WebSphere® WebSphere Process Server for z/OS

IBM

Version 6.1.2

Tutorials

WebSphere® WebSphere Process Server for z/OS

Version 6.1.2

Tutorials

Note

Before using this information, be sure to read the general information in the Notices section at the end of this document.

27 June 2008

This edition applies to version 6, release 1, modification 2 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, send an e-mail message to doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Tutorials

Tutorials for common tasks are available in the IBM® Education Assistant and in the IBM WebSphere® Process Server documentation.

1. The information presented here is available in Adobe® PDF format at the following link: WebSphere Process Server documentation (in PDF format).
2. Business Process Management information roadmaps on IBM developerWorks® organize information about WebSphere Process Server and the other products in the portfolio.

IBM Education Assistant tutorials

The IBM Education Assistant site provides tutorials that you can use at your convenience. To view this educational content, see the IBM Education Assistant at http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0.

WebSphere Process Server tutorials

WebSphere Process Server documentation contains tutorial topics to assist you with some administrative, security, and monitoring tasks.

Tutorial: Relationship manager administration

The relationship manager can be used to add, modify, and remove instances of relationships, which correlate identifiers from different environments for the same item of data. This tutorial demonstrates the basic functions of the relationship manager.

This tutorial demonstrates the basic functions of the WebSphere Process Server relationship manager. Relationships are used to correlate identifiers from different environments for the same item of data. For example, in one environment, states are identified by two-letter abbreviations (AZ, TX). In another environment, different abbreviations are used (Ariz., Tex.). A relationship would be created to correlate ″AZ″ in the first environment to ″Ariz″ in the second environment.

The sample relationship referenced here correlates customer IDs. Many business applications maintain databases of customers, and most of these applications assign their own ID to each customer. In an enterprise environment, the same customer likely has a different ID in each business application. In this tutorial, a relationship is defined to correlate customer IDs. The relationship name is ″SampleCustID″. Two roles are defined for this relationship. One role is for the Customer Information System (CIS), and the other role is for the General Ledger (GL) application. This relationship was created by the relationship services sample along with the roles and a small amount of sample data.

The relationship manager is designed to add, modify, and remove role instances of a relationship instance as well as add, modify, and remove relationship instances. WebSphere Integration Developer should be used to create and deploy new relationship definitions. The definitions are stored as XML files that are deployed as part of a J2EE application to a particular server.

Objectives of this tutorial

After completing this tutorial, you will be able to change the values of relationship instances.

Time required to complete this tutorial

This tutorial requires approximately 10 minutes to complete.

Prerequisites

This tutorial uses a relationship that is created by the relationship services technical sample. Before following the steps of this tutorial, go to the samples gallery and perform the steps described in the relationship services sample to create the required relationship and roles.

Related information

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.612.doc/doc/tovw_access_samples.html

Example: Changing the values of a relationship instance

For a relationship instance, the values of key attributes can be changed on the Relationship Instances page of the administrative console. This example shows the use of that page to change a value for a relationship instance.

About this task

One of your customers has a customer ID of A004 in your CIS application. This same customer has a customer ID of 801 in your GL application. However, due to a data entry error, the relationship instance that correlates the customer IDs of this customer currently has a value of 901 instead of 801 for the GL customer ID. This tutorial takes you through the steps to correct this entry in the relationship.

Procedure

1. Open the administrative console.
2. If security is enabled, log in as a user with administrator privileges.
3. In the navigation pane, click **Integration Applications** → **Relationship Manager**.
4. Open the relationships page for the server you want to manage. Click **Relationships** next to that relationship services MBean.

   A relationship named SampleCustID should be visible.
5. Select the radio button next to SampleCustID, then click **Query**.
6. Locate the relationship instance for the customer
   a. Click the query **By role** tab
   b. In the **Role name** field, select MyGLCustomer_0 from the drop-down list.
   c. In the **Value** field under **Key attributes**, enter`901`
   d. Click **OK**

   This locates the relationship instance for the requested customer and opens the Relationship Instances page.
7. Click the relationship instance ID.

   This displays the relationship instance data for customer ID 901 in the GL application, including all the associated role instances.

8. In the MyGLCustomer_0 role table, select the role instance ID with the key attribute value 901, then click **Delete** below the role table.

   Note: It should not have any property values associated with it. If any other data appears, you need to look at the role instance and record any data you want to keep.
9. Click **Create** to open the New Role Instance page for creating a new role instance for this relationship instance.
10. Enter `801` in the **Value** field under **Key attributes**, then click **OK**.

    The new role instance is saved, and you should see a new role instance in the table.

Results

You now have the correct customer ID value in the relationship instance for the GL application.

Creating end to end security

There are many potential end to end security scenarios. Each of these might involve differing security steps. Several typical scenarios, with the necessary security options, are presented.

Before you begin

These scenarios all assume that global security is enforced.

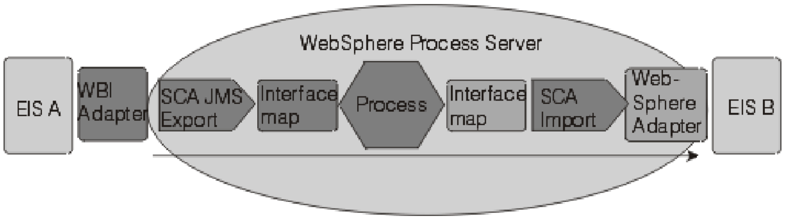
About this task

Procedure

1. Determine which of the examples provided in this section, most closely match your security needs. In some instances, your needs might involve a combination of information from more than one of the scenarios.
2. Read the security information for the relevant scenarios and apply it to your security needs.

Classic integration scenario - inbound and outbound adapters

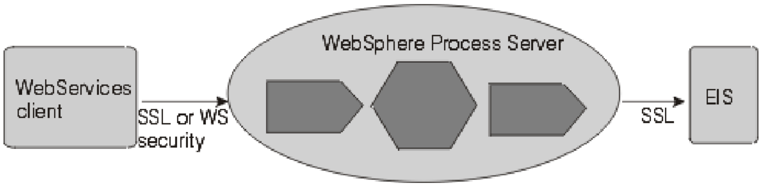
An inbound request comes in from a WebSphere Business Integration Adapter. The service component architecture (SCA) invokes an interface map based on the SCA export. The request flows through a process component, a second interface map and is then passed on to a second EIS (B), via a WebSphere Adapter. These are SCA invocations, with one component invoking a method on the next component.



There is no authentication mechanism for the inbound adapter. You can establish the security context by defining the SecurityIdentity qualifier on the first component - in this instance, the first interface map component. From that point, SCA will propagate the security context from each component to the next. Access control for each component is defined by use of the SecurityPermission qualifier.

Inbound Web service request

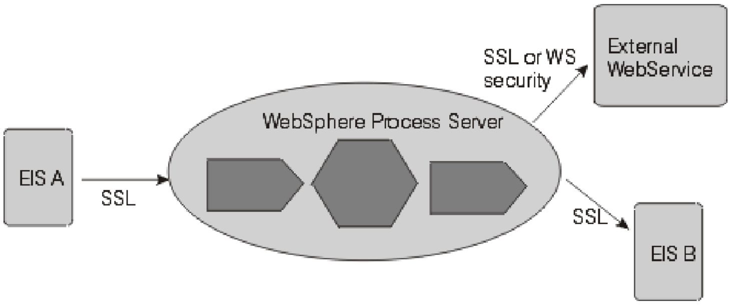
In this scenario a Web service client invokes a WebSphere Process Server component. The request passes through several components in the WebSphere Process Server environment before being passed to an EIS by an adapter.



You can authenticate the Web service client as a SSL client, using HTTP Basic authentication or using WS-Security authentication. When the client is authenticated, access control is applied based on the SecurityPermission qualifier. Between the client and the WebSphere Process Server instance, you can secure the data integrity and privacy using SSL or WS-Security. SSL secures the entire pipe, whereas with WS-Security you can encrypt or digitally sign parts of the SOAP message. For Web services, WS-Security is the preferred standard.

Outbound Web service request

In this scenario the inbound request can be from an adapter, a Web service client, or a HTTP client. A component in WebSphere Process Server, for example a BPEL component, invokes an external Web service.



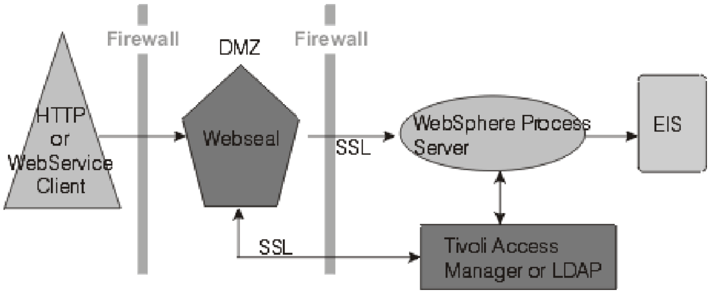
As for the inbound Web service request, you can authenticate with the external Web service as a SSL client, using HTTP Basic authentication or using WS-Security authentication. Use LTPACallBackHandler as the callback mechanism to extract the usernameToken from the current RunAs subject. Between WebSphere Process Server and the target Web service, you can ensure data privacy and integrity using WS-Security.

Web application - HTTP inbound request to WebSphere Process Server

WebSphere Process Server supports three types of authentication for HTTP:

- HTTP basic authentication
- HTTP forms based authentication
- HTTPS SSL-based client authentication.

In addition, to protect your intranet from intruders, you can place the Web server in the demilitarized zone (DMZ), and the WebSphere Process Server inside the inner firewall. In this example, WebSEAL is used as the reverse proxy, which performs the authentication. It has a trust association with WebSphere Process Server behind the firewall and can forward authenticated requests.



Tutorial: Writing a Jacl script that lists security roles

A Jacl script for the wsadmin tool can be used to perform a variety of administrative actions. This tutorial addresses how to write and execute a simple Jacl script that can access and manage a JMX MBean.

The Jacl script used in this tutorial is concerned with calling roles when global security is enabled. Using this script, you can print the role name for each role in a relationship.

Objective of this tutorial

After completing this tutorial, you will be able to:

- Write a Jacl script that calls a JMX MBean requesting a list of all relationships.

For more information about writing scripts, refer to ″Using scripting (wsadmin)″ in the .

Time required to complete this tutorial

This tutorial requires approximately 15-30 minutes to complete.

Prerequisites

This tutorial uses a script that is included with the JMX Security sample. This sample demonstrates the MBean function of printing out a list of role relationships.

Note: To use this script, you must select the option to install code samples during the installation of WebSphere Process Server.

The sample Jacl script is located in:

The name of the script is: `RelServicesAdmin.jacl`.

To run the script, enter:script, enter:

```
wsadmin -f install_root/samples/JMXSample/scripts/RelServicesAdmin.jacl
 -server servername -node nodename
```

This script will call up to 10 relationships in your environment and up to 10 roles for each relationship will be printed on the console.

Example: Writing a Jacl script

A Jacl script can be used to affect the methods and attributes on any MBean in the system. This example shows the use of a Jacl script to manage the JMX Security MBean.

About this task

The basic concepts in this script can be used to communicate with any MBean in the system. All that is required is the name and type of the MBean and the methods and attributes available on the MBean. The getAttribute and setAttribute commands are used for attributes. The invoke command is used for methods. Follow these steps to create a `.jacl` script that manages the JMX Security MBean.

Note: The code in each step is prefaced with a statement explaining what the code does.

Procedure

1. Determine the **nodename**

   The first part of the script shown below determines the nodename. If the `nodeName` is not specified correctly, the correct syntax is printed and the script exits.

   ```
   # read and validate arguments

       if {{$argc == 1 } && { [lindex $argv $i] == "-nodeName" } {
               set nodeName [lindex $argv $i]
   ```
2. Identify the **MBean**

   An MBean is identified by a type and a name.

   Note: The name and type are hard coded in this case since you know the specific MBean you want to use.
   The second part of the script identifies the MBean.

   ```
   # these two variables, mbeanName and mbeanType are used
     to uniquely identify the mbean.
          # for this sample, the mbean that access relationship
            services will be used.

          set mbeanName"RelService"
          set mbeanType"WBIRelServices"
   ```
3. Locate and set the **reference** to the MBean.

   You use the code shown here to set the reference for the MBean.

   ```
   # locate the mbean and set a reference to it in "relSvcsMBean" variable

          set relSvcsMBean [$AdminControl queryNames
              name=$mbeanName,node=$nodeName,type=$mbeanType,*]
   ```
4. Call the **relationship** using the getAttribute command.

   The documentation of this specific MBean defines an attribute named `allRelationshipNames`. Ask the MBean for that attribute using the getAttribute command. The attribute value will be a list that you step through in the next step that invokes the command.

   ```
   # request the list of relationships from the mbean

          set relationships
           [$AdminControl getAttribute $relSvcsMBean allRelationshipNames]
   ```
5. Invoke the **command** for each relationship name, you print the name, and then go back to the MBean for additional information.

In this example the MBean defines a method named getAllRoleNames with a single parameter for the specific relationship name. You use the invoke command to call this method, passing the current relationship name. For each role in the relationship, a role name is printed.

```
# loop through the list of role names and print name

         foreach roleName $roles {
           puts "    Role: $roleName"
         }
      }
    } else {
         # arguments were not correct, print correct syntax
            puts "Usage: wsadmin -f RelServicesAdmin.jacl -nodeName nodeName"
    }
```

Results

You have now written a script to call relationships.

Tutorial: Using the Common Event Infrastructure server for event monitoring

For service component event points that you monitor, events can be published to the Common Event Infrastructure (CEI) server and stored in the CEI server database. This tutorial guides you through an example of setting up monitoring with the CEI server, and how to view events stored in the database.

This tutorial demonstrates how to set up service component event monitors that are published to the CEI server, and how to use the Common Base Event browser to view those stored events. The example you will use in this scenario does not involve static monitoring, whereby an application deployed with a .mon file will continually monitor specific service components event points. For information about how to perform static monitoring, consult the IBM WebSphere Integration Developer Information Center.

The scenario you will follow for this example, instead, will show you how to select for monitoring event points on service components in applications already deployed and running on a server. You will see how the monitoring function fires an event whenever the processing of an application reaches one of those event points. Each of those fired events are published to the CEI server, which will store the event information about its database. You will use the Common Base Event browser to view the events.

Objectives of this tutorial

After completing this tutorial you will be able to:

- Select service component event points to monitor, with events published to the CEI server.
- View the stored events with the Common Base Event browser.

Time required to complete this tutorial

This tutorial requires approximately 15-20 minutes to complete.

Prerequisites

In order to perform this tutorial, you must have:

- Configured and started a server.
- Configured the CEI and its database.
- Enabled the diagnostic trace service on the server.
- Installed and started the Samples Gallery application on the server.
- Installed and started the business rules sample application on the server. Follow the instructions on the Samples Gallery page to set up and run the business rules sample application.

After all of these prerequisites have been completed, run the business rules sample application from the Samples Gallery at least once before proceeding with the tutorial.

Example: Monitoring with the Common Event Infrastructure server

For monitoring with the CEI server, you can use the administrative console to manage the details for event types and to display recorded events in the Common Base Event browser. This example shows the use of the console to change the level of detail recorded for some event types and to use the Common Base Event browser to view the information for individual events.

About this task

You will use the business rules sample application for this scenario; consequently, you should already have the Web page containing this application already open. Keep it open, since you will be running the sample after you specify monitoring parameters. Ensure that you have already run the sample at least once, because that will cause it to appear in the list of functions that you can select to monitor.

Procedure

1. Open the administrative console.
2. In the navigation pane, click **Servers** → **Application Servers**.
3. Click *server_name*.
4. Under Troubleshooting, click **Logging and tracing**
5. Click **Change Log Detail levels**
6. Select the **Runtime** tab.
7. Expand the tree for **WBILocationMonitor.CEI.BR** and you will see five event types under the **WBILocationMonitor.CEI.BR.brsample.*** element. Each event type includes the name **WBILocationMonitor.CEI.BR.brsample_module.DiscountRuleGroup**, appended by the function **Operation._calculateDiscount**, and the following natures:
   - ENTRY
   - EXIT
   - FAILURE
   - SelectionKeyExtracted
   - TargetFound
8. Click on each of the events and select **finest**.
9. Click **OK**.

10. Switch the business rules sample application page, and run the application once.
11. Go back to the administrative console, and select **Integration Applications** → **Common Base Event Browser** from the navigation pane.
12. If you are running your server on node within a Network Deployment environment, then you may need to modify the **Event Data Store** field to include to the names of your server and node. Enter the string in the following form: 'cell/nodes/*node_name*/servers/*server_name*/ejb/com/ibm/events/access/EventAccess'.
13. Press **Get Events**.

Results

You should now see a list in the upper pane of the Common Base Event browser of the four business rule events that were published to the CEI server when you ran the sample application. Select one of the events, and you will be shown the contents of the event in the lower pane. Compare this to the events published to the loggers. Notice that the browser has parsed the original XML string that was published to the CEI server, and that the business object code in the ENTRY and EXIT events was converted from the original hexadecimal format to readable XML. You may want to go back through this exercise and change the level of detail from **finest** to **fine** or **finer**, and compare the differences between the events.

After completing this exercise, you should understand how to select service component event points for monitoring using the CEI server. You have seen that the events fired in this type monitoring have a standard format, and that the results are published to a database. You should also be able to use the Common Base Event browser to retrieve events from the database, and view the information for individual events in a parsed table format on the browser.

What to do next

If you no longer want to monitor the business rules sample application, you can go back to through the steps outlined here and reset the level of detail for the sample events to **info**.

Tutorial: Logging service component events

For service component event points that you monitor, events can be published to the logging facilities of the underlying WebSphere Application Server. This tutorial guides you through an example of setting up monitoring with logging, and how to view events stored in a log file.

The scenario you will follow for this example will show you how to select service component event points for monitoring in applications already deployed and running on a server. You will see how the monitoring function fires an event whenever the processing of an application reaches one of those event points. Each of those fired events takes the form of a standardized Common Base Event, which is published as an XML string directly to a log file.

Objectives of this tutorial

After completing this tutorial you will be able to:

- Select service component event points to monitor, with the output published to the server loggers.

- View the stored events in the log files.

Time required to complete this tutorial

This tutorial requires approximately 15-20 minutes to complete.

Prerequisites

In order to perform this tutorial, you must have:

- Configured and started a server.
- Configured Common Event Infrastructure.
- Enabled the diagnostic trace service on the server.
- Installed and started the Samples Gallery application on the server.
- Installed and started the business rules sample application on the server. Follow the instructions on the Samples Gallery page to set up and run the business rules sample application.

After all of these prerequisites have been completed, run the business rules sample application from the Samples Gallery at least once before proceeding with the tutorial.

Example: Monitoring events in the logger

For monitoring with logging, you can use the administrative console to manage the details for event types. This example shows the use of the console to change the level of detail recorded for some event types and to use a text editor to open the trace.log file to view the information for individual events.

About this task

You will use the business rules sample application for this scenario, so you should already have the web page containing this application already open. Keep it open, since you will be running the sample after you specify monitoring parameters. Ensure that you have already run the sample at least once, so that it will appear in the list of functions that you can select to monitor.

Procedure

1. Open the administrative console.
2. In the navigation pane, click **Servers** → **Application Servers**.
3. Click *server_name*.
4. Under Troubleshooting, click **Logging and tracing**
5. Click **Change Log Detail levels**
6. Select the **Runtime** tab.
7. Expand the tree for **WBILocationMonitor.LOG.BR** and you will see seven event types under the **WBILocationMonitor.LOG.BR.brsample.*** element. The first event is called **WBILocationMonitor.LOG.BR.brsample_module.DiscountRuleGroup**, which includes a single function, named **Operation._calculateDiscount**, with the following natures:
   - ENTRY
   - EXIT
   - FAILURE
   - SelectionKeyExtracted

- TargetFound

8. Click on each of the events and select **finest**.
9. Click **OK**.
10. Switch the business rules sample application page, and run the application once.
11. Use a text editor to open the trace.log file located in the *profile_root*/logs/*server_name* folder on your system.

Results

You should see lines in the log containing the business rule events fired by the monitor when you ran the sample application. The main thing you will probably notice is that the output consists of lengthy, unparsed XML strings conforming to the Common Base Event standard. Examine the ENTRY and EXIT events, and you will see that business object — which was included because you selected the **finest** level of detail — is encoded in hexadecimal format. Compare this output with events published to the Common Event Infrastructure server, which parses the XML into a readable table and decodes any business object data into a readable format. You may want to go back through this exercise and change the level of detail from **finest** to **fine** or **finer**, and compare the differences between the events.

After completing this exercise, you should understand how to select service component event points for monitoring to the logger. You have seen that the events fired in this type monitoring have a standard format, and that the results are published as a string in raw XML format directly to a log file. Viewing the published events is simply a matter of opening the log file in a text editor, and deciphering the contents of individual events.

What to do next

If you no longer want to monitor the business rules sample application, you can go back to through the steps outlined here and reset the level of detail for the sample events to **info**.

Tutorial: Service component performance monitoring

For service component event points that you monitor, you can publish and view performance statistics. This tutorial guides you through an example of setting up performance monitoring, and how to view the resulting statistics.

This tutorial guides you through an example of how to set up service component event monitors that are published to the Performance Monitoring Infrastructure (PMI), and how to view the resulting performance statistics on the Tivoli® Performance Viewer (TPV). This exercise will demonstrate how performance monitoring of service component event points differs from monitoring using the Common Event Infrastructure (CEI) server and loggers. The major difference that you will notice is that you select an entire service component element for performance monitoring, instead of individual events with specific natures. Because WebSphere Process Server can monitor performance only on service component elements having events with ENTRY, EXIT, and FAILURE natures, you will have only those kinds of service component elements available to you to select for monitoring.

While the service component event points ENTRY, EXIT, and FAILURE are identical for all monitoring types, the performance monitoring function in the server fires ″minimized″ events that do not contain all of the information encompassed in CEI events. These events are sent to the PMI, which calculates these performance statistics from corresponding sets of events:

- Successful invocation — the firing of an event of nature type EXIT that follows a corresponding ENTRY event.
- Failed invocation — the firing of an event with a FAILURE nature following a corresponding ENTRY event.
- Time for successful completion — the elapsed time between the firing an ENTRY event and the firing of the corresponding EXIT event point.

The PMI publishes the statistics to the TPV, which presents cumulative counters for the number of successful and failed invocations and a running average of the completion response times.

Objectives of this tutorial

After completing this tutorial you will be able to:

- Select the performance statistics of service component elements that you want to monitor.
- View and interpret the resulting performance statistics.

Time required to complete this tutorial

This tutorial requires approximately 15-20 minutes to complete.

Prerequisites

In order to perform this tutorial, you must have:

- Configured and started a server.
- Enabled the PMI on the server.
- Installed and started the Samples Gallery application on the server.
- Installed and started the business rules sample application on the server. Follow the instructions on the Samples Gallery page to set up and run the business rules sample application.

After all of these prerequisites have been completed, run the business rules sample application from the Samples Gallery at least once before proceeding with the tutorial.

Example: Monitoring service component performance

For monitoring performance, you can use the administrative console to select service components for monitoring and view performance measurements. This example shows the use of the console to monitor performance statistics.

About this task

You will use the business rules sample application for this scenario, where you will monitor all three of the performance statistics: successes, failures, and response times. You should have the Web page containing this application already open; keep it open, because you will be running the sample several times after you begin monitoring. Ensure that you have already run the sample at least once, which causes it to appear in the list of functions that you can select to monitor.

Procedure

1. Open the administrative console.
2. Select the cluster or server to monitor.
   - To monitor a cluster, click **Servers** → **Clusters** → ***cluster_name***.
   - To monitor a single server, click **Servers** → **Application Servers** → ***server_name***.
3. Click the Runtime tab.
4. Under Performance, click **Performance Monitoring Infrastructure**.
5. Select **Custom**.
6. Expand **WBIStats.RootGroup** → **BR** → **brsample_module.DiscountRuleGroup** → **Operation**.
7. Select **_calculateDiscount**
8. Select the check boxes next to BadRequests, GoodRequests, and ResponseTime.
9. Click **Enable**
10. In the navigation pane, click **Monitoring and Tuning** → **Performance Viewer** → **Current Activity**.
11. Select the check box next to *server_name*, then click **Start Monitoring**.
12. Click *server_name*.
13. Expand **WBIStats.RootGroup** → **BR** → **brsample_module.DiscountRuleGroup** → **Operation**.
14. Select the check box next to _calculateDiscount

Results

You should now see a blank graph, and underneath that the names and values for the three statistics. Select the check boxes next to the statistic names, if they are not already checked. The PMI is now ready to publish performance data for the selected event, and the Tivoli Performance Viewer is ready to present the results.

Run the business rules sample application several times, and then watch the performance viewer as it periodically refreshes. Notice that there are now lines on the graph, representing the cumulative number of successful requests and the average response time for each successful request. You can also see the values next to the name for each statistic below the graph. The line for the number of successes should continue to rise as you perform additional invocations of the sample, while the response time line should level off after a few refreshes.

After you have completed this example, you should understand how WebSphere Process Server implements performance monitoring of service components. You should know how to select service components for monitoring, and how the performance statistics are calculated. You will also be able to start the performance monitors, and view the performance measurements for your applications as they are being used.

Performance monitoring can tax system resources; therefore, after you have completed this task you should stop the monitors. To do this, click on the Tivoli Performance Viewer link, select both the node and the server, and press **Stop Monitoring**.

Notices

This information was developed for products and services offered in the U.S.A.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at ″Copyright and trademark information″ at www.ibm.com/legal/copytrade.shtml.

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (http://www.eclipse.org).

IBM WebSphere Process Server for z/OS, Version 6.1.2

Printed in USA