Keep the default Linux and UNIX systems under **Operating Systems**.

Because you already defined the 0 return code globally, it is already applied to this command, and you do not have to define it now. You do must define the failed return code for this command.

46. Repeat the previous steps to create the local failed return code on Linux according to the following specifications:

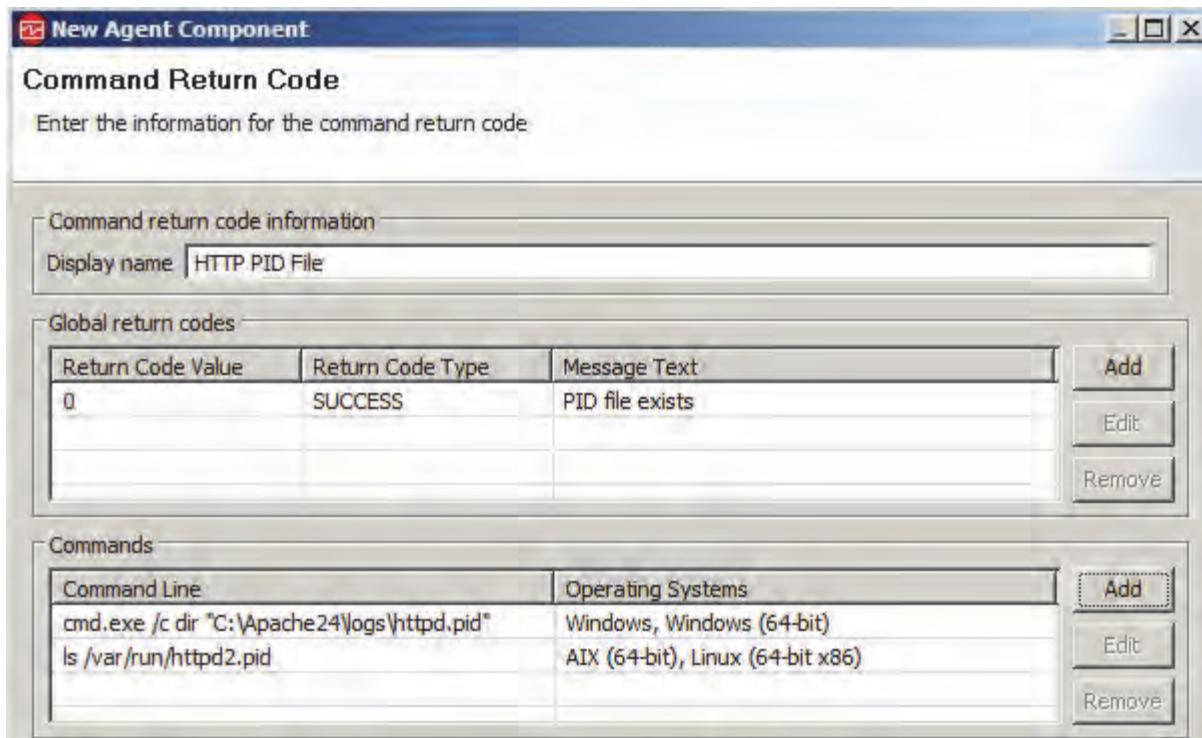
- Return Code type: **NOT_RUNNING**
- Return code: **2**
- Message Type: **Error**
- Message Text: **PID file does not exist**

Your final command window looks like the following example.



47. Click **OK** to save this Linux command.

The Command Return Code window is displayed.

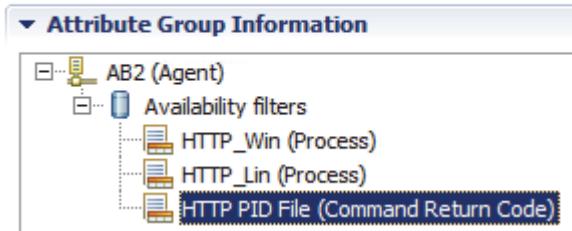


48. Click **Finish** to close the Command Return Code window.

The **Data Source Definition** tab is displayed.

49. If needed, expand **Availability Filters** and confirm your new command return code availability filter.

Data Source Definition



50. Save your agent project.

You successfully added a command return code data source to your agent.

Exercise 2 Install and confirm the AB2 agent in an IBM Performance Management environment

In this exercise, you install the AB2 agent onto WIN1 and LIN4. Then you confirm that data is being gathered for the new attribute groups you added.

1. Confirm that time is synchronized between the APM, LIN4, and WIN1 servers.
2. Confirm that all APM services are running on APM.

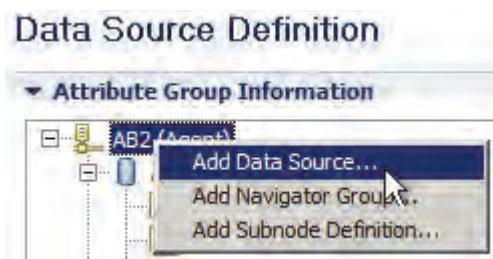
Create data points for the summary dashboard

In this section, you create a single row data source that gathers the data for the summary dashboard. Your goal is to create a summary dashboard like this one.

WIN1 - My HTTP Server	
Status	UP
Name	httpd.exe
Percent Processor Time	0

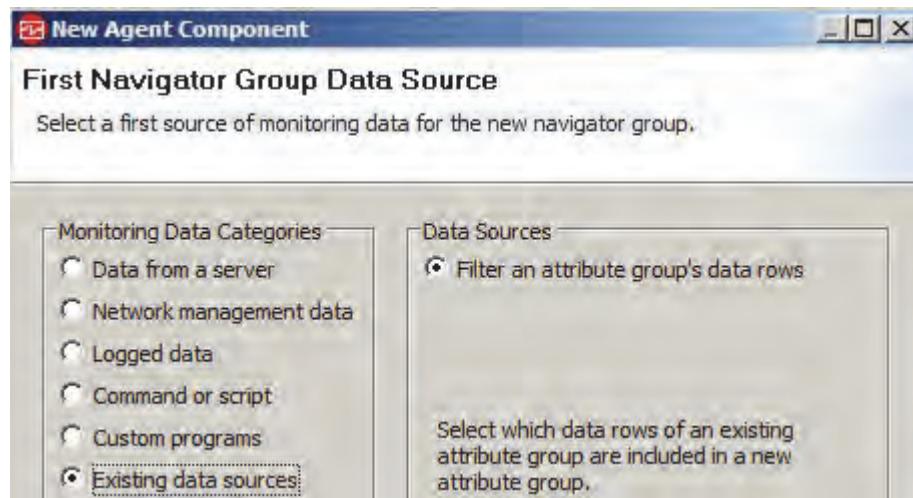
While your agent is already gathering this data, it is part of a multiple-row data set that reports the status of multiple entities. A Summary dashboard can pull data from only a single row data set. In this section, you create a custom data set that gathers only a single row of data.

3. Create a filter data source that changes your availability data to performance data and reduces it to a single row with the status of the HTTP server.
 - a. Click the **Data Sources** tab.
 - b. Right-click **AB2 (Agent)** in **Data Source Definition** and select **Add Data Source**.

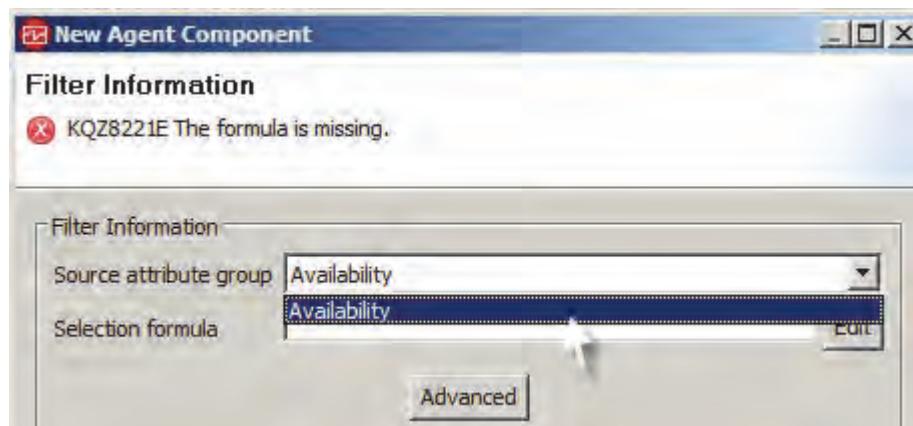


The New Agent Component window opens.

- c. Select **Existing data sources** > **Filter an attribute groups data rows**.

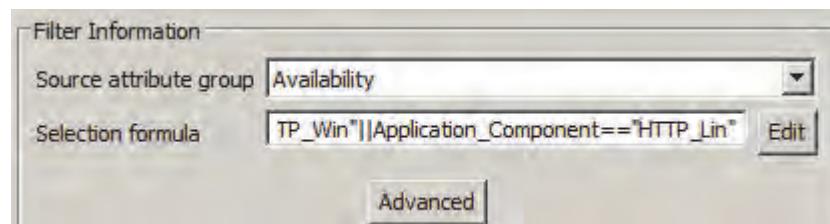


- d. Click **Next**.
e. Select **Availability** from the **Source attribute group** menu.



- f. Enter the following text in the **Selection Formula** field.

Application_Component=="HTTP_Win" || Application_Component=="HTTP_Lin"



This command gathers availability data for the first process whose name is either HTTP_Win or HTTP_Lin.



Hint: Look to the top of the window to see whether the editor identifies errors in your formula.

- g. Click **Finish**.
4. Edit the filtered data source to be a single row of data and to not have a keyed attribute.
 - a. Click **Availability_Filtered (Filter)** and select **Produces a single data row**.

Attribute Group Information

- AB2 (Agent)
 - Availability filters
 - Availability_Filtered (Filter)**

Filtered Attribute Group Information

Attribute group name: Availability_Filtered

Help: Filtered data rows from attribute group Availability.

Produces a single data row Can produce more than one data row
 Produces large number of data rows

- b. Expand **Availability_Filters (Filter)**.
- c. Select **Application_Component** and clear the **Key attribute** check box.

Attribute Group Information

- Availability_Filtered (Filter)
 - Application_Component**
 - Name
 - Status

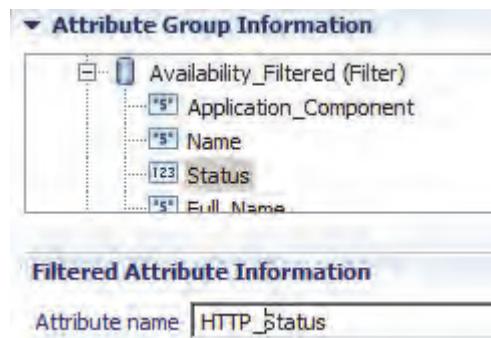
Filtered Attribute Information

Attribute name: Application_Component

Help: The descriptive name of a part of the application.

Hidden - can only be used in derived attribute
 Key attribute

5. Change the **Status** attribute to **HTTP_Status**.



6. Scroll down in the Attribute Information pane and set the following **Severity** criteria for the **HTTP_Status** attribute under **Availability_Filtered (Filter)**.

Severity	
Normal	1
Warning	
Critical	0
Not defined	

7. Save your agent project.

Create dashboards and resource definitions

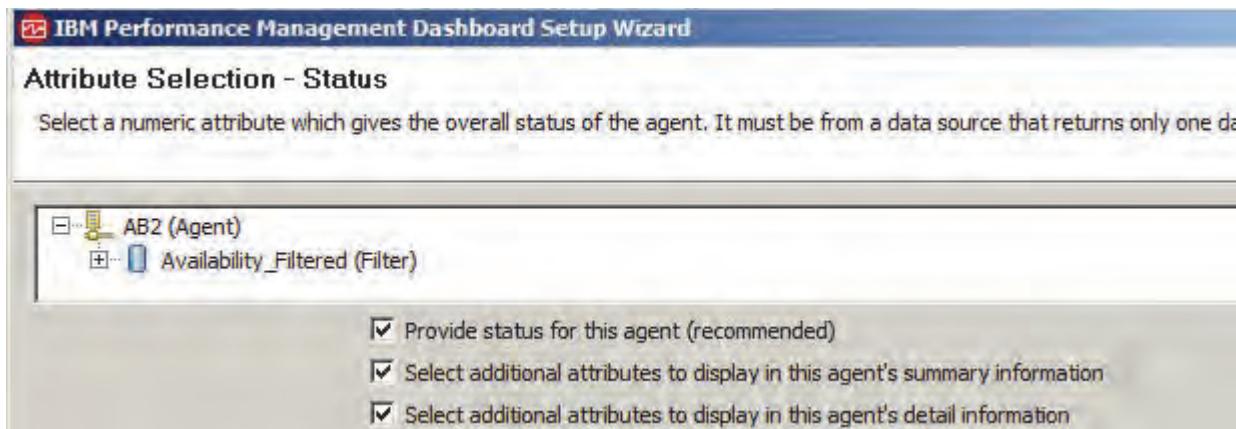
Use the Agent Builder Dashboard Setup wizard to define summary dashboard, details dashboards, and OSLC resources for your agent. These resources determine how the agent is displayed in the navigator Components group of the Application Performance Dashboard. They also specify the IP address and port of primary connection that the monitored server uses.

8. Select **Dashboards** from the Agent Builder Outline view.

The Dashboards Overview opens.

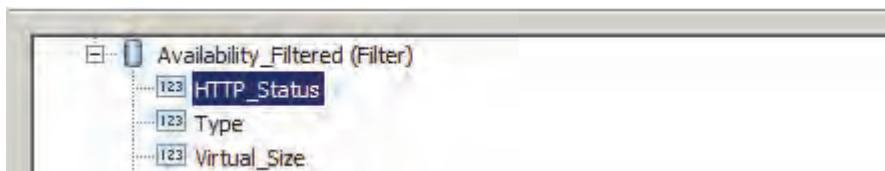
9. Click the **Dashboard Setup wizard** link in the Overview.

The IBM Performance Management Dashboard Setup wizard opens.



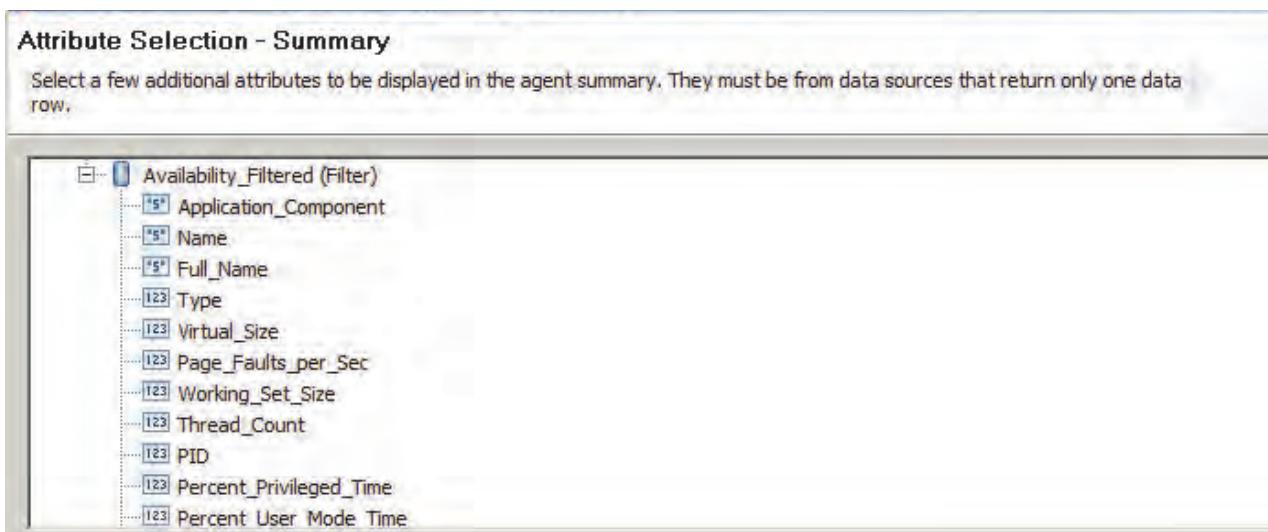
You are prompted to select a numeric attribute that gives the overall status of the agent.

10. Expand **Availability_Filtered (Filter)** and select **HTTP_Status** as attribute for the overall agent status.



11. Click **Next**.

The Attribute Selection - Summary view opens.

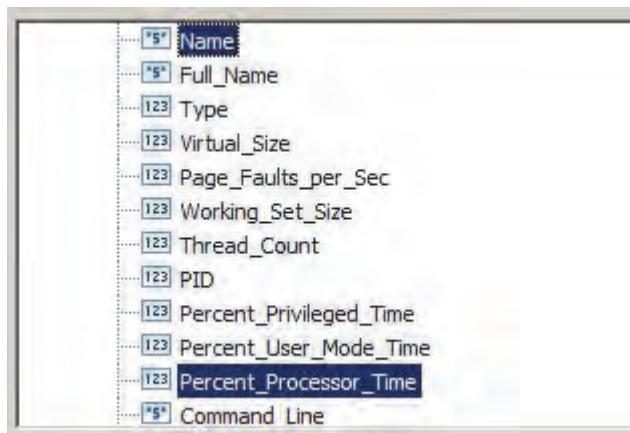


You are prompted to select more attributes for the Summary dashboard.

12. Select **Name** and **Percent_Processor_Time**.



Hint: Hint: Press the CTRL key to select multiple attributes.



13. Click Next.

The Attribute Selection - Details view opens.

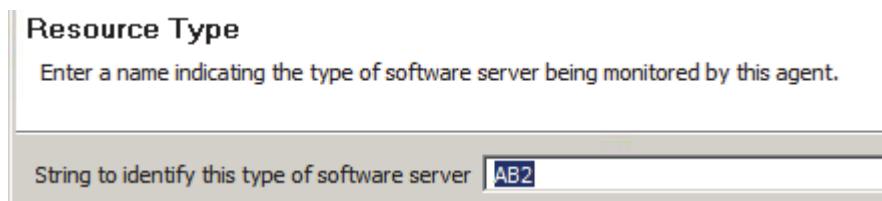
You are prompted to identify data groups and items to display in the details dashboards.

14. Select Availability (Availability).



15. Click Next.

The Resource Type view opens.

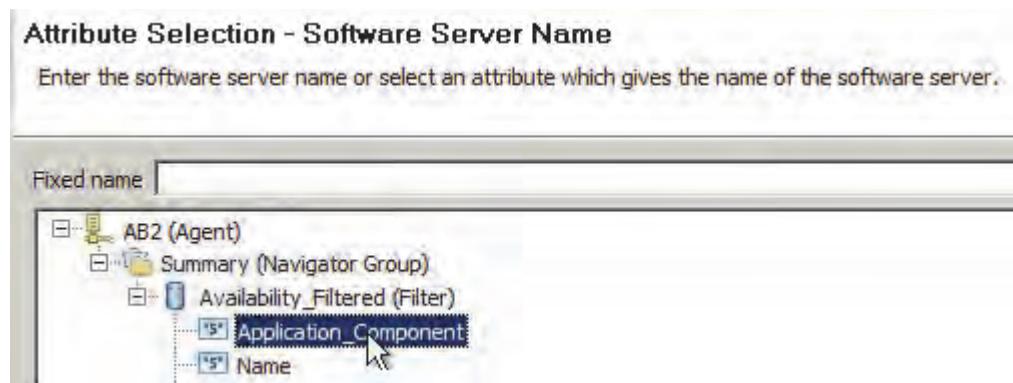


You are prompted for the Resource Type.

16. Keep AB2 in the resource type name field and click Next.

The Attribute Selection - Software Server Name view opens.

17. Expand the **Summary** and **Availability** nodes and select **Application_Component** as the Software Server Name.



18. Click **Next**.

The Attribute Selection - IP Address view opens.

19. Select the check box to use the agent's IP address is selected.



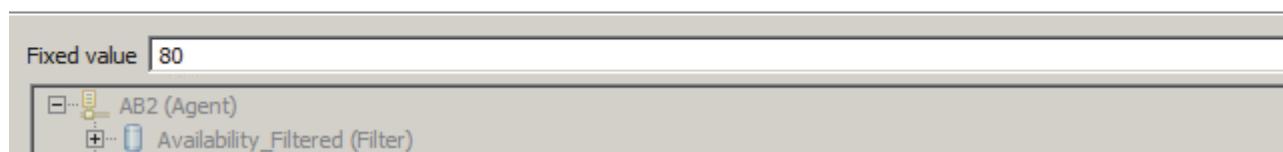
20. Click **Next**.

The Attribute Selection - Port view opens.

21. Enter **80** in the **Fixed value** field.

Attribute Selection - Port

Tell how the port on which the software server listens for requests is known: either from a numeric attribute or from a fixed value.



22. Click **Finish** to complete the wizard.

23. Save your agent project.

You successfully defined the summary, detailed dashboards, and the monitored resource data within OSLC.

Configure the agent for quick identification within the Performance Management Console

This change ensures that the new agents show promptly in the Performance Management Console so that you can confirm your agent.

24. Click the **itm_toolkit_agent.xml** tab.

25. Go to the bottom of the XML file and locate the **<DataModel>** tag.

```
</MESSAGE_LIST>
<DataModel>
    <Resource name="HTTP_Server_IP" oslcType="crtv:IPAddress"/>
```



Hint: Press CTRL+F to search for DataModel.

26. Add the following to the DataModel tag:

```
refreshInterval="5"
```

```
</MESSAGE_LIST>
<DataModel refreshInterval="5">
    <Resource name="HTTP_Server_IP" oslcType="crtv:IPAddress"/>
```

The default refresh interval is 60 minutes.



Important: In a production environment, a refresh interval of 5 minutes is probably too frequent and is not recommended.

27. Save your agent project.

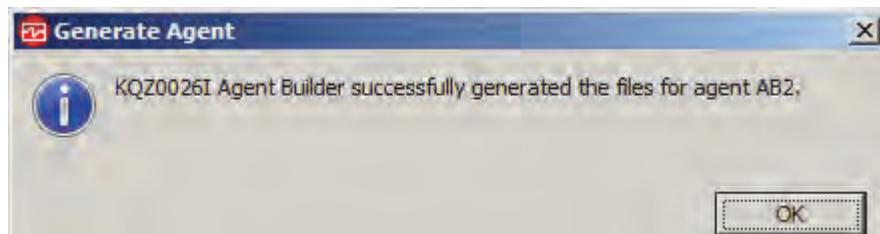
Create the installation files

28. Select **Agent Editor > Generate Agent** from the main menu.

29. Ensure that **Create a ZIP file** and **Create a TAR file** are checked.

30. Click **Finish**.

A progress information window opens. The installation files are created in approximately 1 minute. A confirmation window opens when the installation files are successfully generated.

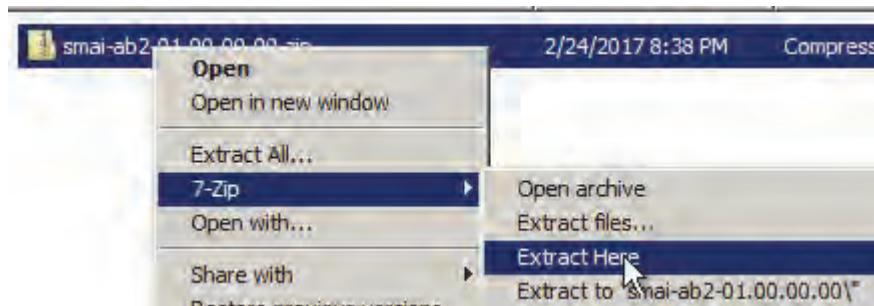


31. Click **OK** to close the confirmation window.
32. Open a command prompt or Explorer window and go to **C:\Users\Administrator** and confirm that the **smai-ab2-01.00.00.00.tgz** and **smai-ab2-01.00.00.00.zip** files are created.

Install the agent on WIN1

In this section, you install the agent on WIN1.

33. On the WIN1 server, create a folder **K01** in **C:\share**.
34. Copy the new **smai-ab2-01.00.00.00.zip** file into the **C:\share\K01** directory.
35. From Windows Explorer, right-click the **smai-ab2-01.00.00.00.zip** file and select **7-Zip > Extract Here**.



The agent installation files are extracted.

36. Open a command prompt and change to the **C:\share\K01** directory on the share drive.
37. Run the following command:

```
installIraAgent.bat C:\IBM\APM
```

```
C:\share\K01>installIraAgent.bat C:\IBM\APM
Installing agent into C:\IBM\APM
Installing K01 .....
Install of K01 Agent successful.

C:\share\K01>
```

38. Open the IPM utility and select **View > Refresh**.

The screenshot shows the IBM Performance Management interface. The title bar reads "IBM Performance Management - [C:\IBM\APM]". The menu bar includes "Actions", "Options", "View", and "Help". Below the menu is a toolbar with icons for search, refresh, and other functions. The main area is a table with the following columns: Service/Application, Task/SubSystem, Configured, Status, Connection status, Configuration, Startup, and Account. There are three rows of data:

Service/Application	Task/SubSystem	Configured	Status	Connection status	Configuration	Startup	Account
Monitoring Agent for Windows OS	Primary	Yes	Started	Connected	up-to-date	Auto	LocalSystem
Monitoring Agent for AB2	Primary	Yes	Started	Connected	up-to-date	Auto	LocalSystem
Monitoring Agent for AB1	Primary	Yes	Started	Connected	up-to-date	Auto	LocalSystem

Your agent is visible and started.

Install the agent on LIN4

39. Access LIN4 and log in as **root** with password **object00**.

40. Open a terminal window and run the following commands to confirm that the **installation scripts are** visible to LIN4:

```
mount /mnt/share
ls /mnt/share
cd /mnt/share/K01
ls
```

41. Confirm the **installIraAgent.sh** script exists.

```
lin4:/mnt/share/K01 # ls
01_dd_010000000.xml  getarch.sh           installIraAgentTEPS.bat   KO1WINNT.dsc
01_dd.properties      help                 installIraAgentTEPS.sh   KO1WX64.dsc
ab2-agent.bat         installIraAgent.bat   installIra.bat          reports
ab2-agent.sh          installIraAgent.sh    installIra.sh           smai-ab2-01.00.00.00.zip
agent.yaml            installIraAgentTEMS.bat ira                  support
getarch.ksh           installIraAgentTEMS.sh k0lunix.dsc
```

The file "installIra.sh" is highlighted with a red box.

42. Install the agent onto this computer by running the following command:

```
./installIraAgent.sh /opt/ibm/apm/agent
```

The screenshot shows a terminal window titled "Terminal". The command "./installIraAgent.sh /opt/ibm/apm/agent/" is entered and executed. The output shows the installation process:

```
lin4:/mnt/share/K01 # ./installIraAgent.sh /opt/ibm/apm/agent/
Installing k01 .....
copying agent config files now....
Processing. Please wait...
Starting the Monitoring Agent for AB2...
Monitoring Agent for AB2 started
Automatic start at system initialization has been configured.
Automatic stop at system shutdown has been configured.
Install of K01 Agent successful.
lin4:/mnt/share/K01 #
```

The AB2 agent does not require configuration and is automatically started.

43. Run the following command to confirm that the HTTP Server agent is running:

```
/opt/ibm/apm/agent/bin/ab2-agent.sh status
```

```
Lin4:/mnt/share/K01 # /opt/ibm/apm/agent/bin/ab2-agent.sh status
Agent status:
Agent is running. Process ID is 14692
Server connection status: Connected
For more information, see the following file: /opt/ibm/apm/agent/logs/01_ServerConne
Lin4:/mnt/share/K01 #
```

Confirm the agents in the Performance Management Console

Because your AB2 agent is a new resource type, it can take up to 30 minutes for the Performance Manager to fully process the agent and its data. Consider taking a break or continuing with the next lecture and returning to this part of the lab after 30 minutes.

44. Log in to the **APM** server as user **root** with password **object00**.

45. Open a browser and go the following URL:

<https://apm:9443>



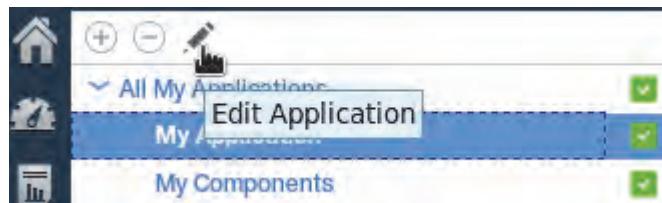
Hint: A bookmark to this URL is in the browser.

46. Log in to the PM console as user **apmadmin** with password **object00**.

47. Open the Application Performance dashboard.

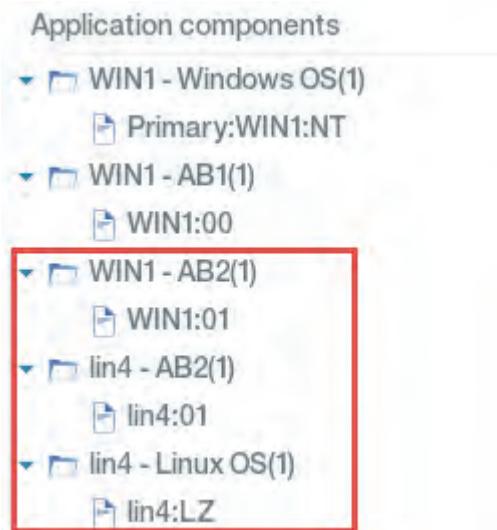
48. Click **My Application** in the Applications explorer.

49. Click the **Edit Application** button.



The Edit Application windows opens.

50. Add the following components to your application:

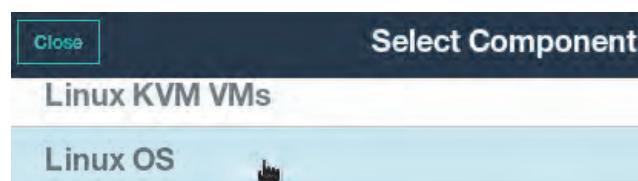


- a. Click the **Add components** button next to the Application components pane.



The Select Component window opens.

- b. Scrolling down, locate and select **Linux OS**.



The Component Editor windows opens.

- c. Select **lin4:LZ** and click **Add**.



The agent is removed from the Component Editor window added to your application definition.

- d. Click **Back** to return to the Select Component window.
- e. Scroll up and select **AB2**.



The Component Editor windows opens.

- f. Select both **WIN1:01** and **lin4:01** agents and click **Add**.



Important: It can take 10 - 15 minutes from when the agent is installed until it shows in the Component Editor window. If the agents are not listed, click **Back** and **AB2** until they are shown. Also, verify the date and time on APM, WIN1, and LIN4 hosts are the same.

The agents are removed from the Component Editor window added to your application definition.

The screenshot shows the "Component Editor" window. On the left, there is a sidebar titled "Application components" with a tree view:

- WIN1 - Windows OS(1)
 - Primary:WIN1:NT
- WIN1 - AB1(1)
 - WIN1:00
- WIN1 - AB2(1)
 - WIN1:01
- lin4 - AB2(1)
 - lin4:01
- lin4 - Linux OS(1)
 - lin4:LZ

Three red arrows point from the bottom of the tree view to the "lin4" nodes: one to "lin4:01" and two to "lin4:LZ". On the right, the main panel of the "Component Editor" window shows:

Component name *

Multiple instances are selected.

Select instances

No instance available.

There are three icons (magnifying glass, copy, delete) above the instance list.

- g. Click **Back** to return to the Select Component window.
- h. Click **Close** to close the Select Component window.



You are returned to the Edit Application window.

A screenshot of the "Edit Application" window. At the top, there are "Cancel" and "Save" buttons. The main area has fields for "Application name *" (containing "My Application") and "Description". Below these is a section "Application read from:" with a "Read..." button. A "Template *" section shows "Custom Application" selected. The bottom half of the window displays a hierarchical tree of "Application components":

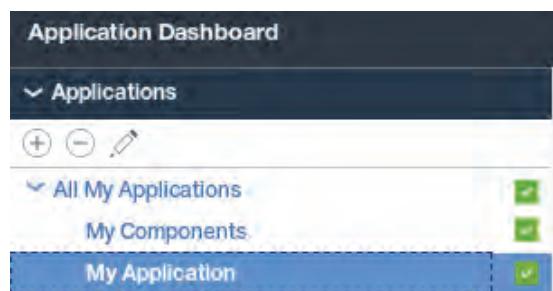
- WIN1 - Windows OS(1)
 - Primary:WIN1:NT
- WIN1 - AB1(1)
 - WIN1:00
- WIN1 - AB2(1)
 - WIN1:01
- lin4 - AB2(1)
 - lin4:01
- lin4 - Linux OS(1)
 - lin4:LZ

On the right side of the component tree, there are three buttons: a plus sign (+), a minus sign (-), and a refresh/circular arrow icon.

51. Click **Save** and **OK** to save your application definition and exit the Edit Application utility.

You are returned to the Application Dashboard.

52. Click **My Application** in the Applications explorer.



53. Refresh this view until the Linux OS and AB2 component bars show and turn green.

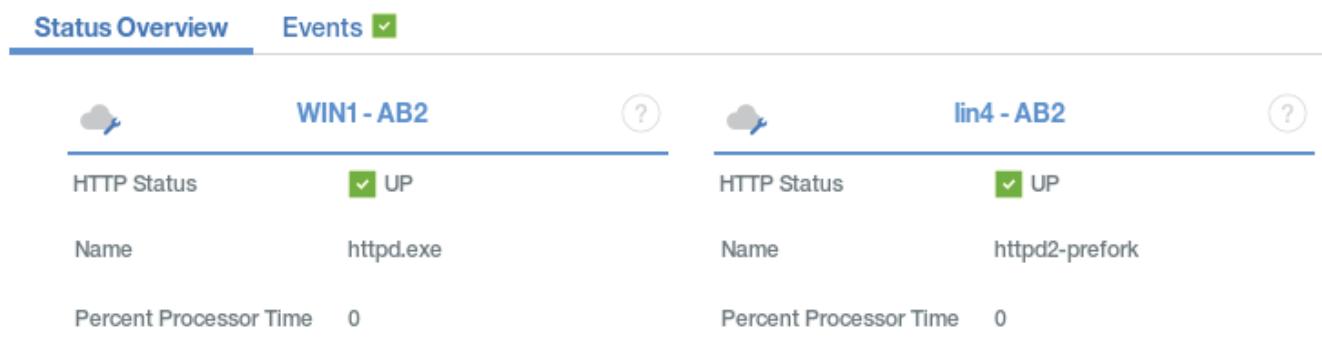


Hint: It can take up to 30 minutes for the Summary dashboard and initial data to be displayed.

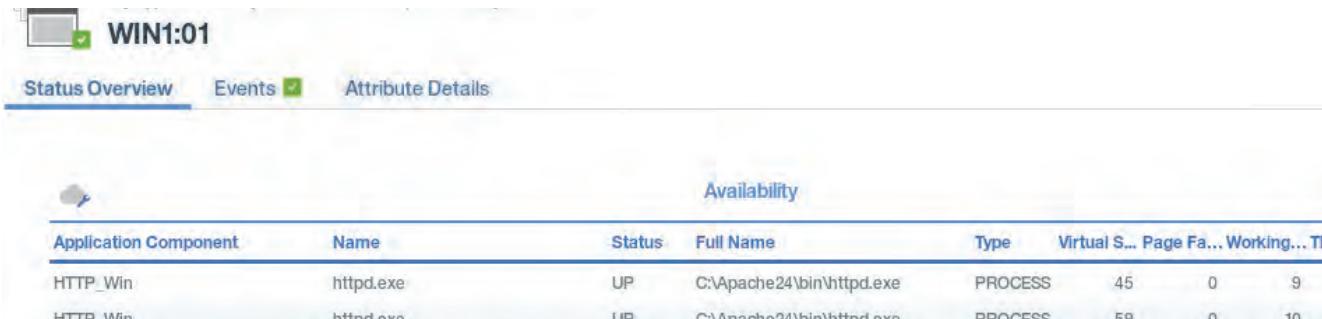


54. Click the **AB2** component bar in Current Components Status.

The AB2 Status Overview window opens with a Summary dashboard for each component.

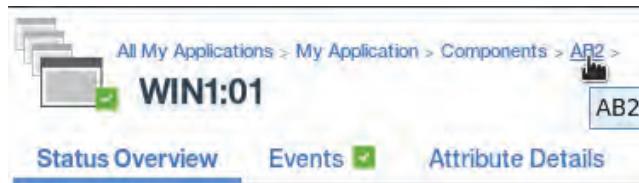


55. Click the **WIN1 - AB2 Summary** dashboard widget to access the detailed dashboards.



56. Confirm that the LIN1 AB2 agent is also sending data.

- a. Click the **AB2** navigation link at the top of the page.



- b. Click the **lin4 - AB2 Summary** dashboard.

You successfully confirmed your agent in a Performance Management Console. Because the agent was available to be added to an application, the Summary dashboard shows the core metrics that you added. A subset of attributes is visible in the **Details dashboard**.

Unit 6 Monitoring custom data sources exercises

In this unit, you modify your agents to gather and monitor data with custom instrumentation.

In the first solution, you modify the AB2 agent to deploy and run a script that generates monitoring data. The agent then monitors and returns the data that is generated by the script. The key to this solution is to create a script or command that generates the target data in the correct format. You also enable the agent to establish a socket connection with an application and monitor the application.

In the second solution, you modify the AB1 agent to monitor a log file. The agent must be able to parse the log file to pull the IP address, packets sent, packets received, and packets lost. Then the agent creates an attribute that states the status of the host that is based on the total number of packets received. Specifically, it states the host status is Good if four packets are received and Failure if no packets are received. Additionally, you add a Java API data source that enables the agent to monitor an application or resource with custom Java code.

Exercise 1 Create an agent to monitor local and remote systems with scripts and socket connections

Agent Builder can create an agent that carries one or more scripts, runs each script on the host, and returns the output data.

In this exercise, you do the following tasks:

- Create an agent that distributes a script, runs the script, and returns the output data to the agent
- Create an agent that listens to a socket connection and monitors data and error codes sent to that connection
- Create the installation archive files with command line interface (CLI) commands

Confirm the script

To return the script output correctly, the data must be placed in a delimited format with no unnecessary data. This task requires that a script that generates and formats the data properly be used by the agent.

Start with a command for the initial formatting of the data. Then, use a script that parses the basic data into only the data you want returned to the monitoring system.

In this exercise, you use a script that uses the VMSTAT command. VMSTAT is a good command for this exercise because it displays interesting data in a space-delimited format, which reduces the amount of extra parsing that you must do on the data.

1. On LIN4, open a terminal window and run the following command:

```
vmstat 1 2
```

```
Lin4:~ # vmstat 1 2
procs -----------memory----- swap-----io--- system--cpu-----
r b swpd   free   buff   cache   si   so    bi    bo   in    cs us sy id wa st
0 0      0 1261992 17132 380340   0   0   144     7 114 154 1 1 82 16 0
0 0      0 1261992 17132 380352   0   0     0   0 101 144 0 0 100 0 0
Lin4:~ #
```

The VMSTAT reports information about processes, memory, paging, block IO, traps, and processor activity. The first report provides averages since the last restart. More reports provide information about a sampling period of length delay. Because you are not interested in the average since the last restart, your command requests two reports with a 1-second delay between them, making the second report a good current interval.

For this exercise, you reduce the data that you gathered to just the processor information on the far right. The columns *us*, *sy*, *id*, and *wa* show the percentage of time the processor spent on user processes and system processes, idle, or waiting.

To parse the data correctly, you need a command or script that removes the first two rows and then everything except those four data points. A script for this task is provided.

2. Change to the following directory:

```
/root/AB_Files
```

3. Run the following command to see the contents of your script:

```
cat script1.sh
```

```
lin4:~/AB_Files # cd /root/AB_Files/
lin4:~/AB_Files # ls
AgentGenerator.log  SocketTestDL.pl  SocketTestDR.pl  SocketTestEL.pl  SocketTest.pl
script1.sh          SocketTestDL.pl~  SocketTestDR.pl~  SocketTestER.pl
lin4:~/AB_Files # cat script1.sh
vmstat 1 2 | tail -1 | awk '{print $13","$14","$15","$16}'
lin4:~/AB_Files #
```

The tail section outputs only the last row. The *awk* section produces the 13th - 16th values, delimited with a comma.

- Run the following command:

```
./script1.sh
```

```
lin4:~/AB_Files # ./script1.sh
8,8,67,17
```

This example has a properly formatted script output for your agent: delimited data points. Notice that the delimiting character is a comma. You now have a script that you can build script data source around.

- Copy the script to the WIN1 share with the following command:

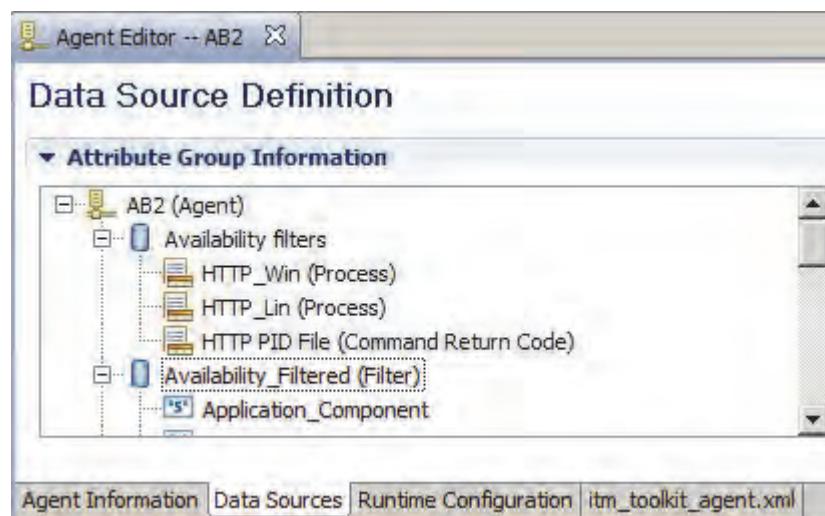
```
cp script1.sh /mnt/share
```

You confirmed the script and made it available to Agent Builder on WIN1.

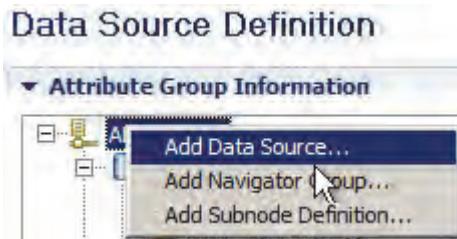
Modify AB2 to monitor the script

Complete the following steps:

- On WIN1, start Agent Builder, if it is not already running.
- If the **AB2** agent is not already open, expand **Agent 2** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
- Click the **Data Sources** tab.

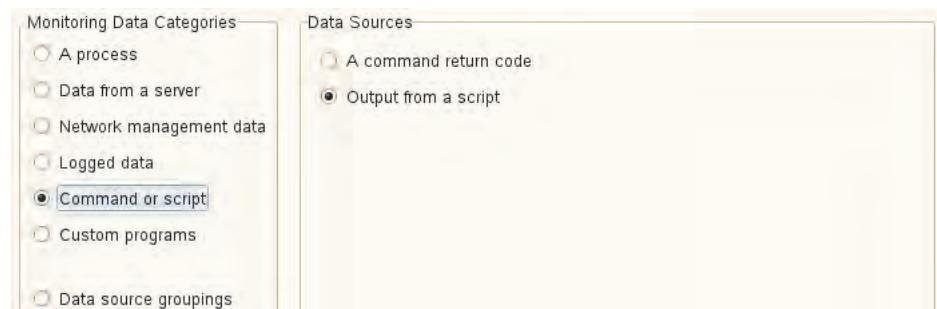


9. Right-click **AB2 (Agent)** under **Attribute Group Information** and click **Add Data Source**.



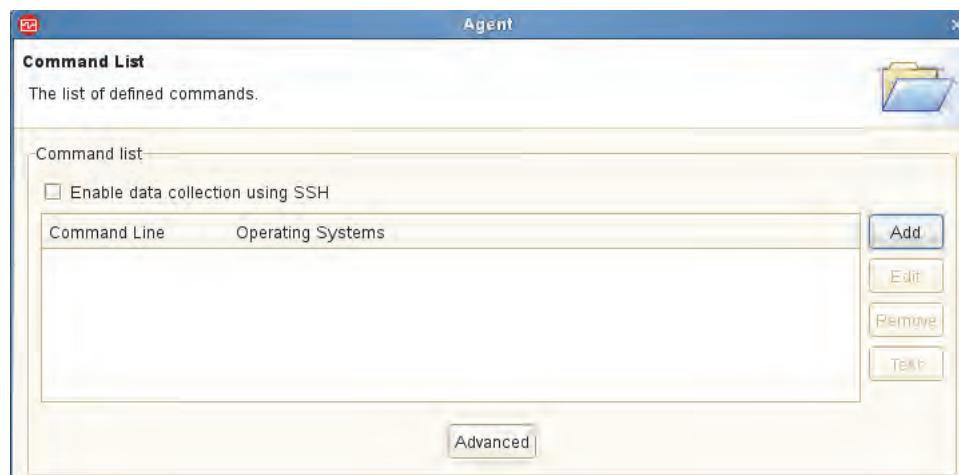
The New Agent Component window opens to Data Source Location.

10. Click **Command or script** under **Monitoring Data Categories** and **Output from a script** under **Data Sources**.



11. Click **Next**.

The Command List window opens.



From this window, you can add one or more scripts to this agent.

12. Click **Add**.

13. Enter **script1.sh** in the command line.



Notice the semicolon in the Separator section.

14. Change the separator from a semicolon to a comma.



15. Keep the **Agent default** operating systems.

16. Click **Add** to the right of the **Command files** section.

17. Navigate to and locate the following file:

C:\share\script1.sh



18. Click **script1.sh** and click **Open**.

The Script Information window opens with the **script1.sh** file that is displayed in the Command files section.



19. Click **Test**.



20. If prompted, click **Yes** to stop the agent.

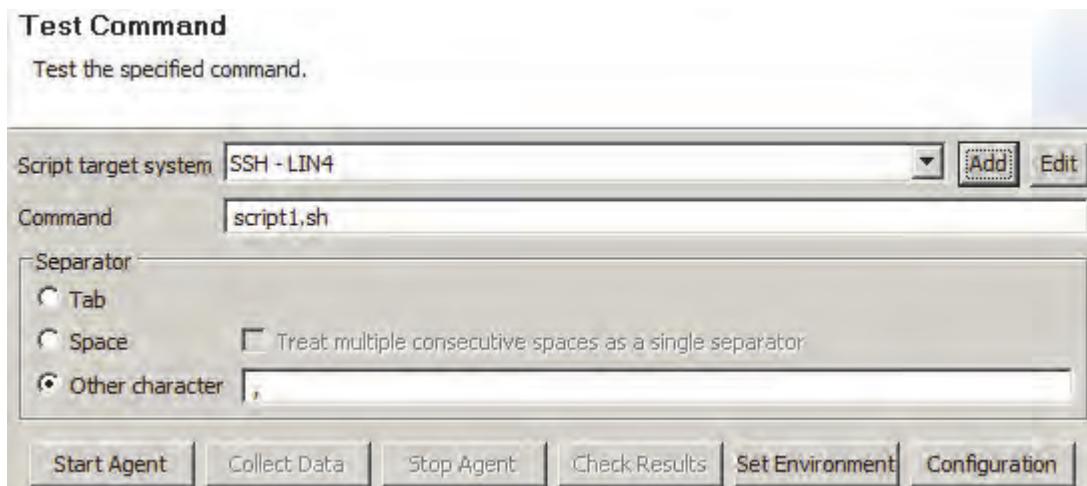
The Test Command window opens.



21. Select **SSH - LIN4** from the **Script target system** menu.



22. Ensure that the target system is LIN4, and the command name and separator are correctly listed.



23. Click **Start Agent**.

24. Click **Collect Data**.

A status indicator is displayed.

25. Select **No** to the prompt to view suggested data types.

The script results are captured and attributes are defined to hold the data.

Results			
<input checked="" type="checkbox"/> Show hidden attributes			
Attribute_1	Attribute_2	Attribute_3	Attribute_4
12	5	75	9

Notice that this test was successful and attributes are correctly created to match the data that is retrieved. Had your script test been unsuccessful, you might find errors in the lower part of the Test Command window and you might see the attributes that are created to be inadequate. For example, you might find two or more data points that are combined into one attribute.



Note: If you do not test a script in the Agent Builder tester, the Agent wizard will not define attributes and you must create them manually outside of the wizard.

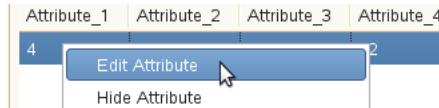
Up to this point in this course, you created agents with predefined attribute groups and attributes. Because Agent Builder cannot know the attributes that are gathered, you must define them now.

26. Click **Stop Agent**.

27. Edit the four attributes for the four values that are generated by this script.

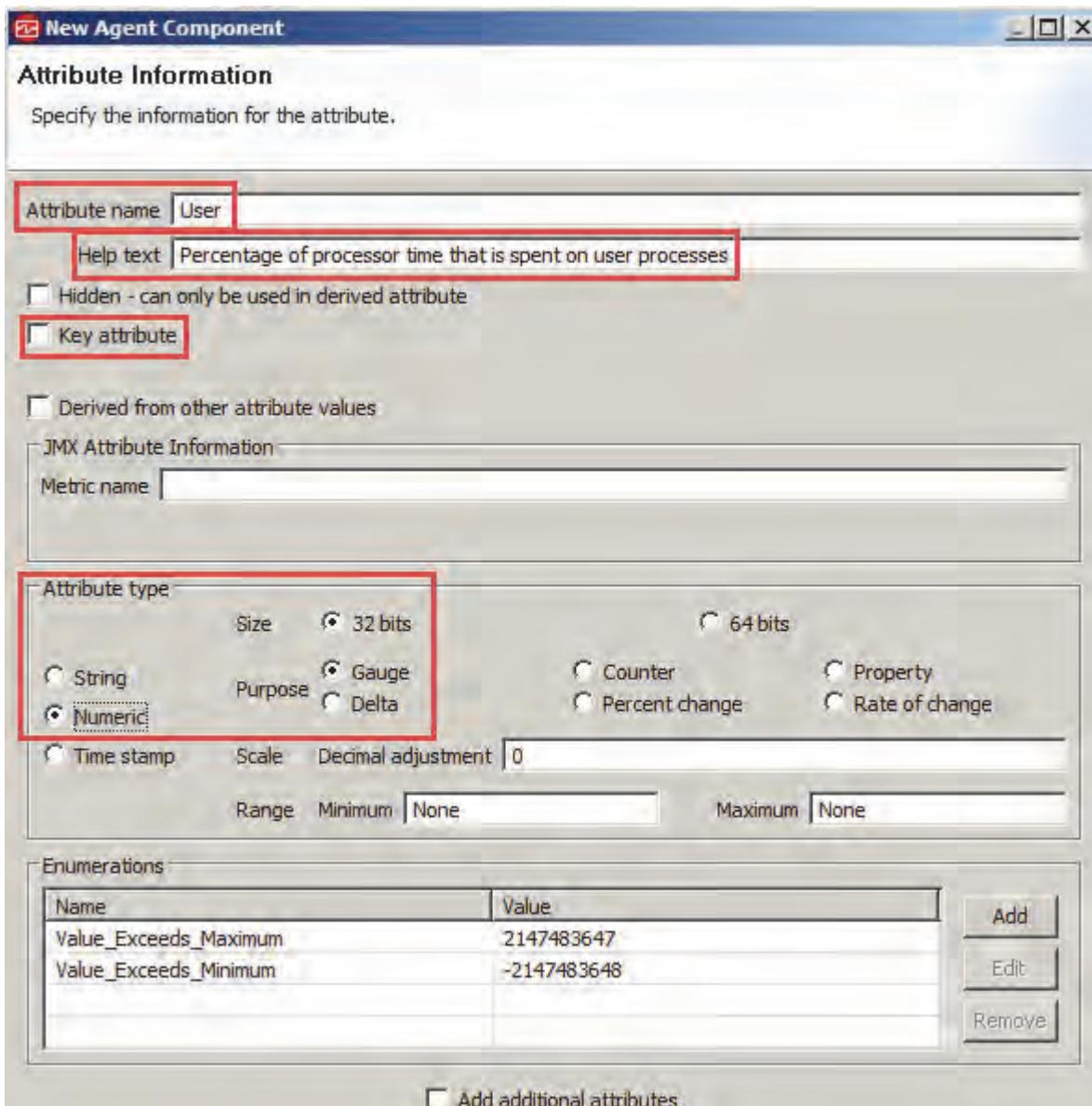
Current Attribute	New Name	Help text	Key?	Type
1	User	Percentage of processor time that is spent on user processes	No	Numeric, Gauge, 32-bit
2	System	Percentage of processor time that is spent on system processes	No	Numeric, Gauge, 32-bit
3	Idle	Percentage of processor time that is spent idle	No	Numeric, Gauge, 32-bit
4	Wait	Percentage of processor time that is spent waiting	No	Numeric, Gauge, 32-bit

- a. Right-click the cell in the attribute column and click **Edit Attribute** or click the column header.



The Attribute Information window opens.

- b. Complete the Attribute Information window based on the information in the preceding table.



- c. Click **OK** after you finish.

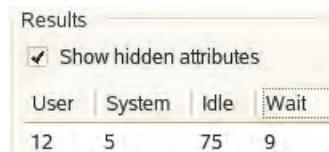
The new column name is displayed in the Test Command window.

Results				
<input checked="" type="checkbox"/> Show hidden attributes				
User	Attribute_2	Attribute_3	Attribute_4	
12	5	75	9	

- d. Select **No** to the prompt to view suggested data types.

- e. Repeat these steps for each attribute.

When completed, you see the following results in your Results pane.



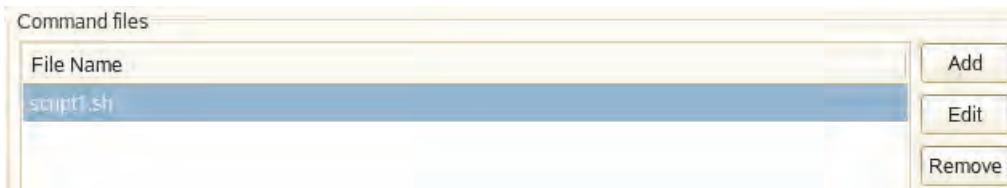
In the Test Command window, you successfully tested your script and you created the attributes that are needed to capture the data.



Important: Clicking **OK** from the Test utility exits the utility and saves your attribute modifications. Clicking **Cancel** exits without saving the attribute definitions.

28. Click **OK** to close the **Test Command** window.

The Command Information window is displayed with the **script1.sh** script in the Command Files section.

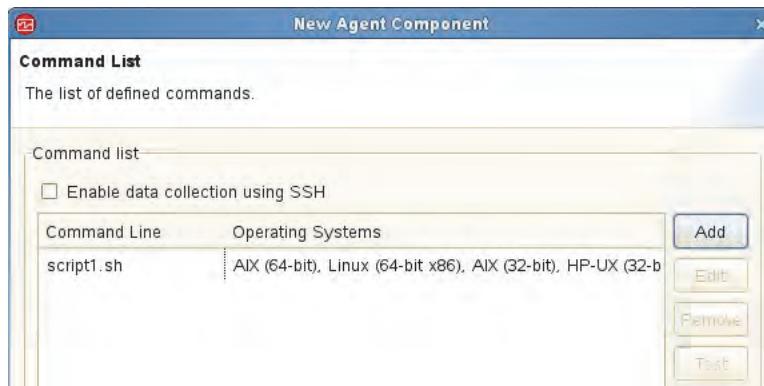


29. Select **All operating systems**.



30. Click **OK** to save your command and close the **Command Information** window.

The Command List window is displayed with the **script1.sh** script added.



You can add more scripts to this agent, edit an existing script, or remove an existing script. Your agent uses the only this script.

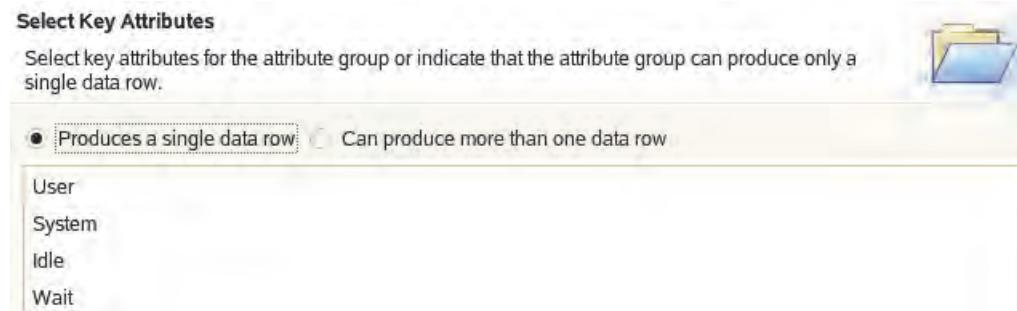
31. Click **Enable data collection using SSH** to enable the agent to collect the data remotely over SSH.



32. Click **Next**.

The Select Key Attributes window opens.

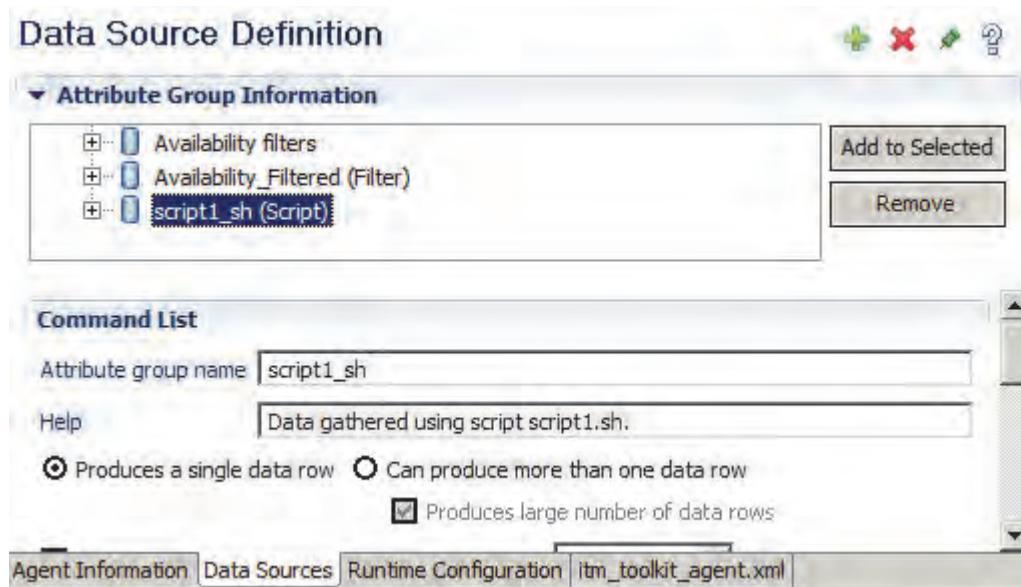
33. Click **Produces a single data row**.



Both the **Next** and **Finish** buttons are active. You completed enough steps to create this agent. If you want to continue working in the wizard to modify this agent, such as adding more data sources, you click **Next**.

34. Click **Finish**.

You are returned to the Data Sources tab showing your new Script data source.



35. Save your agent project.

Notice that the attribute group name has the script name.

36. Modify the agent by renaming the attribute group to Processor Utilization.

- Click **script1_sh (Script)** in the **Attribute Group Information** pane.
- Replace **script1.sh** with **Processor_Utilization** in the **Attribute group name** field and press Enter to activate the change.



37. Save your agent project.

You successfully created an agent that deploys and monitors data that is created by a script.

Add derived attributes

In this section, you add derived attributes that do the following tasks:

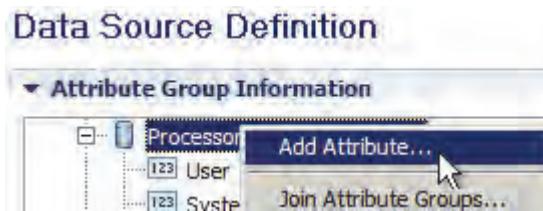
- Get the host name from an environment variable of the monitored host
- Resolve the IP address from the host name
- Concatenate the HostName and IP address fields

38. Add a derived attribute that identifies the agent host according to the following specifications:

- Attribute name: **HostName**
- Help text: **Capture the host name from the hostname environment variable**
- Attribute Type: **String, Maximum size: 64**
- Derived Attribute Formula:

`getenv("hostname")`

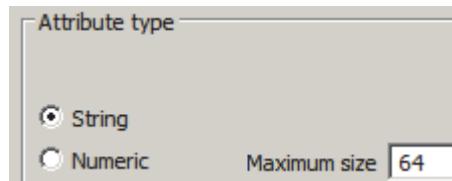
- a. In the **Data Sources** tab, right-click **Processor_Utilization (Script)** attribute group and select **Add Attribute**.



- b. Enter the **Attribute name** and **Help text** values.

Attribute name	HostName
Help text	Capture the host name from the hostname environment variable

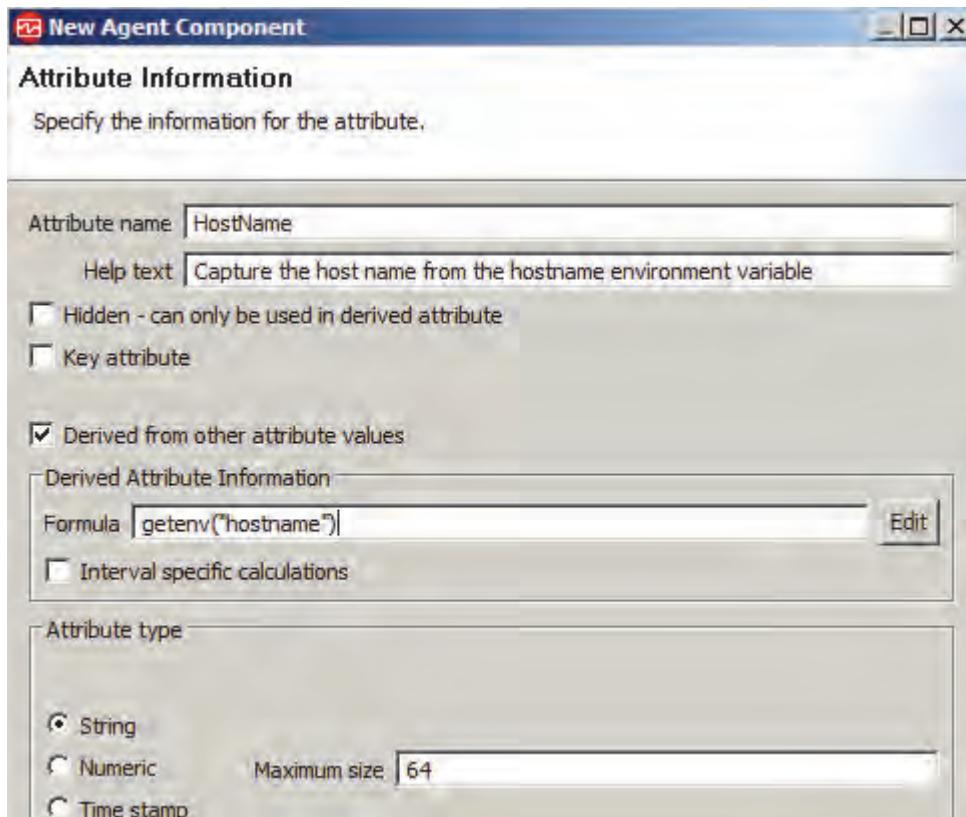
- c. Select **String** in **Attribute Type**.



- d. Select **Derived from other attribute values** and enter the formula.

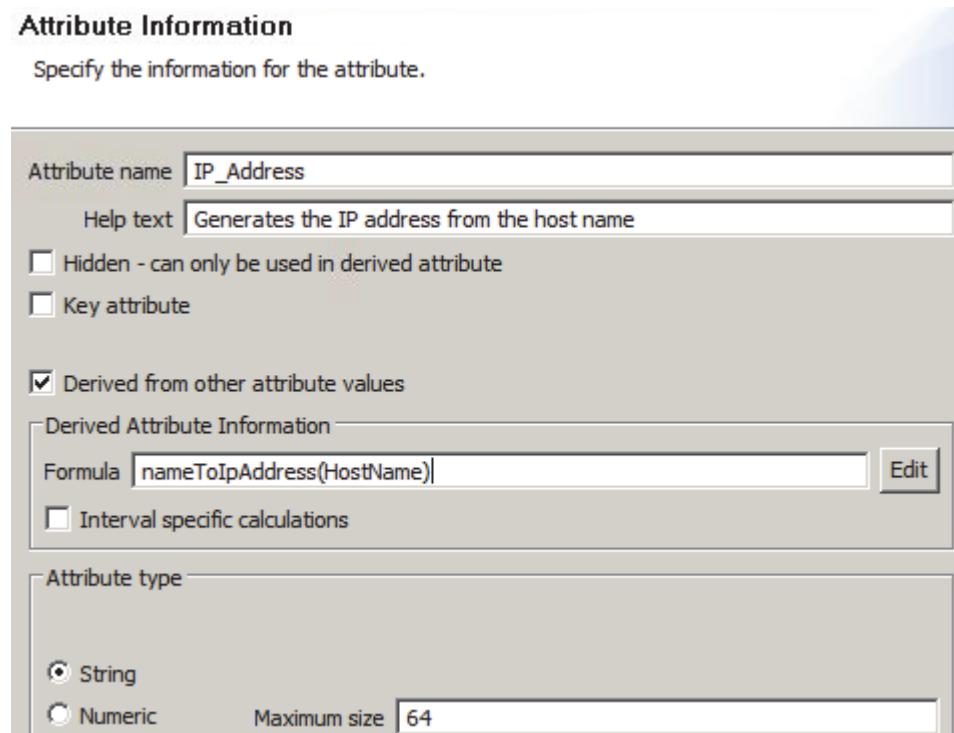
<input checked="" type="checkbox"/> Derived from other attribute values	
Derived Attribute Information	
Formula <input type="text" value="getenv('hostname')"/>	<input type="button" value="Edit"/>
<input type="checkbox"/> Interval specific calculations	

The Attribute Information window looks like the following screen capture.



- e. Click **Finish** to close the Attribute Information window.
39. Add an attribute to Processor_Utilization that derives the IP address from the host name captured by the HostName attribute. Complete the attribute as shown here:
- Attribute name: **IP_Address**
 - Help text: **Generates the IP address from the host name**
 - Attribute Type: **String, Maximum size: 64**
 - Derived Attribute Details Formula:
`nameToIpAddress (HostName)`

The Attribute Information window looks like the following screen capture.



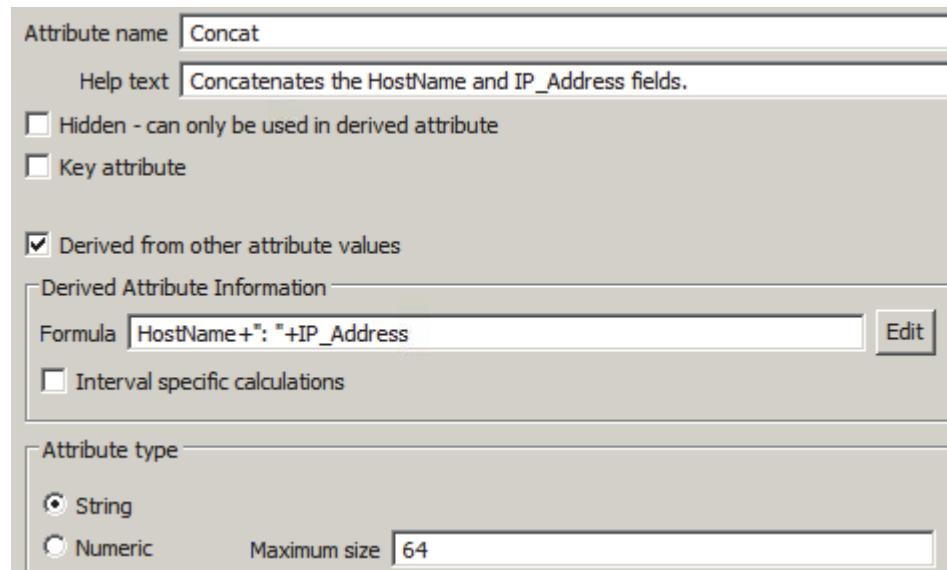
Hint: Click **Finish** to close the Attribute Information window or select **Add additional attributes** to create the derived attribute in the next step.

40. Add a derived attribute that concatenates the HostName and IP_Address attributes according to the following specifications:

- Attribute name: **Concat**
- Help text: **Concatenates the HostName and IP_Address attribute values.**
- Attribute Type: **String, Maximum size: 64**
- Derived Attribute Details Formula:

HostName+": "+IP_Address

The Attribute Information window looks like the following screen capture.



41. Test Processor_Utilization and confirm that it returns one row of data and generates the derived attributes.
 - a. Select **Processor_Utilization**.
 - b. Scroll down in **Command List** and select **script1.sh**.
 - c. Click **Test**.

The screenshot shows the 'Command List' interface. On the left, there is a tree view with nodes 'Processor_Utilization (Script)' and 'AppDataSocket (Socket)'. On the right, a table lists command details. The first row, 'script1.sh', is selected and highlighted in blue. The table has two columns: 'Command Line' and 'Operating Systems'. The 'Command Line' column contains 'script1.sh' and the 'Operating Systems' column lists 'AIX (32-bit), AIX (64-bit), HP-UX (32-bit), ...'. To the right of the table are four buttons: 'Add', 'Edit', 'Remove', and 'Test'. A cursor arrow is pointing at the 'Test' button.

Command Line	Operating Systems
script1.sh	AIX (32-bit), AIX (64-bit), HP-UX (32-bit), ...

- d. Select **SSH - LIN4** in **Script target system**.
- e. Click **Start Agent** and **Collect Data**.

The results looks like this screen capture.

Test Command

1 data row returned at Feb 27, 2017 5:50:43 PM.

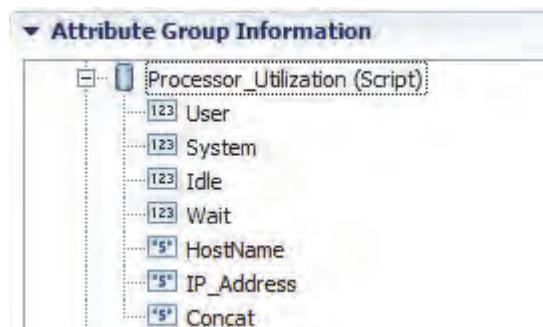
The screenshot shows the 'Test Command' configuration window. The 'Script target system' is set to 'SSH - LIN4'. The 'Command' field contains 'script1.sh'. Under 'Separator', the 'Other character' option is selected with a comma (,) chosen. In the 'Results' section, the 'Show hidden attributes' checkbox is checked. A table displays the data row: User (1), System (0), Idle (99), Wait (0), HostName (WIN1), IP_Address (192.168.1.103), and Concat (WIN1: 192.168.1.103). Buttons for 'Start Agent' and 'Collect Data' are visible.

User	System	Idle	Wait	HostName	IP_Address	Concat
1	0	99	0	WIN1	192.168.1.103	WIN1: 192.168.1.103

f. Click **OK** to close the Test Command window.

g. Click **OK** to close the Command Information window.

The final Processor_Utilization data source looks like the following screen capture:



42. Save your agent project.

Enable the agent to monitor local and remote sockets

In this section, you add a data source that receives data from an application through a socket connection. You install the agent locally and remotely to the monitored application to see the different configuration requirements.

The application sends four data points: a message and three numeric values.

Message	MsgCode	MsgCnt	QueueSize
A message from perl host name	1	2	123
More from perl	456	123	789

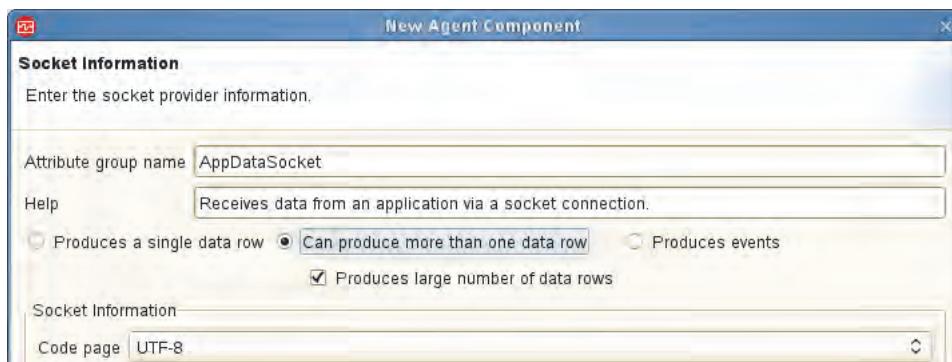
Furthermore, the agent must be able to process two error codes: one informational and the other an error.

43. On the **Data Sources** tab, right-click **AB2 (Agent)** and select **Add Data Source**.
44. On the Agent Initial Data Source page or the Data Source Location page, click **Custom programs** in the Monitoring Data Categories area.
45. In the Data Sources area, click **Socket**.



46. Click **Next**.

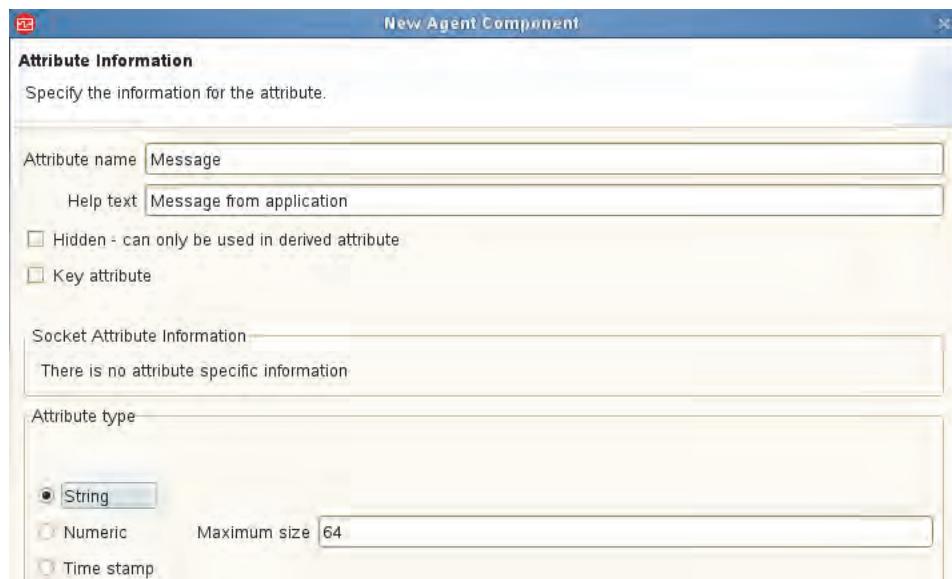
47. On the Socket Information page, enter the following information about the first attribute.
 - Attribute group name: **AppDataSocket**
 - Help text: **Receives data from an application via a socket connection.**
 - Can produce more than one data row: **Selected**
 - Code page: **UTF-8**



48. Click **Next**.

49. On the Attribute Information page, enter the following information about the first attribute:

- Attribute name: **Message**
- Help text: **Message from application**
- Hidden: **Not selected**
- Key attribute: **Not selected**
- Attribute type: **String**
- Maximum size: **64**



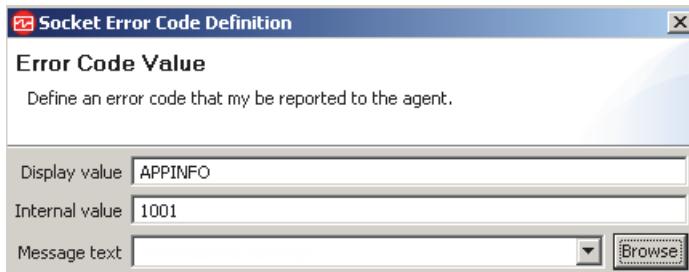
50. Click **Next**.



51. Create the following error codes:

Display value	Internal value	Error type	Message text
APPINFO	1001	Informational	Informational message.
APPERROR	1002	Error	Error message.

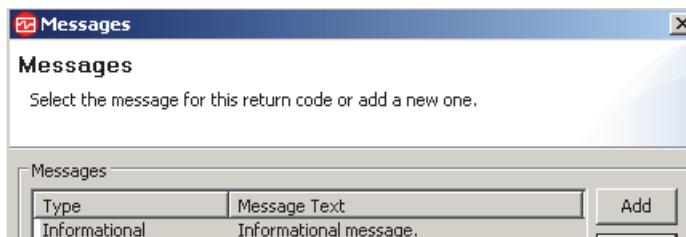
- a. Click **Add**.
- b. Enter the **Display value** and the **Internal value**.



- c. Click **Browse**.
- d. Click **Add**.
- e. Set the message type in the **Type** list and enter the message text in the **Message text** field.



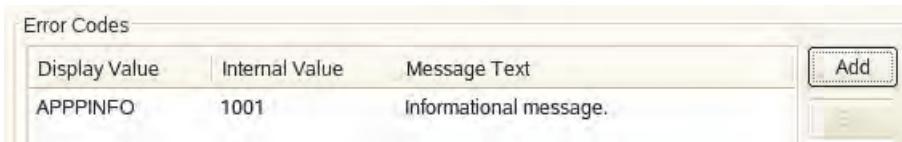
- f. Click **OK** to save the message definition.



- g. Click **OK** to close the Messages window.

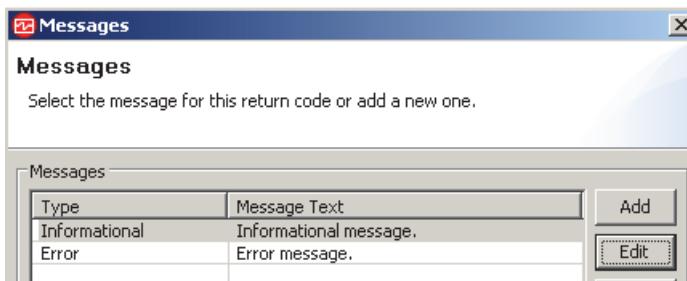


- h. Click **OK** to save the error code.

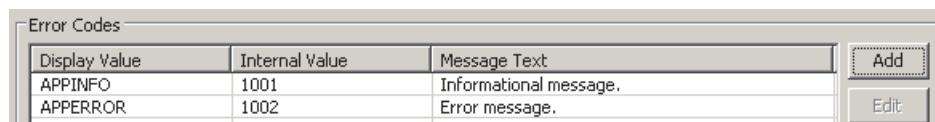


- i. Repeat these steps for the second message.

When you create the second message, the Messages window looks like the following screen capture.



When you create the second error code, the Error Codes section of the Global Socket Data Source Information window looks like the following screen capture.



52. Click **Finish**.

The wizard closes and the **Data Sources** tab opens.

Attribute Group Information

- AB2 (Agent)
 - Availability filters
 - Availability_Filtered (Filter)
 - AppDataSocket (Socket)**
 - Processor_Utilization (Script)

53. Save your agent project.

54. Add the following three attributes to the AppDataSocket data source.

Attribute name	Help text	Key attribute	Attribute type
MsgCode	Message code.	Yes	Numeric, Gauge, 32-bit
MsgCnt	Message count.	No	Numeric, Gauge, 32-bit
QueueSize	Message queue size.	No	Numeric, Gauge, 32-bit

- a. Right-click **AppDataSocket (Socket)** and click **Add Attribute**.
- b. Enter the attribute information and click **Finish**.
- c. Repeat for remaining two attributes.



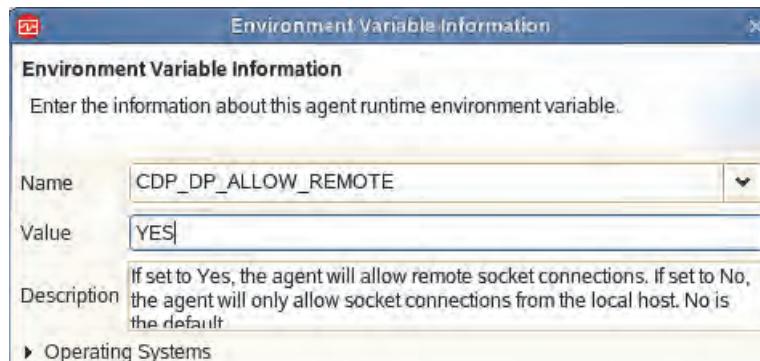
55. Enable the agent to receive socket connections from remote clients.

- a. Go to the Agent Information page and select **Environment Variables**.



Hint: If necessary, scroll down.

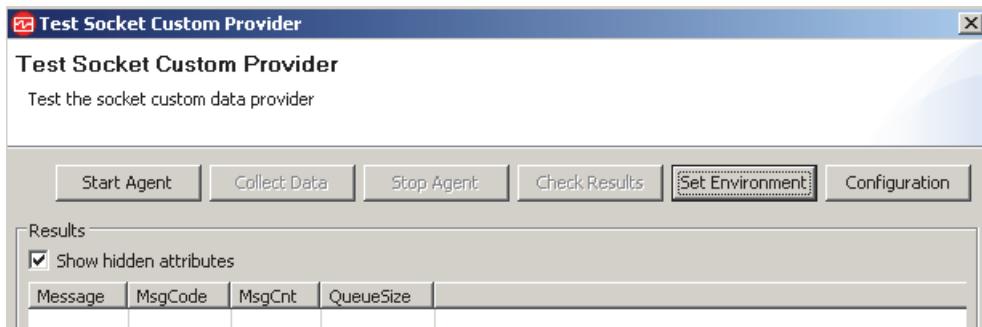
- b. Click **Add**.
- c. In the **Name** field, select **CDP_DP_ALLOW_REMOTE** from the list of environment variables.
- d. Set the **Value** field to **YES**.



- e. Click **OK** to save.

56. Test the AppSocketData data source.

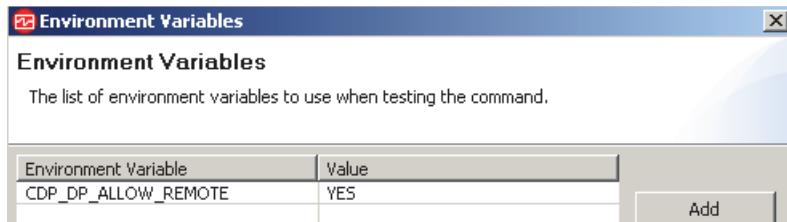
- Go to the **Data Sources** tab, click **AppDataSocket (Socket)**.
- Click **Test**. If necessary, scroll down.



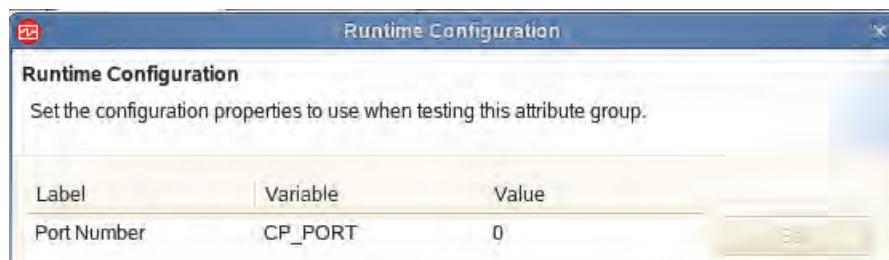
- Click **Set Environment** and confirm the CDP_DP_ALLOW_REMOTE variable is set to **Yes**.



Note: You might set this variable here for the test session.



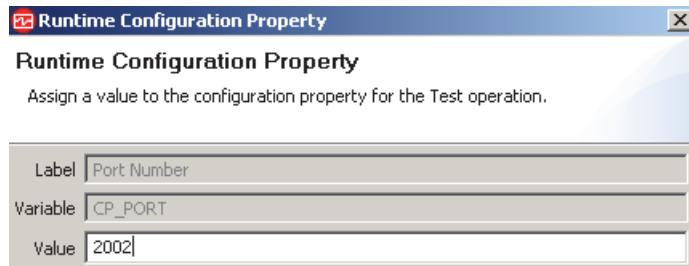
- Click **OK** to close the Environment Variables window.
- Click **Configuration**.



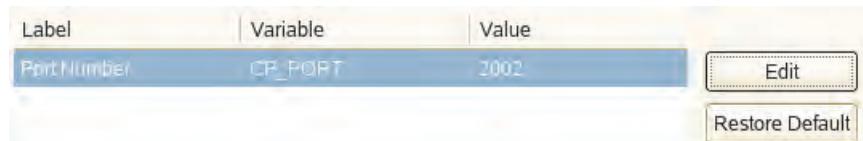
A CP_PORT value of 0 means that the agent uses any available port and writes that port number to the **k02_cps.properties** file. For the test to work, either your application must be able to read the **k02_cps.properties** file to send its data to that port, or you must configure the agent to listen on the port that the application sends to.

- Click the **Port Number** row and click **Edit**.

- g. Change the port value to **2002**.



- h. Click **OK** to save the new port number.



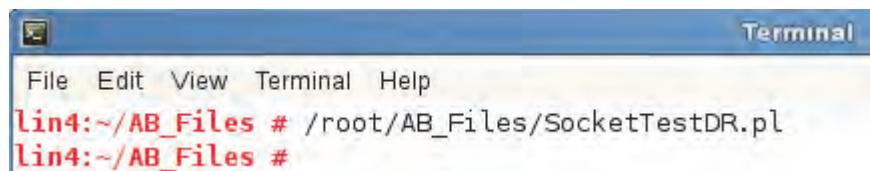
- i. Click **OK** to close the Runtime Configuration window.

- j. Click **Start Agent**.

The monitored application is on LIN4 and must connect to this agent here on WIN1.

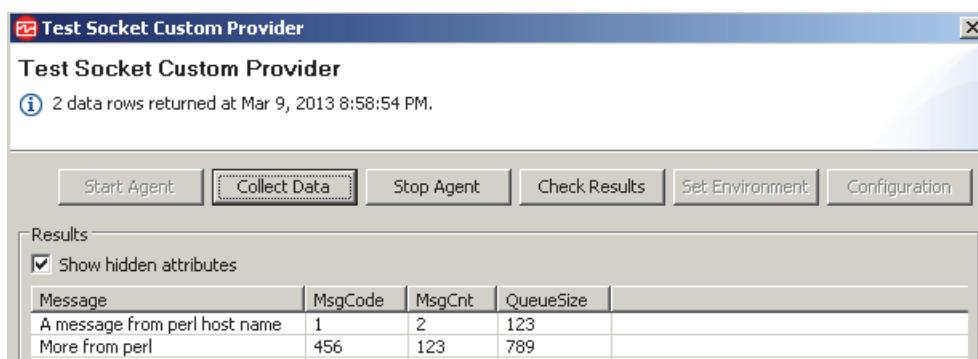
- k. On LIN4, open a terminal window and run this command.

```
/root/AB_Files/SocketTestDR.pl
```



SocketTestDR.pl sends two rows of data to WIN1 through port 2002.

- l. Return to Agent Builder on WIN1 and click **Collect Data**.



- m. Confirm that two rows of data are returned with values in each column that matches the **SocketTestDR.pl** script output.

- n. Click **OK** to close the test window.

57. Save your agent project.

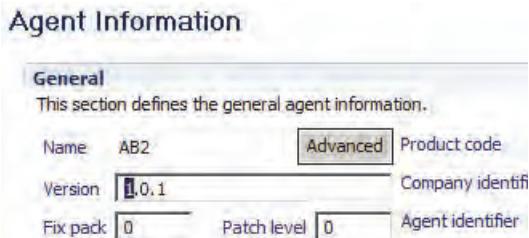
You successfully added a socket data source and enabled it to gather and monitor data from an application with a socket connection.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

58. Select the **Agent Information** tab.

59. Change the version for 1.0.0 to **1.0.1**.



Important: If your initial version number is not 1.0.1, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

60. Save your agent project.

Exercise 2 Install and confirm the updated AB2 agent in an IBM Performance Management environment

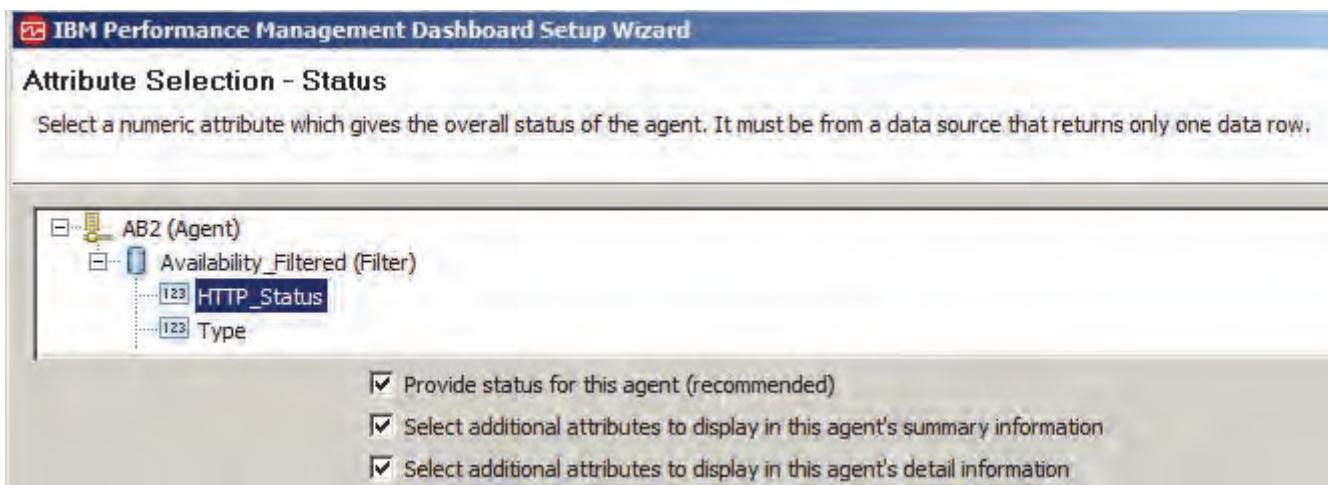
In this exercise, you install the AB2 agent onto WIN1 and LIN4. Then you confirm that data is being gathered for the new attribute groups you added.

1. Confirm that time is synchronized between the APM, LIN4, and WIN1 servers.
2. Confirm that all APM services are running on APM.

Add the script and socket data sources to the AB2 detailed dashboard

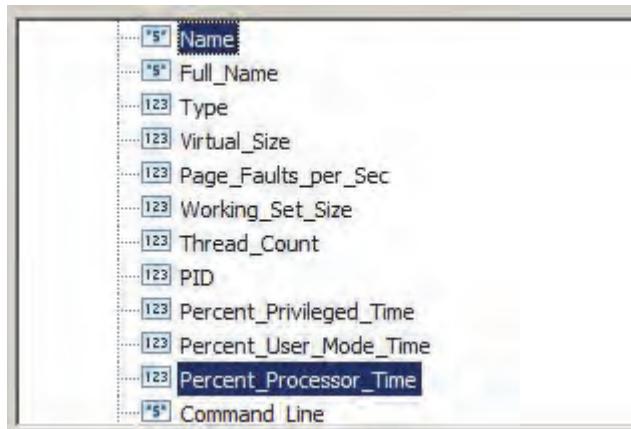
The AB2 agent already has a Summary dashboard and OSLC resource definitions. Use the Agent Builder Dashboard Setup wizard to add the script and socket data sources to the AB2 detailed dashboard.

3. Select **Dashboards** from the Agent Builder Outline view.
The Dashboards Overview opens.
4. Click the **Dashboard Setup wizard** link in the Overview.
The IBM Performance Management Dashboard Setup wizard opens.



5. Keep the current status attribute and click **Next**.

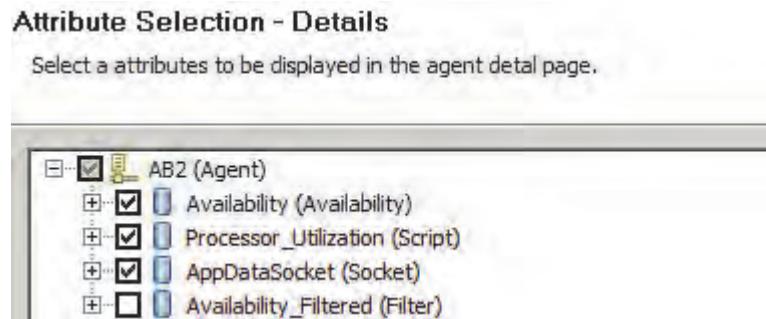
The Attribute Selection - Summary view opens.



6. Keep the current attributes and click **Next**.

The Attribute Selection - Details view opens.

7. Select **Processor_Utilization (Script)** and **AppDataSocket (Socket)**.



You do not need to change any other dashboard or resource properties.

8. Click **Finish** to complete the wizard.

9. Save your agent project.

You successfully modified the summary dashboard, detailed dashboard, and the monitored resource data within OSLC.

Create the updated AB2 agent installation scripts

1. In Agent Builder on WIN1, generate the AB2 agent installation scripts. (See ["Create the installation files"](#) on page 26 for detailed steps.)
2. Delete the contents of the **C:\share\K01** directory.
3. Copy the .zip installation archive in the **C:\Users\Administrator** directory into the **C:\share\K01** directory.

4. Extract the .zip installation archive into the K01 directory.



Hint: From Windows Explorer, right-click the **smai-ab2-01.01.00.00.zip** file and select **7-Zip > Extract Here**.

You successfully generated your installation archives and extracted an installer to the WIN1 share.

Reinstall the AB2 agent on WIN1

In this section, you reinstall the agent on WIN.

5. On WIN1, stop the AB2 agent with either of the following actions:
 - Open the IPM utility, right-click the agent, and select **Stop**.



- Open a command prompt and run the following command:

```
\IBM\APM\BIN\ab2-agent.bat stop
```

```
C:\Users\Administrator>\IBM\APM\BIN\ab2-agent.bat stop
Stopping Monitoring Agent for AB2 ...
Monitoring Agent for AB2 stopped.

C:\Users\Administrator>
```

6. Open a command prompt and change to the **C:\share\K01** directory.
7. Install the agent by running the following command:

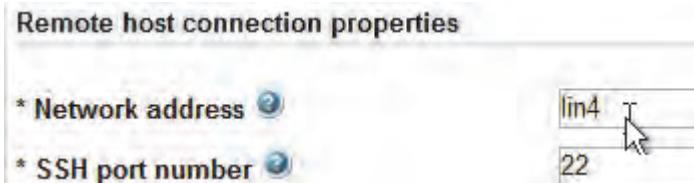
```
installIraAgent.bat C:\IBM\APM\
```

8. Configure the agent.

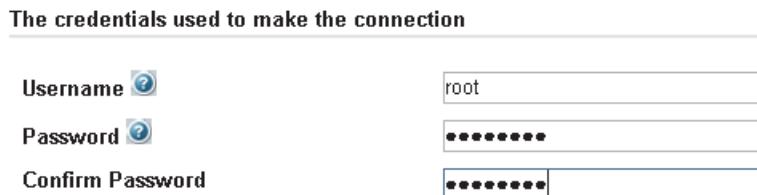
- Locate the AB2 agent in the IPM utility. If necessary, refresh the view.
- Right-click the agent and select **Reconfigure**.



- Enter **LIN4** in **Network address**.



- Click **Next**.
- Enter the following values in the Password section:
 - ◆ Username: **root**
 - ◆ Password: **object00**
 - ◆ Confirm password: **object00**



- Click **Next**.
- Enter **2002** in the **Port Number** field.



- Click **OK** to close the agent configuration window and save your changes.

9. Start the agent.

You completed the installation of the AB2 agent (K02) agent on WIN1.

Reinstall the agent on LIN4

In this section, you reinstall the agent onto the LIN4 system from the WIN1 share directory.

10. On LIN4, run the following command to stop the AB2 agent:

```
/opt/ibm/apm/agent/bin/ab2-agent.sh stop
```

```
lin4:~ # /opt/ibm/apm/agent/bin/ab2-agent.sh stop
Processing. Please wait...
Stopping Monitoring Agent for AB2 ...
Monitoring Agent for AB2 was stopped gracefully.
lin4:~ #
```

11. Change to the **/mnt/share/K01** directory where you extracted the agent on the WIN1 server.

12. Install the agent on this computer by running the following command:

```
./installIraAgent.sh /opt/ibm/apm/agent
```

The AB2 agent now requires configuration for the script and socket data sources.

13. Configure the agent on LIN4.

- Open a terminal window and start the agent configuration by running the following commands:

```
cd /opt/ibm/apm/agent/bin/
./ab2-agent.sh config
```

```
lin4:~ # cd /opt/ibm/apm/agent/bin/
lin4:/opt/ibm/apm/agent/bin # ./ab2-agent.sh config
Configuring Monitoring Agent for AB2
```

```
Edit 'Monitoring Agent for AB2' settings? [ 1=Yes, 2=No ] (default is: 1):
```

- Press Enter to confirm that you want to configure the agent.

You are prompted for the host name or IP address of the system where the monitoring script (script1.sh) is run.

- Enter **LIN4** and press **Enter**.

```
SSH Remote Connection :
Remote host connection properties

Host name or IP address of remote system
Network address (default is: ): lin4
```

- d. Press Enter to accept port 22.

```
The port number used for SSH communication.  
SSH port number (default is: 22): █
```

- e. Type **1** and press Enter to set a password.

```
The type of authentication to use to make the connection  
Authentication Type [ 1=Password, 2=Public Key ] (default is: ): 1
```

- f. Press Enter to select **No** to disconnecting after each collection interval.

```
Disconnect from the remote system after each data collection interval. By default,  
the SSH communication socket will establish an authenticated session at the first  
connection and will leave the connection active until the agent is stopped.  
Disconnect from the remote system after each collection interval [ 1=Yes, 2=No ]  
(default is: 2):
```

- g. Press Enter to select **No** to deleting the script after each collection interval.

```
Delete the script(s) from the remote system after each data collection interval.  
By default, the script(s) will be uploaded to the remote system at the first connection  
and will only be refreshed if the local copy changes. The script(s) will be removed  
from the remote system immediately before the agent is stopped.  
Remove script(s) from the remote system after each collection interval [ 1=Yes,  
2=No ] (default is: 2):
```

- h. Enter **root** for the user name and press Enter.

```
Password :  
The credentials used to make the connection  
  
Username for the remote system  
Username (default is: ): root█
```

- i. Enter **object00** twice for the password.

```
Password used for the remote system.  
Enter Password (default is: ):
```

```
Re-type : Password (default is: ):
```

You are prompted for the socket data source port. Since the socket application runs locally and it is configured to read the **k02_cps.properties** file, you can use the default port of 0, which uses any available port.

- j. Do not enter a value and press **Enter** to keep the default value of 0.

```
Socket :  
Socket Data Source
```

The port that the agent will use to listen on for data from socket clients. A value of 0 indicates an ephemeral port will be used.

Port Number (default is: 0):

You receive a prompt that indicates the agent configuration is complete.

14. Start the agent by running the following command:

```
./ab2-agent.sh start
```

15. Run the following command to confirm that the AB2 agent is running:

```
./ab2-agent.sh status
```

You successfully installed the agent on LIN4.

Confirm the agent changes in the Performance Management Console

In this section, you confirm that data from your script and socket data sources appears in the Performance Management console.

16. Create monitoring data for the socket data source of your agents. Use the Perl scripts on LIN4 to send data to the AppDataSocket data source on both the LIN4 and WIN1 servers.

- a. On LIN4, run the following commands to send monitoring data through a socket to the WIN1 and LIN1 agents:

- ◆ /root/AB_Files/SocketTestDL.pl
- ◆ /root/AB_Files/SocketTestDR.pl

```
Lin4:~/AB_Files # /root/AB_Files/SocketTestDL.pl  
Lin4:~/AB_Files # /root/AB_Files/SocketTestDR.pl  
Lin4:~/AB_Files #
```

The DL script sends monitoring data to the local host (LIN4) and determines the correct port by reading the k01_cps.properties file that is generated by the agent. The DR script sends monitoring data to the WIN1 host through port 2002.

- b. If you are interested in the commands that are used to determine the port and send the data, open each script in a text editor.

17. Log in to the **APM** server as user **root** with password **object00**.

18. Open the Performance Management console by opening a browser and go the following URL:

```
https://apm:9443/
```



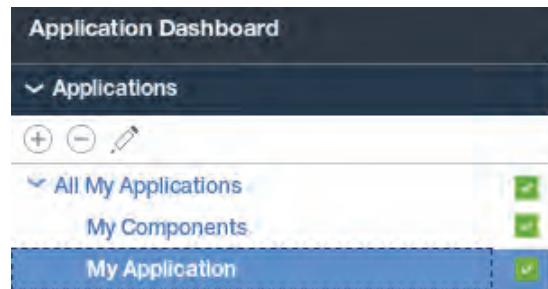
Hint: A bookmark to this URL is in the browser.

19. Log in to the Performance Management console as user **apmadmin** with password **object00**.

20. Open the Application Performance dashboard.

Because the AB2 agent is already a part of My Application, you do not need to edit the application definition.

21. Click **My Application** in the Applications explorer.



22. Click the **AB2** component bar in Current Components Status.



The AB2 Status Overview window opens with a Summary dashboard for each component.

Status Overview	Events
	WIN1 - AB2
HTTP Status	UP
Name	httpd.exe
Percent Processor Time	0
	lin4 - AB2
HTTP Status	UP
Name	httpd2-prefork
Percent Processor Time	0

23. Confirm the LIN1 script and socket data.

- Click the **lin4 - AB2 Summary** dashboard widget to access the detailed dashboards.
- Scroll down and confirm the Processor_Utilization and AppDataSocket data.

Processor_Utilization	
User	0
System	0
Idle	100
Wait	0
HostName	lin4
IP Address	192.168.1.107
Node	lin4:01
Timestamp	4/12/17, 8:12 PM
Concat	lin4:192.168.1.107

The Processor_Utilization dashboard shows the data that is generated by the script1.sh script and the derived attributes that you created. Notice that both the LIN4 and WIN1 agents are monitoring the same host: LIN4. The WIN1 agent connects to LIN4 and runs the script remotely.

AppDataSocket					
Message	MsgCode	MsgCnt	QueueSi..	Node	Timestamp
A message from perl 1002	1	2	123	lin4:02	5/10/16, 1:48 PM
More from perl	456	123	789	lin4:02	5/10/16, 1:48 PM

The AppDataSocket dashboard shows the data that is submitted to the agent by the monitored application. Notice that both the LIN4 and WIN1 agents are monitoring applications on LIN4.

One application sends its data to the LIN4 agent. The other application, running on LIN4 also, sends its data to the WIN1 agent.

24. Confirm the WIN1 agent shows data.

You successfully confirmed your agent in the Performance Management console. That the agent was available to be added to an application, the Summary dashboard shows the core metrics that you added. A subset of attributes is visible in the **Details dashboard**.

Exercise 3 Create an agent to monitor a log file and a custom Java data source

In this exercise, you do the following tasks:

- Create an agent that extracts data from a log file
- Create an agent that gathers data from a Java application with the Java API data source
- Create a custom runtime configuration property that captures the location of the log file to be monitored
- Install the agent and application support

You complete all steps in this exercise on the WIN1 server.

Determine the log file data for the agent to gather

In this section, you look at the log file from which the agent extracts data to better understand why the agent is configured the way that it is. Complete the following steps:

1. On the WIN1 server, open a command prompt and change to the following directory:
C:\AB_Files\Scripts
2. Use the **type** command to display the contents of the **pinginterval.bat** file.

```
C:\Users\Administrator>cd \AB_Files\Scripts
C:\AB_Files\Scripts>type pinginterval.bat
:START
c:\windows\system32\ping -w 100 LIN4>>c:\AB_Files\Output\ping.txt
c:\windows\system32\ping -w 100 WIN1>>c:\AB_Files\Output\ping.txt
GOTO START
C:\AB_Files\Scripts>_
```

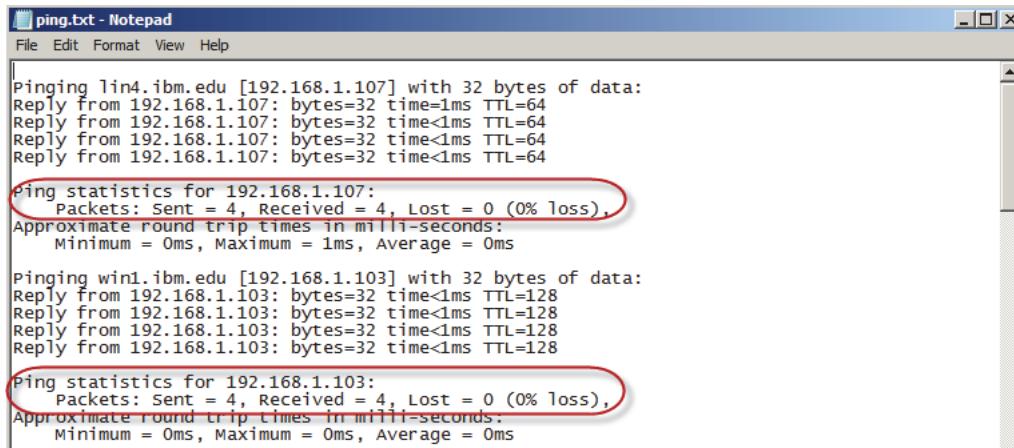
This script pings two hosts, appends the results to a mock log file (C:\AB_Files\Output\ping.txt) and repeats this process until stopped.

3. Run the **pinginterval.bat** file from a command prompt for 15 seconds to gather log data and then cancel (Ctrl + C, and Y) the batch file.

```
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 LIN4 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 WIN1 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>GOTO START
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 LIN4 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 WIN1 1>>c:\AB_Files\Output\ping.txt
C:\AB_Files\Scripts>GOTO START
C:\AB_Files\Scripts>c:\windows\system32\ping -w 100 LIN4 1>>c:\AB_Files\Output\ping.txt
^CTerminate batch job (Y/N)? Y
```

4. Find the **ping.txt** file in C:\AB_Files\Output and view its contents.

You monitor the **ping.txt** log file. A sample of the content of the **ping.txt** log file that is created by **pinginterval.bat** is shown here.



```

ping.txt - Notepad
File Edit Format View Help

Pinging lin4.ibm.edu [192.168.1.107] with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time=1ms TTL=64
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

Pinging win1.ibm.edu [192.168.1.103] with 32 bytes of data:
Reply from 192.168.1.103: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

The circled rows contain the data that you want to extract; specifically, the IP address, packets sent, packets received, and packets lost.



Note: Even though **ping.txt** is not a true log file and Agent Builder has a data source specifically built to monitor devices with ICMP ping, **ping.txt** provides an excellent target for learning how to monitor a log file. With it, you can see some of the complex parsing abilities you can use in an agent.

The goal of this exercise is to create an agent that pulls specific information to help you monitor the status of the two hosts LIN4 and WIN1.

The data that you want the agent to extract from the log file is shown in the following screen capture.

Node	Timestamp	IP Address	Sent	Received	Lost	Status	Avg Lost	HostName	Concat	Path
ITM:03	07/12/15 18:24:31	192.168.1.107	4	4	0	Good	0	lin4.ibm.edu	lin4.ibm.edu: 192.168.1.107	C:\IBM\ITM\GSK8_x64\lib64
ITM:03	07/12/15 18:24:28	192.168.1.103	4	4	0	Good	0	win1.ibm.edu	win1.ibm.edu: 192.168.1.103	C:\IBM\ITM\GSK8_x64\lib64
ITM:03	07/12/15 18:24:25	192.168.1.107	4	4	0	Good	0	lin4.ibm.edu	lin4.ibm.edu: 192.168.1.107	C:\IBM\ITM\GSK8_x64\lib64
ITM:03	07/12/15 18:24:22	192.168.1.103	4	4	0	Good	0	win1.ibm.edu	win1.ibm.edu: 192.168.1.103	C:\IBM\ITM\GSK8_x64\lib64

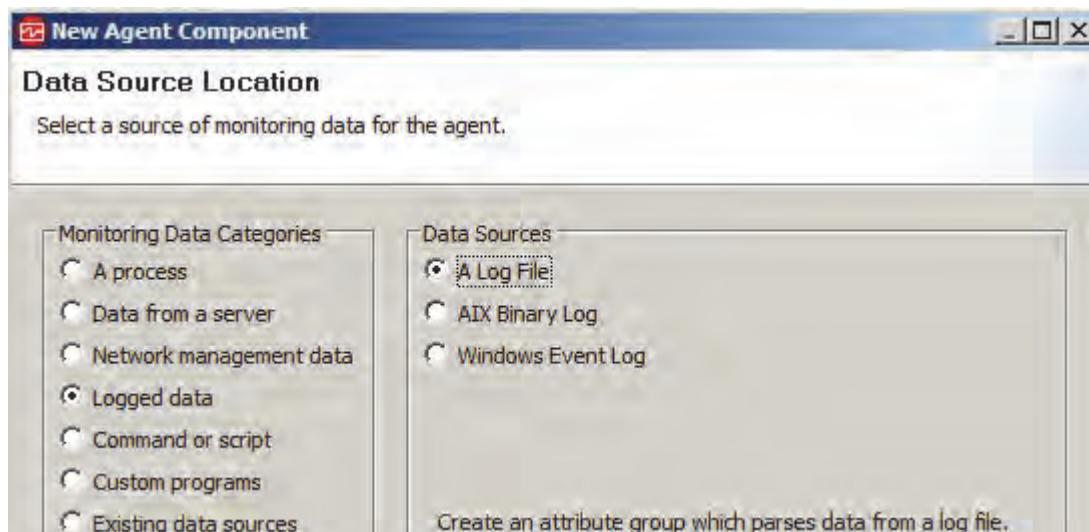
The agent must do the following parsing tasks to the log file:

- Recognize multiple lines in the log file as one record
- Extract data that is not delimited in a consistent way
- Ignore all the unwanted lines of data

Modify the AB1 agent to monitor the ping.txt log file

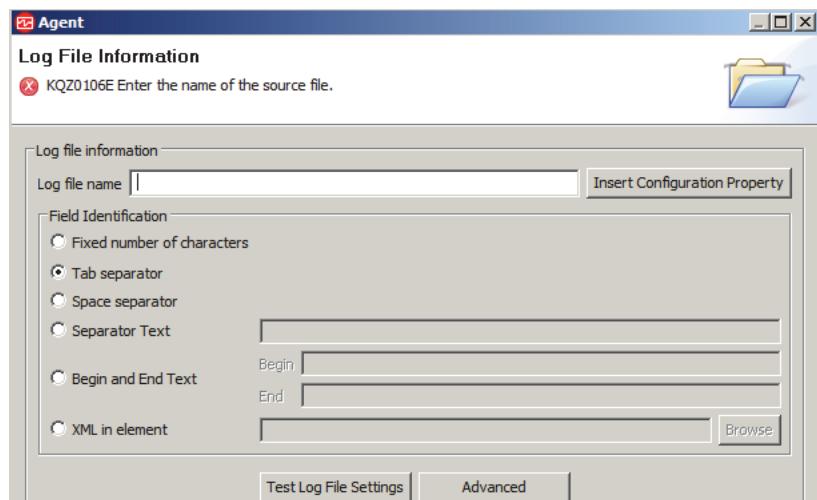
Complete the following steps:

5. If Agent Builder on the WIN1 server is not already open, start it.
6. Close the **AB2 Agent Editor** window if it is open.
7. If the **AB1** agent is not already open, expand **Agent 1** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
8. Click the **Data Sources** tab.
9. Right-click **AB1 (Agent)** under **Attribute Group Information** and click **Add Data Source**.
10. Select **Logged data** under Monitoring Data Categories and **A Log File** under Data Sources.



11. Click **Next**.

The Log File Information window opens.



First, you identify the log file to monitor. Enter the log file name in one of the following ways:

- **Full path and file name:** If the log file on each host to be monitored is the same name and in the same location, enter the full path and file name in the **Log file name** field.
- **Different paths or file names per platform:** If the file name or path to the file differs for different operating systems, you can identify the different files names and paths in the Advanced Log File Attribute Group Information window. Each log file entry requires a unique label.
- **Entered at agent configuration:** Part or all the log file path and name can come from a runtime configuration property that is entered in the agent configuration panes after the agent is installed on a specific host. To do this, click **Insert Configuration Property**.
- **Dynamic File Name:** For applications that create output files with dynamic file names, you can configure the **File log name** field to match patterns. For example, **IN{#####}.log** matches file names that start with IN followed by six numeric characters and file extension **.log**.
- **Regular Expression:** Like dynamic file naming, you can use regular expressions to define a pattern to select the correct log file or files to monitor.

You use a simple full path and file name.

12. Create and insert a runtime configuration property that inserts the location of the **ping.txt** log file. This property is set during agent configuration with a value that is appropriate for the agent host.

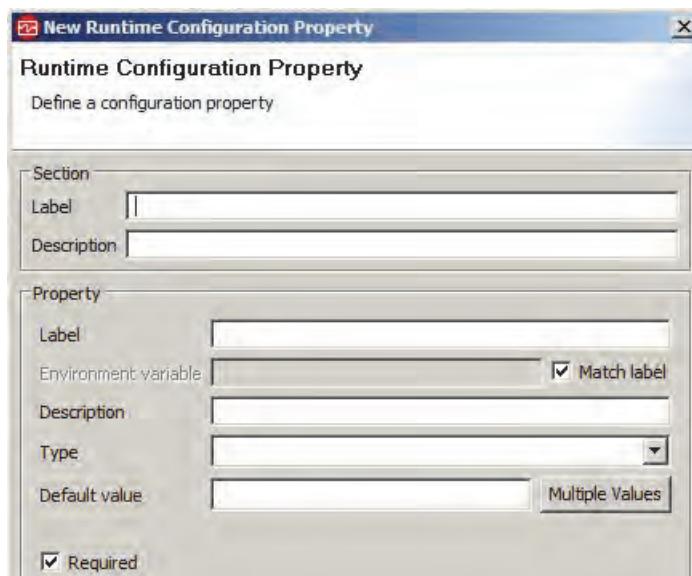
- a. Click in the **Log file name** field and click **Insert Configuration Property**.

The Configuration Properties window opens.



- b. Click **Add**.

The Runtime Configuration Property window opens.



- c. Enter the following information:

◆ Section

Label: **Log_File**

Description: **Runtime configuration properties for log file monitoring**

◆ Property

Label: **File_location**

Description: **Location of the log file**

Type: **String**

◆ Default value: **C:\AB_Files**



Note: For this exercise, the default value is intentionally entered incorrectly. You correct the value when you install and configure the agent.

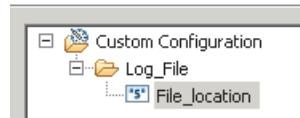
Section		
Label	Log_File	
Description	Runtime configuration properties for log file monitoring	
Property		
Label	File_location	
Environment variable	K00_FILE_LOCATION	<input checked="" type="checkbox"/> Match label
Description	Location of the log file	
Type	String	
Default value	C:\AB_Files	<input type="button" value="Multiple Values"/>
<input checked="" type="checkbox"/> Required		

- d. Click **OK** to save the property.

Your File_location property is listed in the Configuration Properties window.

Configuration Properties

Select the property to insert.



- e. Ensure the **File_location** property is selected and click **OK**.

The Log File Information window is displayed with the File_location property inserted into the **Log file name** field.

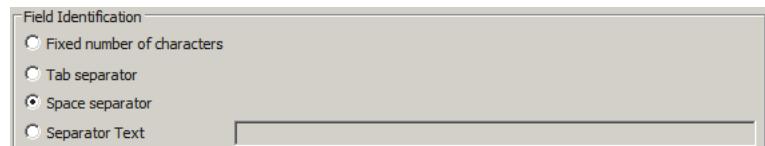
Log file information
Log file name \${K00_FILE_LOCATION}
Field Identification

13. Type \ping.txt in the **Log file name** field after the **File_location** property.

Log file information
Log file name \${K00_FILE_LOCATION}\ping.txt
Field Identification

Field Identification defines the default method of identifying each data point to be retrieved. The default is **Tab**, but the **ping.txt** file uses spaces.

14. Click **Space separator**.



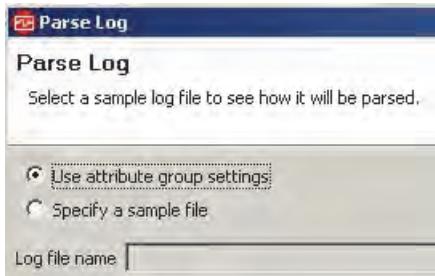
15. Test your log file filter.

- Click **Test Log File Settings**.



Note: If **Test Log File Settings** is unavailable, you missed a setting, such as the space separator from [Step 14](#) on page 6-42.

The Parse Log window opens. Notice that your log file name and path are not displayed.



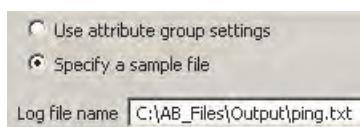
If you entered a specific directory path and file name, it is listed in the **Log file name** field. You do not have to test the exact path and file name that is used by the agent. If the log file you want to test is different from the one shown in the **Log file name** field, click **Specify a sample file** and change the value to the appropriate path and file name. In this lab exercise, you browse and locate the **ping.txt** file that you created earlier in this exercise.

- Click **Specify a sample file** and **Browse**.
- Locate and click the **C:\AB_Files\Output\ping.txt** file.



- Click **Open** to select the file and close the browser.

You are returned to the Parse Log window.



- Click **Start Agent**.

A status window opens and closes. You are notified that the agent is started.

The test agent has been started. Log files can be found in C:\DOCUMENTS\ADMINISTRATOR\LOCALS\Temp\KQZ_1352920363564\TMAITM6\logs.

f. Click **Collect Data**.

g. If prompted, click **No** to view suggested data types.

The resulting data is displayed.

The screenshot shows the 'Parse Log' application interface. At the top, it displays 'Parse Log' and a message indicating '2702 data rows returned at May 12, 2016 6:23:17 PM.' Below this, there are two radio button options: 'Use attribute group settings' (unchecked) and 'Specify a sample file' (checked). A 'Log file name' input field contains 'C:\AB_Files\Output\ping.txt' with a 'Browse' button to its right. A row of buttons includes 'Start Agent' (disabled), 'Collect Data' (highlighted in blue), 'Stop Agent', 'Check Results', 'Set Environment', and 'Configuration'. The main area is titled 'Results' and contains a table with the following data:

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7
Pinging	lin4.ibm.edu	[192.168.1.107]	with	32	bytes	of
Reply	from	192.168.1.107:	bytes=32	time<1ms	TTL=641	
Reply	from	192.168.1.107:	bytes=32	time<1ms	TTL=641	

The parser treats each line as a record (row) and creates an attribute that is based on the space separator. The initial goal is to ensure that you can find the line that you want and that all of the data you want is on one line. Find rows that contain the data you want.

The screenshot shows the 'Parse Log' application interface with the 'Results' table. Four specific rows are highlighted with a red border: the first row ('Pinging'), the second row ('Reply'), the third row ('Reply'), and the fourth row ('Approximate'). These rows represent the data extracted from the log file.

Reply	from	192.168.1.107:	bytes=32	time<1ms	TTL=641	
Ping	statistics	for	192.168.1.107:			
Packets:	Sent	=	4,	Received	=	4,
Approximate	round	trip	times	in	milli-seconds:2	

Unless defined otherwise, each row of a log file is considered an independent record. For the **ping.txt** log file, the data that you want is spread across two lines.

```
-- Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging lin4.ibm.edu [192.168.1.107] with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

For this log file, you must define a record as consisting of at least two rows. But notice all the extra lines of data that does not represent exact data that you want to return. Not only must you define how to identify a record, but you also must configure the agent to ignore unwanted data. One way is to expand your record definition to include more rows as part of the record.

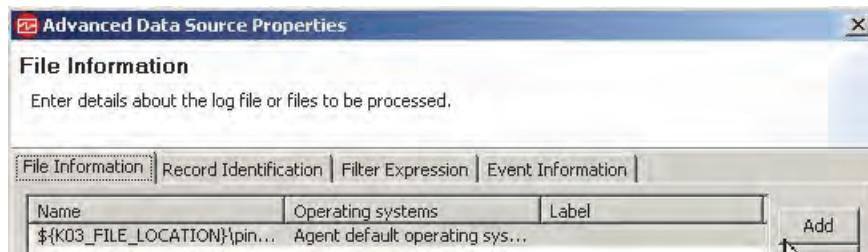
For example, this image shows that the **ping.txt** file uses ten text lines for a record, from Pinging to Minimum.

```
-- Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging lin4.ibm.edu [192.168.1.107] with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging win1.ibm.edu [192.168.1.103] with 32 bytes of data:
Reply from 192.168.1.103: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

16. Configure the agent to identify ten rows as one record.

- Click **Cancel** to exit the Parse Log utility without saving any attribute configurations.
- Click **Advanced** at the bottom of the window.

The Advanced Data Source Properties window opens to the **File Information** tab.

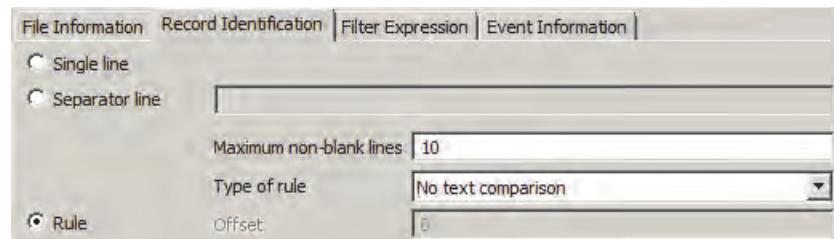


In the **File Information** tab, you enable the agent to process different log files on different operating systems. The other options define how the agent reads the log file and processes the records.

- c. Click the **Record Identification** tab.

The **Record Identification** defines a record.

- d. Click **Rule** in **Record Identification**.



- e. Enter **10** in **Maximum non-blank lines**.

- f. Click **OK** to save these settings.

The Log File Information window is displayed. The log file definition is not yet completed, but you can start testing it.

17. Test your log file filter again (see [Step 15](#) for detailed steps).

The resulting data is displayed.

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7	Attribute_8
Pinging	lin4.ibm.edu	[192.168.1.107]	with	32	bytes	of	data:
Pinging	win1.ibm.edu	[192.168.1.103]	with	32	bytes	of	data:
Pinging	lin4.ibm.edu	[192.168.1.107]	with	32	bytes	of	data:

All the data elements are on a single row in the parser.

18. Scroll right to find the data you want.

```
time<1ms TTL=641 Reply from 192.168.1.107: bytes=32 time<1ms TTL=641 Ping statistics for 192.168.1.107: Packets: Sent = 4, R
time<1ms TTL=1282 Reply from 192.168.1.103: bytes=32 time<1ms TTL=1282 Ping statistics for 192.168.1.103: Packets: Sent = 4,
time<1ms TTL=643 Reply from 192.168.1.107: bytes=32 time<1ms TTL=643 Ping statistics for 192.168.1.107: Packets: Sent = 4, R
time<1ms TTL=1284 Reply from 192.168.1.103: bytes=32 time<1ms TTL=1284 Ping statistics for 192.168.1.103: Packets: Sent = 4,
```

Ensure that you find rows that contain Ping statistics with all your data points on the same row, including the IP address, Sent, Received, and Lost packets.

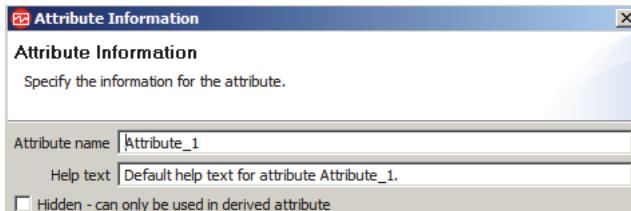
Notice any extra lines of data that do not represent your record or data you want to return. There are many ways to configure the agent to ignore that data.

Next, you define the first four attributes to filter for the correct data. It is best to work from left to right. You want the IP address to display in Attribute_1. The idea is to configure Attribute_1 to ignore data from the start of the record to the data you want in this attribute.

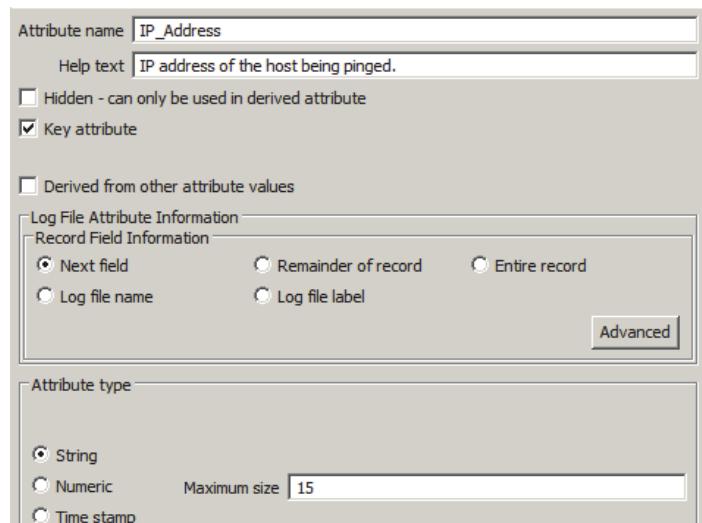
19. Define Attribute_1 so that it is defined correctly for your target data (IP address), is a key attribute, and retrieves the IP address.

- Stop the agent.
- Right-click a cell in the **Attribute_1** column and select **Edit Attribute**.

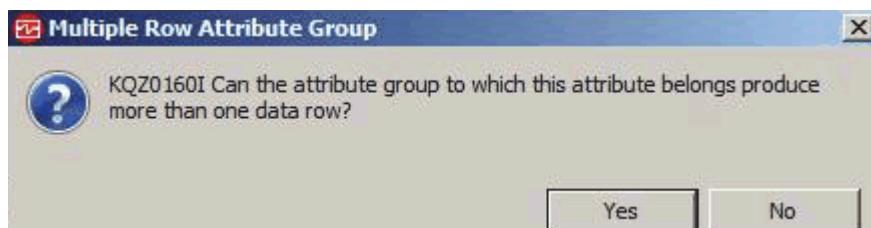
The Attribute Information window opens.



- Modify the following properties:
 - ◆ Attribute name: **IP_Address**
 - ◆ Help text: **IP address of the host being pinged.**
 - ◆ Key attribute: **Selected**
 - ◆ String Maximum size: **15**



- If prompted, select **Yes** the attribute group can produce more than one row.



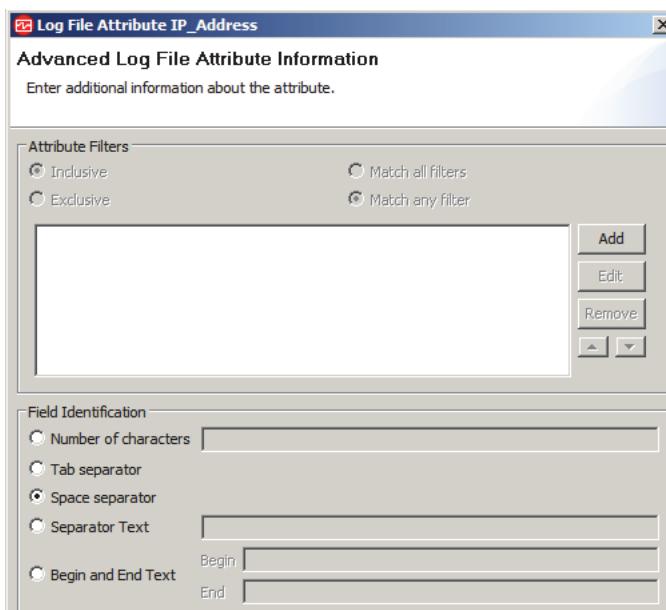
The Record Field Information is set to **Next field** and the attribute type is **String**. Because the global delimiter is a space, this definition does not skip over the first fields to retrieve the

IP address. You need a unique filter. Specify the information that comes before and after the text you want.

```
Reply from 192.168.1.13: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
```

e. Click **Advanced**.

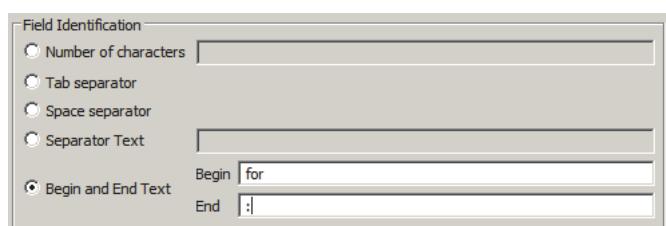
The Advanced Log File Attribute Information window opens.



The Field Identification pane defines how to distinguish the data for this attribute. This setting overrides the global delimiter, which is a space for your agent. In this case, you configure the Field Identification items to get whatever lies between the word **for** and a colon (:).

f. Select **Begin and End Text**.

g. Enter **for** in the **Begin** field and a colon (**:**) in the **End** field.



h. Click **OK** to save the Advance Log File Attribute Information.

i. Click **OK** to save your new IP_Address attribute.

- j. If prompted, click **No** to any prompts to change the data type after redefining an attribute.
 The Parse Log window opens with IP_Address as the first column header.
20. Click **Start Agent** and **Collect Data** to reparse the ping.txt file based on your new attribute definition.
21. If prompted, click **No** to change the data type after redefining an attribute.

Results						
<input checked="" type="checkbox"/> Show hidden attributes						
IP_Address	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7
192.168.1.107	Packets:	Sent	=	4,	Received	=
192.168.1.103	Packets:	Sent	=	4,	Received	=
192.168.1.107	Packets:	Sent	=	4,	Received	=

You now see all rows that start with the IP address and no rows that start with words you do not want.

22. Edit the other attributes so that they select for their specific values and that they are configured to the data they hold. Modify Attributes 2, 3, and 4 to match the information in the following table:

	Attribute_2	Attribute_3	Attribute_4
Attribute name	Sent	Received	Lost
Help Text	Number of packets sent.	Number of packets received.	Number of packets lost.
Key	No	No	No
Attribute Type	Numeric > Gauge	Numeric > Gauge	Numeric > Gauge
Begin delimiter	Sent =	Received =	Lost =
End delimiter	,	,	(



Note: For the Begin delimiter, include a space between *Sent*, *Received*, *Lost*, and the equal sign.

- ◆ Answer **No** to any prompts to change the data type after redefining an attribute.
- ◆ You can click **Start Agent** and **Collect Data** as you finish each attribute definition to ensure that it parses correctly.
- ◆ Click **Stop Agent** before editing the next attribute.

After you define all four attributes, your results look like the following example.

Results									
<input checked="" type="checkbox"/> Show hidden attributes									
IP_Address	Sent	Received	Lost	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	
192.168.1.107	Packets:	Sent	=	4,	Received	=	4,	Lost	

After you parse the log file again, your output looks like the following example.

Results									
<input checked="" type="checkbox"/> Show hidden attributes									
IP_Address	Sent	Received	Lost	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	
192.168.1.107	4	4	0	0%	loss),	Approximate	round	trip	tir

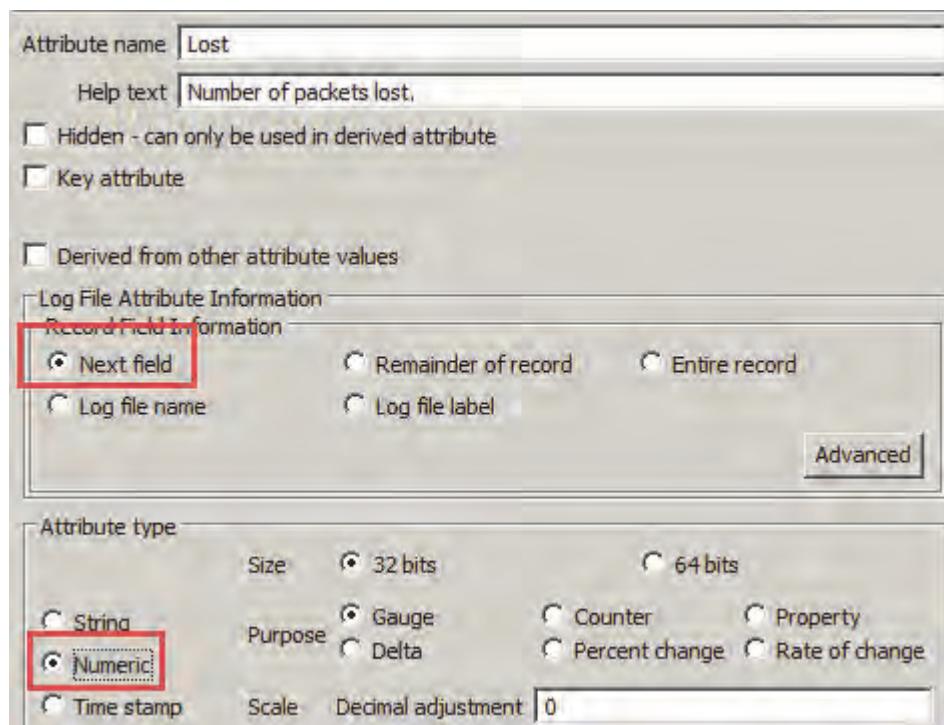
In some instances, you might have blank lines in your Parse Log results. There are several ways to remove blank links. Attribute filters are a common way to remove unwanted rows (records). You might add a filter to IP_Address, or you might set a filter on another attribute. In this instance, you modify the Sent attribute to exclude rows where Sent is equal to zero.

23. Right-click the **Lost** cell for a data row and select **Remove Subsequent Attributes**.

Results									
<input checked="" type="checkbox"/> Show hidden attributes									
IP_Address	Sent	Received	Lost	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	
192.168.1.107	4	4	0	0%	loss),	Approximate	round	trip	tir

Edit Attribute
 Hide Attribute
 Insert Attribute Before
 Insert Attribute After
 Remove
Remove Subsequent Attributes
 Remove All

The final attribute is redefined to hold the remainder of the record. Therefore, it gets changed to String instead of what Lost was, Numeric

24. Reset **Lost** to the **Next field** and **Numeric**.25. Click **OK** to close the Parse Log window and save these attribute definitions.

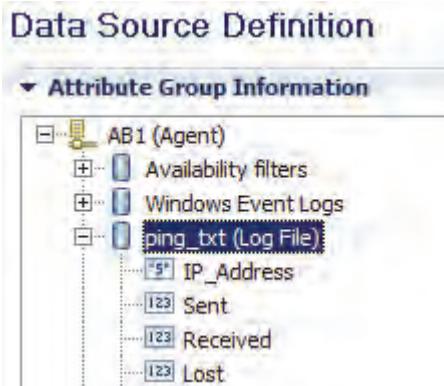
Note: Clicking **OK** to exit is important; it saves your attribute definitions. If you click **Cancel**, you lose the attribute definitions and must return to the Parse Log utility to redo them.

The **Next** and **Finish** buttons are active. You completed enough steps to finish this agent. If you wanted to continue working in the wizard to modify this agent, such as adding more data sources, you would click **Next**.

26. Click **Finish** to exit the Agent wizard and save your changes.

You are returned to the **Data Sources** tab.

27. Expand **ping_txt (Log File)**.



You see your four defined attributes.

28. If you still have unwanted attributes, you can remove them here.

29. Replace the attribute group name, **ping_txt**, with **Host_Availability**.

a. Click **ping_txt (Log File)**.

b. Change **Attribute group name** to **Host_Availability**.

The screenshot shows the 'Attribute Group Information' section of the Data Source Definition tool. The 'ping_txt (Log File)' group has been renamed to 'Host_Availability (Log File)'. The 'Log File Attribute Group Information' panel at the bottom displays the 'Attribute group name' as 'Host_Availability'.

c. Press **Enter** to activate your change.

30. Save your agent project.

Add derived attributes

In this section, you add derived attributes to Host_Availability that do the following tasks:

- Give a status of Good or Failure to a pinged server based on the number of received packets.
- Calculate the average number of lost packets for the last 10 pings.



Note: The addition and testing of these derived attributes might be done at the same time you defined the original four attributes in the Parse Log utility. It is done here to show that you can define new attributes and test your log file from the **Data Source** tab in the Agent Editor.

31. Add a derived attribute to Host_Availability according to the following specifications:

- Attribute name: **Status**
- Help text: **Displays status of host based on the number of received pings.**
- Derived Attribute Formula: **Received**
- Attribute Type: **Numeric > Gauge**
- Enumerations:

Name	Value	Severity
Good	4	Normal
Poor	3	Warning
Poor	2	Warning
Poor	1	Warning
Failure	0	Critical

a. Right-click **Host_Availability (Log File)** and click **Add Attribute**.

The Attribute information window opens.

b. Enter the **Attribute name** and **Help text** values.

Attribute name: Status
Help text: Displays status of host based on the number of received pings.
 Hidden - can only be used in derived attribute
 Key attribute

c. Set the attribute type to **Numeric > Gauge** to match the Received attribute.

d. Select **Derived from other attribute values**.

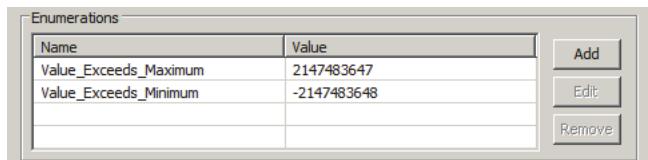
Derived from other attribute values
Derived Attribute Information
Formula: Edit
 Interval specific calculations

- e. Enter **Received** in the **Formula** field.

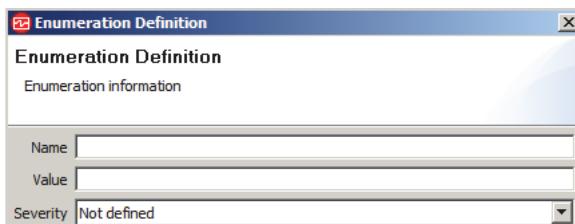


This action sets the new attribute to the value of the Received attribute.

- f. In the **Enumerations** section, click **Add**.

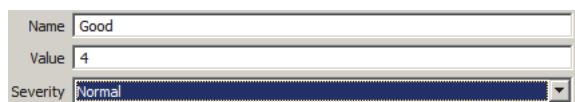


The Enumeration Definition window opens.



- g. Enter the following values:

- ◆ Name: **Good**
- ◆ Value: **4**
- ◆ Severity: **Normal**

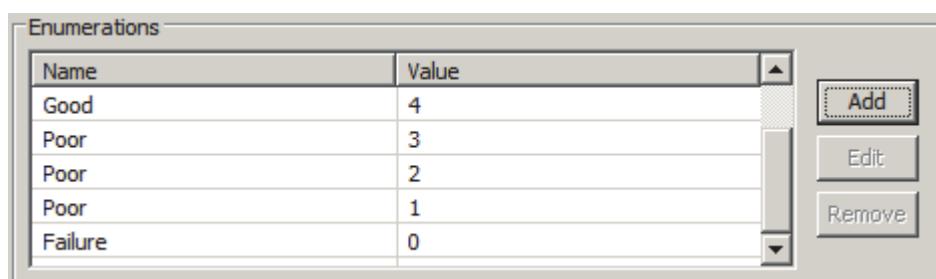


If Received is 4, Status is Good.

- h. Click **OK** to save the enumeration.

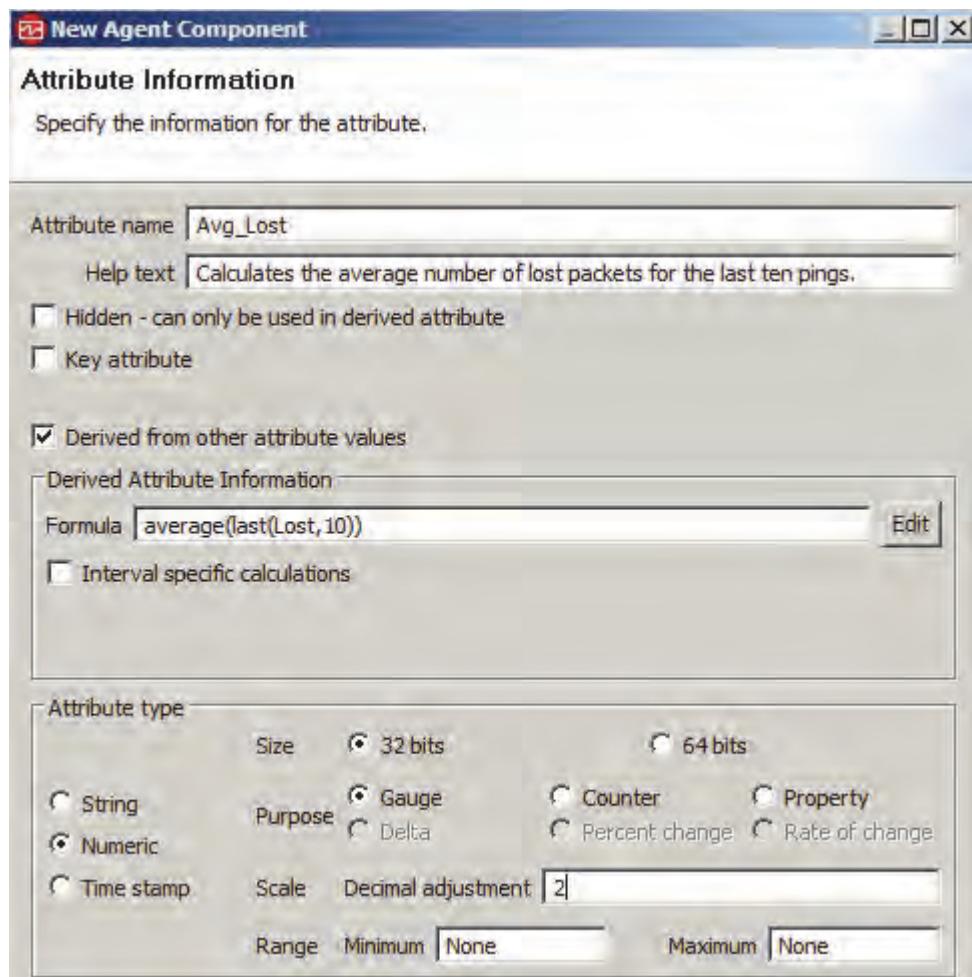
- i. Repeat the previous three steps for the other enumerations.

Your final **Enumerations** pane looks like the following example (order is not important).



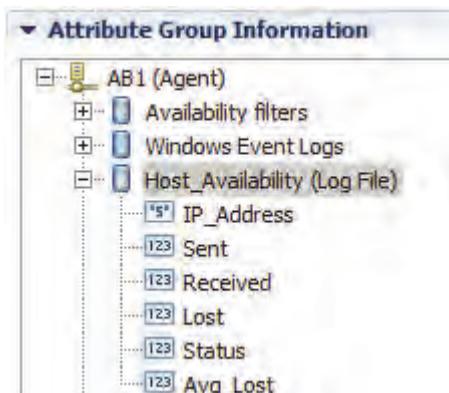
- j. Click **Finish** to close the Attribute Information window and save the **Status** attribute.
32. Add a derived attribute to Host_Availability according to the following specifications:
 - Attribute name: **Avg_Lost**
 - Help text: **Calculates the average number of lost packets for the last ten pings.**
 - Derived Attribute Formula:
`average(last(Lost,10))`
 - Attribute Type: **Numeric, Gauge, 32-bit, Decimal adjustment: 2**
- a. Right-click **Host_Availability (Log File)** and click **Add Attribute**.
The Attribute information window opens.
- b. Enter the **Attribute name** and **Help text** values.
- c. Set the attribute type to **Numeric > Gauge**.
- d. Set **Decimal adjustment** to **2**.
- e. Select **Derived from other attribute values**.
- f. Enter the following formula in the **Formula** field.
`average(last(Lost,10))`

The Attribute Information window looks like the following screen capture.



33. Click **Finish** to close the Attribute Information window.

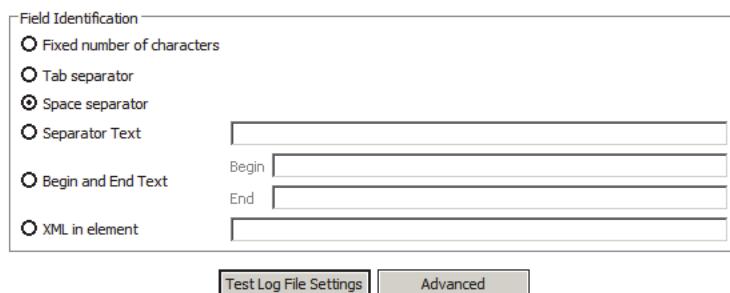
The new attributes are displayed on the **Data Source Definition** tab.



34. Save your agent project.

35. Test your log file parsing.

- Click **Host_Availability (Log File)**.
- Scroll down and click **Test Log File Settings** in the **Log File Attribute Group Information** section.



The Parse Log window opens.

- Click **Specify a sample log file**.
- Browse and locate the **ping.txt** file.
- Start the agent and collect data.

Your parsed log file looks like the following example.

Results						
<input checked="" type="checkbox"/> Show hidden attributes						
IP_Address	Sent	Received	Lost	Status	Avg_Lost	Attribute_7 (Remainder of record)
192.168.1.107	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
192.168.1.103	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
192.168.1.107	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
192.168.1.103	4	4	0	Good	0	0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms

Status shows Good for any rows with a Received value of 4 and Failure for any rows with a Received of 0. For rows with a Received value of 1, 2, or 3, the actual value is shown.

- Confirm that the derived values are shown.
- Scroll to the right as far as you can.

Attribute_7 (Remainder of record)
0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms
0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms

Notice the new attribute that holds the remainder of the record. You do not need this attribute. You might delete it now. Because you already defined all of your attributes before you came into the Parse Log utility, another solution is to click **Cancel** and exit without saving the new attribute.

- Click **Cancel** to close the **Parse Log** window and not save the new attribute.

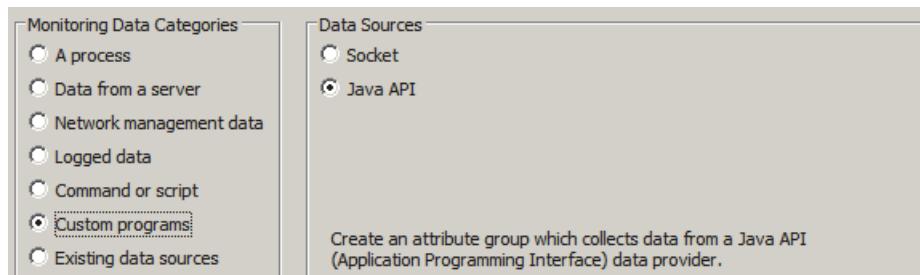
You created an agent that monitors a log file for specific data and derives new monitored data.

Add a Java API data source

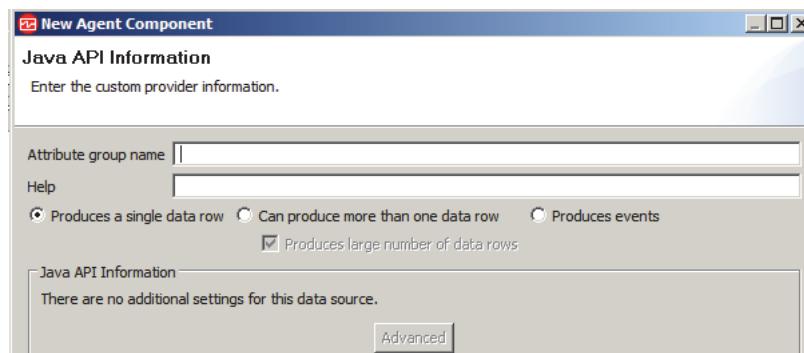
In this section, you add a Java API data source that gathers four data points from a custom Java application. The custom Java code that creates the monitored data is provided and you do not need to create it. An example of the data it generates is shown here.

Name	Integer Value	String Value	Float Value
One_SSN_arg1_One	14	One_SSN_arg1_One	11.12
One_SSN_arg2_One	15	One_SSN_arg2_One	12.12
One_SSN_arg3_One	16	One_SSN_arg3_One	13.12

36. Add a Java API data source that gathers one text string attribute called Name.
- In the **Data Sources** tab, right-click **AB1 (Agent)** and click **Add Data Source**.
The Data Source Location window opens.
 - Click **Custom Programs** under **Java API**.



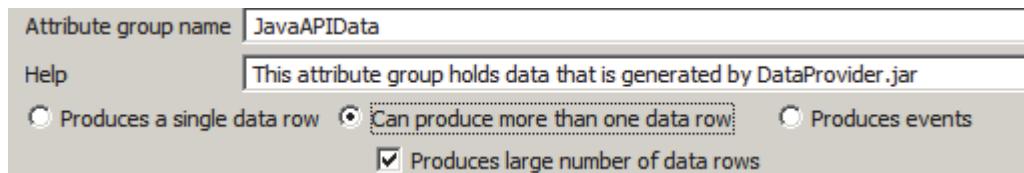
- Click **Next**.
The Java API Information window opens.



Here, you define an attribute group for data that is gathered by the custom Java application, **DataProvider.jar** in this instance.

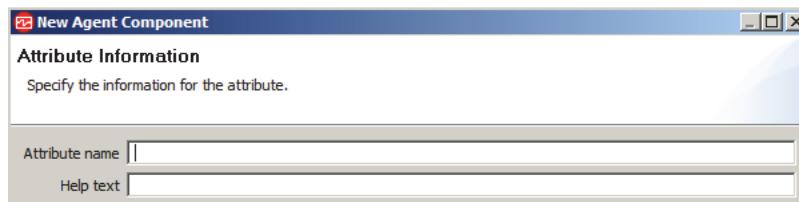
- Enter the following values:
 - ◆ Attribute group name: **JavaAPIData**
 - ◆ Help: **This attribute group holds data that is generated by DataProvider.jar**

- e. Select **Can produce more than one data row**.



- f. Click **Next**.

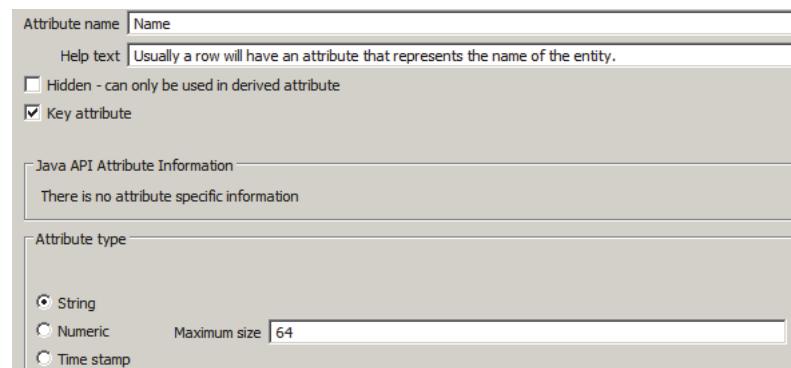
The Attribute Information window opens.



You must define one attribute while in the Agent wizard. Other attributes can be created when you leave the Agent wizard.

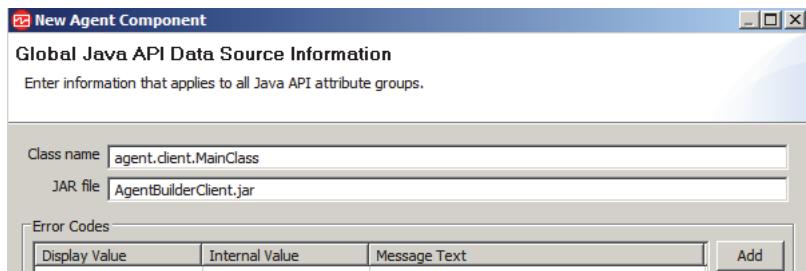
- g. Set the following values:

- ◆ Attribute name: **Name**
- ◆ Help text: **Usually a row will have an attribute that represents the name of the entity.**
- ◆ Key attribute: **Selected**
- ◆ Attribute type: **String, Maximum size: 64**



- h. Click **Next**.

The Global Java API Data Source Information window opens.

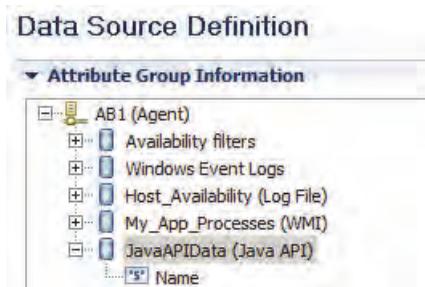


These files are provided with Agent Builder.

- i. Click **Finish**.

You are returned to the **Data Source Definition** tab.

37. Expand **JavaAPIData** and confirm the one attribute, **Name**, was created.



You add more attributes later in this exercise.

38. Save your agent project.

39. Test the JavaAPIData data source and confirm that it returns data the expected sample data.

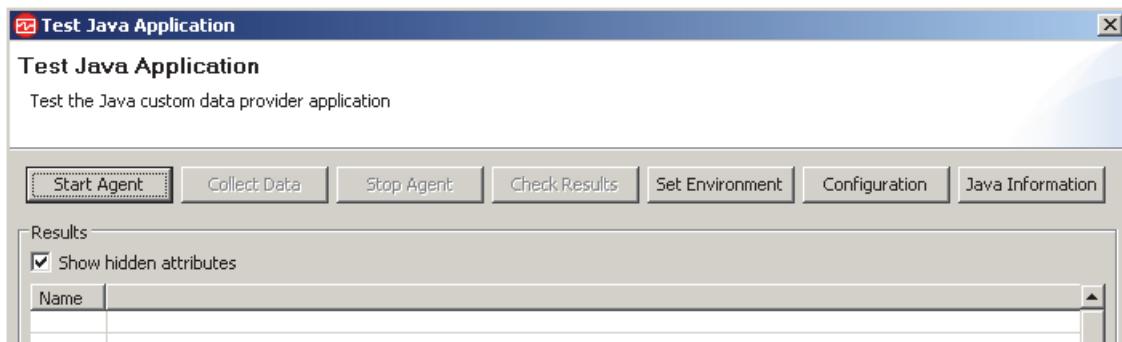
- a. Click **JavaAPIData (Java API)** in the **Data Source Definition** tab.
- b. Click **Test**.



Hint: If necessary, scroll down to see the **Test** button.

Attribute group name	JavaAPIData	
Help	Help: This attribute group holds data that is generated by DataProvider.jar	
<input type="radio"/> Produces a single data row	<input checked="" type="radio"/> Can produce more than one data row	<input type="radio"/> Produces events
<input checked="" type="checkbox"/> Produces large number of data rows		
<input checked="" type="checkbox"/> Add this attribute group to a reporting category		Performance
<input checked="" type="checkbox"/> Add this attribute group to the warehouse		
Refresh interval	minutes	
Java API Information		
There are no additional settings for this data source.		
<input type="button" value="Edit Java Source"/> <input type="button" value="Global Settings"/> <input type="button" value="Test"/> <input type="button" value="Advanced"/>		

The Test Java Application window opens.



The single attribute that you added to this data source, Name, is shown.

- c. Click **Start Agent** and **Collect Data**.

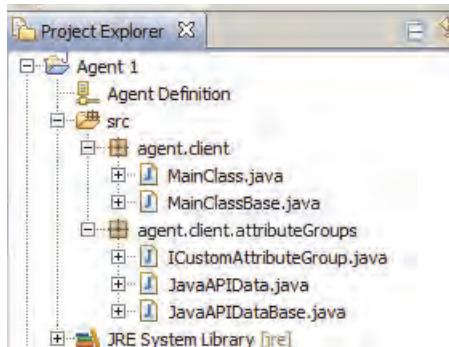
Results	
<input checked="" type="checkbox"/> Show hidden attributes	
Name	
arg1	
arg2	
arg3	

Confirm that three rows of data are returned. The JavaAPIData.java class that is created by Agent Builder generates this sample data. To enable this data source to gather real data, you must edit or replace this Java class.

- d. Click **OK** to close the Test Java Application window.

40. Identify the code that generated the sample data.

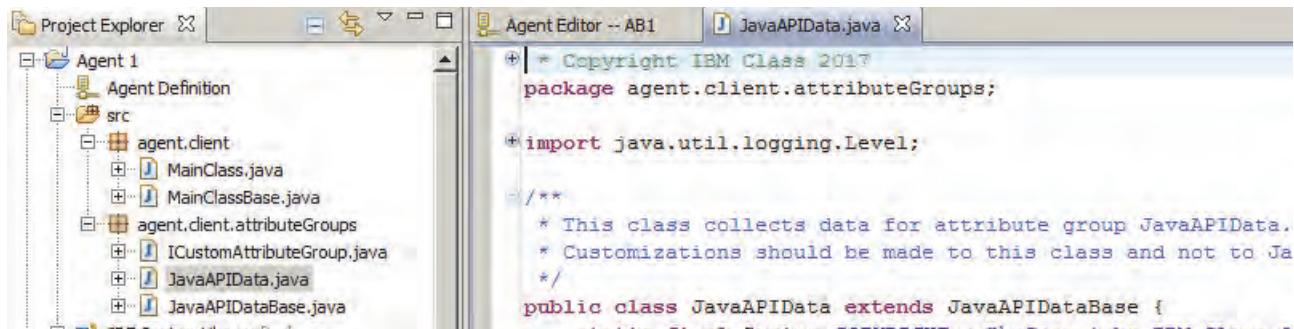
- In the Project Explorer, expand **Agent 1 > src**.
- Expand both **agent.client** and **agent.client.attributeGroups**.



The **agent.client** package contains the main code that interacts with the agent: a base class (**MainClassBase.java**) and a subclass that you can modify. The **agent.client.attributeGroups** package contains two classes for each attribute group. Again, the package contains a base class (<attr group name>**Base.java**) and a subclass that you can modify.

The Agent Builder always regenerates the base classes. The Agent Builder does not regenerate the subclasses unless they are not present.

- In the Project Explorer, double-click **JavaAPIData.java** to open the class in an editor.



- d. Locate the **collectData** section and the lines that generate the sample attributes for the Name attribute.

```

| public void collectData(Request request) throws Exception {
|     final String METHOD = "collectData";
|     logger.entering(CLASS, METHOD, request);
|     AgentConnection agentConnection = mainClient.getAgentConnection();
|     Attributes row;
|
|     // This example sends 3 data rows with arbitrary values.
|     // Replace the rows with data from the monitored resource.
|     for (int rowNum = 1; rowNum <= 3; ++rowNum) {
|         row = new Attributes(
|             "arg" + rowNum
|         );
|         logger.logp(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
|         row.setAttributeValues(request.getID());
|     }
}

```

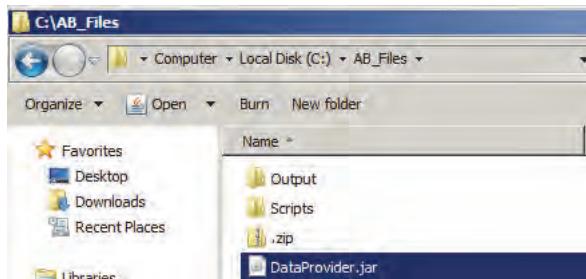
Agent Builder creates Java class similar to JavaAPIData for each Java API attribute group that you define and configures it to create sample data for each attribute that you initially add to the data source. You must modify this class to either generate the actual data you want collected and monitored or modify it to collect that data from custom Java code that generates the data.

You add a custom .jar file to this agent that generates the monitored data. You must modify the **JavaAPIData.java** class to gather the data from this custom .jar file.

- e. Close the JavaAPIData.java editor.

41. Locate the following file:

C:\AB_Files\DataProvider.jar



The DataProvider.jar file is a custom Java application that is created to generate custom data that this agent monitors. It represents custom code that you must create to generate the data that you want to gather and monitor. **DataProvider.jar** specifically generates four attributes that are called Name, Integer Value, String Value, and Float Value.

Name	Integer Value	String Value	Float Value
One_SSN_arg1_One	14	One_SSN_arg1_One	11.12
One_SSN_arg2_One	15	One_SSN_arg2_One	12.12
One_SSN_arg3_One	16	One_SSN_arg3_One	13.12

To enable your Agent Builder agent, you must do the following tasks:

- Add **DataProvider.jar** to your agent so that the agent can use it to generate the monitored data
- Add the three new attributes that are created by **DataProvider.jar**
- Re-create the **JavaAPIData.java** class to enable it to gather the new attributes
- Modify the **JavaAPIData.jar** class to gather the four attributes from **DataProvider.jar**

42. Add **DataProvider.jar** to your agent.

- a. In Agent Builder, go to the **Data Sources** tab of your agent.
- b. Click **JavaAPIData (Java API)**.
- c. Scroll down in the attribute group information pane and select **Global Settings**.



- d. Click **Add** to the right of **Supplemental Files**.
- e. Locate and open the **DataProvider.jar** file in the **C:\AB_Files** directory.



- f. Save your agent project.

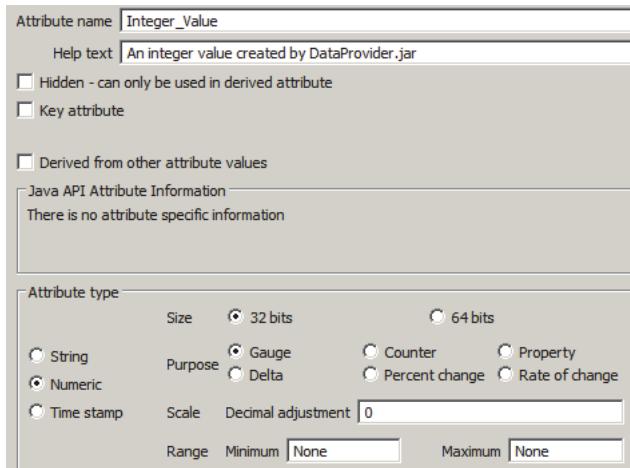
43. Add the three new attributes.

- a. In the **Data Sources** tab, ensure **JavaAPIData (Java API)** is selected and then click **Add to Selected** on the right.

The Attribute Information window opens.

- b. Set the following attribute properties:
 - ◆ Attribute name: **Integer_Value**
 - ◆ Help text: **An integer value created by DataProvider.jar**
 - ◆ Attribute type: **Numeric, 32-bit, Gauge**

The attribute information window looks like this screen capture.

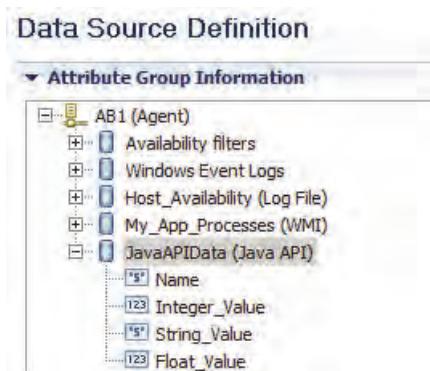


- c. Select **Add additional attributes** and click **Next**.
- d. Repeat this step for the remaining two new attributes according to this table.

Attribute name	Help text	Attribute type
String_Value	A string value that is created by DataProvider.jar	String with maximum size of 64
Float_Value	A floating-point value that is created by DataProvider.jar	Numeric, 64-bits, Counter, Decimal adjustment of 2

- e. Click **Finish** when you are done with the **Float_Value** attribute.

The **Data Source Definition** tab looks like this screen capture.



- f. Save your agent project.
44. Re-create the **JavaAPIData.java** class to enable it to gather the new attributes.
- a. If it is still open, close the JavaAPIData.java editor.
 - b. In the Project Explorer, double-click the **JavaAPIData.java** class to reopen it.

- c. Locate the **collectData** section as you did earlier in this lab.

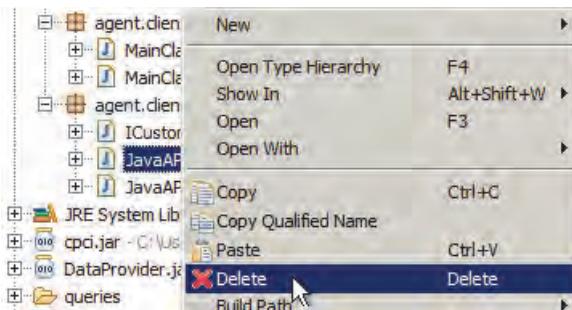
```
| public void collectData(Request request) throws Exception {
    final String METHOD = "collectData";
    logger.entering(CLASS, METHOD, request);
    AgentConnection agentConnection = mainClient.getAgentConnection();
    Attributes row;

    // This example sends 3 data rows with arbitrary values.
    // Replace the rows with data from the monitored resource.
    for (int rowNum = 1; rowNum <= 3; ++rowNum) {
        row = new Attributes(
            "arg" + rowNum
        );
        logger.logp(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
        row.setAttributeValues(request.getID());
    }
}
```

It still generates only data for one attribute. This behavior is expected. The Attribute Group class is created when the attribute group is initially defined and is not modified as you add more attributed. You must re-create this class.

To re-create this class with the new attributes delete the existing class, modify the agent in some manner and save those changes.

- d. Close the JavaAPIData.java editor.
e. Right-click the **JavaAPIData.java** class in the Project Explorer and click **Delete**.

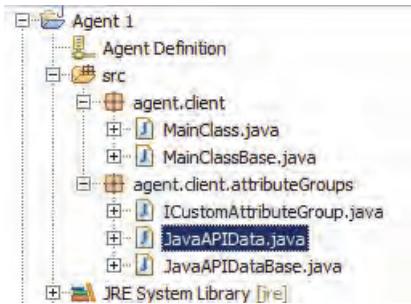


- f. Click **OK** to confirm the deletion.

Notice that you cannot save your agent project, which you need to do to re-create the JavaAPIData class. The next step changes the agent so that you can save your agent project. It then reverses the change because the change made was not needed.

- g. On the **Agent Information** tab, change the product code to **K02** and then change it back to **K00**.
You can now save your agent project.
h. Save your agent project.

The **JavaAPIData.java** class is re-created.



- i. Test the **JavaAPIData** data source and confirm that it returns values to all four attributes.

Results				
<input checked="" type="checkbox"/> Show hidden attributes				
Name	Integer_Value	String_Value	Float_Value	
arg1	11	arg1	11.12	
arg2	12	arg2	12.12	
arg3	13	arg3	13.12	

Sample values are again created by the **JavaAPIData.java** class. Next you modify that class to gather data from your custom Java application, **DataProvider.jar**.

45. Modify the **JavaAPIData.java** class to gather the four attributes from **DataProvider.jar**.

- a. Double-click the **JavaAPIData.java** class in the Project Explorer to open it in an editor.
- b. Locate the **collectData** section.

```
public void collectData(Request request) throws Exception {
    final String METHOD = "collectData";
    logger.entering(CLASS, METHOD, request);
    AgentConnection agentConnection = mainClient.getAgentConnection();
    Attributes row;

    // This example sends 3 data rows with arbitrary values.
    // Replace the rows with data from the monitored resource.
    for (int rowNum = 1; rowNum <= 3; ++rowNum) {
        row = new Attributes(
            "arg" + rowNum,
            10 + rowNum,
            "arg" + rowNum,
            10.12 + rowNum
        );
        logger.logp(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
        row.setAttributeValues(request.getID());
    }
}
```

- c. Replace the row arguments ("arg" through 10.12) with the following text:

```
DataProvider.getStringValue("Name"),
DataProvider.getIntValue("Integer_Value"),
DataProvider.getStringValue("String_Value"),
DataProvider.getFloatValue("Float_Value")
```



Hint: Type "DataProvider." and a menu appears. Double-click the correct **get** option from the menu. Type the value name with quotations.

With this code, the JavaAPIData.java class can gather the four attributes from **DataProvider.jar**.

The row section looks like this screen capture:

```
public void collectData(Request request) throws Exception {
    final String METHOD = "collectData";
    logger.entering(CLASS, METHOD, request);
    AgentConnection agentConnection = mainClient.getAgentConnection();
    Attributes row;

    // This example sends 3 data rows with arbitrary values.
    // Replace the rows with data from the monitored resource.
    for (int rowNum = 1; rowNum <= 3; ++rowNum) {
        row = new Attributes(
            DataProvider.getStringValue("Name"),
            DataProvider.getIntValue("Integer_Value"),
            DataProvider.getStringValue("String_Value"),
            DataProvider.getFloatValue("Float_Value")
        );
        logger.log(Level.FINER, CLASS, METHOD, "Sample data collection complete.");
        row.setAttributeValues(request.getID());
    }
}
```

- d. Save **JavaAPIData.java**.

- e. Return to the AB1 agent editor and test the JavaAPIData data source. Confirm it returns values to all four attributes.

Results				
<input checked="" type="checkbox"/> Show hidden attributes				
Name	Integer_Value	String_Value	Float_Value	
value of Name0	0	value of String_Value1	1.12	
value of Name2	2	value of String_Value3	3.12	
value of Name4	4	value of String_Value5	5.12	

Notice that values returned by the **DataProvider.jar** are different from those values that are originally created by the **JavaAPIData.java** class.

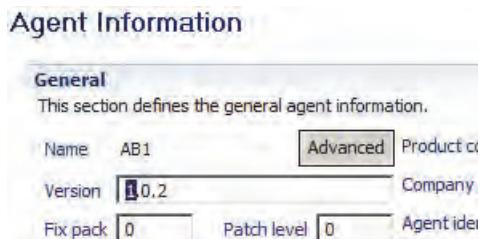
You successfully added a Java API data source and enabled it to gather and monitor data from a custom Java application.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

46. Select to the **Agent Information** tab.

47. Change the version for 1.0.1 to **1.0.2**.



Important: If your initial version number is not 1.0.1, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

48. Save your agent project.

Exercise 4 Install and confirm the updated AB1 agent in an IBM Performance Management environment

In this exercise, you install the AB1 agent onto WIN1. Then, you confirm that data is being gathered for the new attribute groups you added.

1. Confirm that time is synchronized between the APM and WIN1 servers.
2. Confirm that all APM services are running on APM.

Add the log file and Java API data sources to the AB1 agent detailed dashboard

The AB1 agent already has a Summary dashboard and OSLC resource definitions. Use the Agent Builder Dashboard Setup wizard to Add the log file and Java API data sources to the AB1 agent detailed dashboard.

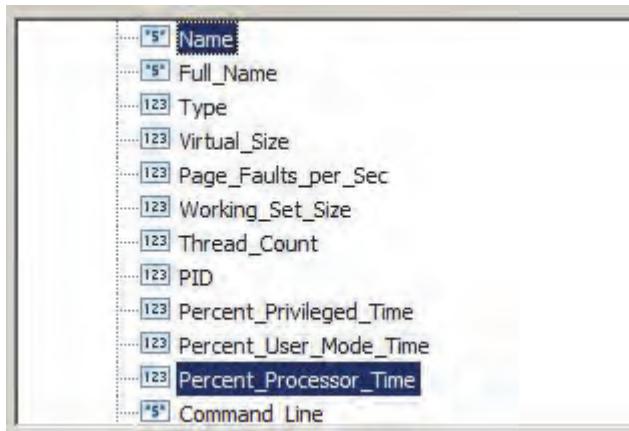
Create dashboards and resource definitions

3. Select **Dashboards** from the Agent Builder Outline view.
The Dashboards Overview opens.
4. Click the **Dashboard Setup wizard** link in the Overview.
The IBM Performance Management Dashboard Setup wizard opens.



5. Keep the status attribute and click **Next**.

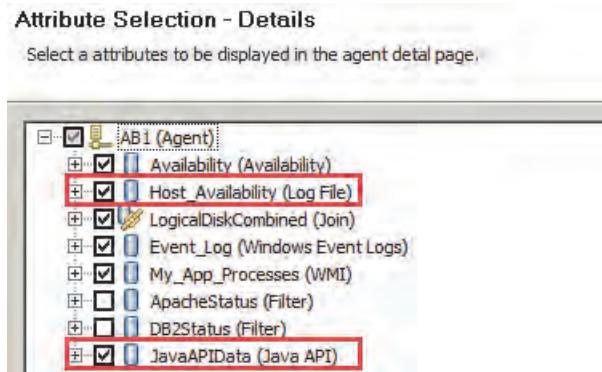
The Attribute Selection - Summary view opens.



6. Keep the current attributes and click **Next**.

The Attribute Selection - Details view opens.

7. Select **Host_Availability (Log File)** and **JavaAPIData (Java API)**.



You do not need to change any other dashboard or resource properties.

8. Click **Finish** to complete the wizard.
9. Save your agent project.

You successfully defined the summary dashboard, detailed dashboard, and the monitored resource data for OSLC.

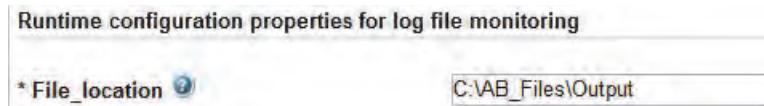
Reinstall the AB1 agent on WIN1

In this section, you install the agent on WIN1.

10. Stop the AB1 agent on WIN1.
11. Create the agent installers and install the agent in WIN1 in any manner you would like.

12. Configure the AB1 agent.

- a. In the IPM utility, right-click the AB1 agent and select **Reconfigure**.
- b. Replace **C:\AB_Files** with **C:\AB_Files\Output** in the **File_location** field.



- c. Click **Next**.

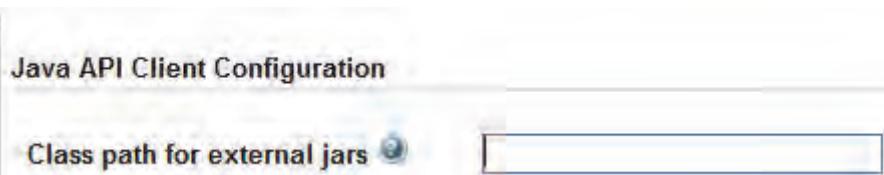
You are prompted for Java configuration parameters. The default location of Java (Java 50) is not valid on this host.

- d. Browse to and change the Java home path to the following directory:

`C:\Program Files (x86)\ibm\Java70`



- e. Click **Next**.



Do not set the Java API class path to external .jar files.

- f. Click **OK** to close the configuration editor and save your changes.

13. Start your **Monitoring Agent for AB1**.

You completed the installation of the AB1 agent (K02) agent on WIN1.

Confirm the AB1 modifications in the Performance Management Console

In this section, you confirm that data from your log file and Java API data sources appears in the Performance Management console.

14. Generate the log file data.

- a. On WIN1, go to the following directory and delete the current **ping.txt** file.

`C:\AB_Files\Output`

- b. Open a command prompt and change to the following directory:
C:\AB_Files\Scripts
- c. Run the following command:
pinginterval.bat

15. Log in to the **APM** server as user **root** with password **object00**.

16. Open the Performance Management console by opening a browser and go the following URL:

<https://apm:9443/>



Hint: A bookmark to this URL is in the browser.

17. Log in to the Performance Management console as user **apmadmin** with password **object00**.

18. Open the Application Performance dashboard.

Because the AB1 agent is already a part of My Application, you do not need to edit the application definition.

19. Click **My Application** in the Applications explorer.

20. Click the **AB1** component bar in Current Components Status.

The AB1 Status Overview window opens with a Summary dashboard for each component.

The screenshot shows the IBM Performance Management interface. At the top, there's a navigation bar with 'All My Applications > My Application > Components >'. Below it, a blue square icon with a white 'AB1' label is selected. Underneath, there are two tabs: 'Status Overview' (which is currently active) and 'Events'. The main area displays a summary for the 'WIN1 - AB1' component. It includes a small icon of a computer monitor, the component name 'WIN1 - AB1', and several status metrics: 'HTTP Status' is 'UP' with a green checkmark; 'Percent Processor Time' is '0'; 'DB2 Status' is 'UP'; and 'Percent Processor Time' is '0' again. There are also small icons for each metric.

21. Click the **WIN1 - AB1 Summary** dashboard widget to access the detailed dashboards.

22. Scroll through the data and confirm the log file (Host_Availability) and Java API (JavaAPIData) data is displayed.

The screenshot shows the IBM Performance Management console interface. At the top, there are three tabs: "Status Overview" (underlined), "Events" (with a green checkmark icon), and "Attribute Details". Below the tabs, there are two main sections:

- Host_Availability:** This section displays network statistics. The table has columns: IP Address, Sent, Received, Lost, Status, Avg Lost, HostName, Concat, and PathEnv. One row is shown: 192.168.1.107, 4, 4, 0 Good, 0.00, lln4.ibm.edu, lln4.ibm.edu: 192.168.1.107, and C:\IBM\.
- JavaAPIData:** This section displays data from the Java API. The table has columns: Name, Integer V... String Value, Float Val... Node, and Timestamp. Two rows are shown: "value of Name288" with value 288 and timestamp 5/11/16, 5:35 PM; and "value of Name290" with value 290 and timestamp 5/11/16, 5:35 PM.

The Host_Availability dashboard shows the data that is generated by the log file data source monitoring the ping.txt file.

The JavaAPIData dashboard shows the data that is created by the DataProvider.jar code and passed to your agent

You successfully confirmed your agent in the Performance Management console.

Exercise 4 Install and confirm the updated AB1 agent in an IBM Performance Management environment

Unit 7 Monitoring remote and optional resources exercises

In this unit, you modify your agents to monitor remote resources and contain optional data sources.

In the first exercise, you modify the AB1 agent to monitor multiple hosts at the same time. It pings a list of hosts to confirm their availability. It tests multiple URLs from multiple HTTP servers to confirm their availability. It uses subnodes to gather the same network data from multiple servers.

In the second exercise, you modify the AB2 agent by placing optional sets of monitored data sources into their own subnode. With this setup, you choose which subnodes to configure when you configure the agent.

Exercise 1 Remotely monitor many resources

Your company has several resources that you must monitor where you are not allowed to or cannot install an agent.

In this exercise, you modify the AB1 agent to monitor the following resources:

- SNMP data with several SNMP data sources within a subnode. On the subnode, you can configure a single agent to monitor multiple hosts.
- HTTP server URLs and their embedded objects
- The network availability of the application servers

Create an agent to monitor multiple hosts from one agent

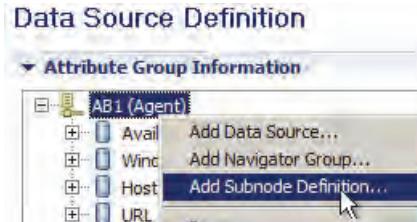
In this section, you modify the AB1 agent to gather process performance information from Simple Network Management Protocol (SNMP) from all three systems in class.

- System
- ifTable
- ipAddrTable

You must complete this exercise from the WIN1 server.

Complete the following steps:

1. If it is not running, start Agent Builder.
2. If the **AB1** agent is not already open, expand **Agent 1** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
3. Click the **Data Sources** tab.
4. Right-click **AB1 (Agent)** under **Attribute Group Information** and click **Add Subnode Definition**.



The Subnode Information window opens.

Subnode Information	
Name	<input type="text"/>
Type	<input type="text"/>
Description	<input type="text"/>
Company identifier	<input type="text"/> IBM_E
Subnode identifier	<input type="text"/> K00_
<input checked="" type="checkbox"/> Show nodes attribute group for this type of subnode	

5. Enter the following information:
 - Name: **SNMP**
 - Type: **SNP**
 - Description: **Subnode to gather SNMP information from multiple hosts**

Name	<input type="text"/> SNMP
Type	<input type="text"/> SNP
Description	<input type="text"/> Subnode to gather SNMP information from multiple hosts
Company identifier	<input type="text"/> IBM_E
Subnode identifier	<input type="text"/> K04SNP
<input checked="" type="checkbox"/> Show nodes attribute group for this type of subnode	

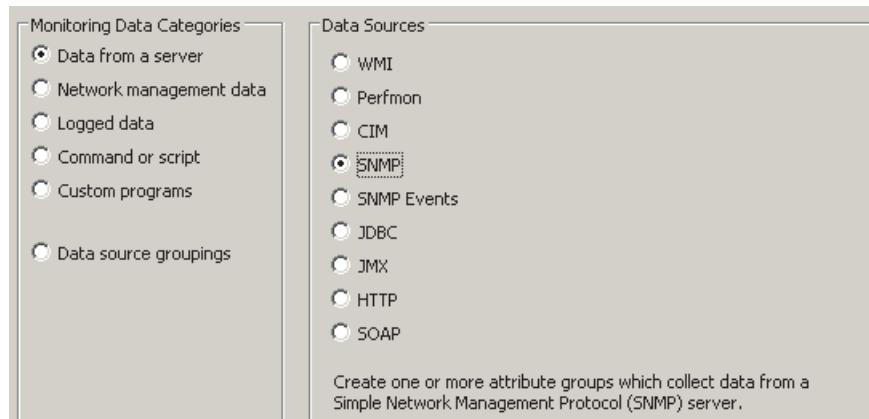


Note: Type is a unique three-character code that identifies this subnode.

6. Click Next.

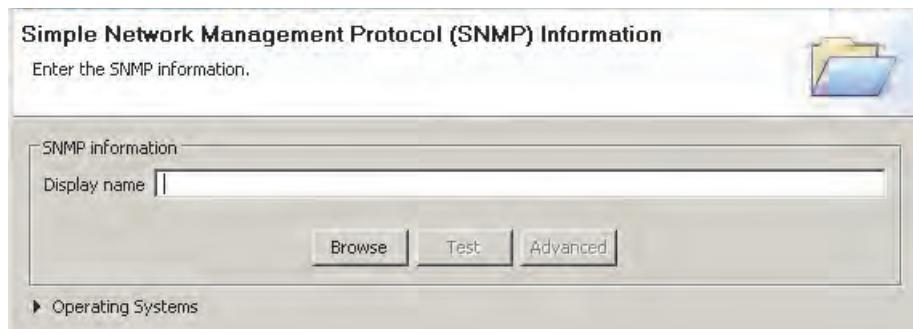
The Agent Initial Data Source window opens.

7. Define the data sources for this subnode. Click **Data from a server under **Monitoring Data Categories** and **SNMP** under **Data Sources**.**



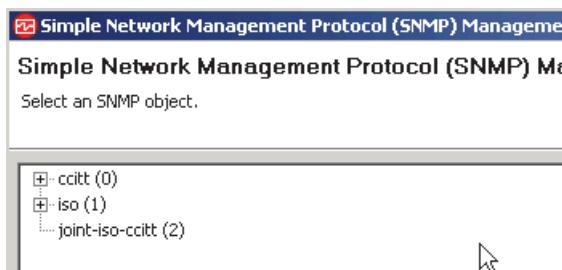
8. Click Next.

The Simple Network Management Protocol (SNMP) Information window opens.



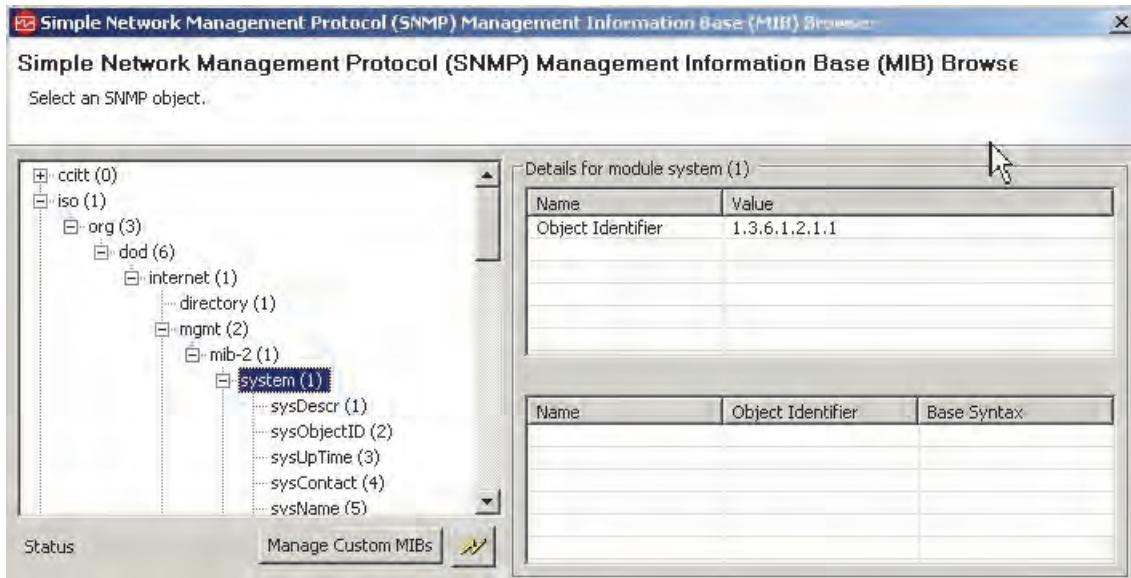
9. Click **Browse in the **SNMP information** pane.**

The SNMP MIB Browser opens.



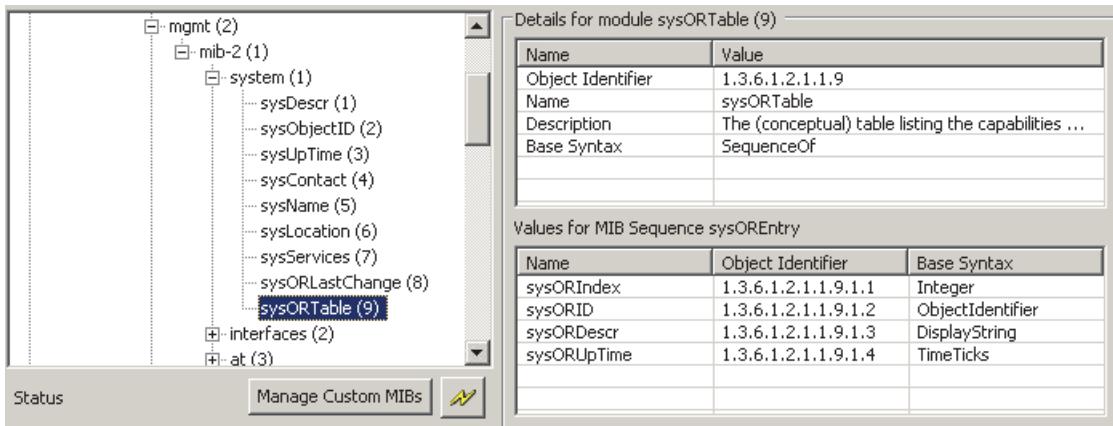
10. Locate and expand the following SNMP object:

iso/org/dod/internet/mgmt/mib-2/system



Notice the properties that are listed below **system**. This data can be gathered from this SNMP object. Agent Builder creates an attribute group for system and an attribute for each of these properties.

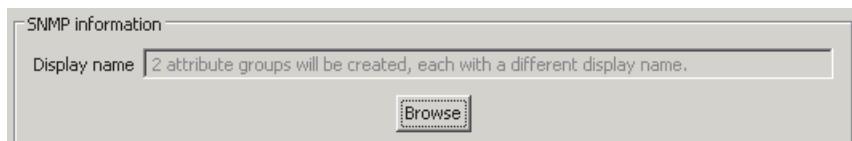
11. Scroll down the SNMP object lists and click **sysORTable**.



Notice the values for MIB Sequence sysOREntry pane. The sysORTable object contains subattributes, and it can be selected as its own data source. By selecting **system** in your agent, the child object sysORTable generates its own attribute group with its own set of attributes.

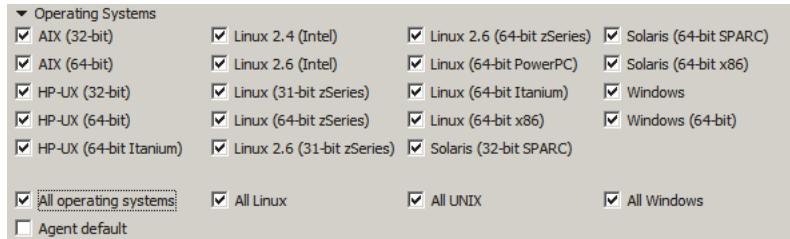
12. Reselect **system** and click **OK**.

The SNMP information window is displayed.



System is not listed in the **Display name** field. The sysORTable object causes your agent to generate two attribute groups, one for system and another for sysORTable. In subsequent steps, as you select other SNMP objects, you see the object name that is listed here.

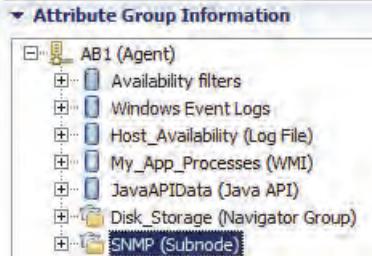
13. Expand **Operating Systems** and select **All operating systems**.



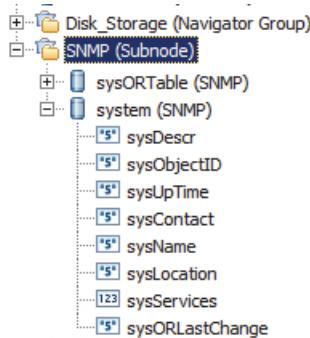
14. Click **Finish**.

You are returned to the **Data Source Definition** tab.

Data Source Definition



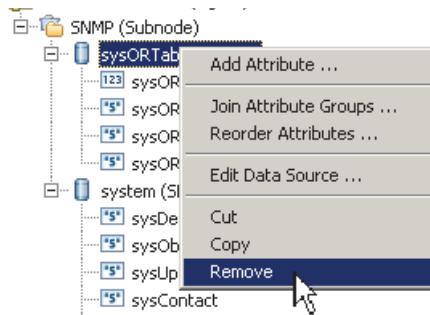
15. Expand **SNMP (Subnode)** and **system**.



The attribute group and attribute names are based on the SNMP objects you saw in the SNMP MIB browser.

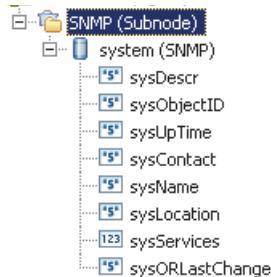
16. Remove the **sysORTable** attribute group.

- Right-click **sysORTable (SNMP)** and click **Remove**.



- Click **Yes** to the prompt to remove one attribute group and four attributes.

Your SNMP subnode looks like the following image.

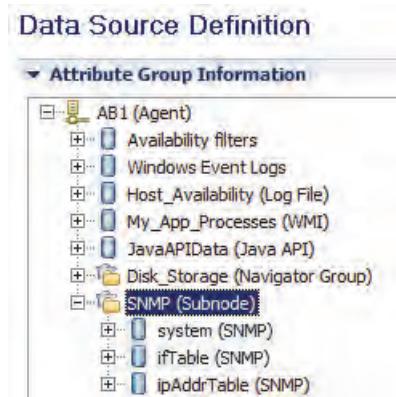


17. Add the following extra SNMP objects by selecting **SNMP (Subnode)**, clicking **Add to Selected** and repeating [Step 7](#) on page 7-3 through [Step 14](#) on page 7-5.

`iso/org/dod/internet/mgmt/mib-2/interfaces/ifTable`

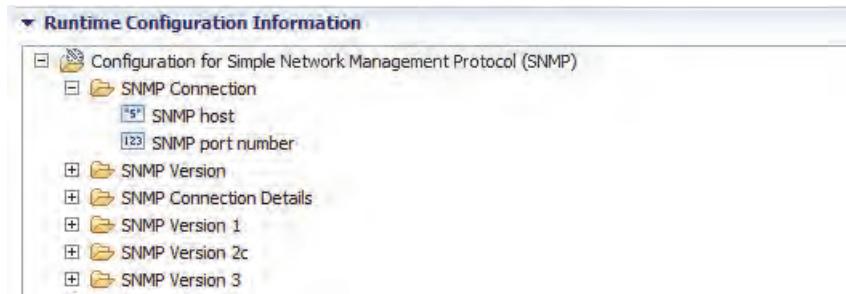
`iso/org/dod/internet/mgmt/mib-2/ip/ipAddrTable`

Your final Data Source Definition window looks like the following example.



18. Click the **Runtime Configuration Information** tab.

19. Expand Configuration for Simple Network Management Protocol (SNMP).

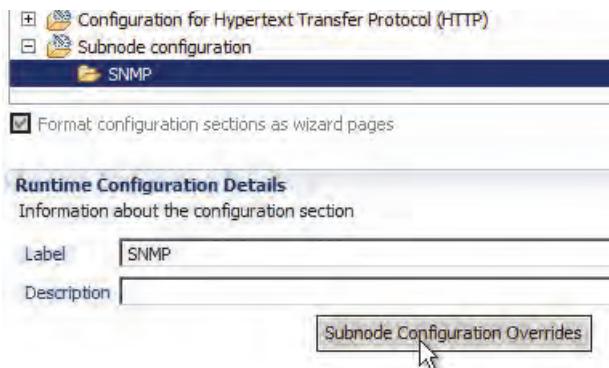


The Configuration for Simple Network Management Protocol object contains global runtime configuration parameters. The Subnode configuration object contains those parameters that must be set for each subnode.

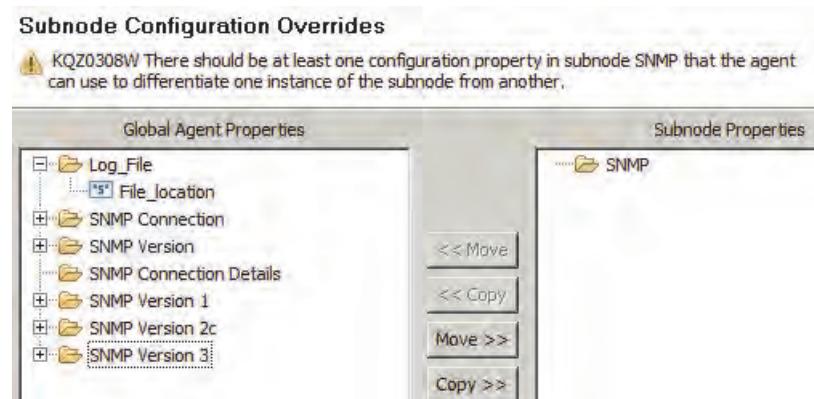
These SNMP properties are set when the agent is configured and establish the connection into the database server, whether it is local or remote. By moving these configuration parameters into the subnode, they can be set differently for each subnode.

20. Set the SNMP Runtime Configuration properties to be configured per subnode.

- Close Configuration for Simple Network Management Protocol (SNMP).
- Expand Subnode Configuration and click SNMP.
- Click Subnode Configuration Overrides.



The Subnode Configuration Overrides window opens.



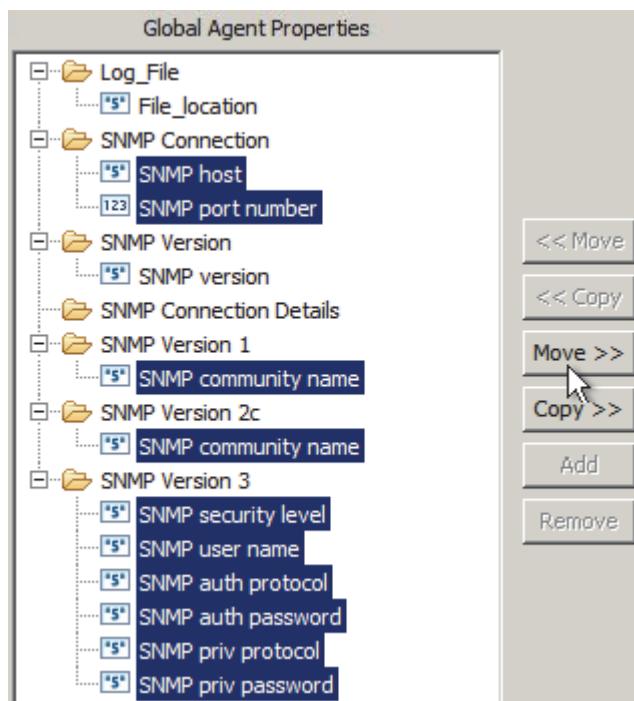
In this window, you can determine which configuration data, which you set during installation of the agent, is configurable for each subnode. Currently, all properties are global. At the time of the installation, you enter them once and they apply to all subnodes. Because you are monitoring multiple remote SNMP servers, this result is not acceptable.

At a minimum, you set the SNMP host name as a subnode property so that you can identify the three host systems. Other information that you might set per subnode include the port that is used to connect to SNMP, the SNMP version, the SNMP community name, and the various SNMP version 3 authentication properties. In this exercise, set all SNMP possible properties configurable for each subnode.

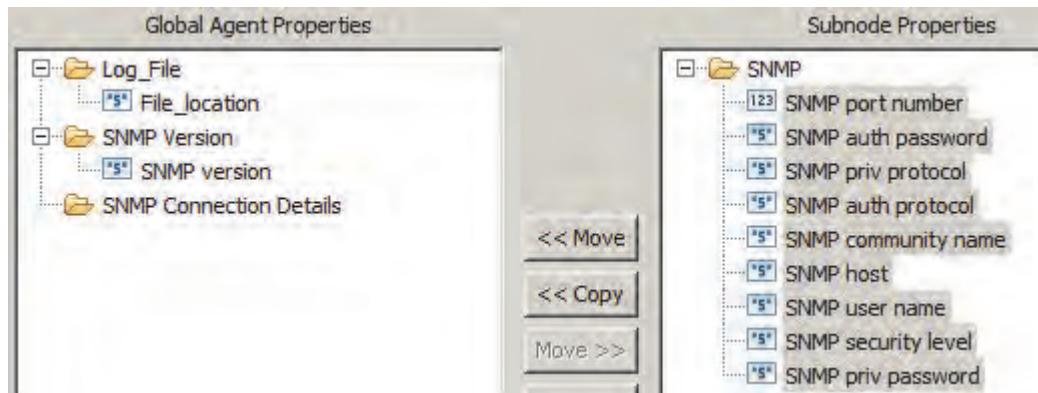
d. Select the following properties:

- ◆ SNMP host
- ◆ SNMP port number
- ◆ SNMP Version 1> SNMP community name
- ◆ SNMP Version 2c > SNMP community name
- ◆ SNMP security level
- ◆ SNMP user name
- ◆ SNMP auth protocol
- ◆ SNMP auth password
- ◆ SNMP priv protocol
- ◆ SNMP priv password

e. Click **Move>>**.



The subnode configuration overrides looks like the following screen capture when you are done.



Now the configuration properties are set when defining subnodes during agent configuration.

21. Click **OK** to save your changes and exit the Subnode Configuration Overrides window.
 You are returned to the **Data Source Definition** tab.
22. Set the default SNMP port to 162.
 - a. Under **Subnode configuration > SNMP**, select **SNMP port number**.
 SNMP port number does not have a default value.
 - b. Enter **161** in **Default Value**.

The screenshot shows the 'Subnode configuration' section with 'SNMP' selected. Under 'SNMP', 'SNMP port number' is highlighted with a red arrow. Below this, there is a checkbox for 'Format configuration sections as wizard pages' which is checked.

Runtime Configuration Details	
Information about the configuration property	
Label	SNMP port number
Environment variable	SNMP_PORT
Description	The port number of the SNMP server.
Type	Numeric
Default value	161

23. Save your agent project.

Add a data source to monitor HTTP pages

In this section, you add an HTTP data source to monitor select pages from the lab web servers.

24. On WIN1, locate and open the following text file:

C:\AB_Files\urls.txt

```
urls.txt - Notepad
File Edit Format View Help
http://win1/
http://win1/AnyBank.htm
http://win1/bad.html
http://win1/AnySiteMap.html
http://lin4
http://lin4/AnyBank.htm
http://lin4/AnyInsurance.htm
http://lin4/AnyInvestment.htm
http://lin4/AnySiteMap.html
http://itm/bad.html
```

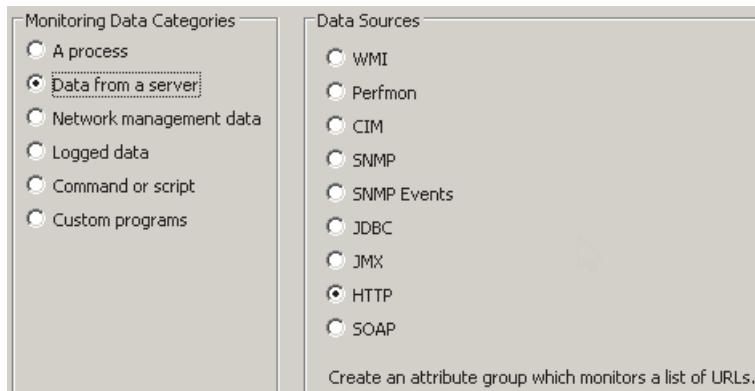
The HTTP data source uses a local text file that lists the URLs to be monitored. For this lab, you use the urls.txt file.

25. In Agent Builder, click the **Data Sources** tab in the AB1 agent editor.

26. Right-click **AB1 (Agent)** and click **Add Data Source**.

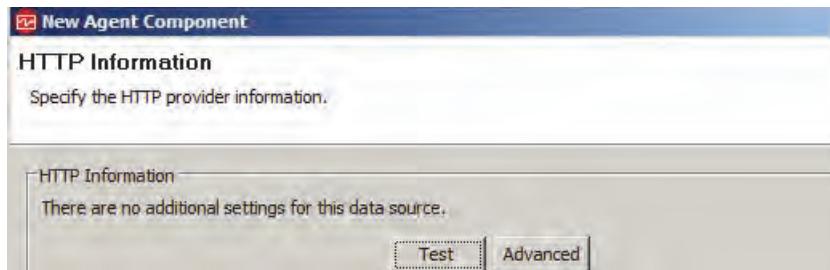
The Data Source Location window opens.

27. Click **HTTP** under **Data from a server**.



28. Click **Next**.

The HTTP Information window opens.

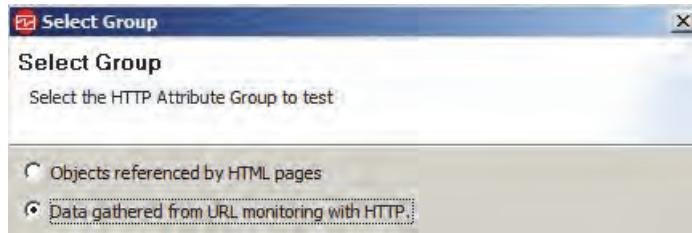


An HTTP data source requires no extra configuration. The files that contain the URLs to monitor are identified when the agent is installed and configured.

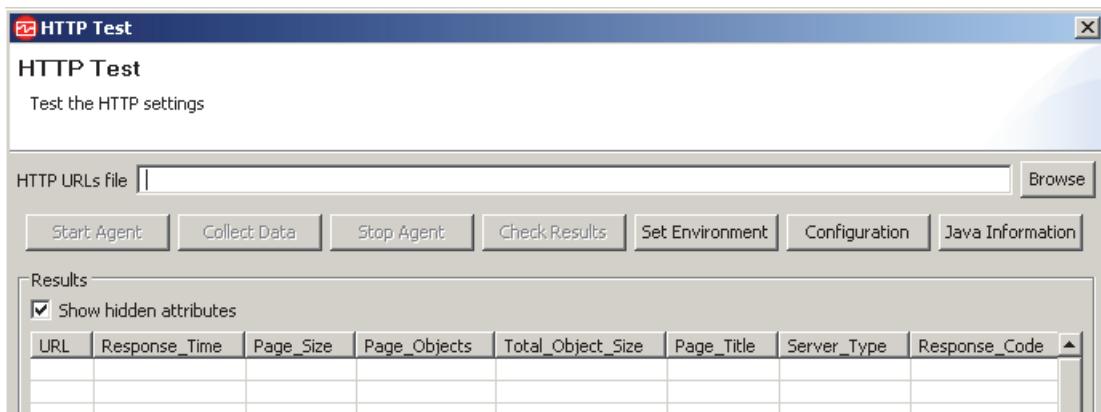
29. Click **Test**.

The Select Group window opens.

30. Click **Data gathered from URL monitoring with HTTP** and click **OK**.

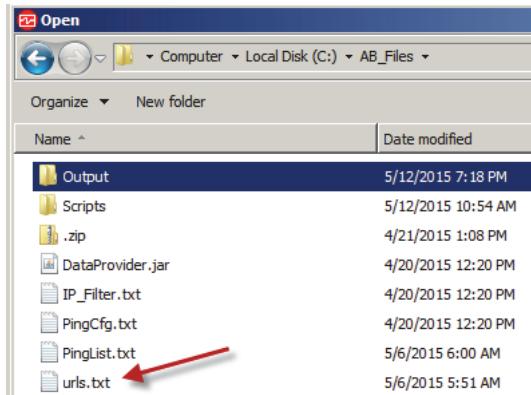


The HTTP Test window opens.

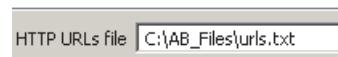


31. Click **Browse**.

32. Locate and open the **C:\AB_Files\urls.txt** file.



You are returned to the HTTP Test window with the **urls.txt** file in the **HTTP URLs file** field.



33. Click Start Agent and Collect Data.

Results						
URL	Response_Time	Page_Size	Page_Objects	Total_Object_Size	Page_Title	
http://win1/	31	1501	0	NOT_COLLECTED	Any Bank Home page	
http://win1/AnyBank.htm	0	861	0	NOT_COLLECTED	Any Bank Banking page	
http://win1/bad.html	0	206	0	NOT_COLLECTED	404 Not Found	
http://win1/AnySiteMap.html	31	848	0	NOT_COLLECTED	Any Bank Site Map	
http://lin4/	16	1501	0	NOT_COLLECTED	Any Bank Home page	
http://lin4/AnyBank.htm	16	861	0	NOT_COLLECTED	Any Bank Banking page	
http://lin4/AnyInsurance.htm	16	868	0	NOT_COLLECTED	Any Bank Insurance page	

34. Confirm that agent successfully gathered data on the monitored URLs.

35. Click **OK** to close the HTTP Test window.

36. Click **Finish** to close the HTTP Information window.

You are returned to the **Data Source Definition** tab with two new data sources: URL_Objects and Managed_URLs.



37. View the configuration properties that you would manage for this data source.

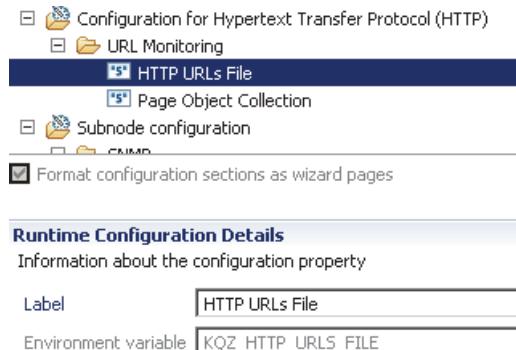
a. Click the **Runtime Configuration** tab.

b. Expand the **Configuration for Hypertext Transfer Proxy > Proxy Server Configuration** nodes.

Label	HTTP Data Provider Proxy Information
Environment variable	KQZ_HTTP_PROXY_TITLE

You must configure these properties if the URLs that the agent monitors go through a proxy server.

- c. Click each property and review the values to confirm that they are configured the way that you want. For example, do you want to require the property, or do you want to set a default value?
- d. Expand the Configuration for Hypertext Transfer Protocol (HTTP) > URL Monitoring nodes.



The **HTTP URLs file** property prompts the user for the name and location of the HTTP URLs file that tells the agent which URL to monitor. The **Page Object Collection** property allows the user to configure the agent to monitor or not monitor embedded objects.

- e. Click each property and review the values to confirm that they are configured the way that you want.

38. Save your agent project.

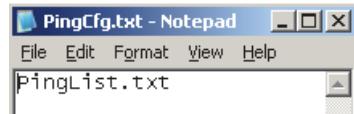
You successfully added an HTTP data source to this agent.

Add a data source to monitor network availability

In this section, you add a Ping (ICMP) data source to the agent that enables it to monitor the network availability of the lab servers.

39. Locate and open the following text file:

C:\AB_Files\PingCfg.txt



The Ping data source uses several local text files to determine which systems to monitor. The ping configuration file is a text file that must be specified as a runtime configuration parameter to the agent. The contents of each line in this file must be the location of a device list file. Each entry in the ping configuration file causes the agent to start a separate thread for monitoring the devices in the provided device list file.

40. Locate and open the following text file:

C:\AB_Files\PingList.txt

A screenshot of a Windows Notepad window titled "PingList.txt - Notepad". The window shows the following text:
WIN1 Tint4
win2
192.168.1.103
Missinghost
192.168.1.107

A device list file is a text file that contains a list of device and host names. The list of device and host names can be separated with spaces and line breaks.

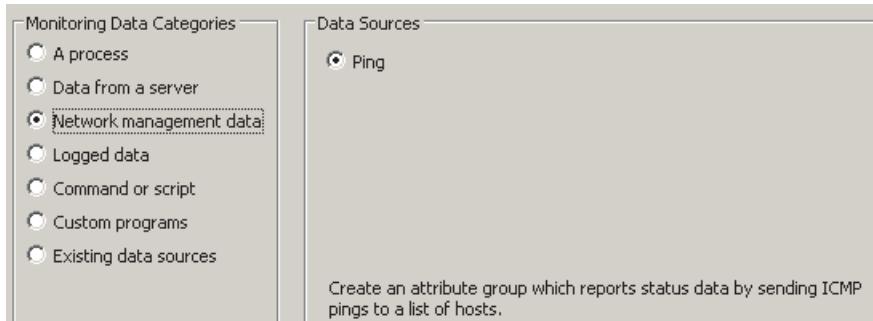
You use these files to identify which hosts to monitor.

41. In Agent Builder, click the **Data Sources** tab in the AB1 agent editor.

42. Right-click **AB1 (Agent)** and click **Add Data Source**.

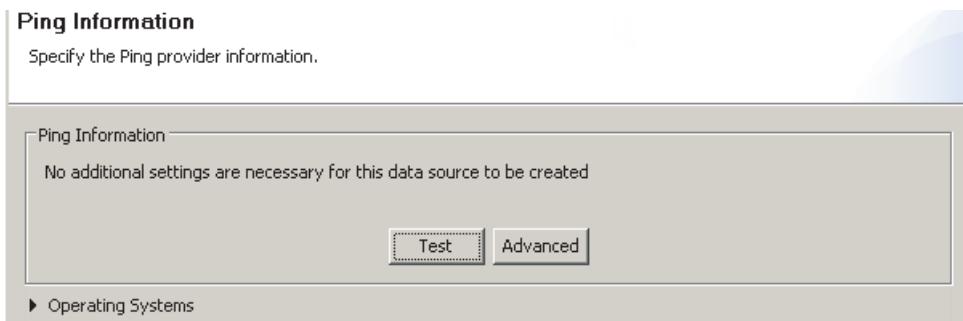
The Data Source Location window opens.

43. Click **Ping** under **Network management data**.



44. Click **Next**.

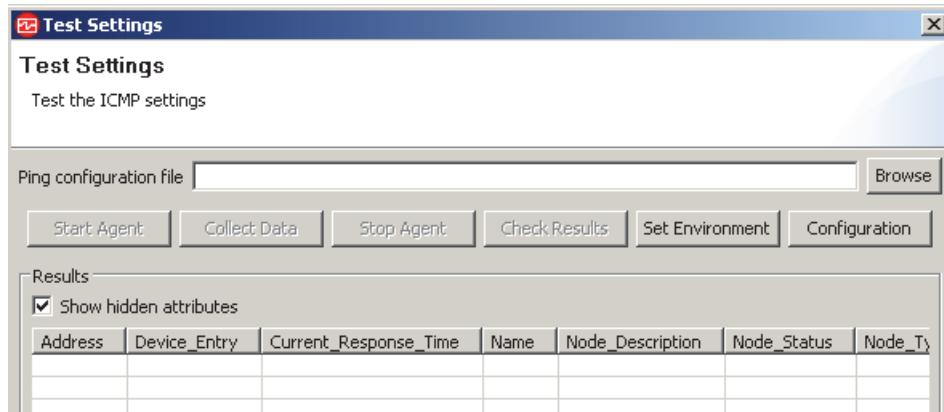
The Ping Information window opens.



A Ping data source requires no extra configuration. The file that contains the host names or IP addresses to monitor is identified when the agent is installed and configured.

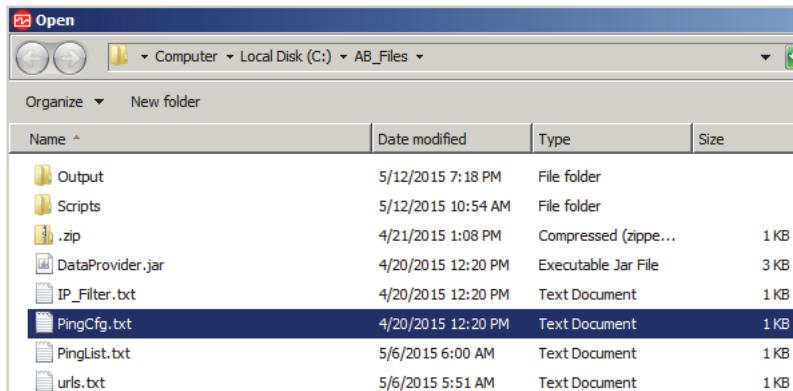
45. Click **Test**.

The Test Settings window opens.



46. Click **Browse**.

47. Locate and open the **C:\AB_Files\PingCfg.txt** file.



You are returned to the Test Settings window with the **PingCFG.txt** file in the **Ping configuration file** field.

Ping configuration file C:\AB_Files\PingCfg.txt

48. Click **Start Agent** and **Collect Data**.

Results							
<input checked="" type="checkbox"/> Show hidden attributes							
Address	Device_Entry	Current_Response_Time	Name	Node_Description	Node_Status	Node_Type	Status_Timestamp
192.168.1.103	192.168.1.103	0	win1.ibm.edu	PingList	Unknown	IP Node	
192.168.1.107	192.168.1.107	0	lin4.ibm.edu	PingList	Unknown	IP Node	
UNKNOWN_ADDRESS	Missinghost	0	UNKNOWN_HOSTNAME	PingList	Unknown	IP Node	
192.168.1.103	win1	0	win1.ibm.edu	PingList	Unknown	IP Node	
192.168.1.109	win2	0	win2.ibm.edu	PingList	Unknown	IP Node	
192.168.1.107	lin4	0	lin4.ibm.edu	PingList	Unknown	IP Node	

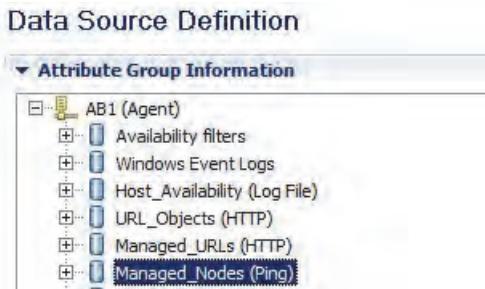
The device **Missinghost** is a name that cannot be resolved and therefore is not pinged.

49. Confirm that agent successfully gathered data on the monitored systems.

50. Click **OK** to close the Test Settings window.

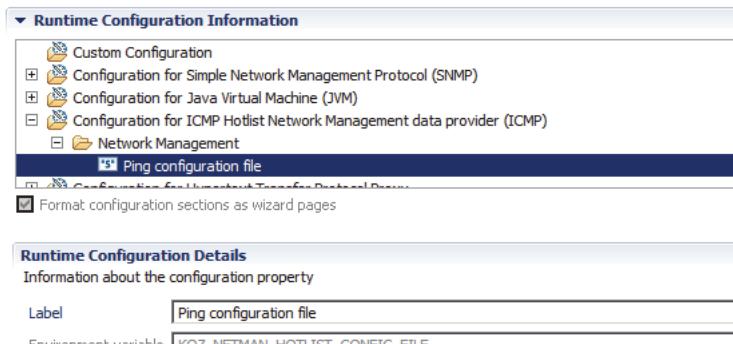
51. Click **Finish** to close the Ping Information window.

You are returned to the **Data Source Definition** tab with two new data sources: URL_Objects and Managed_URLs.



52. View the configuration properties that you would manage for this data source.

- Click the **Runtime Configuration** tab.
- Expand the **Configuration for ICMP Hotlist Network Management data provider (ICMP) > Network Management** nodes.



The only runtime configuration property for this data source is the ping configuration file, which leads the agent to the list or lists of devices it monitors.

53. Save your agent project.

You successfully added a Ping (ICMP) data source to this agent.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

54. Select to the **Agent Information** tab.

55. Change the version for 1.0.2 to **1.0.3**.

Agent Information

General
This section defines the general agent information.

Name	AB1	Advanced	Product o
Version	1.0.3	Company	
Fix pack	0	Patch level	0
Agent ide			



Important: If your initial version number is not 1.0.2, increase the current value by 1 to ensure that the changes are identified by your monitoring server.

56. Save your agent project.

Exercise 2 Install and confirm the updated AB1 agent in an IBM Performance Management environment

In this exercise, you reinstall the AB1 agent onto WIN1. Then you confirm that data is being gathered for the new attribute groups you added.

1. Confirm that time is synchronized between the APM and WIN1 servers.
2. Confirm that all APM services are running on APM.

Create dashboards and resource definitions

The base AB1 agent already has a Summary dashboard and OSLC resource definitions. Use the Agent Builder Dashboard Setup wizard to add the HTTP and Ping data sources to the base AB1 agent detailed dashboard. Furthermore, define dashboards and OSLC resources for the SNMP subnodes.

3. Select **Dashboards** from the Agent Builder Outline view.

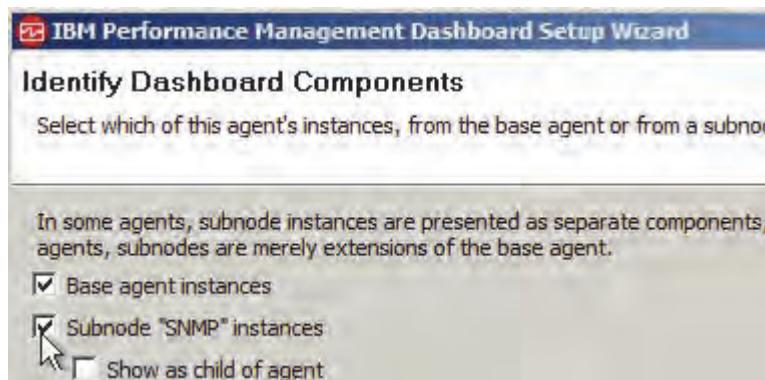
The Dashboards Overview opens.

4. Click the **Dashboard Setup wizard** link in the Overview.

The IBM Performance Management Dashboard Setup wizard opens.

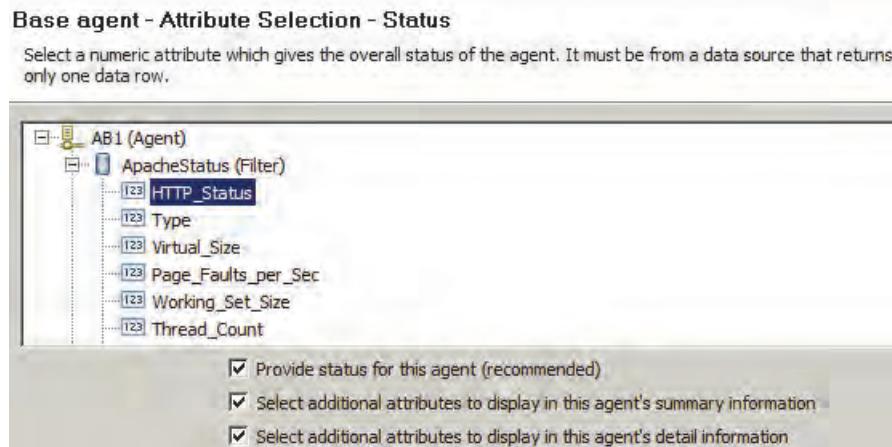
Because this agent is configured for subnodes, you are prompted whether to create a dashboard for the base agent and each subnode.

5. Select the **Subnode “SNMP” instances** check box.



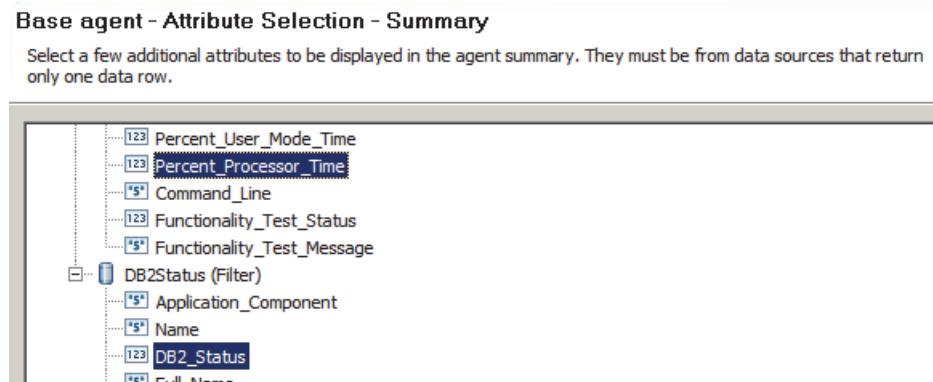
6. Click **Next**.

The Base agent - Attribute Selection - Status view opens. You are prompted to select a numeric attribute that gives the overall status of the base agent.



7. Keep the status attribute and click **Next**.

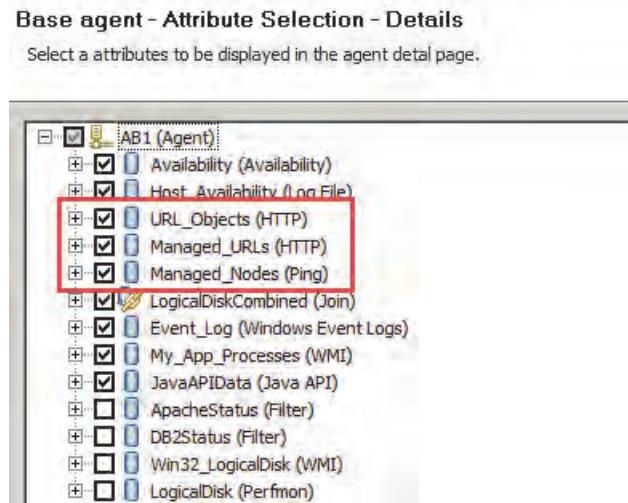
The Base agent - Attribute Selection - Summary view opens. You are prompted to select more attributes for the base agent Summary dashboard.



8. Keep the current attributes and click **Next**.

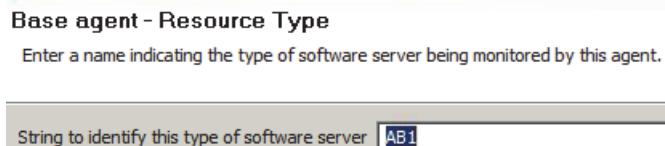
The Base agent - Attribute Selection - Details view opens. You are prompted to identify data groups and items to display in the base agent details dashboards.

9. Select the following attribute groups:
 - URL_Objects (HTTP)
 - Managed_URLs (HTTP)
 - Managed_Nodes (Ping)



10. Click **Next**.

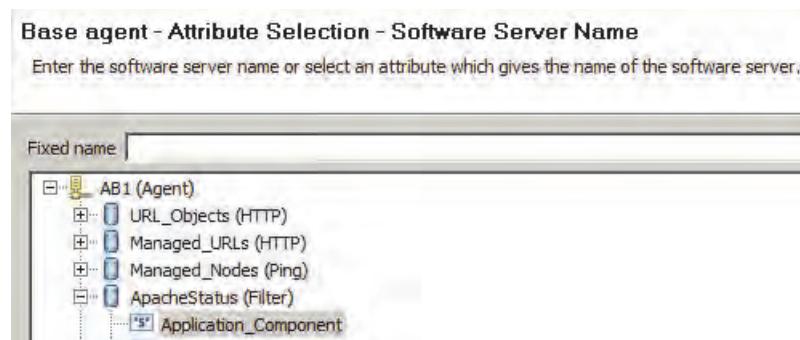
The Base agent - Resource Type view opens.



You are prompted for the Resource Type.

11. Keep **AB1** in the resource type name field and click **Next**.

The Base agent - Attribute Selection - Software Server Name view opens.



12. Keep the Application_Component attribute and click **Next**.

The Base agent - Attribute Selection - IP Address view opens.

Base agent - Attribute Selection - IP Address

Select the source of the IP address of the software server: either from an attribute or from the



13. Keep the current setting and click **Next**.

The Attribute Selection - Port view opens.

14. Keep the current value and click **Next**.

Base agent - Attribute Selection - Port

Tell how the port on which the software server listens for requests is kept as a fixed value.



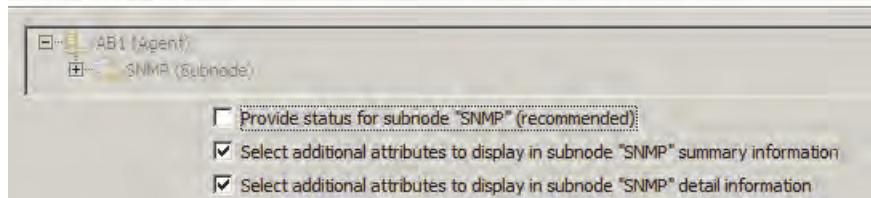
15. Click **Next**.

The Subnode "SNMP" - Attribute Selection - Status view opens. You are prompted to select a numeric attribute that gives the overall status of the SNMP subnode. Do not set an attribute to show the overall subnode status.

16. Clear **Provide status of subnode "SNMP"**.

Subnode "SNMP" - Attribute Selection - Status

Select a numeric attribute which gives the overall status of subnode "SNMP". It must be from a data source that returns only one data row.

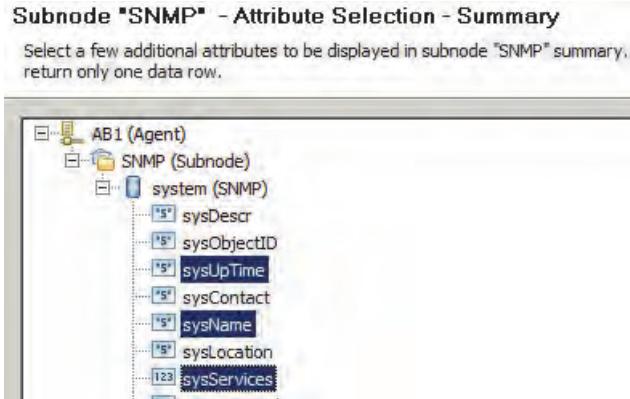


17. Click **Next**.

The Subnode "SNMP" - Attribute Selection - Summary view opens. You are prompted to identify attributes to display in the Summary dashboard.

18. Expand **System** and select the following attributes:

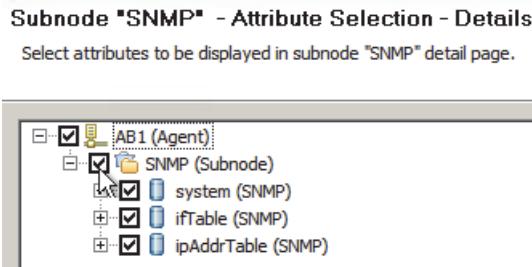
- sysUpTime
- sysName
- sysServices



19. Click **Next**.

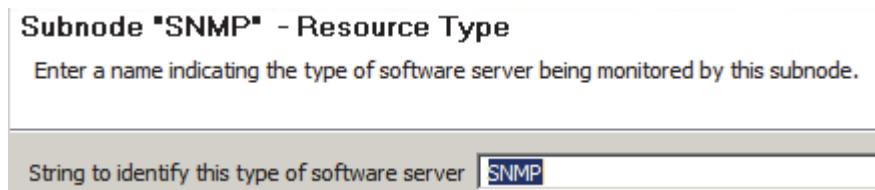
The Subnode “SNMP” - Attribute Selection - Details view opens. You are prompted to identify data groups and items to display in the SNMP subnode details dashboards.

20. Select **SNMP (Subnode)** which selects all the attribute groups.



21. Click **Next**.

The Subnode “SNMP”- Resource Type view opens.

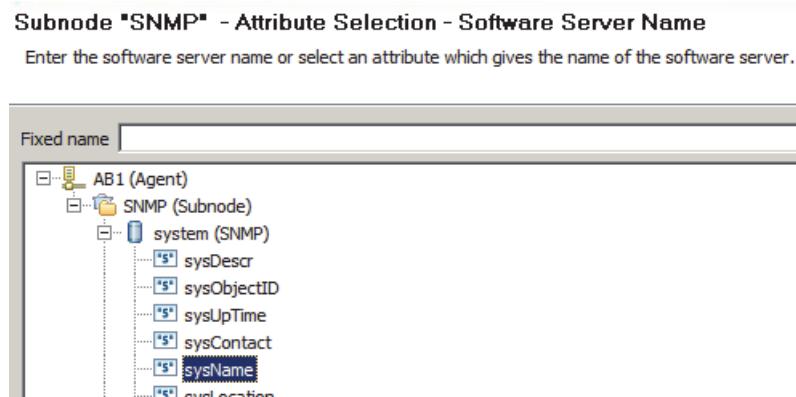


You are prompted for the resource type.

22. Keep **SNMP** in the resource type name field and click **Next**.

The Subnode “SNMP” - Attribute Selection - Software Server Name view opens. Enter a fixed name or select an attribute.

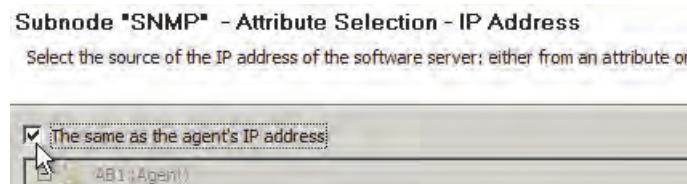
23. Select **SNMP > system (SNMP) > sysName**.



24. Click **Next**.

The Subnode “SNMP” - Attribute Selection - IP Address view opens.

25. Select the check box to use the agent’s IP address.



26. Click **Next**.

The Subnode “SNMP” - Attribute Selection - Port view opens.

27. Enter **0** in the **Fixed value** field.



28. Click **Finish** to complete the wizard.

29. Save your agent project.

You successfully defined the summary dashboard, detailed dashboard, and the monitored resource data for OSLC.

Reinstall the AB1 agent on WIN1

30. Stop the AB1 agent on WIN1.

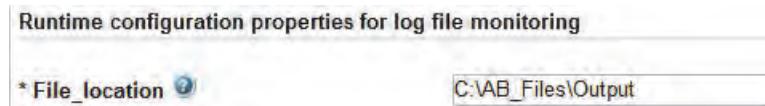
31. Create the agent installers and install the agent on WIN1 in any manner you would like.



Important: When generating the agent installers, ignore the warning that the SNMP subnode has no status attribute.

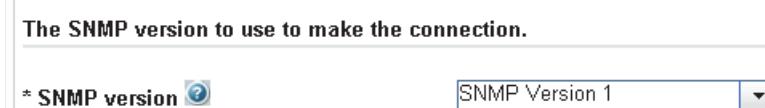
32. Configure your Monitoring Agent for AB1.

- In the IPM utility, right-click the agent and select **Reconfigure** to start the configuration.



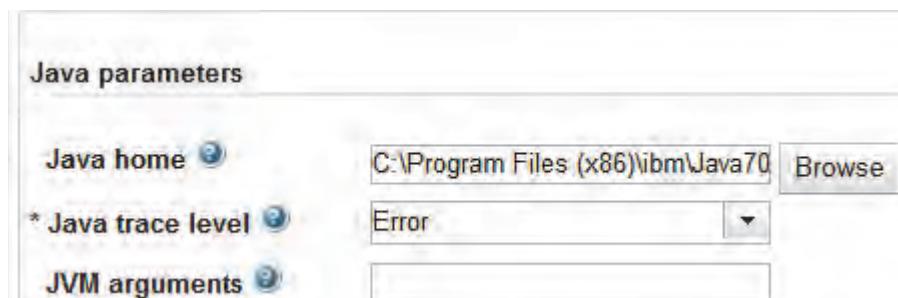
You are prompted for the log file location.

- Keep the current value and click **Next**.



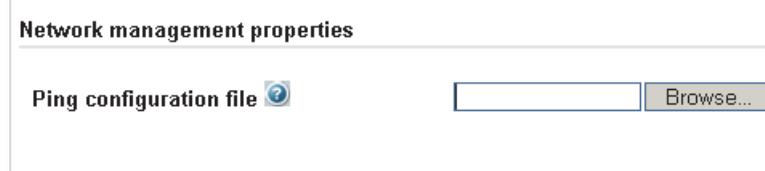
From this pane, you specify the version of SNMP of the monitored system. Selecting a version of SNMP brings forward the configuration parameters of that version in the next pane, and you can set the default values. Because this property was defined only as global, and not per subnode, you cannot change this property per subnode.

- Keep the default **SNMP version** and click **Next**.



You are prompted to Java parameters.

- Keep the current value and click **Next**.



Here you identify the ping configuration file that identifies which hosts to monitoring.

- e. Click **Browse** and open the C:\AB_Files\PingCfg.txt file.

Ping configuration file  C:\AB_Files\PingCf...

- f. Click **Next**.

Configuration for a proxy server used by HTTP providers

HTTP Data Provider Proxy Information

Proxy Hostname 	<input type="text"/>
Proxy Port 	80
Proxy User Name 	<input type="text"/>
Proxy Password 	<input type="password"/>
Confirm Proxy Password	<input type="password"/>

Here you configure the connection through a proxy server for HTTP monitoring. This agent does not use a proxy server.

- g. Click **Next**.

Configuration for monitoring a list of URLs

* HTTP URLs File 

<input type="text"/> <input type="button" value="Browse..."/>

* Page Object Collection 

Yes	<input type="button" value="▼"/>
-----	----------------------------------

Here you define which HTTP URLs to monitor and whether to monitor embedded objects.

- h. Click **Browse** and open the C:\AB_Files\urls.txt file.

* HTTP URLs File 

<input type="text"/> C:\AB_Files\urls.txt <input type="button" value="Browse..."/>
--

* Page Object Collection 

Yes	<input type="button" value="▼"/>
-----	----------------------------------

- i. Keep the default **Yes** in Page Object Collection and click **Next**.

SNMP	New...
SNMP port number	161
SNMP auth password	
Confirm SNMP auth password	
SNMP priv protocol	DES
SNMP auth protocol	MD5
SNMP community name	
Confirm SNMP community name	
SNMP host	
SNMP user name	
SNMP security level	noAuthNoPriv
SNMP priv password	
Confirm SNMP priv password	

Here you define and configure a subnode for each remote system you want to monitor with the SNMP subnode. You can set default values for SNMP host and port number. In this lab, you monitor all three lab systems with the SNMP subnode.

33. Enter the following field to be set for each subnode:

- SNMP community name: **ibm**
- Confirm SNMP community name: **ibm**
- SNMP host: **192.168.1**.

SNMP auth protocol	MD5
SNMP community name	***
Confirm SNMP community name	***
SNMP host	192.168.1
SNMP user name	



Hint: The fields are case-sensitive.

34. Add the following three hosts as SNMP nodes:

- WIN1, 192.168.1.103
- WIN2, 192.168.1.109
- LIN4, 192.168.1.107

a. Click **New**.

The subnode is created.

A screenshot of a configuration dialog box. At the top left is a 'Delete' button. Below it is a section labeled '* SNMP' with a red box around it. To the right of this section are four input fields: 'SNMP port number' containing '161', 'SNMP auth password' (empty), 'Confirm SNMP auth password' (empty), and another 'SNMP auth password' (empty) field below it.

Notice that values that you entered are repeated for each subnode. SNMP is the name that is shown in the Tivoli Enterprise Portal. SNMP host is the host name or IP address of the remote system to be monitored.

- b. Enter the host name in the **SNMP** field.
- c. Enter the last numbers of the IP address in the **SNMP host** fields.

A screenshot of the same configuration dialog box, but with different values entered. The 'SNMP' field now contains 'WIN1' (highlighted with a red box). The 'SNMP host' field at the bottom now contains '192.168.1.103' (also highlighted with a red box). The other fields remain empty.

- d. Repeat these steps for the remaining two hosts by clicking **New** and entering the **SNMP** and **SNMP host** values for the other hosts.

SNMP configuration for host WIN2:

<input type="button" value="Delete"/>	* SNMP	WIN2
	SNMP port number	161
	SNMP auth password	
	Confirm SNMP auth password	
	SNMP priv protocol	
	SNMP auth protocol	
	SNMP community name	***
	Confirm SNMP community name	***
	* SNMP host	192.168.1.109

SNMP configuration for host LIN4:

<input type="button" value="Delete"/>	* SNMP	LIN4
	SNMP port number	161
	SNMP auth password	
	Confirm SNMP auth password	
	SNMP priv protocol	
	SNMP auth protocol	
	SNMP community name	***
	Confirm SNMP community name	***
	* SNMP host	192.168.1.107

35. Click **OK** to save the agent configuration.

The IPM utility main window is displayed.

36. Start your **Monitoring Agent for AB1**.

Confirm the AB1 modifications in the Performance Management Console

In this section, you confirm that data from your SNMP, HTTP, and ping data sources show in the Performance Management console.

37. Log in to the **APM** server as user **root** with password **object00**.

38. Open the Performance Management console by opening a browser and go the following URL:

<https://apm:9443/>



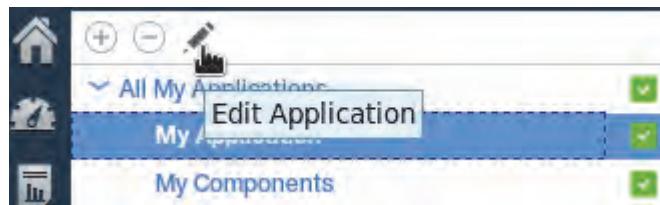
Hint: A bookmark to this URL is in the browser.

39. Log in to the Performance Management console as user **apmadmin** with password **object00**.

40. Open the Application Performance dashboard.

41. Click **My Application** in the Applications explorer.

42. Click the **Edit Application** button.



The Edit Application window opens.

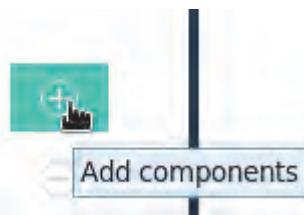
43. Add the following components to your application:

Application components

- lin4 - Linux OS(1)
 - lin4:LZ
- WIN1 - Windows OS(1)
 - Primary:WIN1:NT
 - 00:LIN4 - SNMP(1)
 - 00:LIN4:SMP
 - 00:WIN1 - SNMP(1)
 - 00:WIN1:SMP
 - 00:WIN2 - SNMP(1)
 - 00:WIN2:SMP

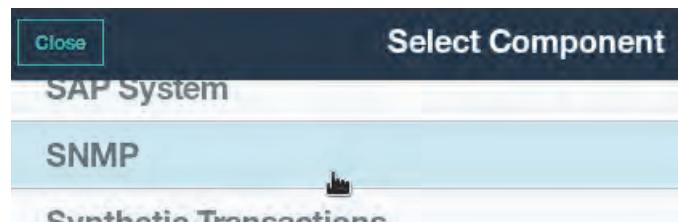
Show all unaccepted component changes.

a. Click the **Add components** button next to the Application components pane.



The Select Component window opens.

- b. Scrolling down, locate and select **SNMP**.



The Component Editor windows opens.

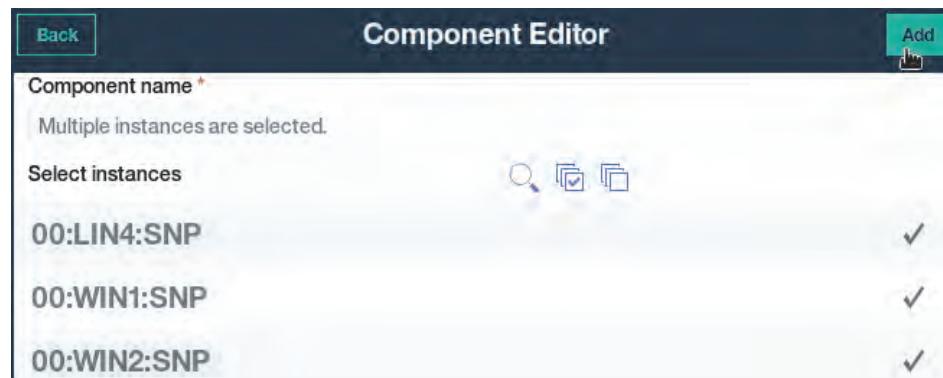
- c. Select the following components:

- ◆ 00:LIN4:SNP
- ◆ 00:WIN1:SNP
- ◆ 00:WIN2:SNP



Important: It can take 10 - 15 minutes from when the agent is installed until the subnodes appear in the Component Editor window. If the subnodes are not listed, click Back and SNMP until they are shown. Also, ensure the date and time on APM, WIN1, and LIN4 hosts are the same.

- d. Click **Add**.



The subnodes are removed from the Component Editor window added to your application definition.

- e. Click **Back** and **Close** to close the Select Component window.

You are returned to the Edit Application window with the SNMP subnodes in the Application components window.



44. Click **Save** and **OK** to save your application definition and exit the Edit Application utility.

You are returned to the Application Dashboard.

Confirm the base agent data sources

45. Click **My Application** in the Applications explorer.

46. Click the **AB1** component bar in the Current Components Status widget to open the Summary dashboard.

47. Click the **WIN1 - AB1** Summary dashboard widget to access the detailed dashboards.

48. Scroll down and confirm **URL_Objects**, **Managed_URLs** and **Managed_Nodes** data is shown.

URL_Objects					
URL	Object Name	Object S..	Object R..	Node	
http://lin4/AnyInvestment.htm	http://lin4/Images/AnyCorpWeb...	84607	4	WIN1:00	
http://lin4/AnyInvestment.htm	http://lin4/Images/AnyCorpWeb...	7825	1	WIN1:00	
http://win1/	http://win1/Images/AnyCorpWe...	47232	2	WIN1:00	

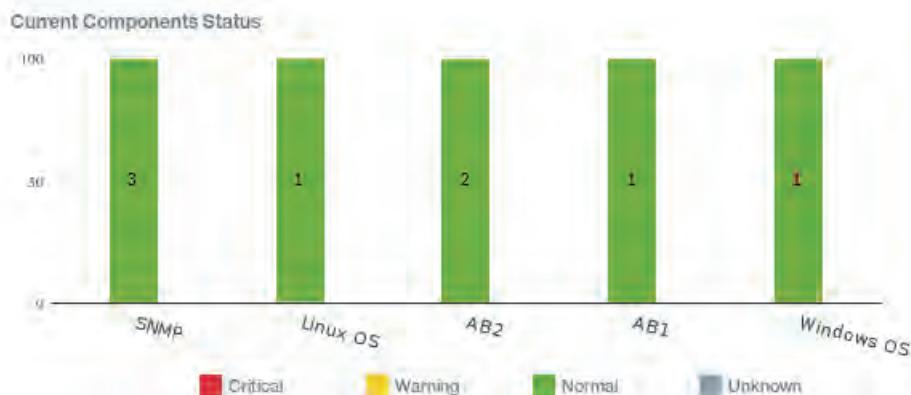
Managed_URLs						
URL	Respons..	Page Size	Page Ob..	Total Obj..	Page Title	Server Type
http://win1/	22	1501	6	92572	Any Bank home page	Apache/2.4.25 (Win64) OpenSS...
http://win1/	22	1501	6	92572	Any Bank home page	Apache/2.4.25 (Win64) OpenSS...
http://win1/AnyBank.htm	68	861	2	92949	Any Bank Banking page	Apache/2.4.25 (Win64) OpenSS...

Managed_Nodes					
Address	Device Entry	Current ...	Name	Node Description	Node St..
192.168.1.103	WIN1	1	win1.ibm.edu	PingList	Active
192.168.1.103	192.168.1.103	1	win1.ibm.edu	PingList	Active
192.168.1.107	192.168.1.107	1	lin4.ibm.edu	PingList	Active

Confirm the SNMP subnode data sources

49. Click **My Application** in the Applications explorer.

50. Refresh this view until the **SNMP** component is shown in the Current Component Status view.



Hint: It can take up to 10 minutes for the component to be displayed.

51. Click the **SNMP** component bar in the Current Components Status widget to open the Summary dashboard.



The Summary dashboard shows selected values from your SNMP data source. Seeing valid data here confirms that data source in this agent.

52. Click the **LIN4: 04 - SNMP Summary** dashboard widget to access the detailed dashboards.

The screenshot shows a dashboard titled "04:LIN4:SNP". At the top, there are three tabs: "Status Overview" (which is underlined in blue), "Events" (with a green checkmark icon), and "Attribute Details". Below the tabs, there is a section titled "system" with a small cloud icon. A table lists various system attributes:

Attribute	Value
sysDescr	Linux lin4 3.0.101-0.40-default #1 SMP Thu Sep 18 13:09:38 ...
sysObjectID	1.3.6.1.4.1.8072.3.2.10
sysUpTime	543341 - 1 hours, 30 minutes, 33.41 seconds
sysContact	Sysadmin (root@localhost)
sysName	lin4
sysLocation	Server Room
sysServices	0
sysORLastChange	21 - 0.21 seconds
Node	04:LIN4:SNP
Timestamp	2/8/17, 9:29 PM

You successfully confirmed your agent in the Performance Management console. That the agent was available to be added to an application, the Summary dashboard shows the core metrics that you added. A subset of attributes is visible in the **Details dashboard**.

Exercise 3 Allow data sources to be optional

Your company has many servers with similar and dissimilar resources that need to be monitored. In this lab, you build an agent to monitor these resources:

- Queue status that is gathered directly from the application server with the Java Management Extension (JMX) data source
 - Processor information that is gathered from the application servers with the Common Information Model (CIM) data source
- Both JMX and CIM are multiplatform and can provide the data that is needed from both Windows and Linux.
- Software code status in a database

Modify the AB2 agent so that the data sources that are monitored are determined when the agent is installed and configured on a specific server. Enable this ability by putting each optional data source in its own subnode.

Add monitoring of Common Information Management data sources

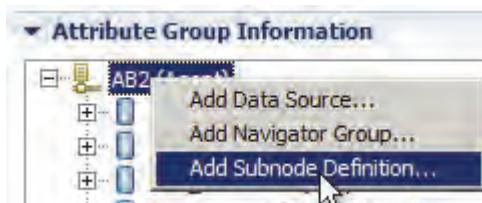
In this section, you enable the agent to monitor processors that run on a system that contains a Common Information Management data source.

1. If it is not running, start Agent Builder.
2. Close all open Agent Editor windows.
3. If the **AB2** agent is not already open, expand **Agent 2** in the Project Explorer and double-click the **itm_toolkit_agent.xml** file.
4. Click the **Data Sources** tab.



Note: If you deployed the AB2 agent into the IBM Performance Management environment, you also see the Availability_Filtered (Filter) data source.

- Right-click **AB2 (Agent)** and select **Add Subnode Definition...**.



The Subnode Information window opens.

- Enter the following information:
 - Name: **AB_Processor**
 - Type: **CIM**
 - Description: **Subnode to gather processor data through CIM**

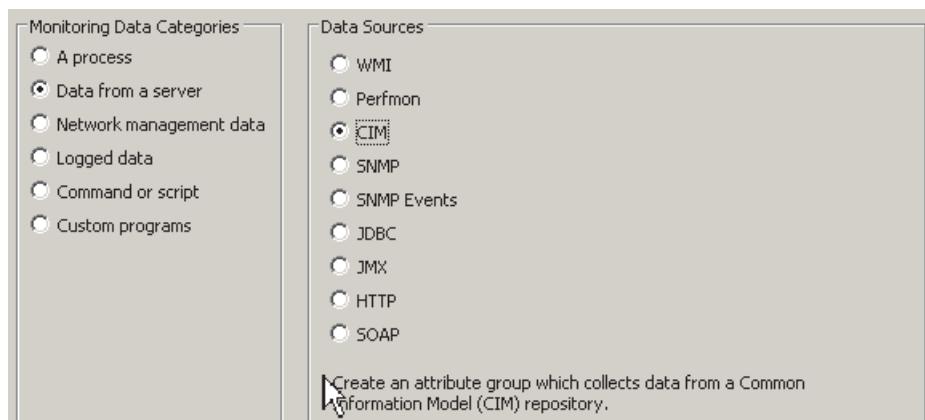
Subnode Information	
Name	<input type="text" value="AB_Processor"/>
Type	<input type="text" value="CIM"/>
Description	<input type="text" value="Subnode to gather processor data through CIM "/>
Company identifier	<input type="text" value="IBM_E"/>
Subnode identifier	<input type="text" value="K01CIM"/>
<input checked="" type="checkbox"/> Show nodes attribute group for this type of subnode	

Note: Type is a unique three-character code identifying this subnode.

- Click **Next**.

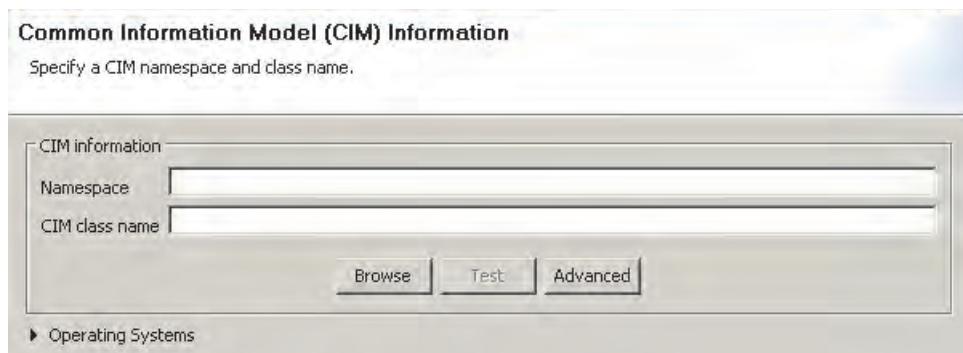
The Initial Subnode Data Source window is displayed.

- Click Data from a server in the Monitoring Data Categories section and CIM in the Data Sources section.



- Click Next.

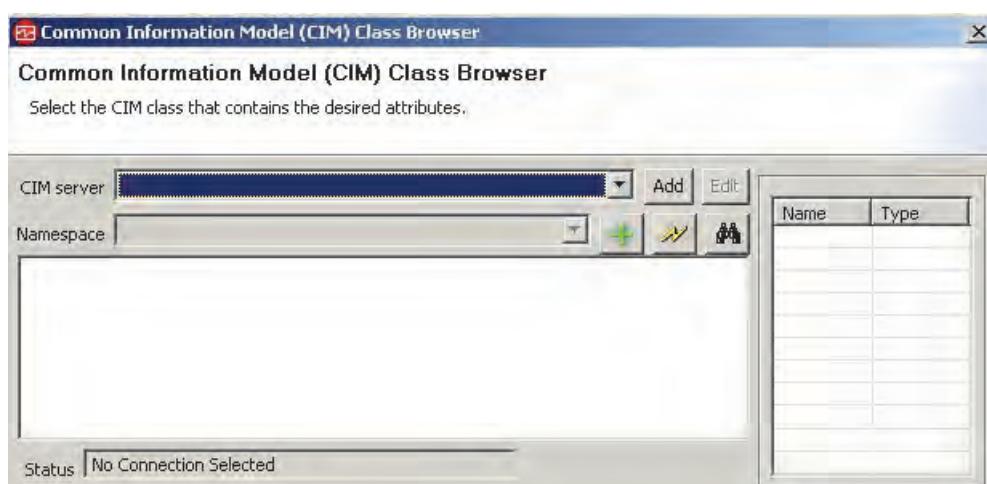
The Common Information Model (CIM) Information window opens.



This data source works similarly to WMI except that it is multiplatform.

- Click Browse.

The Common Information Model (CIM) Browser opens.



- Click Add.

12. Select **CIM server** and click **Next**.



Both WIN1 and WIN2 run a CIM managed object server.

13. Enter the following values:

- **Host name:** WIN2
- **User name:** Administrator
- **Password:** object00
- Save the password: **Selected**



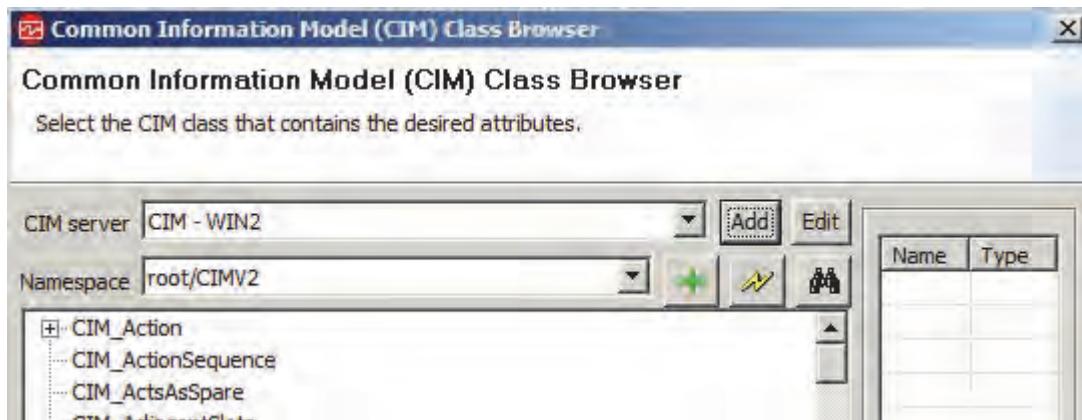
14. Click **Test Connection**.

You receive an indication that the connection was successful.



15. When you have a successful test connection, click **Finish**.

The browser is displayed with the root **Namespace** selected.



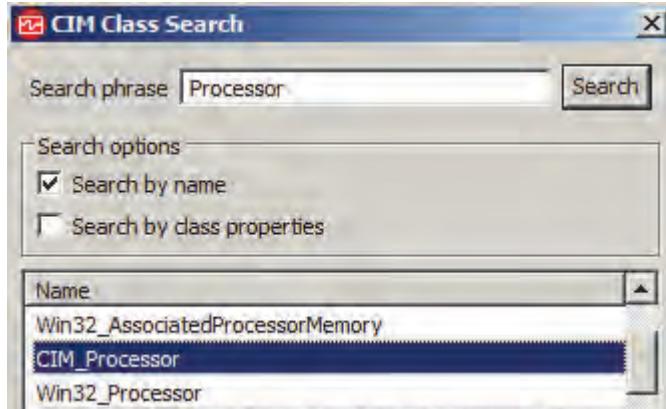
16. Browse the namespaces by clicking the **Namespace** menu.

You see a list of all the CIM namespaces on WIN2. The list of namespaces is similar to the WMI namespaces from [Unit 4, Exercise 1](#) on page 4-3. For example, notice the CIMV2. It is the same namespace from which you monitored the HTTP Server and DB2 process information and the disk space information. Because WMI namespaces are built on the CIM model, the CIM agent you are using, IBM Director, presents all WMI namespaces as CIM namespaces.

17. Select the **root/CIMV2** namespace.

18. Using the search tool, search for class names with the word **Processor** in them.

19. Select **CIM_Processor** and click **OK**.

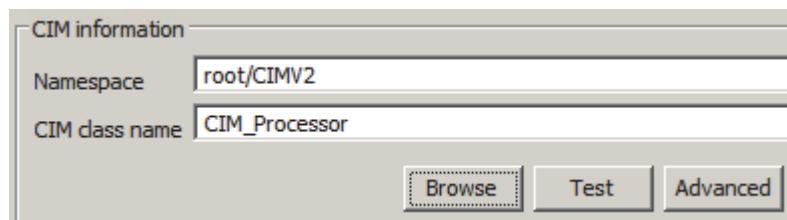


The CIM browser is displayed.

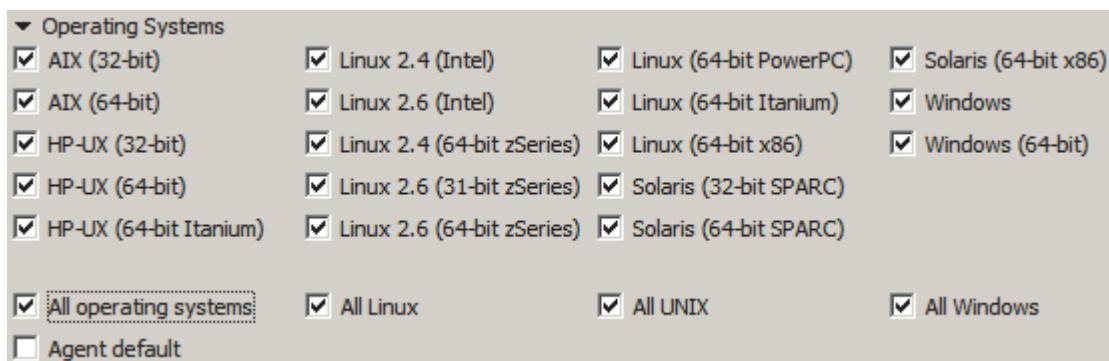
CIM_Processor properties	
Name	Type
AddressWidth	uint16
Availability	uint16
Caption	string
ConfigMan...	uint32
ConfigMan...	boolean

20. Click **OK**.

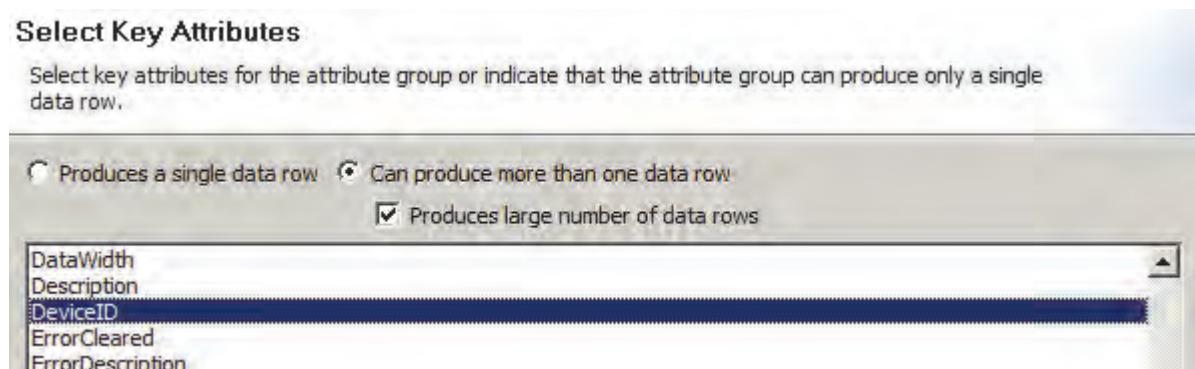
Your CIM information is displayed in the Common Information Model (CIM) Information window.



Because CIM is platform independent, unlike WMI, this monitor can collect CIM_Memory from any host with a CIM agent.

21. Expand Operating Systems and select **All operating systems** in the **Operating Systems** pane.22. Click **Next**.

The Select key attributes window opens.

23. Keep **Can produce more than one data row** selected and click **DeviceID** from the list.24. Click **Finish**.

You return to the **Data Source Definition** tab and the new AB_Processor subnode is displayed.

Data Source Definition

Attribute Group Information

- AB2 (Agent)
 - + Availability filters
 - + Processor_Utilization (Script)
 - + AppDataSocket (Socket)
 - AB_Processor (Subnode)
 - + CIM_Processor (CIM)

25. Expand **AB_Processor (Subnode)**.

The new CIM data source is displayed.

Data Source Definition

Attribute Group Information

- AB2 (Agent)
 - + Availability filters
 - + Processor_Utilization (Script)
 - + AppDataSocket (Socket)
 - AB_Processor (Subnode)
 - + CIM_Processor (CIM)

26. Browse the attributes of the **CIM_Processor** attribute group.

27. Click the **Runtime Configuration** tab.

A list of new CIM runtime configuration objects is displayed.

Runtime Configuration Information

Runtime Configuration Information

- Custom Configuration
- + Configuration for Common Information Model (CIM)

28. Expand **Configuration for Common Information Model (CIM)** and **CIM Connection Configuration**.

Runtime Configuration Information

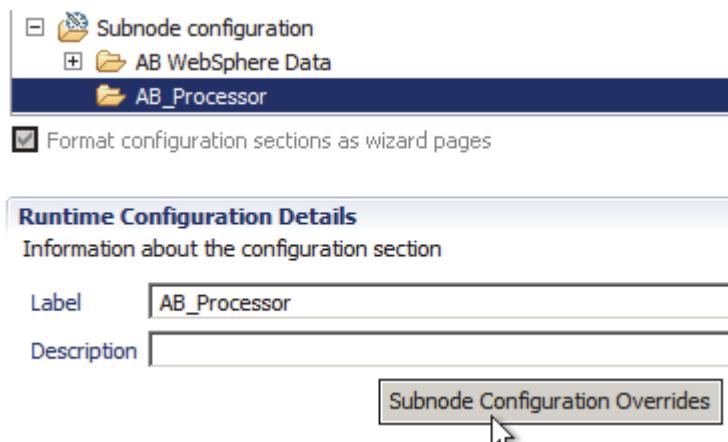
Runtime Configuration Information

- Configuration for Common Information Model (CIM)
 - CIM Connection Configuration
 - + CIM Local or Remote
 - + CIM user ID
 - + CIM password
 - + CIM host name
 - + CIM over SSL

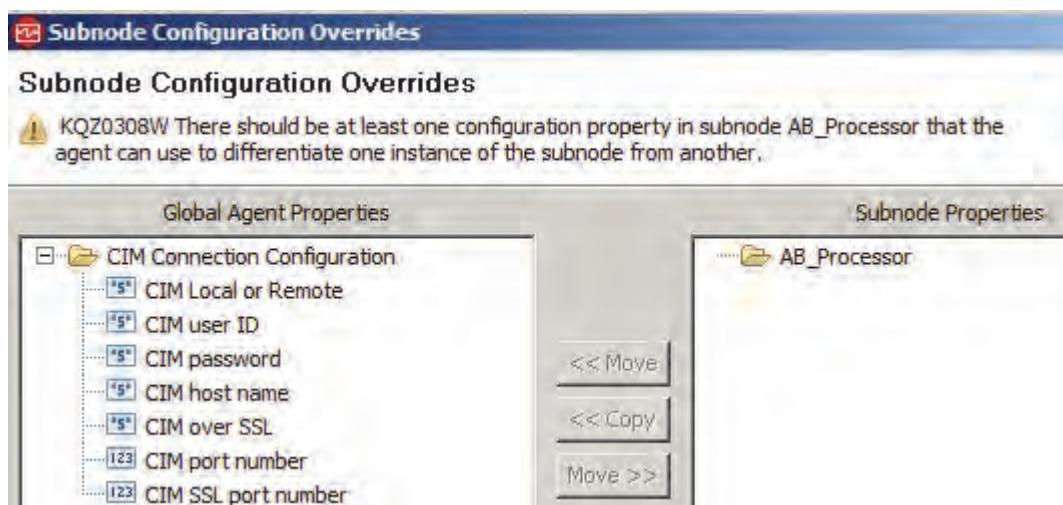
Format configuration sections as wizard pages

These CIM parameters are set when the agent is configured. You can require values, set default values, and configure if properties are prompted for when configuring the agent. By moving these configuration parameters into the subnode, you can set them differently for each subnode.

29. Set the Configuration for Common Information Model properties to be configured per subnode.
 - a. Close **Configuration for Common Information Model (CIM)**.
 - b. Expand **Subnode Configuration** and click **AB_Processor**.
 - c. Click **Subnode Configuration Overrides**.



The Subnode Configuration Overrides window opens.



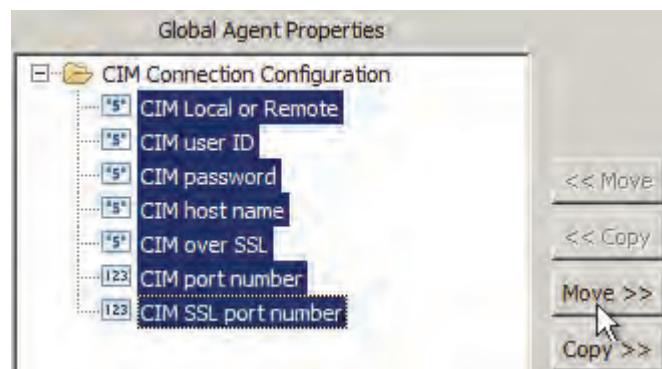
In this window, you can determine which configuration data, which you set during installation of the agent, is configurable for each subnode. Currently, all properties are global. At the time of the installation, you enter them once and they apply to all subnodes. Because the agent might monitor multiple remote CIM servers, this result is not acceptable.

At a minimum, you set the CIM host name as a subnode property so that you can identify the host systems. Other information that you might set per subnode includes the user ID, password, and ports to be used. In this exercise, set all CIM possible properties configurable for each subnode.

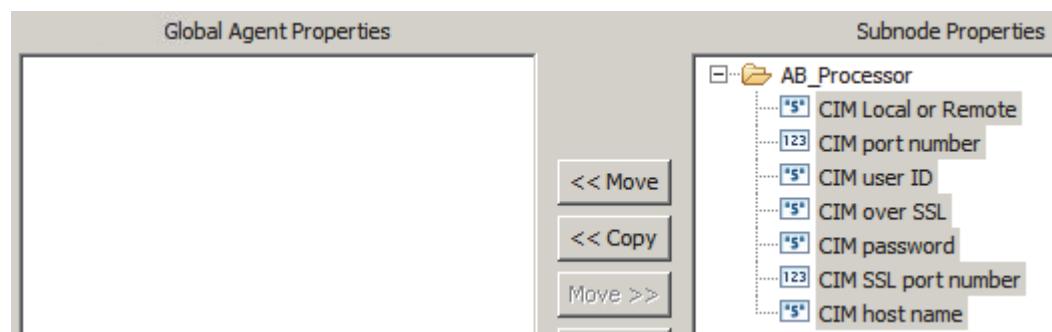
d. Select the following properties

- ◆ CIM Local or Remote
- ◆ CIM user ID
- ◆ CIM password
- ◆ CIM host name
- ◆ CIM over SSL
- ◆ CIM port number
- ◆ CIM SSL port number

e. Click **Move>>**.



The subnode configuration overrides looks like the following screen capture when you are done.



You set the configuration properties when you define subnodes.

30. Click **OK** to save your changes and exit the Subnode Configuration Overrides window.

You are returned to the Data Source Definition tab.

31. Save your agent project.

Add monitoring of remote WebSphere Application Server Java Management Extensions

In this section, you enable the agent to gather and monitor Active MQ queues within a WebSphere server.

Because this data source cannot be used on all servers where the agent is installed, you use a subnode to define the data to be retrieved. At run time, if you want to monitor this JMX data source, configure the agent to create a subnode.

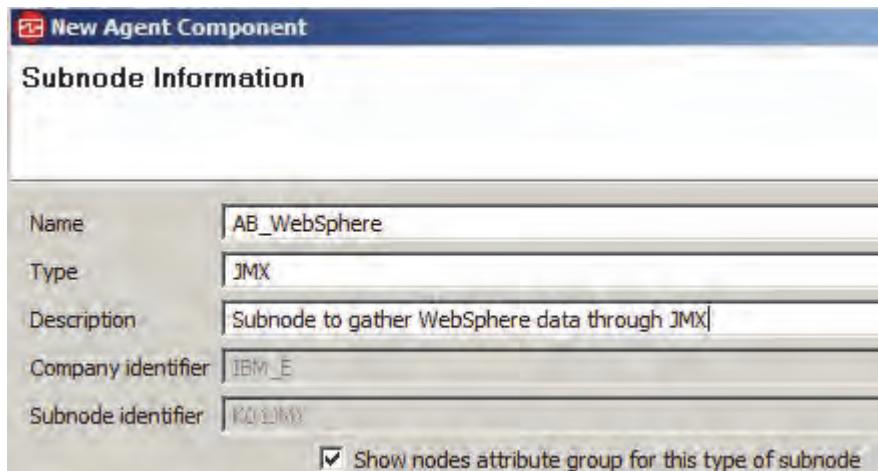
Complete the following steps:

32. In the **Data Sources** tab, right-click **AB2 (Agent)** and select **Add Subnode Definition**.

The Subnode Information window opens.

33. Enter the following information:

- Name: **AB_WebSphere**
- Type: **JMX**
- Description: **Subnode to gather WebSphere data through JMX**

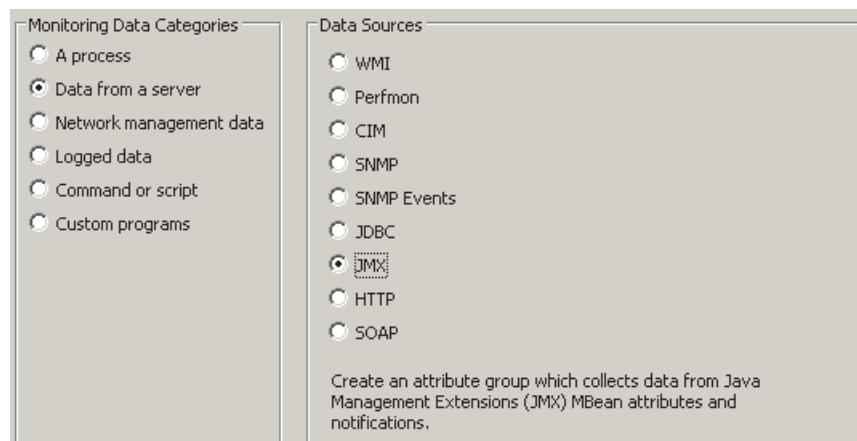


Note: Type is a unique three-character code identifying this subnode.

34. Click **Next**.

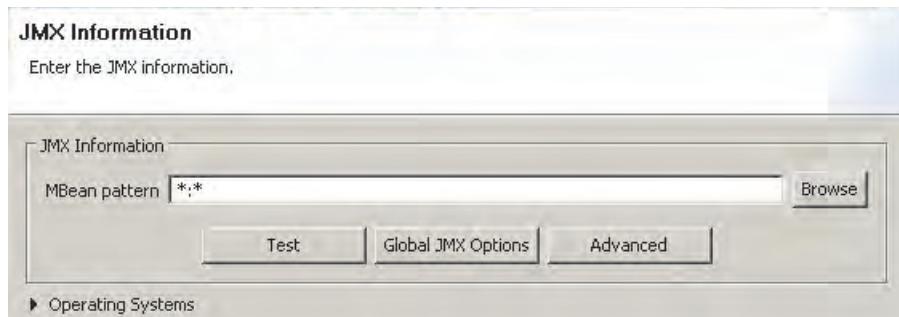
The Agent Initial Data Source window is displayed.

35. Click Data from a server under Monitoring Data Categories and JMX under Data Sources.



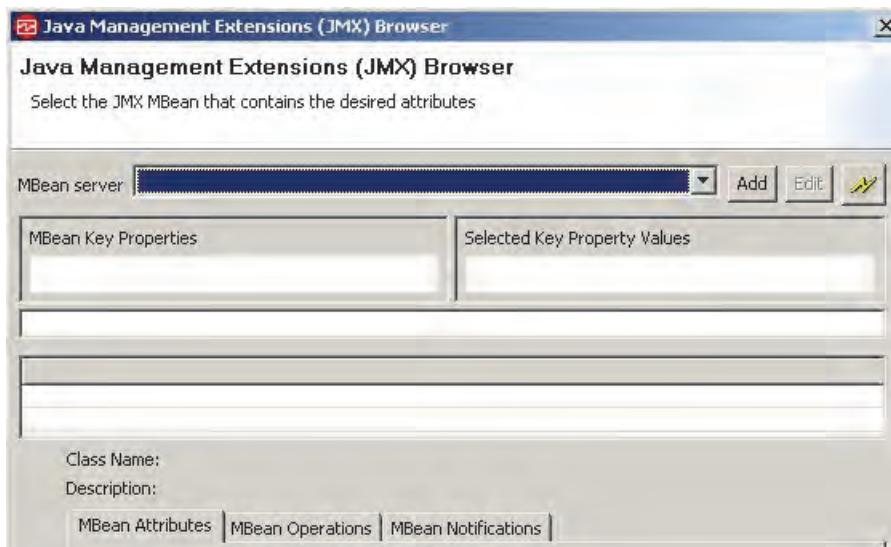
36. Click Next.

The JMX Information window opens.



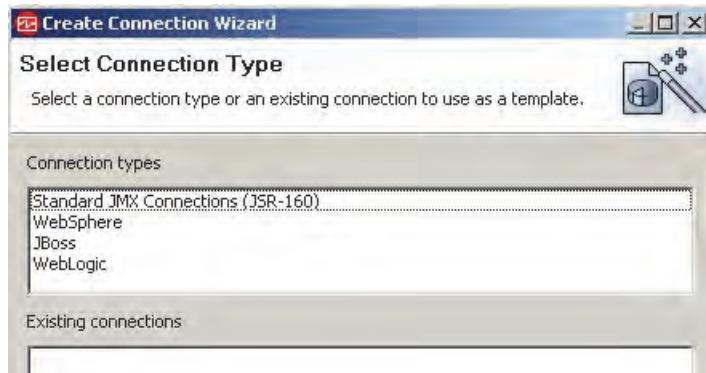
37. Click Browse.

The Java Management Extensions (JMX) Browser opens.

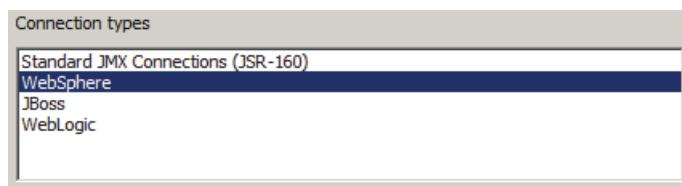


38. Click Add.

The JMX Connection Selection window opens.

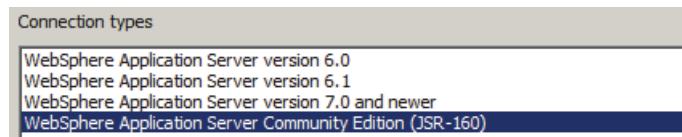


39. Click **WebSphere** in the **Connection Types** pane and click **Next**.

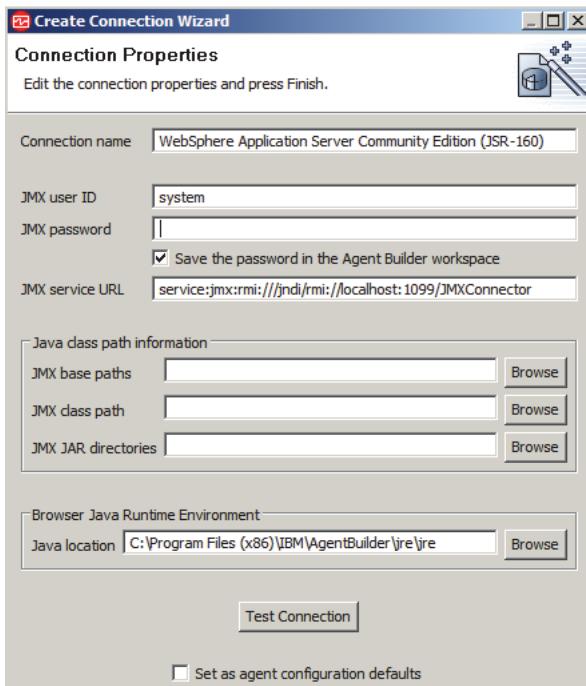


The connections templates are displayed in the window.

40. Click **WebSphere Application Server Community Edition** and click **Next**.



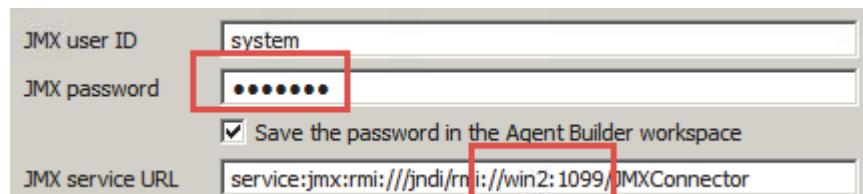
The Connection Properties window opens.



41. Enter **manager** in JMX password.

Because WIN2 is the only host with WebSphere Application Server Community Edition installed, it is the target for browsing.

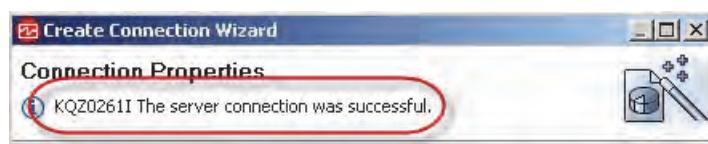
42. Replace **localhost** in JMX service URL with **WIN2**.



43. Click **Test Connection**.



You receive an indication that the connection was successful.

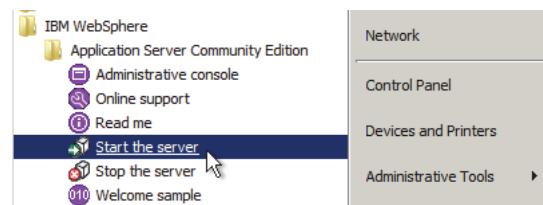


If your connection was not successful, do the following tasks:

- Confirm that WIN2 is running, its network card is enabled, and that the WIN1 server can be pinged.
- Confirm that WAS Community Edition is running.



- If it is not running, you must start it.



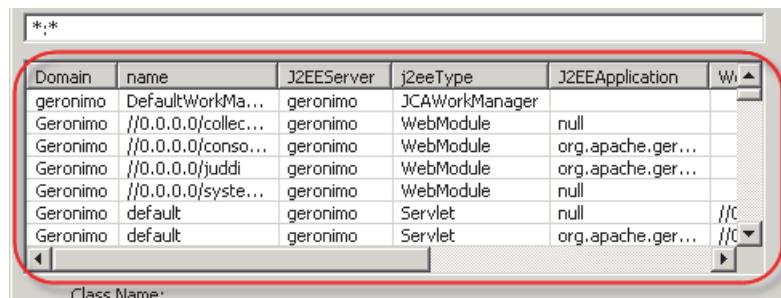
44. When you have a successful test connection, click **Finish**.

The Java Management Extensions (JMX) Browser is displayed with data now shown in the various windows.

The screenshot shows the JMX Browser interface. At the top, there is a toolbar with buttons for Add, Edit, and a refresh symbol. Below the toolbar, there are two tables: "MBean Key Properties" and "Selected Key Property Values". The "MBean Key Properties" table lists properties like [Domain], name, J2EEServer, j2eeType, J2EEApplication, WebModule, and type. The "Selected Key Property Values" table is currently empty. At the bottom, there is a table showing data for three domains: Geronimo. The columns are Domain, name, J2EEServer, j2eeType, J2EEApplication, WebModule, and Other Key. The data rows are:

Domain	name	J2EEServer	j2eeType	J2EEApplication	WebModule	Other Key
Geronimo	//0.0.0.0/	io	geronimo	WebModule	null	
Geronimo	//0.0.0.0/activ...	geronimo		WebModule	null	
Geronimo	//0.0.0.0/CAHe...	geronimo		WebModule	null	

45. Browse the MBeans to familiarize yourself with the tool. The current pattern (*.*) retrieves all MBeans.
- Before selecting any MBean key properties, scroll through the MBean list to see the MBeans from which you can pull data.



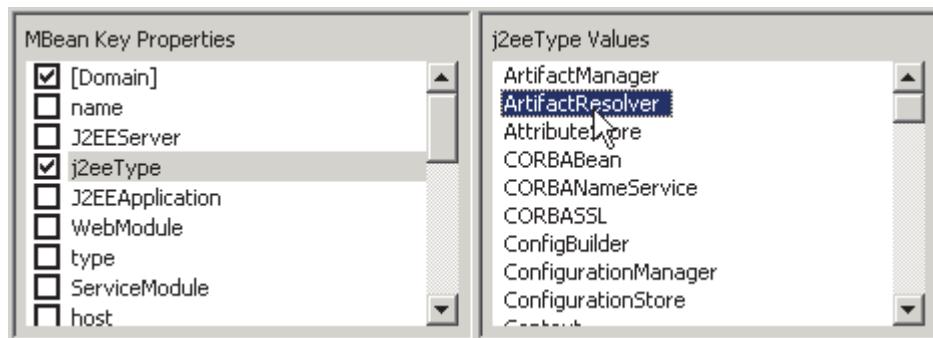
The screenshot shows a table titled 'MBeans' with a red box highlighting the first seven columns: Domain, name, J2EEServer, j2eeType, J2EEApplication, and WebModule. The table lists various Geronimo MBeans, such as DefaultWorkManager, JCAWorkManager, and several WebModules and Servlets.

Domain	name	J2EEServer	j2eeType	J2EEApplication	WebModule
geronimo	DefaultWorkMa...	geronimo	JCAWorkManager		
Geronimo	//0.0.0.0/collect...	geronimo	WebModule	null	
Geronimo	//0.0.0.0/consol...	geronimo	WebModule	org.apache.ger...	
Geronimo	//0.0.0.0/juddi	geronimo	WebModule	org.apache.ger...	
Geronimo	//0.0.0.0/system...	geronimo	WebModule	null	
Geronimo	default	geronimo	Servlet	null	//0.0.0.0/default
Geronimo	default	geronimo	Servlet	org.apache.ger...	//0.0.0.0/default

- Scroll to the right and review the column names.

Notice that the first set of column names matches the first names in the MBean Key Properties list. The final column, Other Key Properties, shows all the other MBean Key Properties.

- Select a single **MBean Key Property**. If **Key Property Values** are shown, click them one at a time.

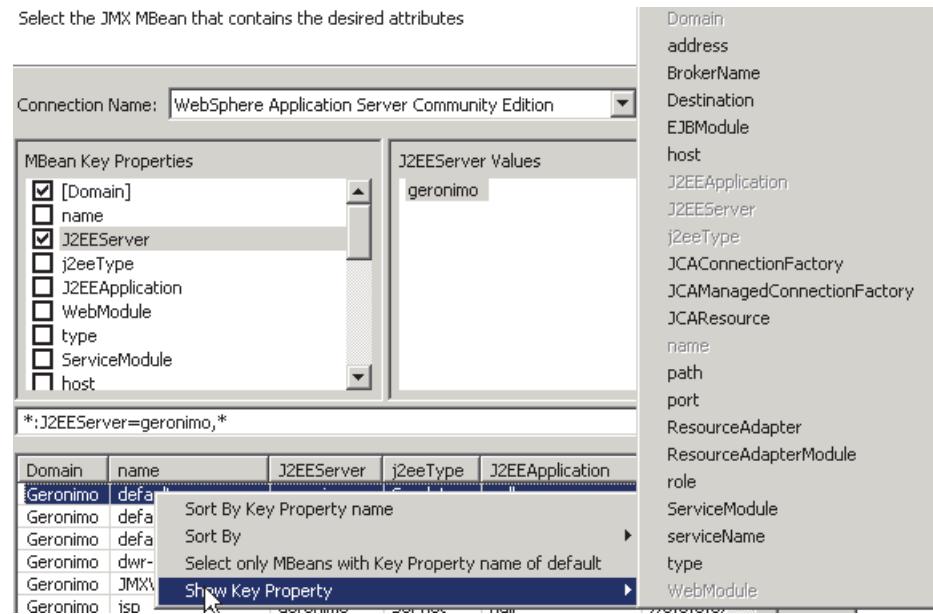


When you select an item from the Key Properties list, the item modifies the Object Name pattern. In the following table, the item shows the pattern all of the MBeans that match the new pattern. Selecting a key property also displays the list of values of that key property in the list on the right. Selecting a key property value further modifies the Object Name pattern and further filters the list of matching MBeans.

- Clear the **MBean Key Property** that you selected, select another property, and select its values one at a time. Repeat this process for several key properties.

You can select one or more key properties, but you can select only one value for each key property. You can also sort each column.

- e. Right-click any cell in the list of MBeans.



A context-sensitive menu is displayed with other options that can help you locate the MBean that you want.

The goal is to create an MBean pattern that filters all unwanted MBeans and then to select a specific MBean from the list that contains the attributes that you want to monitor. You can select only one MBean from the list, regardless of the object pattern.

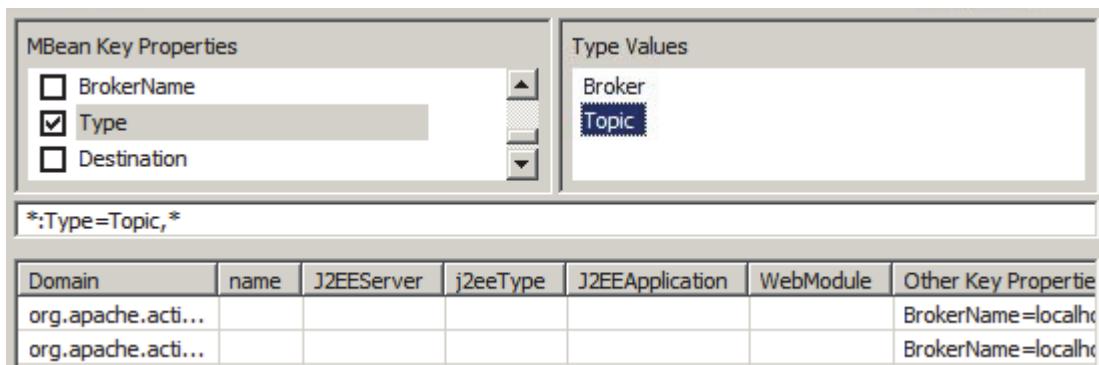
Use the Object Name pattern, the sorting feature, and the menu to locate the MBean you want.

- f. When you located a specific MBean of interest, select it in the list.

On the **MBean Attributes** tab is a list of attributes that are collected for the MBean attribute group you selected. Now it is time to select a specific MBean to monitor.

46. Clear any **MBean Key Properties** selections you made, but keep **Domain** selected.

47. Select **Type** in **MBean Key Properties** and **Topic** in **Type Values**.





Important: A type and Type MBean Key Property exist. Select **Type**.

48. Click the first MBean in the MBean list.

The screenshot shows the JMX Information window with the following details:

- MBean List:** The search bar shows "*:Type=Topic,*". The table lists two MBeans:

Domain	name	J2EEServer	j2eeType	J2EEApplication	WebModule	Other Key Properties
org.apache.activemq.broker.jmx						BrokerName=localhost
org.apache.activemq.broker.jmx						BrokerName=localhost
- MBean Details:**
 - Class Name:** org.apache.activemq.broker.jmx.TopicView
 - Description:** Information on the management interface of the MBean
 - MBean Attributes Tab:** Contains a table of attributes:

Name	Description	Type	Read/Write
Name	Attribute exposed for ma...	java.lang.String	Read Only
EnqueueCount	Attribute exposed for ma...	long	Read Only
DispatchCount	Attribute exposed for ma...	long	Read Only
DequeueCount	Attribute exposed for ma...	long	Read Only
ConsumerCount	Attribute exposed for ma...	long	Read Only
QueueSize	Attribute exposed for ma...	long	Read Only
MemoryPercentageUsed	Attribute exposed for ma...	int	Read Only
MemoryLimit	Attribute exposed for ma...	long	Read/Write

The attributes that are listed in the **MBean Attributes** tab are the ones to be monitored.

49. Click **OK** to save your selection.

The JMX Information window is displayed with the MBean pattern completed.

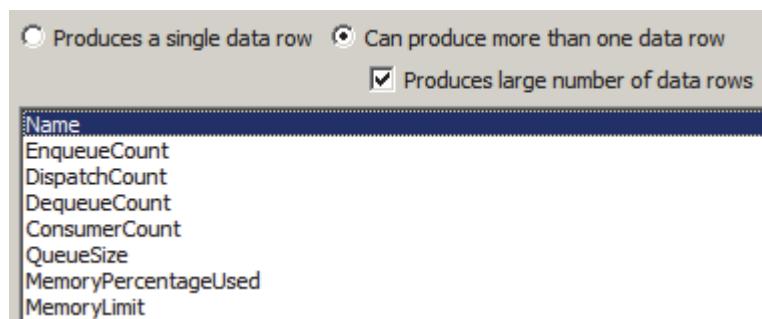
The screenshot shows the JMX Information window with the following details:

- MBean pattern:** *:Type=Topic,*
- Buttons:** Test, Global JMX Options, Advanced

50. Click **Next**.

The Select Key Attributes window opens and lists specific attributes for the MBean that you selected.

51. Keep **Can produce more than one data row** selected and click **Name** from the list.



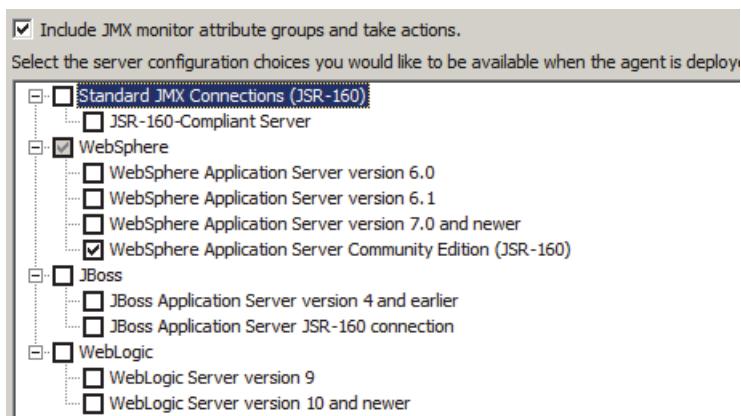
52. Click **Next**.

The JMX Agent-Wide Options window opens.



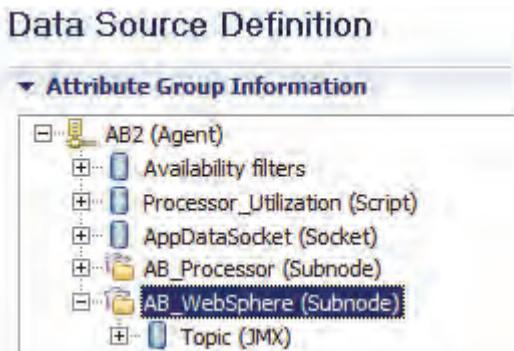
Even though you browsed MBeans from a WebSphere Community Edition application server, this MBean might be used by several types of application servers. With this window, you can identify all the types of application servers that you can monitor for this MBean.

53. Expand **WebSphere** and select **WebSphere Application Server Community Edition**.



54. Click **Finish**.

You are returned to the **Data Source Definition** tab.



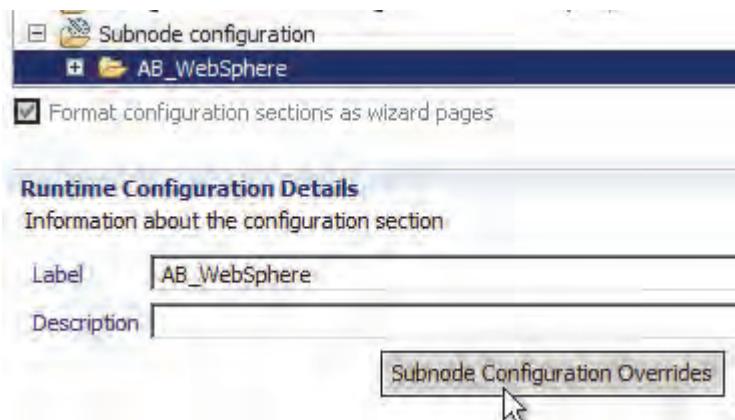
55. Browse the attributes of each new JMX attribute group.
56. Move the JMX Runtime Configuration properties to the subnode.
 - a. Click the **Runtime Configuration Information** tab.
 - b. Expand **Configuration for Java Management Extensions** and **WebSphere Application Server Community Edition**.



You set these WebSphere Application Server Community Edition properties when you configure the agent and establish the connection into the JMX server, whether it is local or remote. By moving these configuration parameters into the subnode, you can set them differently for each subnode.

- c. Close **Configuration for Java Management Extensions**.
- d. Expand **Subnode Configuration** and click **AB_WebSphere**.

e. Click **Subnode Configuration Overrides**.



f. Select the following properties and click **Move>>**.

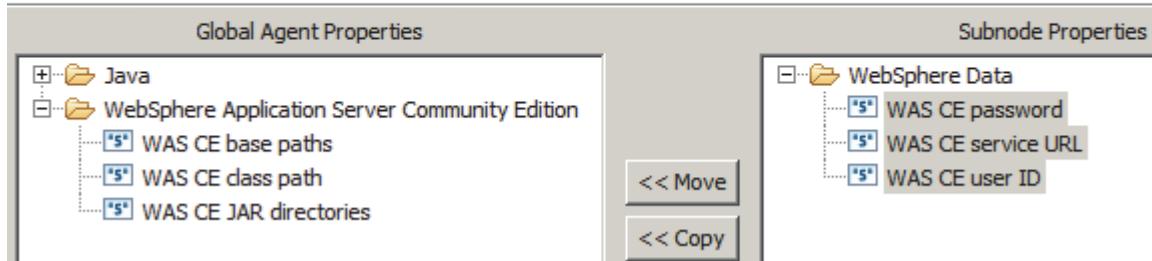
- ◆ WAS CD user ID
- ◆ WAS CE password
- ◆ WAS CE service URL



The subnode configuration overrides looks like the following screen capture when you are done.

Subnode Configuration Overrides

Add configuration properties to subnode WebSphere Data which will differentiate one instance of the subnode from another when the agent runs.



- g. Click **OK** to save your changes and exit the window.
 57. Save your agent project.

Add a JDBC data source to the agent

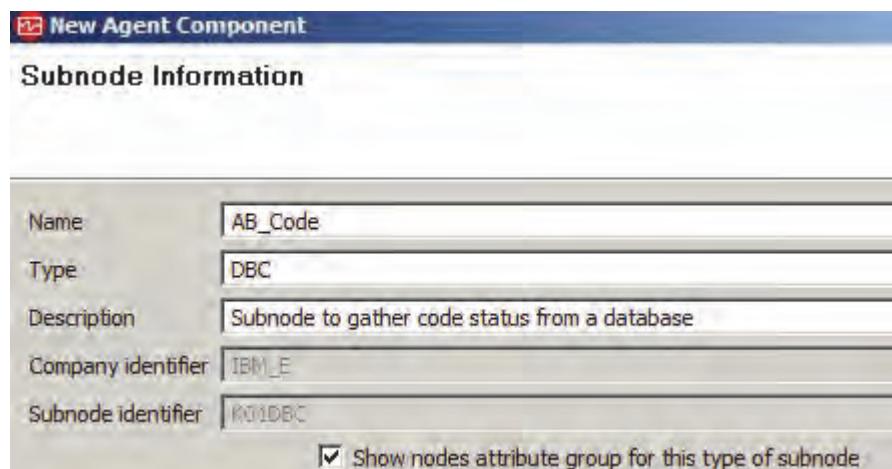
In this section, you enable the agent to gather user data from a database with JDBC.

58. In the **Data Sources** tab, right-click **AB2 (Agent)** and select **Add Subnode Definition**.

The Subnode Information window opens.

59. Enter the following information:

- Name: **AB_Code**
- Type: **DBC**
- Description: **Subnode to gather code status from a database**

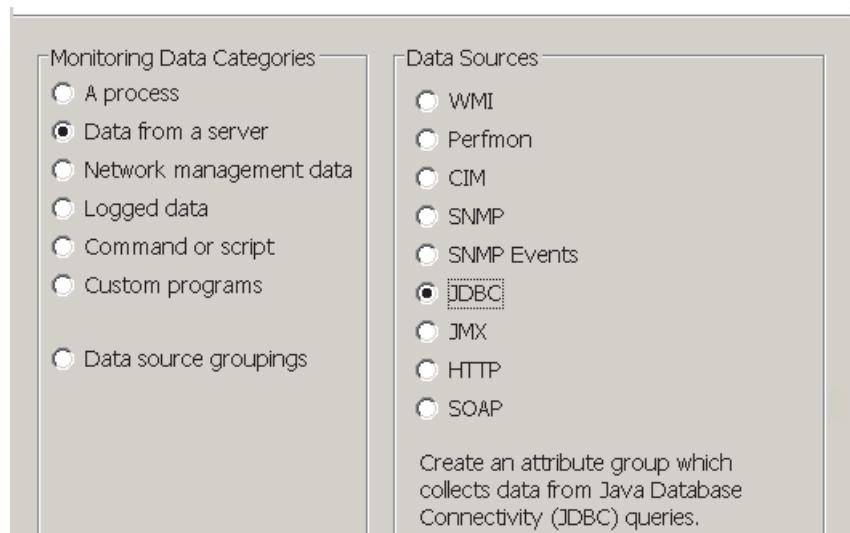


Note: **Type** is a unique three-character code that identifies this subnode.

60. Click Next.

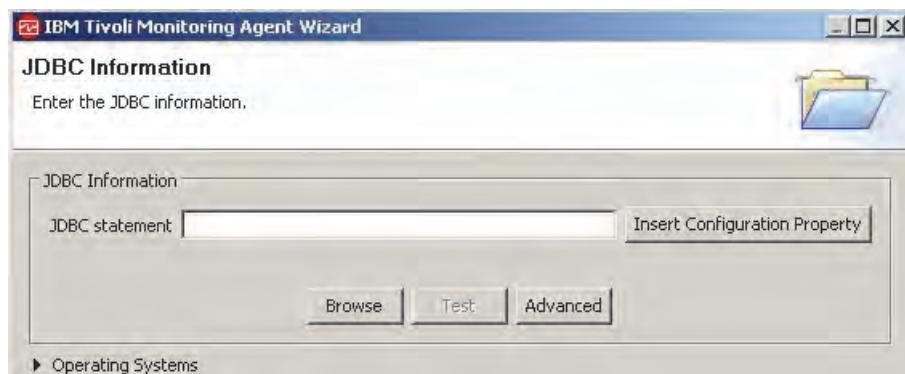
The Data Source Location window opens. Set your agent to monitor a database through JDBC.

61. Select **Data from a server from **Monitoring Data Categories** and **JDBC** from **Data Sources**.**



62. Click Next.

The JDBC Information window opens.

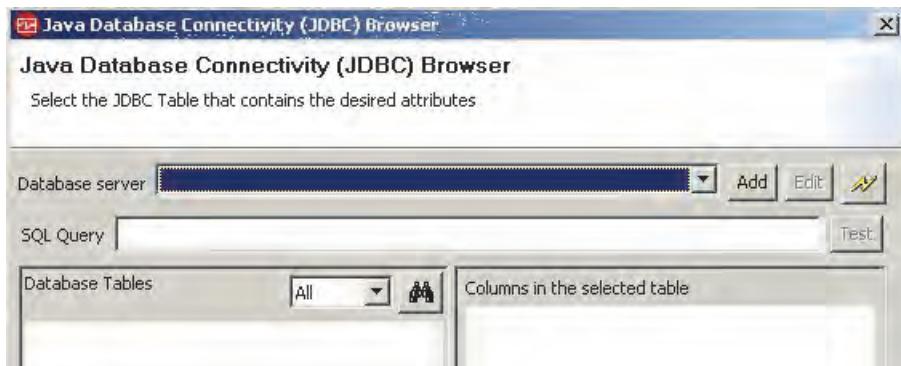


From this window, you can enter the JDBC statement to retrieve data from a database.

Typically, this statement is either an SQL command or a command that runs a stored procedure in the database. While you can enter the command manually, it is best to use the browser, which completes several tasks at once.

63. Click **Browse.**

The Java Database Connectivity (JDBC) Browser window opens.



To use the JDBC browser, you must first establish a JDBC connection with a local or remote database.

64. Click **Add**.



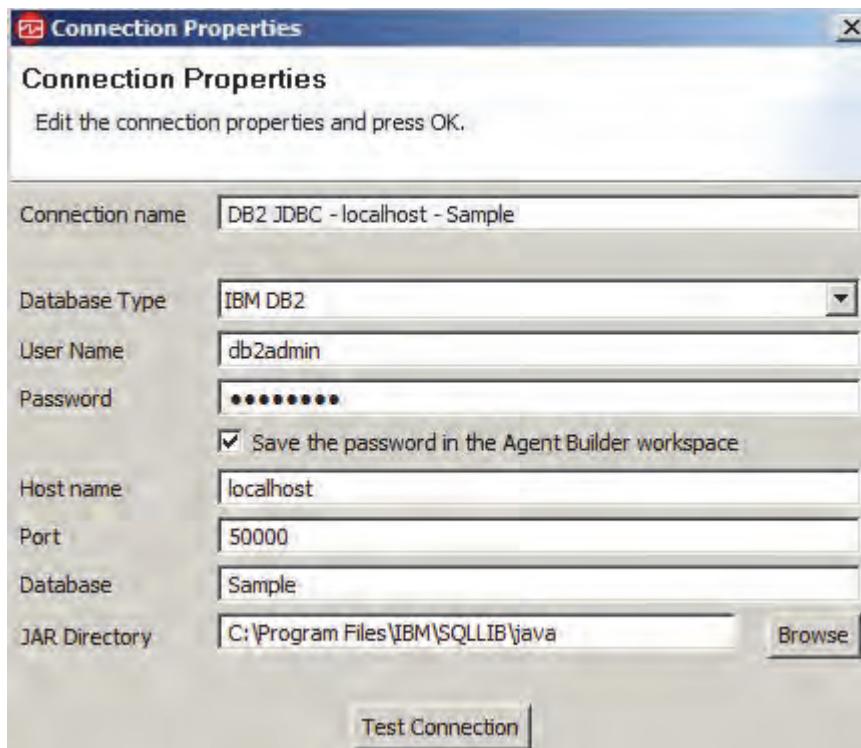
The Create Connection Wizard window opens.



65. Click **JDBC Connection** in the Connection types pane and click **Next**.

66. Keep the **Connection name** value as is and enter the following values:

- Password: **object00**
- Database: **Sample**



Notice that you can connect to a database on a remote host by entering the appropriate host name or IP address in the **Host name** field.

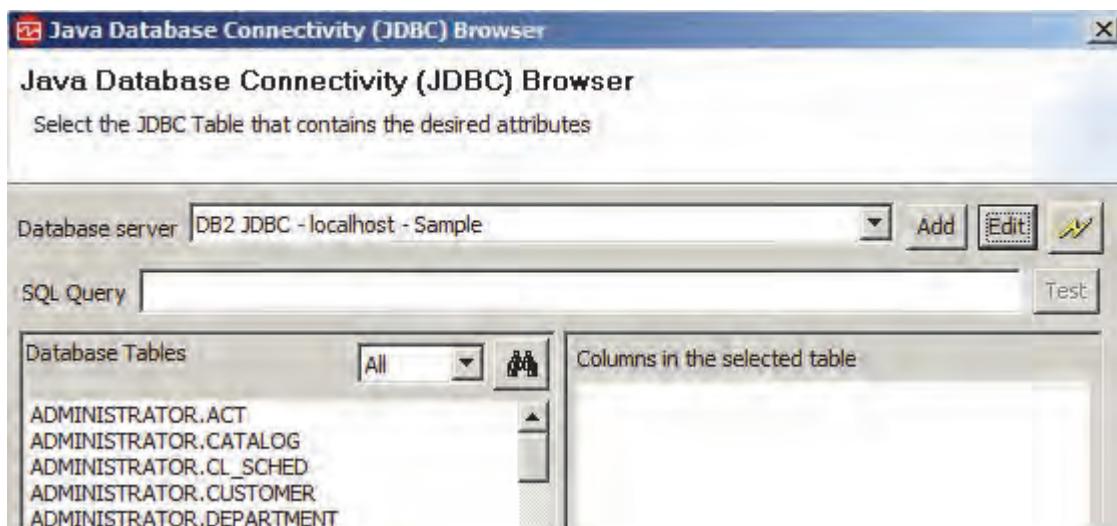
67. Click **Test Connection**.

A message that indicates a successful connection is displayed at the top of the window.

Connection Properties
(i) KQZ0261I The server connection was successful.

68. Click **Finish** to close the Connection Properties window.

You are returned to the JDBC browser, and the tables in the database are listed.



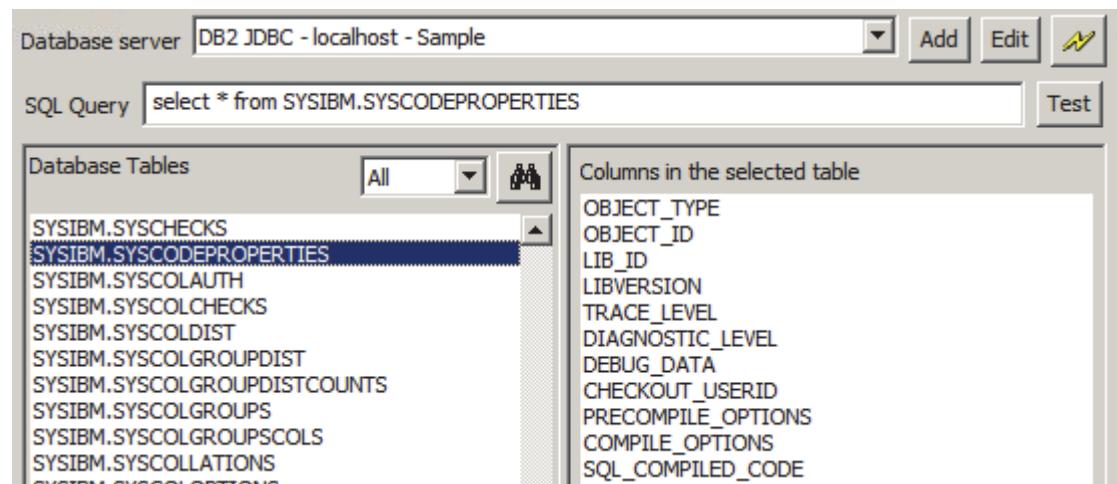
A browser does several important tasks:

- It confirms that your agent can connect to a target data source.
- It defines the exact data that you want gathered.
- It confirms that the data can be gathered.
- Additionally, the browser retrieves information about the data and the metadata from the data source. Using this metadata, Agent Builder can define attribute groups and attributes to hold the data, which is a critical task when defining an agent.

The purpose of browsing is not to identify a target host to monitor. The target host is identified when the agent is installed.

If you click a table in the Database Tables pane, the table column names are displayed in the Columns in the Select Table pane. An SQL SELECT statement is shown in the **SQL Query** field.

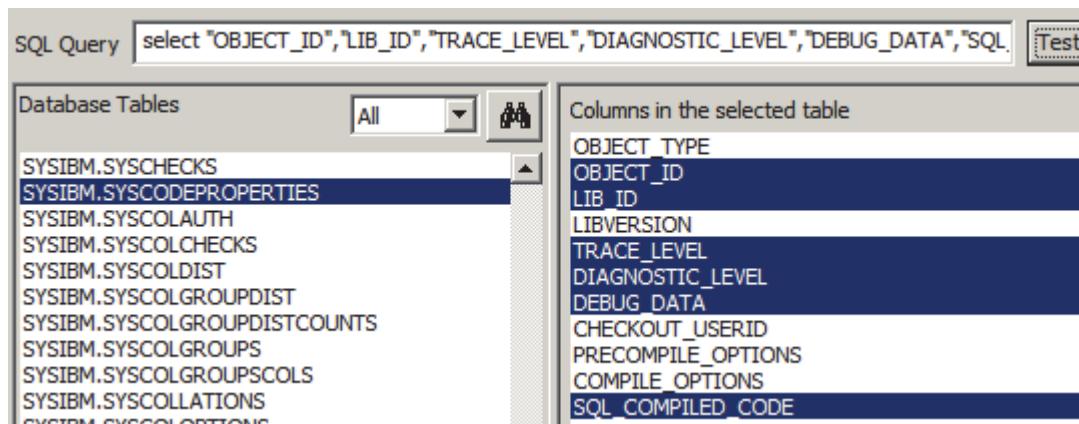
69. Find and select the **SYSIBM.SYSCODEPROPERTIES** table.



Notice that the SELECT statement includes an asterisk (*), which indicates that the command retrieves all the columns. You can manually edit this command to suit your needs. You can also select columns and modify the select statement.

70. Holding the CTRL key, click the following column names to select them. This action modifies the select statement to retrieve only data from the columns.

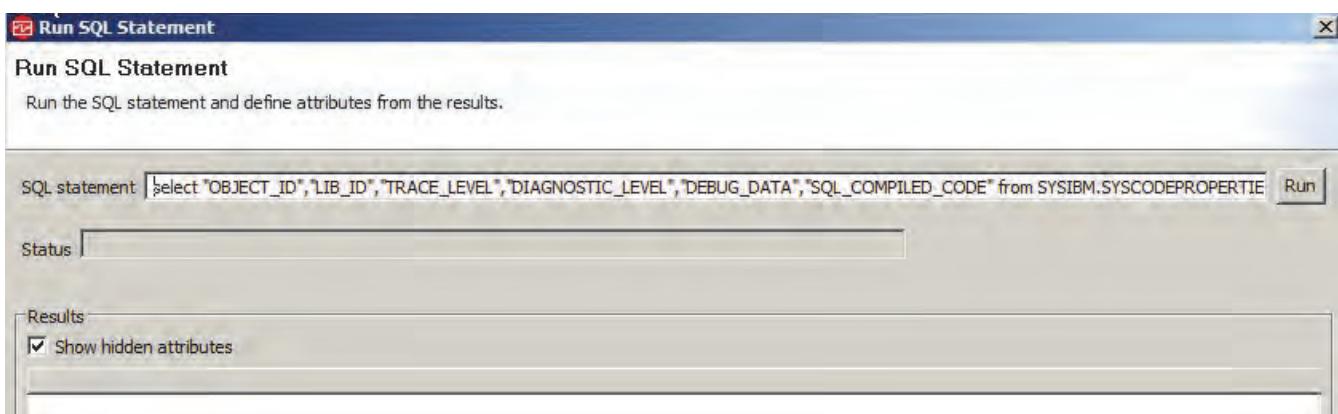
- OBJECT_ID
- LIB_ID
- TRACE_LEVEL
- DIAGNOSTIC_LEVEL
- DEBUG_DATA
- SQL_COMPILED_CODE



Notice that the SQL statement changed to retrieve only the columns that you selected.

71. Click **Test** to open the test utility.

The Test JDBC Statement window opens.



Notice that the SQL command you built is displayed. You can edit it manually here, if you want to test a different command.

72. Click **Run**.

73. If prompted, click **Yes** to view any suggested data types, and click **OK** to accept those data types.

The screenshot shows a utility window with a status bar at the top stating "Status Displaying the first 50 rows of 240 total rows." Below this is a section titled "Results" with a checked checkbox for "Show hidden attributes". A table displays six rows of data with columns: OBJECT_ID, LIB_ID, TRACE_LEVEL, DIAGNOSTIC_LEVEL, DEBUG_DATA, and SQL_COMPILED_CODE. The data is as follows:

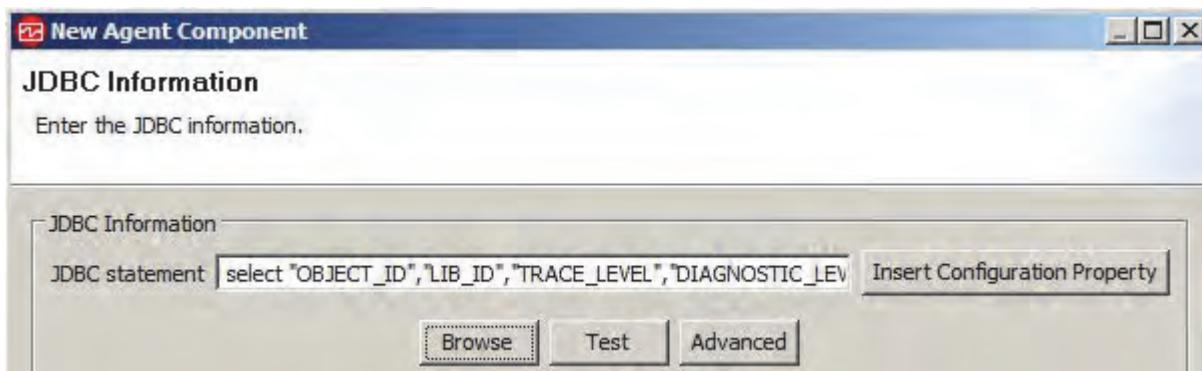
OBJECT_ID	LIB_ID	TRACE_LEVEL	DIAGNOSTIC_LEVEL	DEBUG_DATA	SQL_COMPILED_CODE
66124	1756752525	0	0	0	1356
66135	1854528969	0	0	0	1088
66139	1589140265	0	0	0	1008
66145	1864981170	0	0	0	1672

The utility displays the data that is retrieved. You can manually edit the statement and test different commands.

If you click **Cancel**, you return to the JDBC browser and the metadata that is gathered by this test is not kept. This action can result in an attribute group not being created or your changes not being saved. If you click **OK**, you exit the test utility and the JDBC browser, and the column definition information you viewed is used to create an attribute group and its attributes in your agent.

74. Click **OK** to close the test window.

The JDBC Information window opens.



Your SQL statement is shown in the **JDBC statement** field.

Next, you identify the operating systems on which this query is used.

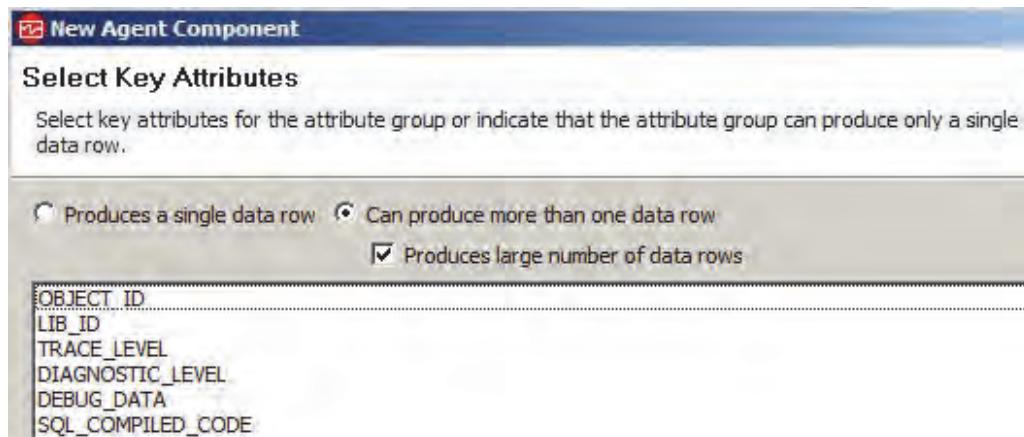
75. Expand **Operating Systems**.

As you can see, Agent Builder uses the default operating systems set for the agent, but because this agent monitors a database through a JDBC connection, this data source can be platform-independent.

76. Select **All operating systems**.

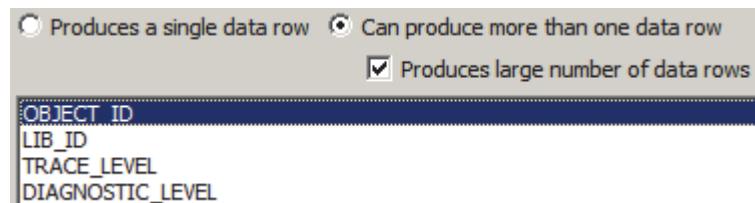
77. Click Next.

The Select Key Attributes window opens.



If the retrieved data consists of more than one row of data, you must define key attributes. Then, the agent can identify unique records and manage the data. For example, this SQL command retrieves information about Tivoli Enterprise Portal users. The agent needs to know which attribute uniquely identifies each user.

78. Select **OBJECT_ID.**

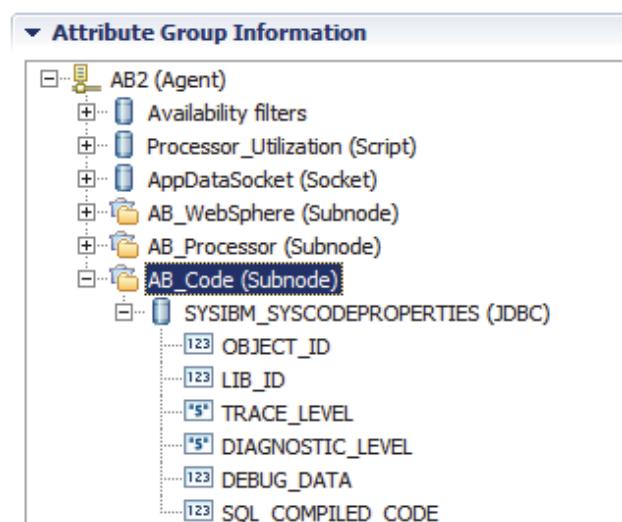


79. Click Finish.

You are returned to the **Data Source Definition** tab with the new subnode and attribute group.

80. Expand **AB_Code (Subnode).**

81. Expand **SYSIBM_SYSCODEPROPERTIES (JDBC).**



Notice that the attribute group name matches the target table name in the database. The attribute list is the subset of columns that you identified in your SQL command, not the full list of columns in the database table.

82. Move the JDBC Runtime Configuration properties to the subnode.

- Click the **Runtime Configuration Information** tab.

- Expand **Subnode Configuration** and click **AB_Code**.

You set these JDBC properties when you configure the agent and establish the connection into the database server, whether it is local or remote. By moving these configuration parameters into the subnode, you can set them differently for each subnode.

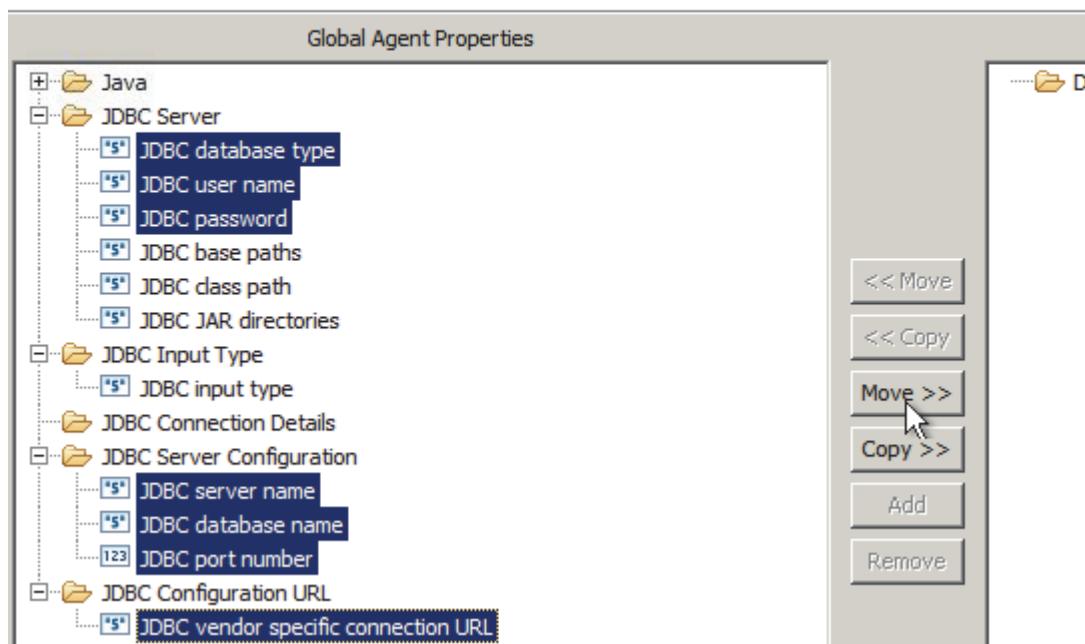
- Click **Subnode Configuration Overrides**.

- Select the following properties and click **Move>>**.

- ◆ JDBC database type
- ◆ JDBC user name
- ◆ JDBC password
- ◆ JDBC server name
- ◆ JDBC database name
- ◆ JDBC port number
- ◆ JDBC vendor specific connection URL

Subnode Configuration Overrides

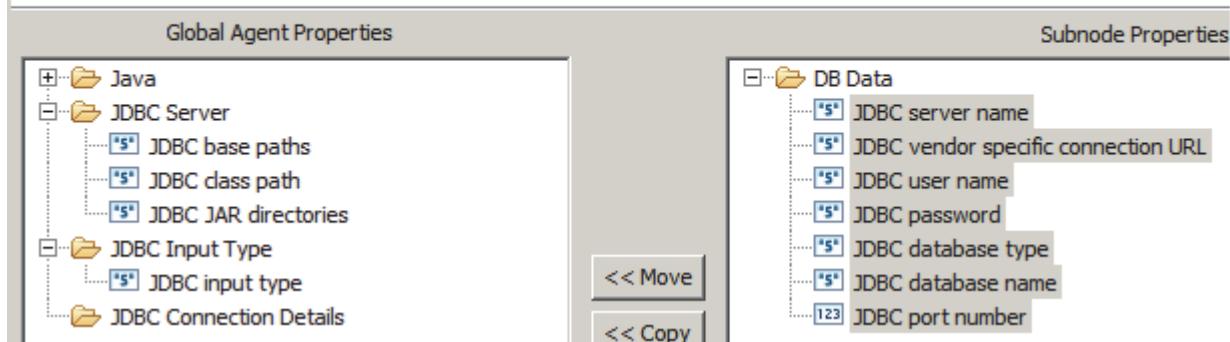
 KQZ0308W There should be at least one configuration property in subnode DB Data that the agent can use



The subnode configuration overrides looks like the following screen capture when you are done.

Subnode Configuration Overrides

Add configuration properties to subnode DB Data which will differentiate one instance of the subnode from another when the



e. Click **OK** to the window.

83. Save your agent project.

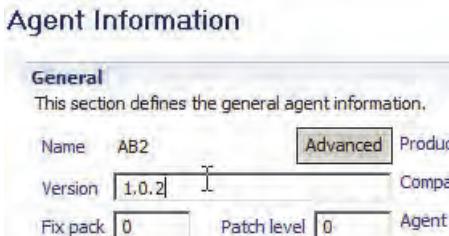
You successfully added a JDBC data source to this agent.

Change the agent version

Whenever you change an agent that is deployed into a monitoring environment, update the version number to ensure the monitoring environment processes any server updates.

84. Select to the **Agent Information** tab.

85. Change the version for 1.0.1 to **1.0.2**.



Important: If your initial version number is not 1.0.1, increase the current value by 1 to ensure that your monitoring server identifies the changes.

86. Save your agent project.

Exercise 4 Install and confirm the updated AB2 agent in an IBM Performance Management environment

In this exercise, you reinstall the AB2 agent onto WIN1 and LIN4. Then you confirm that data is being gathered for the new attribute groups you added.

1. Confirm that time is synchronized between the APM and WIN1 servers.
2. Confirm that all APM services are running on APM.

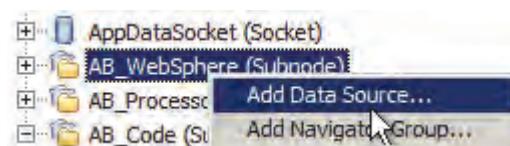
Create Summary Dashboard data points for the AB_WebSphere subnode

In this section, you create a single row data source that gathers the data for the AB_WebSphere subnode summary dashboard.

Your goal is to create a summary dashboard like this one:

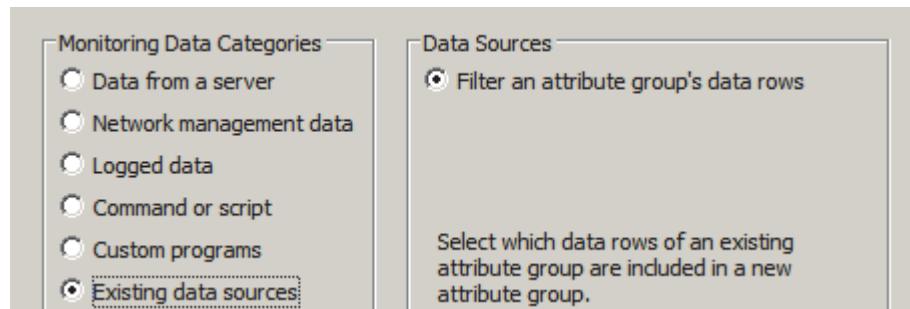


3. Create a filter data source in the AB_WebSphere subnode that selects only the **ActiveMQ.Advisory.Topic** queue data from **Topic**.
 - a. Click the **Data Sources** tab.
 - b. Right-click **AB_WebSphere (subnode)** and select **Add Data Source**.



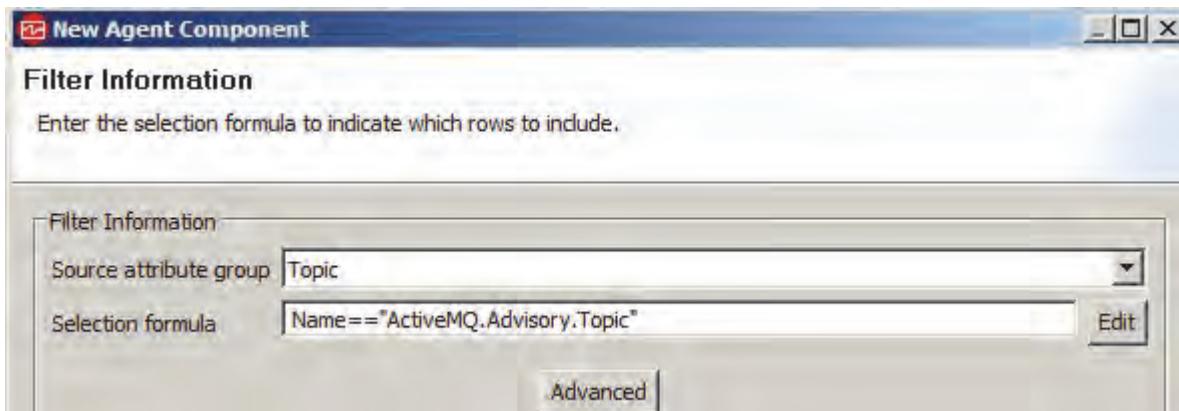
The New Agent Component window opens.

- c. Select **Existing data sources** > **Filter an attribute groups data rows**.



- d. Click **Next**.
e. Select **Topic** from the **Source attribute group** menu.
f. Enter the following text in the Selection Formula row.

Name=="ActiveMQ.Advisory.Topic"



This command pulls the single row from Topic where the queue name equals the ActiveMQ.Advisory.Topic. The status of this queue is used as the overall status on the Summary dashboard.



Hint: Look to the top of the window to see whether the editor identifies errors in your formula.

- g. Click **Finish**.

- Set the **Topic_Filtered (Filter)** attribute group to produce a single data row.

Filtered Attribute Group Information

Attribute group name **Topic_Filtered**

Help Filtered data rows from attribute group Topic.

Produces a single data row Can produce more than one data row
 Produces large number of data rows

- Select the **Topic_Filtered > Name** attribute and clear **Key attribute**.

Attribute Group Information

	Topic_Filtered (Filter)
	Name
	EnqueueCount
	DispatchCount

Filtered Attribute Information

Attribute name **Name**

Help Attribute exposed for management

Hidden - can only be used in derived attribute
 Key attribute

- Test the full agent and confirm that **Topic_Filtered (Filter)** returns one row of data.

Attribute Group Test

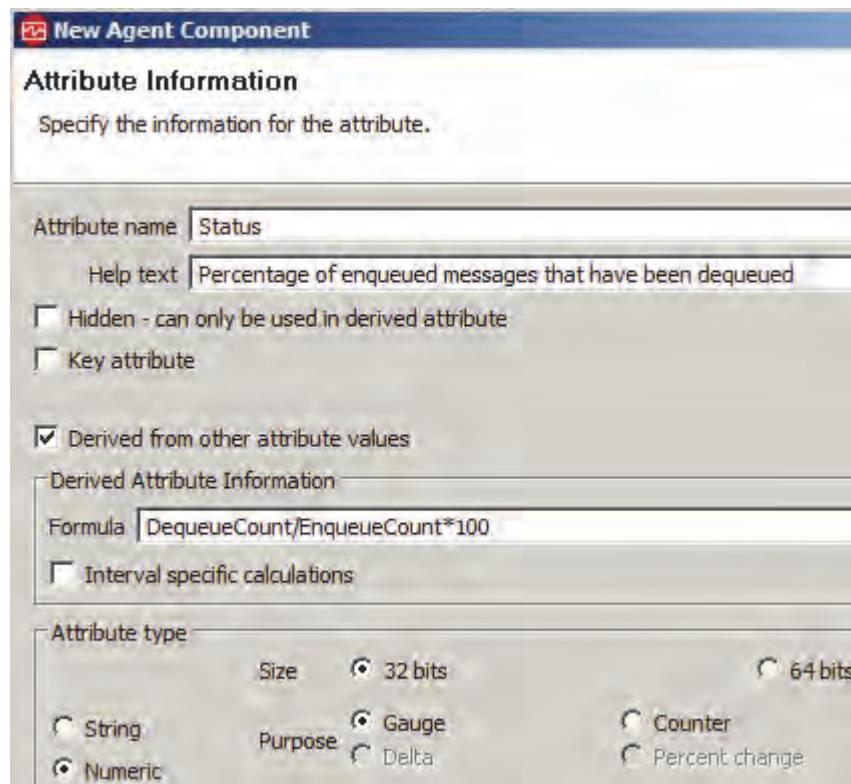
1 data row returned at Feb 23, 2017 4:00:55 PM.

Name	EnqueueCount	DispatchCount	DequeueCount	ConsumerCount	Queue
ActiveMQ.Advisory.Topic	4	0	0	0	0

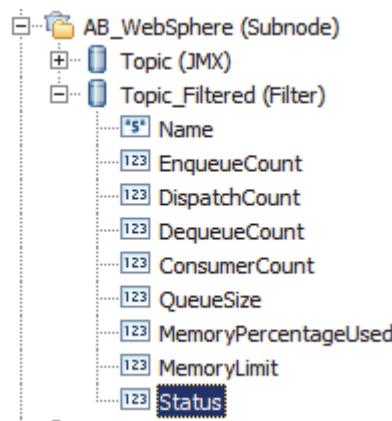


Hint: In Agent Test, you must display Topic data before you can see Topic_Filtered data.

7. Create an attribute on **Topic_Filtered (Filter)** named **Status** that calculates the percentage of enqueued messages that are dequeued.



When the Status attribute is created, the Topic_Filtered (Filter) data source looks like the following screen capture:



- Configure the **Status** attribute under the filtered attribute group to be the main status metric by setting the following Severity criteria:

The screenshot shows the Agent Builder interface with the following details:

- Outline View:** Shows a tree structure with nodes: MemoryLimit, Status, and AB_Processor (Subnode).
- Properties View:** A table titled "Severity" with columns "Severity" and "Value". It contains four rows:

Severity	Value
Normal	60 - 100
Warning	21 - 59
Critical	0 - 20
Not defined	

 Buttons for "Add", "Edit", and "Remove" are visible on the right side of the table.

When you are finished, the Severity table looks like the one shown at the start of this step.

- Save your agent project.

You successfully created an attribute group to build a Summary dashboard for the AB_WebSphere subnode.

Create dashboards and resource definitions

- Select **Dashboards** from the Agent Builder Outline view.

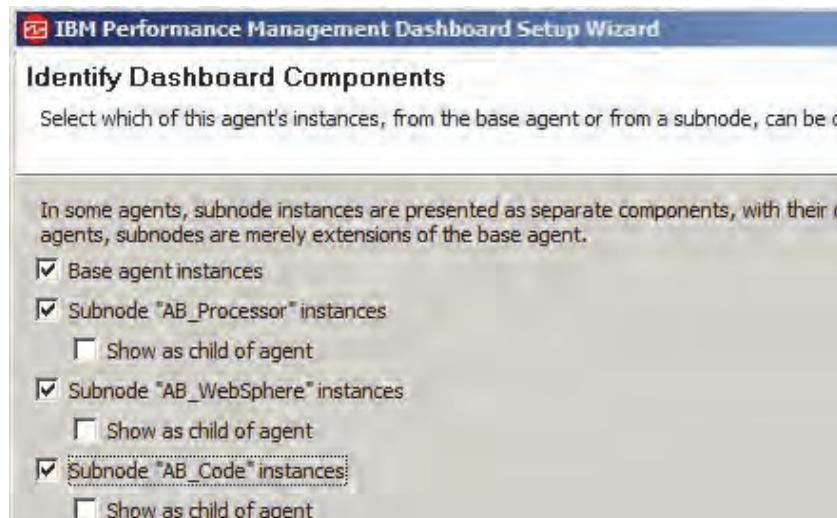
The Dashboards Overview opens.

- Click the **Dashboard Setup wizard** link in the Overview.

The IBM Performance Management Dashboard Setup wizard opens.

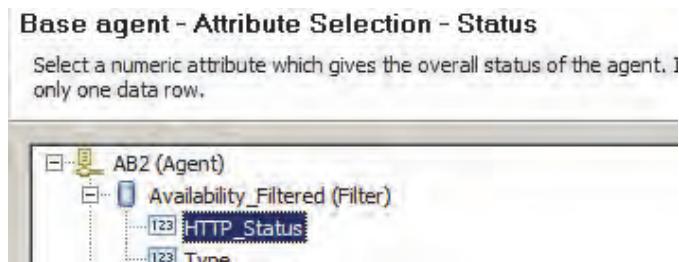
Because this agent is configured for subnodes, you are prompted whether to create a dashboard for the base agent and each subnode.

- Select all three subnode instances.



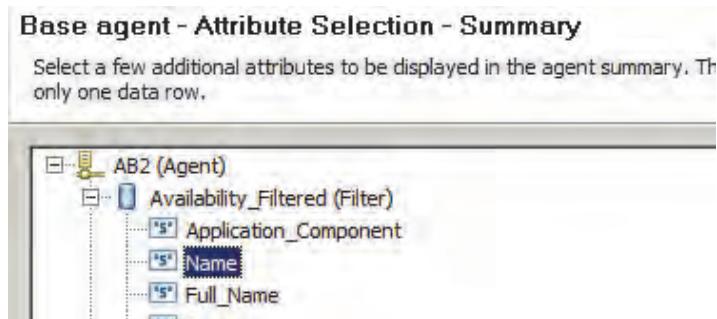
13. Click **Next**.

The Base agent - Attribute Selection - Status view opens. You are prompted to select a numeric attribute that gives the overall status of the base agent.



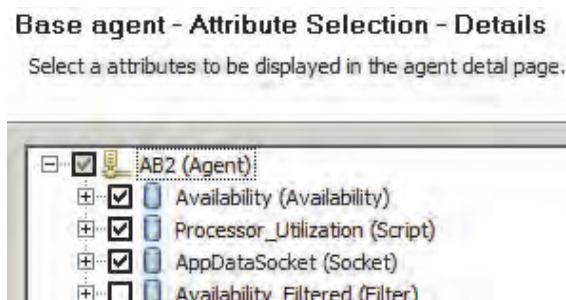
14. Keep the status attribute and click **Next**.

The Base agent - Attribute Selection - Summary view opens. You are prompted to select more attributes for the base agent Summary dashboard.



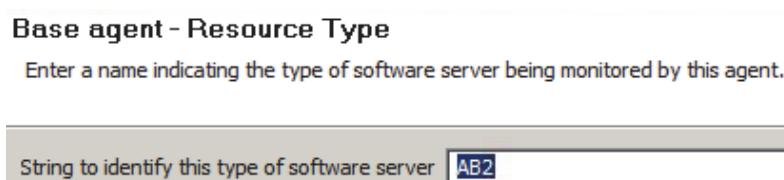
15. Keep the current attributes and click **Next**.

The Base agent - Attribute Selection - Details view opens. You are prompted to identify data groups and items to display in the base agent details dashboards.



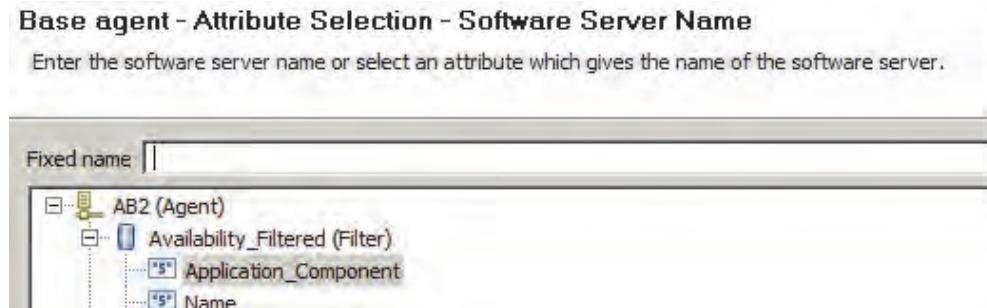
16. Keep the current attribute groups and click **Next**.

The Base agent - Resource Type view opens.



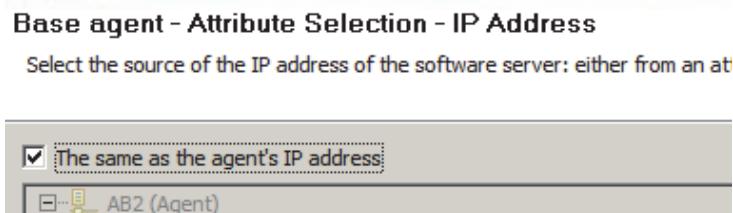
17. Keep **AB2** in the resource type name field and click **Next**.

The Base agent - Attribute Selection - Software Server Name view opens.



18. Keep the current attribute and click **Next**.

The Base agent - Attribute Selection - IP Address view opens.



19. Keep the current setting and click **Next**.

The Attribute Selection - Port view opens.

20. Keep the current value and click **Next**.

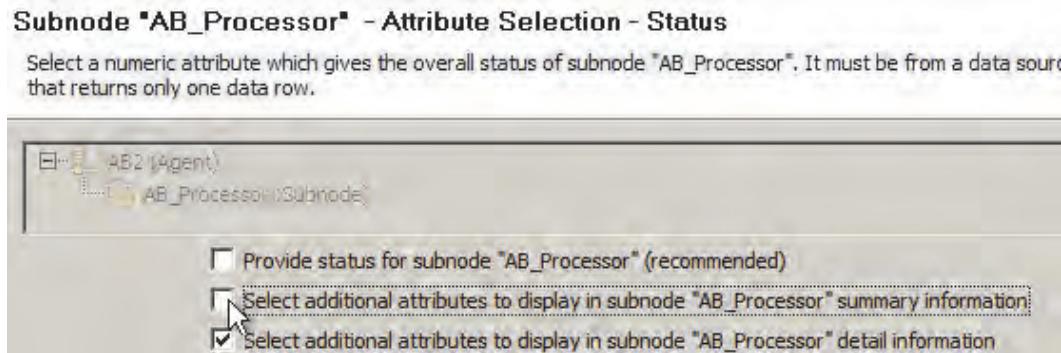


21. Click **Next**.

The Subnode "AB_Processor" - Attribute Selection - Summary view opens. You are prompted to select more attributes for the AB_Processor Summary dashboard.

Do not configure dashboards for this subnode.

22. Clear the first two check boxes to disable the Summary dashboard for this subnode.



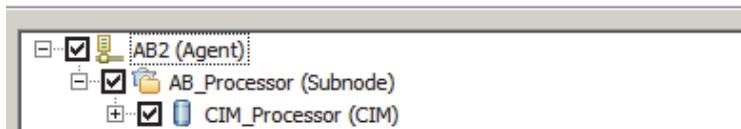
23. Click **Next**.

The Subnode "AB_Processor" - Attribute Selection - Details view opens. You are prompted to identify data groups and items to display in the AB_Processor subnode details dashboards.

24. Select **CIM_Processor (CIM)**.

Subnode "AB_Processor" - Attribute Selection - Details

Select attributes to be displayed in subnode "AB_Processor" detail page.



25. Click **Next**.

The Subnode "AB_Processor"- Resource Type view opens.

Subnode "AB_Processor" - Resource Type

Enter a name indicating the type of software server being monitored by this subnode.

String to identify this type of software server

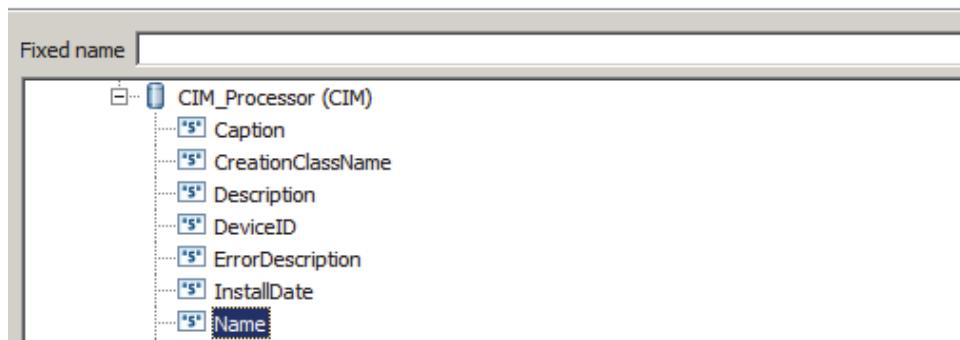
26. Keep **AB_Processor** in the resource type name field and click **Next**.

The Subnode "AB_Processor" - Attribute Selection - Software Server Name view opens.

27. Select **CIM_Processor (CIM) > Name**.

Subnode "AB_Processor" - Attribute Selection - Software Server Name

Enter the software server name or select an attribute which gives the name of the software server.



28. Click **Next**.

The Subnode "AB_Processor" - Attribute Selection - IP Address view opens.

29. Select the check box to use the agent's IP address.

Subnode "AB_Processor" - Attribute Selection - IP Address

Select the source of the IP address of the software server: either from an attribute o



30. Click **Next**.

The Subnode "AB_Processor" - Attribute Selection - Port view opens.

31. Enter **0** in the **Fixed value** field.



The Subnode "AB_WebSphere" - Attribute Selection - Status view opens. You are prompted to select a numeric attribute that gives the overall status of the AB_WebSphere subnode.

Create dashboards for this subnode.

32. Expand **AB_WebSphere (Subnode)** > **Topic Filtered (Filter)** and select **Status**.

Subnode "AB_WebSphere" - Attribute Selection - Status

Select a numeric attribute which gives the overall status of subnode "AB_WebSphere". It must be from a data source that returns only one data row.

[123] EnqueueCount
[123] DispatchCount
[123] DequeueCount
[123] ConsumerCount
[123] QueueSize
[123] MemoryPercentageUsed
[123] MemoryLimit
[123] **Status**

Provide status for subnode "AB_WebSphere" (recommended)
 Select additional attributes to display in subnode "AB_WebSphere" summary information
 Select additional attributes to display in subnode "AB_WebSphere" detail information

33. Click **Next**.

The Subnode "AB_WebSphere" - Attribute Selection - Summary view opens. You are prompted to identify attributes to display in the Summary dashboard.

34. Expand **AB_WebSphere (Subnode)** > **Topic Filtered (Filter)** and select the following attributes:

- **EnqueueCount**
- **DequeueCount**

Subnode "AB_WebSphere" - Attribute Selection - Summary

Select a few additional attributes to be displayed in subnode "AB_WebSphere" summary. sources that return only one data row.

AB2 (Agent)
AB_WebSphere (Subnode)
Topic_Filtered (Filter)
Name
[123] **EnqueueCount**
[123] DispatchCount
[123] **DequeueCount**
[123] ConsumerCount

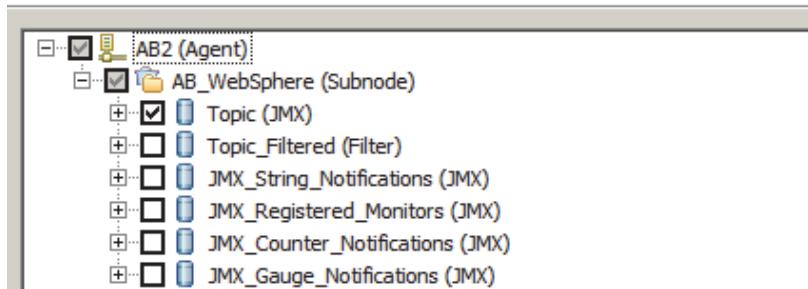
35. Click **Next**.

The Subnode "AB_WebSphere" - Attribute Selection - Details view opens. You are prompted to identify data groups and items to display in the SNMP subnode details dashboards.

36. Expand **AB_WebSphere (Subnode)** and select **Topic (JMX)**.

Subnode "AB_WebSphere" - Attribute Selection - Details

Select attributes to be displayed in subnode "AB_WebSphere" detail page.

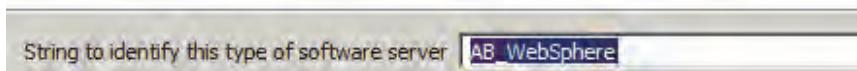


37. Click **Next**.

The Subnode "AB_WebSphere"- Resource Type view opens.

Subnode "AB_WebSphere" - Resource Type

Enter a name indicating the type of software server being monitored by this subnode.



You are prompted for the Resource Type.

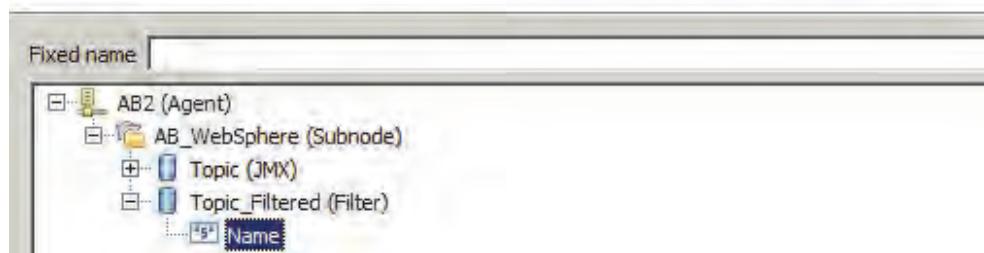
38. Keep **AB_WebSphere** in the resource type name field and click **Next**.

The Subnode "AB_WebSphere" - Attribute Selection - Software Server Name view opens. Enter a fixed name or select an attribute.

39. Select **Topic_Filtered (Filter) > Name**.

Subnode "AB_WebSphere" - Attribute Selection - Software Server Name

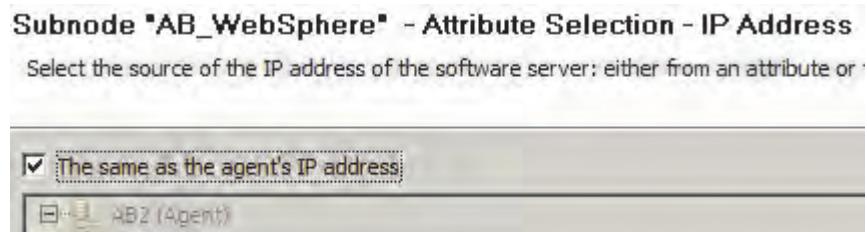
Enter the software server name or select an attribute which gives the name of the software server.



40. Click **Next**.

The Subnode "AB_WebSphere" - Attribute Selection - IP Address view opens.

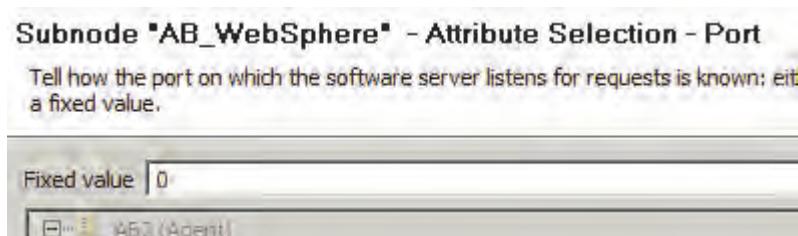
41. Select the check box to use the agent's IP address.



42. Click **Next**.

The Subnode "AB_WebSphere" - Attribute Selection - Port view opens.

43. Enter **0** in the **Fixed value** field.

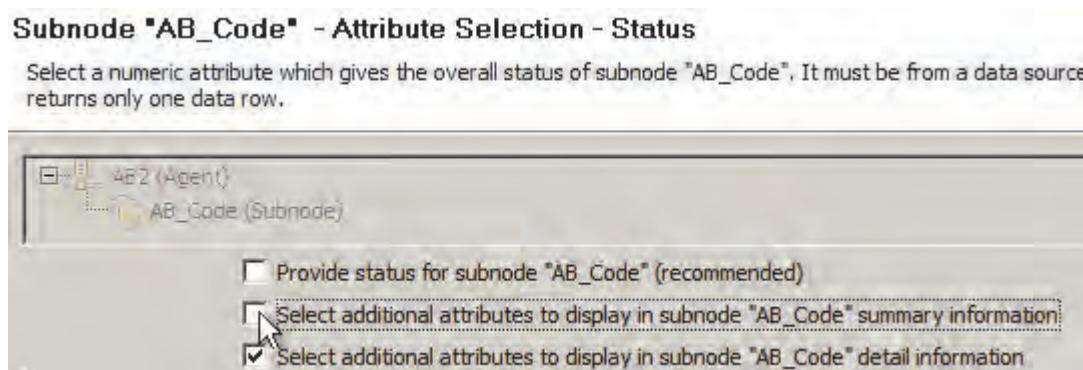


44. Click **Next**.

The Subnode "AB_Code" - Attribute Selection - Summary view opens. You are prompted to select more attributes for the AB_Code Summary dashboard.

Do not configure dashboards for this subnode.

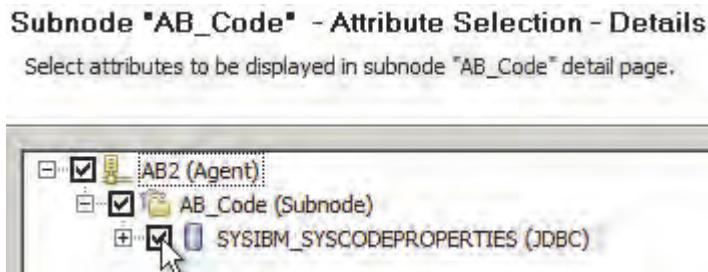
45. Clear the first two check boxes to disable the Summary dashboard for this subnode.



46. Click **Next**.

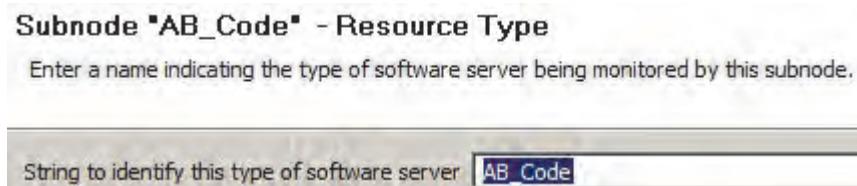
The Subnode "AB_Code" - Attribute Selection - Details view opens. You are prompted to identify data groups and items to display in the AB_Processor subnode details dashboards.

47. Select **SYSIBM_SYSCODEPROPERTIES (JDBC)**.



48. Click **Next**.

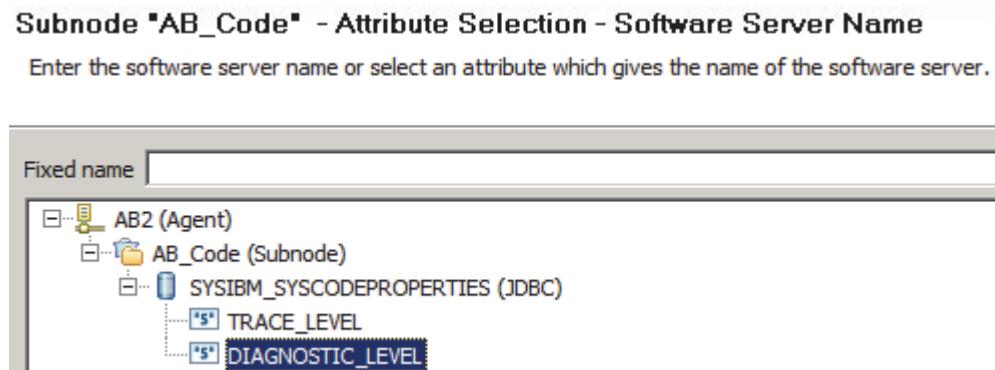
The Subnode "AB_Code"- Resource Type view opens.



49. Keep **AB_Code** in the resource type name field and click **Next**.

The Subnode "AB_Code" - Attribute Selection - Software Server Name view opens.

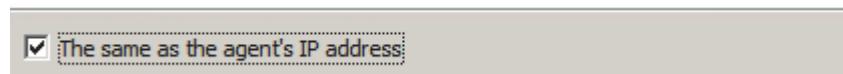
50. Select **SYSIBM_SYSCODEPROPERTIES (JDBC) > Diagnostic Level**.



51. Click **Next**.

The Subnode "AB_Code" - Attribute Selection - IP Address view opens.

52. Select the check box to use the agent's IP address.



53. Click **Next**.

The Subnode "AB_Code" - Attribute Selection - Port view opens.

54. Enter **0** in the **Fixed value** field.



55. Click **Finish** to complete the wizard.

56. Save your agent project.

You successfully defined the summary, dashboard, detailed dashboard, and the monitored resource data for OSLC.

Reinstall and configure the agent on WIN1

In this section, do the following actions:

- Reinstall the AB2 agent on WIN1.
- Create an AB_WebSphere subnode to enable JMX monitoring on WIN2
- Create an AB_Code subnode to enable JDBC monitoring on WIN1
- Do not create an AB_Processor subnode to monitor processors

57. Create the agent installers and install the agent in WIN1 in any manner you would like.



Important: When generating the agent installers, ignore the warning that the subnodes have no status attribute.



Hint: Stop the current AB2 agent before installing the new one.

58. Configure your **Monitoring Agent for AB2** in the IPM utility.

- Create an AB_WebSphere subnode to monitor WebSphere on WIN2
- Create an AB_Code subnode to monitor code data in the Sample database with the JDBC data source
- Do not create an AB_Processor subnode to monitor WIN2.
 - a. Locate the AB2 agent in the IPM utility.
 - b. Right-click the agent and select **Reconfigure**.

You are prompted for the remote host to monitor with the script data source.

The dialog box has a title bar 'Remote host connection properties'. Below it are two input fields: 'Network address' containing 'lin4' and 'SSH port number' containing '22'.

- c. Keep the current values and click **Next**.

You are prompted for the user ID and password of the remote host to monitor with the script data source.

The dialog box has a title bar 'The credentials used to make the connection'. It contains three input fields: 'Username' with 'root', 'Password' with masked text, and 'Confirm Password' with masked text.

- d. Keep the current values and click **Next**.

You are prompted for the socket to use to monitor for data from the socket data source.

The dialog box has a title bar 'Socket Data Source'. It contains one input field 'Port Number' with the value '2002'.

- e. Keep the current values and click **Next**.

You are prompted for Java configuration parameters. The default location of Java (Java 50) is not valid on this host.

- f. Browse to and change the Java home path to the following directory:

C:\Program Files (x86)\ibm\Java70

The dialog box has a title bar 'Java parameters'. It contains three input fields: 'Java home' with 'C:\Program Files (x86)\ibm\Java70', 'Java trace level' with 'Error', and 'JVM arguments' with an empty field.

- g. Click **Next**.

- h. Enter **C:\Program Files\IBM\SQLLIB\java** into the JDBC JAR directories field.

JDBC Data Source

JDBC Class Path Information

JDBC base paths

JDBC class path

* JDBC JAR directories

- i. Click **Next**.

Sources

Select the type of configuration necessary to connect to the server.

* JDBC input type

- j. Keep the JDBC Input Type at **Specify basic JDBC properties** and click **Next**.

WAS CE Class Path Information

WAS CE base paths

WAS CE class path

WAS CE JAR directories

Here you configure the location of extra software for connecting to the WebSphere Community Edition application server the agent monitors. None are needed.

- k. Do not set any **WAS CE Class Path Information** and click **Next**.

You are prompted to manage AB_WebSphere subnodes.

WebSphere Data

New...

WAS CE password

Confirm WAS CE password

* WAS CE service URL

WAS CE user ID

Here you create a subnode to monitor your JMX (WebSphere) data source. Subnodes can contain optional monitoring that you can enable or not, depending on your needs.

- I. Click **New** to create an AB_WebSphere subnode.

* AB_WebSphere	
* WAS CE service URL	service:jmx:rmi:///jndi/rmi://localhost:099/JMXCo...
WAS CE user ID	system
WAS CE password	masked
Confirm WAS CE password	masked

A subnode definition is created carrying down global properties set at the top.

- m. In **WAS CE service URL**, replace **localhost** with **WIN2**.

- n. Enter the following values:

- ◆ AB_WebSphere: **WIN2**
- ◆ WAS CE Password: **manager**
- ◆ Confirm: WAS CE Password: **manager**

* AB_WebSphere	WIN2
* WAS CE service URL	rmi:///jndi/rmi://win2:099/JMXCo...
WAS CE user ID	system
WAS CE password	masked
Confirm WAS CE password	masked

- o. Click **Next**.

You are prompted to manage AB_Processor subnodes.

AB_Processor	New...
CIM Local or Remote	Local
CIM port number	
CIM user ID	
CIM over SSL	Yes
CIM password	
Confirm CIM password	
CIM SSL port number	
CIM host name	

This agent does not monitor processors through the CIM data source, so you do not need to configure any AB_Processor subnodes.

- p. Click **Next**.

You are prompted to manage AB_Code subnodes.

AB_Code

New...

JDBC port number	<input type="text"/>
JDBC vendor specific connection URL	<input type="text"/>
* JDBC password	<input type="password"/>
* Confirm JDBC password	<input type="password"/>
* JDBC user name	<input type="text"/>
JDBC database name	<input type="text"/>
* JDBC database type	<input type="text"/> IBM DB2 Database
JDBC server name	<input type="text"/>

This agent monitors code data that is found in the Sample database on WIN1.

- q. Enter the following global properties:

- ◆ JDBC port number: **50000**
- ◆ JDBC password: **object00**
- ◆ Confirm JDBC password: **object00**
- ◆ JDBC user name: **db2admin**

AB_Code

New...

JDBC port number	<input type="text"/> 50000
JDBC vendor specific connection URL	<input type="text"/>
* JDBC password	<input type="password"/> *****
* Confirm JDBC password	<input type="password"/> *****
* JDBC user name	<input type="text"/> db2admin
JDBC database name	<input type="text"/>
* JDBC database type	<input type="text"/> IBM DB2 Database
JDBC server name	<input type="text"/>

- r. Click **New** to create an AB_Code subnode.

The screenshot shows a configuration window for creating a new subnode named 'AB_Code'. The fields are as follows:

Delete	
* AB_Code	
JDBC port number ?	50,000
JDBC vendor specific connection URL ?	
* JDBC password ?	*****
* Confirm JDBC password	*****
* JDBC user name ?	db2admin
JDBC database name ?	
* JDBC database type ?	IBM DB2 Database
JDBC server name ?	

- s. Enter the following JDBC connection properties.

- ◆ AB_Code: **WIN1**
- ◆ JDBC database name: **Sample**
- ◆ JDBC server name: **WIN1**

The screenshot shows the same configuration window with the following fields highlighted by red boxes:

Delete	
* AB_Code	WIN1
JDBC port number ?	50000
JDBC vendor specific connection URL ?	
* JDBC password ?	*****
* Confirm JDBC password	*****
* JDBC user name ?	db2admin
JDBC database name ?	Sample
* JDBC database type ?	IBM DB2 Database
JDBC server name ?	WIN1

- t. Click **OK** to close the configuration window and save your changes.

The IPM utility main window opens.

59. Start your **Monitoring Agent for AB2**.

Reinstall and configure the AB2 agent on LIN4

In this section, do the following actions:

- Reinstall the AB2 agent on LIN4
- Create an AB_Processor subnode to monitor processors on WIN2 with the CIM data source
- Do not create an AB_WebSphere subnode to enable JMX monitoring
- Do not create an AB_Code subnode to enable JDBC monitoring

60. Install the agent on LIN4.

- a. To get the installer to LIN4, copy it from **C:\Users\Administrator** into the **C:\Share** directory on the ITM server. On LIN4, locate the installer file in the /mnt/share directory.
Run **mount /mnt/share** if the file is not visible.
- b. Extract the files into their own directory with **tar -xzvf**.
- c. Install the agent into the **/opt/ibm/apm/agent** directory



Hint: Stop the current AB2 agent before installing the new one.

61. Configure the AB2 agent.

- a. Open a terminal window and run the following command:

```
/opt/ibm/apm/agent/bin/ab2-agent.sh config
```

You are first prompted if you want to configure the agent.

- b. Press Enter to select the default, **Yes**.

```
Lin4:/mnt/share/K01 # /opt/ibm/apm/agent/bin/ab2-agent.sh config
Configuring Monitoring Agent for AB2
```

```
Edit 'Monitoring Agent for AB2' settings? [ 1=Yes, 2=No ] (default is: 1):
```

62. Accept the previously configured script data source properties.

- a. Press Enter to monitor remotely.

You are prompted for the host name or IP address of the system where the monitoring script (script1.sh) is run.

- b. Press Enter to accept **lin4** as the target host for the script data source.

```
SSH Remote Connection :  
Remote host connection properties  
  
Host name or IP address of remote system  
Network address (default is: lin4): █
```

- c. Press Enter to accept port **22**.

```
The port number used for SSH communication.  
SSH port number (default is: 22): █
```

- d. Press Enter to set a password.

```
The type of authentication to use to make the connection  
Authentication Type [ 1=Password, 2=Public Key ] (default is: ): 1
```

- e. Press Enter to select **No** to disconnecting after each collection interval.

```
Disconnect from the remote system after each data collection interval. By default, the SSH communication socket will establish an authenticated session at the first connection and will leave the connection active until the agent is stopped.  
Disconnect from the remote system after each collection interval [ 1=Yes, 2=No ] (default is: 2):
```

- f. Press Enter to select **No** to deleting the script after each collection interval.

```
Delete the script(s) from the remote system after each data collection interval.  
By default, the script(s) will be uploaded to the remote system at the first connection and will only be refreshed if the local copy changes. The script(s) will be removed from the remote system immediately before the agent is stopped.  
Remove script(s) from the remote system after each collection interval [ 1=Yes, 2=No ] (default is: 2):
```

- g. Press Enter to accept **root** at the user name.

```
Password :  
The credentials used to make the connection  
  
Username for the remote system  
Username (default is: root): █
```

- h. Press Enter twice to accept the current password.

```
 Password used for the remote system.  
 Enter Password (default is: *****):  
  
 Re-type : Password (default is: *****):
```

63. Press Enter to accept the previously configure socket of 0.

```
Socket :  
Socket Data Source
```

The port that the agent will use to listen on for data from socket clients. A value of 0 indicates an ephemeral port will be used.
Port Number (default is: 0):

64. Set global Java properties for a possible JDBC data source. This agent does monitor the JDBC data source.

- a. Press Enter to accept the default Java home directory.

```
Java :  
Java parameters  
  
The path to where java is installed  
Java home (default is: /opt/ibm/java): █
```

- b. Press Enter to accept the default Java trace level.

- c. Press Enter to not enter JVM arguments.

This parameter allows you to specify the trace level used by the Java providers
Java trace level [1=Off, 2=Error, 3=Warning, 4=Information, 5=Minimum Debug, 6=Medium, 7=Maximum Debug, 8=All] (default is: 2):

This parameter allows you to specify an optional list of arguments to the java virtual machine
JVM arguments (default is:):

The next several fields are used to set alternative paths to files used by the JDBC data source.

- d. Press Enter to select the default JDBC base path.
e. Press Enter to select the default JDBC class path.
f. Enter a forward slash (/) and press Enter to set the JDBC JAR directories.

- g. Press Enter to select the default JDBC input type.

```
JDBC Server :  
JDBC Data Source  
-----  
JDBC Class Path Information  
Directories used to locate any Java class path entry or JAR file directory which  
is not fully-qualified  
JDBC base paths (default is: ):  
  
JAR files which Java will look in to find a needed class or resource  
JDBC class path (default is: ):  
  
Directories which contain JAR files which Java will look in to find a needed class  
or resource. All JAR files in each of these directories will be looked in.  
JDBC JAR directories (default is: ): /  
  
JDBC Input Type :  
Select the type of configuration necessary to connect to the server.  
  
The type of input properties you want to use to connect to the database  
JDBC input type [ 1=Specify basic JDBC properties, 2=Specify the JDBC URL ] (default is: 1):  
  
The next several fields are used to locate extra software that is needed to make the JMX  
connection to WebSphere Community Edition. None is needed in this instance.  
  
65. Set global WebSphere properties for a possible JMX data source. This agent does not monitor  
the JMX data source.  
a. Press Enter to select the default WAS CE base path.  
b. Press Enter to select the default WAS CE class path.
```

- c. Press Enter to select the default WAS CE JAR directories.

WebSphere Application Server Community Edition :

WAS CE Class Path Information

Directories used to locate any Java class path entry or JAR file directory which is not fully-qualified

WAS CE base paths (default is:):

JAR files which Java will look in to find a needed class or resource

WAS CE class path (default is:):

Directories which contain JAR files which Java will look in to find a needed class or resource. All JAR files in each of these directories will be looked in.

WAS CE JAR directories (default is:):

You are prompted to manage AB_WebSphere subnodes. You can create, edit, and delete subnodes.

This agent does not monitor the WebSphere so do not create any AB_WebSphere subnodes.

66. Configure subnodes.

- a. Type **5** and press Enter to exit managing AB_WebSphere subnodes.

```
AB_WebSphere :  
  
No 'AB_WebSphere' settings available.  
Edit 'AB_WebSphere' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit]  
(default is: 5): 5
```

You are prompted to manage AB_Processor subnodes.

This agent monitors processor data through the CIM data source so create an AB_Processor subnode.

- b. Type **1** and press Enter to create an AB_Processor subnode.

```
AB_Processor :  
  
No 'AB_Processor' settings available.  
Edit 'AB_Processor' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit] (default is  
: 5): 1
```

c. Enter the following values:

- ◆ AB_Processor: **WIN2**
- ◆ CIM Local or Remote: **2 (Remote)**
- ◆ CIM port number: **5988**
- ◆ CIM user ID: **Administrator**
- ◆ CIM of SSL: **2 (No)**
- ◆ CIM password: **object00**
- ◆ CIM SSL port: **5989**
- ◆ CIM host name: **WIN2**

```
AB_Processor (default is: ): WIN2
Local or remote authentication to the CIM server
CIM Local or Remote [ 1=Local, 2=Remote ] (default is: ): 2

The port number used for communication that is not secure
CIM port number (default is: ): 5988

The user ID used to access the CIM server
CIM user ID (default is: ): Administrator

Use SSL for communication with the CIM server
CIM over SSL [ 1=Yes, 2=No ] (default is: ): 2

The password to access the CIM server
Enter CIM password (default is: ):

Re-type : CIM password (default is: )

The port number used for secure communication
CIM SSL port number (default is: ): 5989

The host name to be accessed for CIM data
CIM host name (default is: ): WIN2
```

You are prompted again to manage AB_Processor subnodes.

d. Type **5** and press Enter to exit managing AB_Processor subnodes.

```
AB_Processor settings: AB_Processor=WIN2
Edit 'AB_Processor' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit] (default is
: 5): 5
```

You are prompted to manage AB_Code subnodes. This agent does not monitor code data in a database. Do not create any AB_Processor subnodes.

- e. Type **5** and press **Enter** to exit managing AB_Code subnodes.

```
AB_Code :  
  
No 'AB_Code' settings available.  
Edit 'AB_Code' settings, [1=Add, 2>Edit, 3=Del, 4=Next, 5=Exit] (default is: 5):  
5
```

You exit the agent configuration process.

You successfully configured this agent.

67. Start your **Monitoring Agent for AB2**.

Confirm the agents in the Performance Management Console

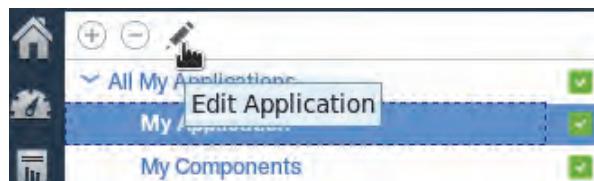
In this section, you confirm your new data sources and subnodes in the Performance Management console.

68. On APM, open the Application Performance dashboard.

69. Add the following components to My Application:

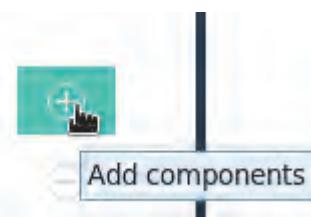
- AB_Code > 01:WIN1:JDBC
- AB_Processor > 01:WIN2:CIM
- AB_WebSphere > 01:WIN2:JMX

- a. Click **My Application** in the Applications explorer.
- b. Click the **Edit Application** icon.



The Edit Application windows opens.

- a. Click the **Add components** button next to the Application components pane.



The Select Component window opens.

- b. Locate and select **AB_Code**.



- c. Select **01:WIN1:DBC** and click **Add**.



Important: It can take 10 - 15 minutes from when the agent is installed until the subnodes appear in the Component Editor window. If the subnodes are not listed, click Back and the subnode until the agent instance is shown.

- d. Click **Back** to return to the Select Component window.

- e. Locate and select **AB_Processor**.

- f. Select **01:WIN2:CIM** and click **Add**.



- g. Click **Back** to return to the Select Component window.

- h. Locate and select **AB_WebSphere**.

- i. Select **01:WIN2:JMX** and click **Add**.



- j. Click **Back** and **Close** to close the Select Component window.

You are returned to the Edit Application window with the subnodes in the Application components window.



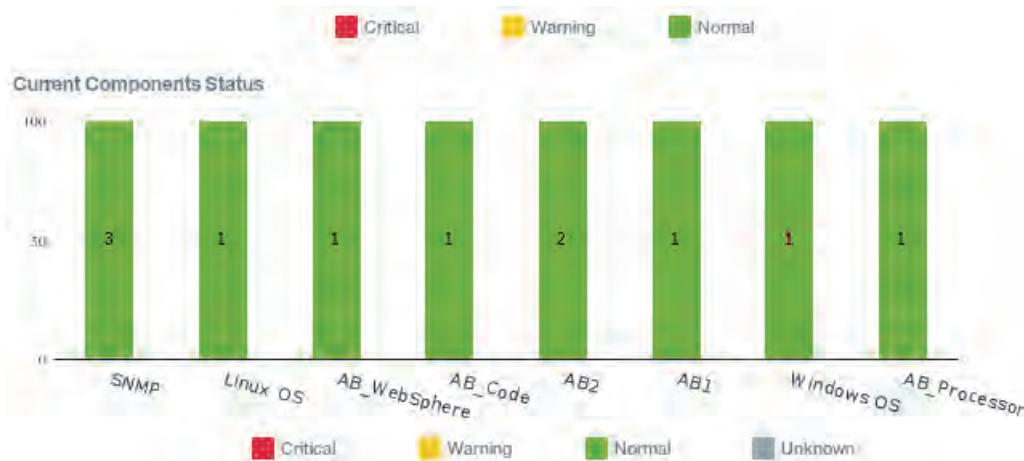
70. Click **Save** and **OK** to save your application definition and exit the Edit Application utility.

You are returned to the Application Dashboard.

Confirm the AB_WebSphere subnode data source

71. Click **My Application** in the Applications explorer.

72. Refresh this view until the **AB_WebSphere** component is shown in the Current Component Status view.



Hint: It can take up to 10 minutes for the component to be displayed.

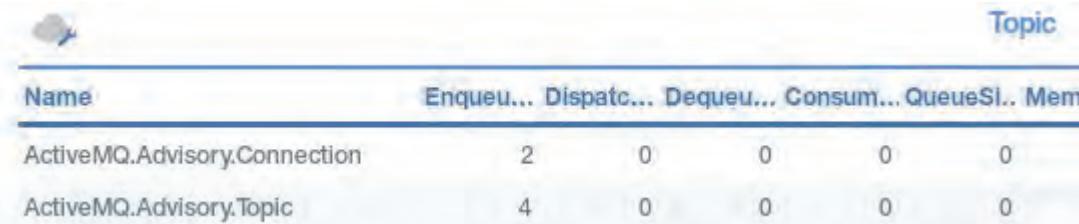
73. Click the **AB_WebSphere** component bar in the Current Components Status widget to open the Summary dashboard.

The screenshot shows a summary dashboard for the "AB_WebSphere" component. At the top, there's a breadcrumb navigation: All My Applications > My Application > Components > AB_WebSphere. Below the breadcrumb, there are two tabs: "Status Overview" (which is selected) and "Events". The main area is titled "01 - AB_WebSphere" and includes a small icon of a computer monitor with a blue square on it, followed by the text "WIN2". Below this, there are three data points:

Status	Value
EnqueueCount	4
DequeueCount	0

The Summary dashboard shows selected values from your JMX data source. Seeing a valid dashboard here confirms your dashboard configuration. Seeing valid data here confirms the filtered data source in this agent.

74. Click the **01 - AB_WebSphere** Summary dashboard widget to access the detailed dashboards.

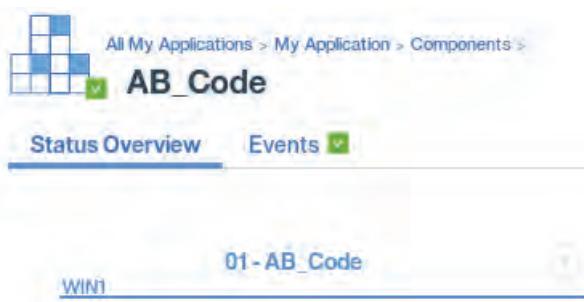


Name	Enqueu...	Dispatch...	Dequeue...	Consum...	QueueSi...	Mem	Topic
ActiveMQ.Advisory.Connection	2	0	0	0	0	0	
ActiveMQ.Advisory.Topic	4	0	0	0	0	0	

The detailed dashboard shows all rows from your JMX data source. Seeing data here confirms your dashboard configuration and data source in this agent.

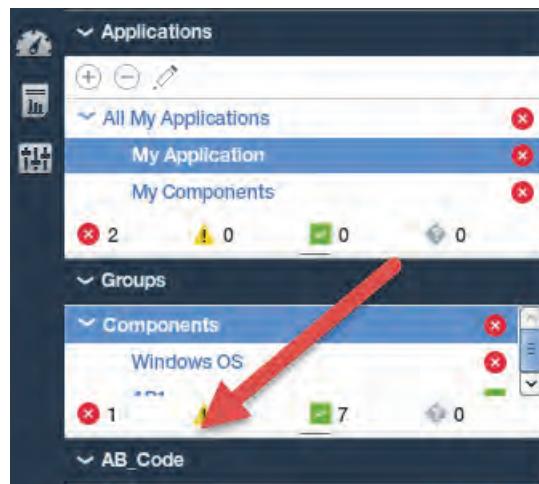
Confirm the AB_Code subnode data source

75. Click **My Application** in the Applications explorer.
76. Refresh this view until the **AB_Code** component is shown in the Current Component Status view.
77. Click the **AB_Code** component bar in the Current Components Status widget to open the Summary dashboard.



The Summary dashboard shows no values because you did not create a Summary dashboard for this subnode. Without a Summary dashboard, you need to take different steps than before to see the monitored data.

78. Locate the **AB_Code** instances pane on the left below the **Applications** and **Groups**.



79. Click **01:WIN1:DBC**.



The 01:WIN1:DBC detailed dashboard opens.

A screenshot of the 01:WIN1:DBC detailed dashboard. The title bar shows "01:WIN1:DBC". Below it, there are tabs for Status Overview, Events, and Attribute Details. The Status Overview tab is selected. At the top, there's a table titled "SYSIBM_SYS CODE PROPERTIES". The table has columns for Object, Lib ID, Trace, Diagnose, Debug, SQL, CO, Node, and Timestamp. There are three rows of data: one for object 66124 and two for object 66135, all timestamped at 3/3/17, 12:11 AM.

The detailed dashboard shows all rows from your JDBC data source. Seeing data here confirms your dashboard configuration and data source in this agent.

Confirm the AB_Processor subnode data source

80. Click **My Application** in the Applications explorer.

81. Refresh this view until the **AB_Processor** component is shown in the Current Component Status view.

82. Click the **AB_Processor** component bar in the Current Components Status widget to open the Summary dashboard.

The screenshot shows the 'Current Components Status' section of the IBM Performance Management interface. A blue bar labeled 'AB_Processor' is highlighted, indicating it has been selected. Above the bar, the navigation path is shown as 'All My Applications > My Application > Components > AB_Processor'. Below the bar, there are two tabs: 'Status Overview' (which is underlined in blue) and 'Events'.

The Summary dashboard shows no values because you did not create a Summary dashboard for this subnode.

83. Locate the **AB_Processor** instances pane on the left below the **Applications** and **Groups**.

84. Click **01:WIN1:CIM**.



The 01:WIN1:CIM detailed dashboard appears.

The screenshot shows the '01:WIN1:CIM' detailed dashboard. At the top, the navigation path is 'All My Applications > My Application > Components > AB_Processor > 01:WIN1:CIM'. Below the path, there are three tabs: 'Status Overview' (underlined in blue), 'Events', and 'Attribute Details'. The 'Attribute Details' tab is currently active. A table titled 'CIM_Processor' displays two rows of processor information:

Address...	Available...	Caption	ConfigM...	ConfigM...	CreationClassName	Current...
64	3	Intel64 Family 6 Model 26 Stepp...	0	FALSE	Win32_Processor	2000
64	3	Intel64 Family 6 Model 26 Stepp...	0	FALSE	Win32_Processor	2000

The detailed dashboard shows all rows from your JDBC data source. Seeing data here confirms your dashboard configuration and data source in this agent.

You successfully confirmed your agent in the Performance Management console. That the agent was available to be added to an application, the Summary dashboard shows the core metrics that you added. A subset of attributes is visible in the **Details dashboard**.



IBM Training



© Copyright IBM Corporation 2017. All Rights Reserved.