IBM SOA Policy Pattern

Contents

Chapter 1. Pattern overview

The IBM® SOA Policy Pattern routes MQ JMS messages based on data held in policy documents retrieved from a service registry.

The IBM SOA Policy Pattern for Red Hat Enterprise Linux V2.0 provisions and manages the IBM PureApplication System (IPAS) hardware or IBM Workload Deployer (IWD) to provide the following features, which are pre-configured as a part of the pattern:

- An enterprise service bus (ESB), IBM WebSphere® Message Broker
- A JMS provider, WebSphere MQ.
- A service registry, WebSphere Service Registry and Repository (WSRR)

What scenarios are enabled by this pattern?

MQ JMS applications send messages to the JMS input queue for this pattern, and those messages are routed to another MQ JMS queue depending on which policy matches that input message. The pattern uses JMS header information to decide which policies are applicable, then evaluates those policies to determine where the message is routed. A response is sent back to the JMS sending application to acknowledge that the message has been routed. As a result, the pattern can support many JMS applications simultaneously, each with their own routing rules expressed through a set of policies.

Policies specify the schedule in terms of the times of the day and the day of the week, and so on, to route messages to different endpoint destinations. No other conditions or actions are supported in this pattern. The pattern uses the WS-MediationPolicy standard to define how and when messages are routed. The namespace for this standard is http://www.ibm.com/xmlns/stdwip/2011/02/ws-mediation. The Web Services Mediation Policy 1.0 domain defines a set of policy assertions for describing mediation requirements for a service.

Each policy is a part of the SOA policy lifecycle. Policies that are applied must be in the Approved, Deprecated, or Superceded governance states. For more information, see “Policy usage in the IBM SOA Policy Pattern” on page 29.

What is included in the pattern?

The IBM SOA Policy Pattern is an example of a virtual system pattern. A virtual system pattern consists of a collection of parts. Each part is a virtual operating system image containing installed IBM software that has been configured based on pattern parameters supplied during the provisioning process.

This pattern provides three parts:

- An image containing WebSphere Message Broker V8.0.0.1 and WebSphere MQ V7.0.1.8.
- An image containing WebSphere Service Registry and Repository V8.0 and WebSphereApplication Server V8.0.
- An image containing DB2® Enterprise Edition (to support WSRR) V9.7.5.

When the IBM PureApplication System hardware or IBM Workload Deployer user creates an instance of the IBM SOA Policy Pattern to provide a pre-configured ESB, three images are created from those parts. This configuration is shown in the following figure:

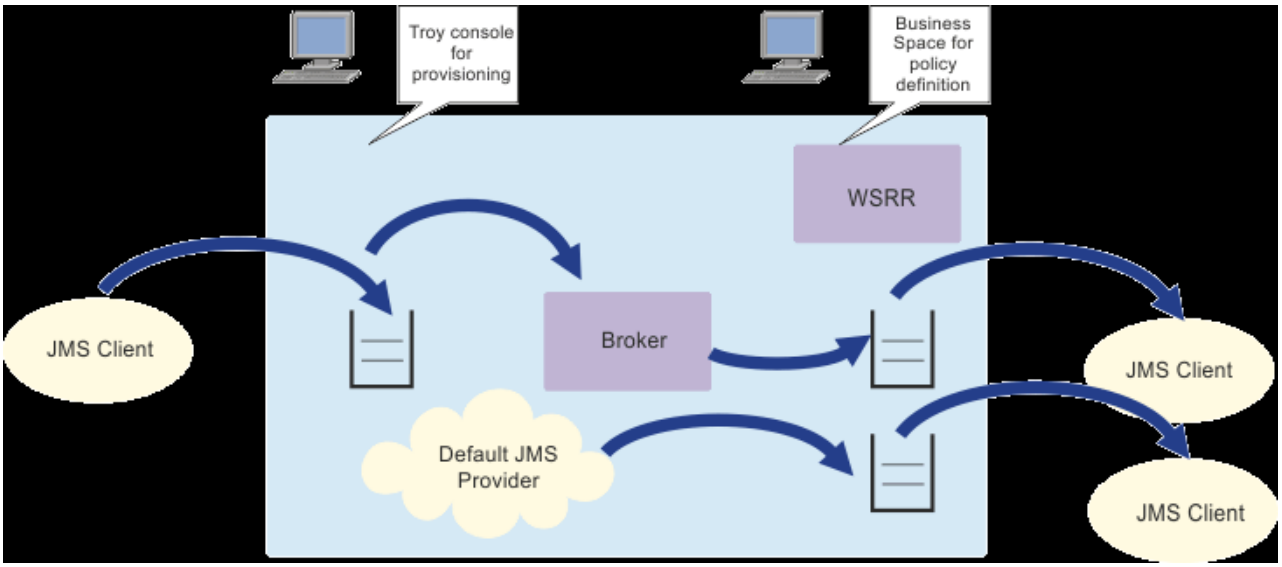

Figure 1. Overview of the IBM SOA Policy Pattern

To create this configuration the user runs the following components:

1. One WebSphere MQ queue manager to provide JMS services and allow JMS programs to connect to the pattern.
2. One preconfigured WebSphere Message Broker to perform the routing between JMS destinations.
3. One preconfigured WSRR instance to define and manage the policies that control routing.
4. One DB2 instance to support WSRR.
5. The IBM Workload Deployer or IBM PureApplication System browser-based user interface used to deploy the pattern.
6. The Business Space browser-based user interface used for creating and managing policies.

What other applications does it integrate with?

You can load your own policy documents into WSRR and these policies define their own JMS end point destinations. At first configuration, the registry is loaded with two example policies that use two example endpoints. The WebSphere Message Broker configuration included with the IBM SOA Policy Pattern provides a message flow that reads JMS messages from an input queue, and based on the policies retrieved from the registry, routes the messages to output queues.

The IBM SOA Policy Pattern includes a JMS provider but does not include JMS applications, so you need to add your existing MQ JMS applications to complete the solution. JMS destinations are defined using standard WebSphere MQ procedures. You can choose how your MQ JMS applications connect to control what sort of messaging topology you build; they can attach remotely a single queue manager hosted by the pattern, using MQ Client bindings, or they can use

MQ distributed messaging techniques to feed messages into the pattern queue manager from an existing remote queue manager.

Draft comment

!!! NOTE !!! this second part is really a new user story, and not something we have described in the documentation, but it is important to explain the difference between the JMS configuration files needed by the broker and those which might be needed by the applications.

How do you control the message routing?

When the pattern has been instantiated, the routing behaviour is controlled by a policy administrator who uses Business Space (provided with WSRR) to define and manage policies that satisfy routing requirements. For each policy, a JMS destination needs to exist so a messaging administrator must ensure that each JMS endpoint defined in a policy also exists on the messaging subsystem. For more information, see Chapter 5, "Working with the deployed instance," on page 25.

Related concepts:

"IBM SOA Policy Pattern" on page 10
The IBM SOA Policy Pattern provides a JMS-based dynamic message routing environment using WebSphere Message Broker and WSRR.

Related information:

IBM WebSphere Message Broker Version 8.0.0.0 Information Center

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

Chapter 2. Getting started with the IBM SOA Policy Pattern

Review the topics in this section to understand what is covered in this scenario, the reasons why a business might want to follow the scenario, the user roles involved, and an overview of the solution proposed by the scenario.

Before you begin

You can use the IBM SOA Policy Pattern on IBM PureApplication System or on the or IBM Workload Deployer appliance.

Procedure

To use the IBM SOA Policy Pattern, complete the following steps:

1. Download and install the IBM SOA Policy Pattern. See "Downloading and installing the pattern" for information about downloading the packages from Passport Advantage®.
2. Configure and deploy the pattern. For more information, see Chapter 3, "Working with the IBM SOA Policy Pattern," on page 9.
   a. Accept the imported virtual system image licenses for WebSphere Message Broker, WSRR, and DB2.
   b. Optional: Configure user access to the imported products Message Broker, WSRR, DB2 images in the catalog.
   c. Deploy the pattern. For more information, see "Deploying instances from the IBM SOA Policy Pattern" on page 18.
   d. Verify the deployment. See "Verify the deployment" on page 19.
3. Use the IBM SOA Policy Pattern images on the Workload Deployer appliance or PureApplication System, see Chapter 5, "Working with the deployed instance," on page 25.

Downloading and installing the pattern

The IBM IBM SOA Policy Pattern images for use with IBM Workload Deployer Version 3.1.0.2 or IBM PureApplication System are packaged for download from Passport Advantage.

Before you begin

Ensure that there is 15 GB of space available for the `CI9G8ML.tar.gz` file that contains the compressed pattern installer, and a further 16 GB of space available for the extracted files.

The image must be downloaded to a system running Microsoft Windows or Linux and with Java™ V1.6 installed.

DB2 Enterprise V9.7.5.0 must be installed on the Cloud appliance before the pattern can be installed.

About this task

The IBM SOA Policy Pattern images and patterns are provided in an open virtual archive (OVA) file. The OVA file and script installers are packaged together for download from Passport Advantage.

Procedure

To download the IBM SOA Policy Pattern images from Passport Advantage, complete the following steps:

1. Access the Passport Advantage web site: IBM Passport Advantage.
2. Download the package file containing the image and patterns. The file is named `CI9G8ML.tar.gz`.
3. Extract the contents of the `CI9G8ML.tar.gz` file to your local Microsoft Windows or Linux system. On Linux, enter:

```
tar -xvzf CI9G8ML.tar.gz
```

4. Open a command prompt and navigate to the directory containing the extracted file content.
5. To install the IBM SOA Policy Pattern into the Cloud appliance, run the installer command. The name of the command is `installer.bat` on Microsoft Windows or `installer` on Linux. Enter the following command: `installer -h <host> -u <username> -p <password>` where the username and password are the Cloud Administrator credentials. For example:

```
./installer -h apiliwd.rchland.ibm.com -u cbadmin -p cbadmin
```

6. When prompted, read and accept the IBM SOA Policy Pattern license.

Results

The scripts and parts are loaded and the virtual system pattern required for this pattern is created and the pattern is added to the pattern catalog.

Note: ..If a virtual system pattern at the correct version used in the IBM SOA Policy Pattern already exists in the catalog, it is not overwritten.

Configuring user access

To enable users to access the images and patterns on the appliance, the Runtime Administrator must first create the user access. You can either create the users first and add the users to the group or create the group first and then create the users and add them to the group.

About this task

Administrative users, usually the Runtime Administrator, can add other users to access and administer the patterns.

Procedure

To configure user access, complete the following steps:

1. Choose one of the following options to configure users and, optionally, user groups:
   - Add and configure a user from the Users window of the interface.
     a. From the menu click **System** > **Users**.

b. Click the Add icon.
c. Provide a short user name as well as the user's actual name, email address, and passwords and click **OK**.
d. Select the user you added in the Users panel to configure access. Configure the access and actions of the user you selected.
e. Add the user to one or more user groups in the **User groups** field.

- Create a user group.
  a. From the menu click **System** > **User Groups**.
  b. Click the Add icon. Provide a name and description for the group.
  c. Select the group you added in the User Groups panel to configure the access.
  d. Add members in the **Group members** field and supply the permissions to apply to the group.

2. Optional: If you have already added the virtual images, provide access for the users or group to the virtual images. From the menu, click **Catalog** > **Virtual images** to open the Virtual Images window. Select a IBM SOA Policy Pattern virtual image from the left panel and then add the users or group in the right panel.

What to do next

If you have not yet added the virtual images, add those and then provide the users or group access to them.

Related information:

IBM PureApplication System: Managing users and groups

IBM Workload Deployer: Managing users and groups

Chapter 3. Working with the IBM SOA Policy Pattern

An IBM SOA Policy Pattern provides a topology definition for repeatable deployment that can be shared. Patterns describe the function provided by each virtual machine in a virtual system; each function is identified as a part in the pattern. Set up and configure the pattern before you deploy it.

To install and deploy the pattern, complete the following steps:

1. Download the IBM SOA Policy Pattern image file from the Passport Advantage web site: IBM Passport Advantage.
2. Install the pattern to the appliance. For more information, see "Downloading and installing the pattern" on page 5.
3. Read and accept the license agreements.
4. Configure the roles and access control for users to administer the image and patterns. The Runtime Administrator configures the appropriate roles and accesses for users to the images and patterns.
5. Deploy the pattern. For more information, see "Deploying instances from the IBM SOA Policy Pattern" on page 18.

Patterns and parts

The IBM SOA Policy Pattern parts are the functional components of a pattern. Each part represents a single virtual machine. A pattern provides a topology definition for repeatable deployment that can be shared.

Patterns describe the function provided by each virtual machine in a virtual system. Each function is identified as a part in the pattern. Patterns take on the characteristics of their associated parts. For example, when a pattern containing a WebSphere Message Broker part is deployed, the user gets a virtual machine with a running WebSphere Message Broker instance.

Parts

Parts describe the components that are configured on a virtual machine. Use parts to create patterns. Each part has a set of properties (parameters) that are used during deployment to help define the overall configuration of the virtual system. You can customize a part by modifying its parameters, by adding a script package, or both. When you load the IBM SOA Policy Pattern images onto IBM Workload Deployer, the parts are included.

Patterns

You can use predefined patterns, create new patterns, and edit existing patterns that have not been completed and locked. For detailed information about using the IBM Workload Deployer to access existing patterns or create custom pattern, see the IBM Workload Deployer, Version 3.1 Information Center.

Note: The patterns differ from the topology patterns described in the WebSphere Message Broker Information Center. While the topology patterns describe the

functions provided by clusters, the WebSphere Message Broker Hypervisor Edition patterns in IBM Workload Deployer describe the functions provided by each virtual machine.

- **Predefined pattern**: When you load WebSphere Message Broker Hypervisor Edition on the IBM Workload Deployer, several predefined patterns are created, which can be deployed without modification to the cloud. One pattern is a Basic pattern supporting WebSphere Message Broker for development and unit test. The second pattern provides additional configuration, potentially required for production and quality assurance environments. For a more detailed description of predefined patterns see,

  Draft comment
  WebSphere Message Broker Hypervisor Edition predefined patterns

  .
- **Deploying patterns**: Use IBM Workload Deployer to deploy a pattern and create virtual systems in the cloud. You can deploy any pattern defined in the IBM Workload Deployer, including the predefined patterns loaded with the WebSphere Message Broker Hypervisor Edition virtual machine image or custom patterns that you create.

Related information:

IBM WebSphere Message Broker Version 8.0.0.0 Information Center

Patterns

Patterns provide a repeatable topology that can be deployed to a cloud. Deployed patterns are virtual systems running in the cloud. Patterns, whether predefined or created, contain parts. Some parts are required for the pattern to function when deployed to the cloud as a virtual system.

When the virtual images have been loaded into IBM Workload Deployer or IBM PureApplication System, and the proper access has been assigned to the users, users can begin to work with the patterns of the images.

The IBM SOA Policy Pattern

This pattern contains the following required parts:

- WebSphere Message Broker Advanced 8.0.0.1
- WSRR Standalone server 8.0.0.0
- DB2 Enterprise 9.7.5.0

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

IBM WebSphere Message Broker Version 8.0.0.0 Information Center

IBM SOA Policy Pattern

The IBM SOA Policy Pattern provides a JMS-based dynamic message routing environment using WebSphere Message Broker and WSRR.

The IBM SOA Policy Pattern pattern requires the following parts:

- **WebSphere Message Broker Basic 8.0.0.1** - In the WebSphere Message Broker Basic 8.0.0.1 virtual system pattern, you can configure the Queue Manager name, Queue Manager listener port number, and mandatory passwords. The rest of the parameters are locked and are inherited from the base virtual system pattern of WebSphere Message Broker with their default values. For more information, see "WebSphere Message Broker Basic part" on page 15.

  The default Message Broker product created is MB8BROKER, and the default administrative user is virtuser.
- **WSRR Standalone server 8.0.0.1** - In the WSRR Standalone server 8.0.0.1 virtual system pattern, you can configure the root password, WebSphere administrative user name, and Websphere administrative password. The rest of the parameters are locked and are inherited from the base virtual system pattern of WSRR Standalone server 8.0.0.1 with their default values. For more information, see "WSRR Standalone server part" on page 15.
- **DB2 Enterprise 9.7.5.0** - You can configure the passwords for root, db2inst1, db2fenc1, dasusr1, and virtuser. The rest of the parameters are locked and are inherited from the base virtual system pattern of DB2 Enterprise 9.7.5 with their default values. For more information, see "DB2 Enterprise part" on page 16.

The pattern configures two sample policies in WSRR and two sample JNDI destinations. For more information about the samples, see "Samples" on page 13. The sample scripts can be replaced with your own policies and JNDI destinations by cloning the pattern and adapting the sample scripts with your own customized settings.

The parts must be instantiated in the following order:

1. DB2 Enterprise 9.7.5.0
2. WSRR Standalone server 8.0.0.0
3. WebSphere Message Broker Basic 8.0.0.1

Scripts and advanced options

Scripts are used to configure the products and each perform a complete configuration step; for example, to load policies into WSRR. The scripts used for this pattern cannot be used outside of a part in this pattern.

The IBM SOA Policy Pattern pattern requires the following scripts on the WSRR Standalone server part:

- **SOA Policy Pattern: Create Sample Policies** - An optional script. This script creates and approves two sample mediation policies with a PolicySelector value and scheduling conditions each with a different JNDI MQ destination. The schedule for the sample policies is specified in terms of time and days of the week. To use a script containing customized new policies, clone the pattern and replace the script.
- **SOA Policy Pattern: WSRR Configuration** - This script package is mandatory for this pattern. The script adds a WSRR modifier plugin that creates the PolicySelector property for a policy when it is created, and makes the necessary Business Space user interface modification that displays the PolicySelector property in the user interface. The script also adds a public key to the `authorized_keys` file which enables the password-less SCP/SSH login. The SCP/SSH session is used to transfer the required configuration files from WSRR to the Message Broker instance to configure security. To enable Message Broker to get cache notifications that are sent by WSRR for any policy updates, a non-secure JMS client connection is enabled using the **`SSL-supported`** transport

option under the IIOP/RMI security of the WebSphere Application Server configuration in this script package. This script is needed for the "SOA Policy Pattern: Connect Broker to Secure WSRR" script on the WebSphere Message Broker Advanced part to configure the WSRR and Message Broker security.

The IBM SOA Policy Pattern requires the following scripts on the WebSphere Message Broker Basic part:

- **SOA Policy Pattern: Create Instance** - This script creates the required JNDI definitions and queues required in this pattern. It creates the required JNDI bindings for REQUEST_IN, REPLY_OUT, and BLACKOUT. The script also creates the respective MQ queues REQUEST_INQ, REPLY_OUTQ, and SYSTEM.DEAD.LETTER.QUEUE. Then, it creates the execution group on the default Message Broker, MB8BROKER, and the default queue manager, MB8QMGR. The execution group name is set to "default". Finally, the message flow BAR file is deployed.
- **SOA Policy Pattern: Create Sample JNDI Bindings** - An optional script. This script creates the sample JNDI bindings and two associated JMS destinations used in this sample. The JNDI definition created are DESTINATION1_OUT and DESTINATION2_OUT. This script also creates the MQ queues, DESTINATION1_OUTQ and DESTINATION2_OUTQ, associated with JNDI objects and then creates the MQ JNDI bindings needed for the IBM SOA Policy Pattern. This script package can be edited in a clone of this pattern, and new QCF definitions and JNDI definitions for the sample policies can be added before being deployed.
- **SOA Policy Pattern: Connect Broker to Secure WSRR** - This script package is mandatory for this pattern. This script performs the security configuration for WebSphere Message Broker and enables a secure connection to WSRR. The `DummyClientKeyFile.jks` and `DummyClientTrustFile.jks` files are copied from the remote WSRR system using password-less SCP. The password-less SCP/SSH configuration is created by the SOA Policy Pattern: WSRR Configuration script on WSRR and when the files have been copied from WSRR, the configuration is deleted from the WSRR instance. The script uses the default port 9443 to connect to WSRR and the default password "WebAS" is used for both keystores.

  The cache notification of the WSRR policy update is enabled in Message Broker. The cache notification uses a non-secure JMS client connection over IIOP to connect to WebSphere Application Server. To enable the non-secure JMS client connection, the "SSL-supported" transport option is set under the IIOP/RMI security of the WebSphere Application Server configuration using the script package "SOA Policy Pattern: WSRR Configuration" on the WSRR Standalone server part.

The scripts must be run in the following order:

1. SOA Policy Pattern: Create Sample Policies
2. SOA Policy Pattern: WSRR Configuration
3. SOA Policy Pattern: Create Instance
4. SOA Policy Pattern: Create Sample JNDI Bindings
5. SOA Policy Pattern: Connect Broker to Secure WSRR

Setting up the pattern

To optionally change the values that have been defaulted, complete the following steps:

1. Load the IBM SOA Policy Pattern.

2. Configure the configurable properties, see “WebSphere Message Broker Basic part” on page 15.
3. Deploy the pattern to the cloud.

Related concepts:

“WebSphere Message Broker Basic part” on page 15
The WebSphere Message Broker Basic part provides some configuration options.

“WSRR Standalone server part” on page 15
The WSRR Standalone server part provides some configuration options.

“DB2 Enterprise part” on page 16
The DB2 Enterprise part provides some configuration options.

Related tasks:

“Customizing the pattern” on page 17
To customize the pattern, clone the pattern and edit the cloned version.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

IBM WebSphere Message Broker Version 8.0.0.0 Information Center

Samples:

Sample policies and applications are provided with the default pattern that configure sample policies and sample JNDI destinations.

Sample policies

Two sample policies are created when this script package is run:

- SampleRoutingSchedule01
- SampleRoutingSchedule02

These policies have the following scheduling conditions:

- **`StartTime`** - 8 a.m.
- **`StopTime`** - 8 p.m.
- **`WeekDays`** - Every day of the week.

The policies in WSRR also have a custom property field added called **`PolicySelector`**. For SampleRoutingSchedule01, this value is set to GID007 and for SampleRoutingSchedule02 this value is set to GID008. The PolicySelector value is provided by the client as part of the JMS message header and the matching policy is enforced by the message flow.

Sample JNDI bindings

The default JNDI destinations created in the “SOA Policy Pattern: Create Sample JNDI Bindings” script package are DESTINATION1_OUT and DESTINATION2_OUT, and the respective MQ queues DESTINATION1_OUTQ and DESTINATION2_OUTQ.

Sample JMS client

Two sample JMS clients are provided with this pattern. The SendJMSMessage and ReceiveJMSMessage Java sample clients are located in `/opt/ibm/mqsi/8.0.0.1/sample/JMSSendReceive` in the installation directory.

- You can use the SendJMSMessage sample JMS client application to send the JMS Message with a specific PolicySelector value. It is a command-line program that uses three mandatory arguments to specify the JNDI bindings location, JMS message body, and the PolicySelector value. The sample JMS client application waits for the reply message from the IBM SOA Policy Pattern message flow, and when the reply is received it shows the reply message in stdout. Comments are included in the sample code to explain the significant things it does in order to interact with the routing flow.

  To run the sample JMS client application, run the following command as virtuser:

```
java com.ibm.jms.SendJMSMessage <Location JNDI Bindings> <InputTextMessage>
 <PolicySelector Value>
```

  For example:

```
java com.ibm.jms.SendJMSMessage file:///home//virtuser//JNDI-DIR
 '<data><msg>11</msg></data>' GID007
```

- You can use the ReceiveJMSMessage sample JMS client application to get the message from the endpoint destination queue where the IBM SOA Policy Pattern message flow routes the messages. The output message is shown in the stdout log with its JMS Header. To access the stdout log to see the output message, click **Instances** > **Virtual system** , expand the **Virtual machines** section, and click **remote_std_out.log**.

  To read the message, the command takes two mandatory parameters including the JMS output destination as its argument:

```
java com.ibm.jms.ReceiveJMSMessage  <Location JNDI Bindings> <Output JMS Destination>
```

  For example:

```
java com.ibm.jms.ReceiveJMSMessage file:///home//virtuser//JNDI-DIR DESTINATION1_OUT
```

Customizing the policies and JNDI bindings

Optionally, you can delete the "SOA Policy Pattern: Create Sample Policies" and "SOA Policy Pattern: Create Sample JNDI Bindings" script packages from the cloned IBM SOA Policy Pattern. After removing these optional script packages, you can deploy the cloned IBM SOA Policy Pattern and use the deployed instance of the pattern. For more details about WSRR policy management and managing JMS destinations on the deployed instance, see Chapter 5, "Working with the deployed instance," on page 25.

Related concepts:

"Policy usage in the IBM SOA Policy Pattern" on page 29
Policies in WSRR are administered using the Business Space user interface. Policies can be added, edited, or removed at any time. Policies are selected based on their PolicySelector property value and governance state. Policies are valid if they are in the Approved, Deprecated, or Superseded governance states, and policies in other governance states are discarded during schedule validation. The Schedule condition is the only condition that this pattern accepts, and the routing action is the only action that is accepted by this pattern.

Related tasks:

"Cloning the IBM SOA Policy Pattern" on page 16
The IBM SOA Policy Pattern cannot be edited. If the topology provided in the IBM SOA Policy Pattern virtual system patterns do not provide the function you need, the pattern can be cloned and then edited to create new patterns.

"Managing the routing behavior of the SOA Policy pattern" on page 28
JMS destinations and policies can be added, edited, or removed at any point. Policies are active if they are in the Approved, Deprecated, or Superseded governance states. Policies in other governance states are discarded during schedule validation.

Parts

The following parts comprise the IBM SOA Policy Pattern.

WebSphere Message Broker Basic part

The WebSphere Message Broker Basic part provides some configuration options.

The WebSphere Message Broker Basic pattern consists of a single part, called WebSphere Message Broker Basic 8.0.0.1. When this part is deployed, it creates a Message Broker instance, a single execution group, and a queue manager with a WebSphere MQ listener defined on port 2414. You can configure the pattern in other ways; for example, by creating additional execution groups and deploying BAR files.

The configurable parameters of the WebSphere Message Broker Basic 8.0.0.1 virtual system image are described in the following table:

Table 1. Configurable parameters

Parameter name	Required	Configurable	Default value	Description
Queue Manager	Yes	Yes	MB8QMGR	Allows the user to set the password for the db2inst1 user.
Queue Manager TCP/IP Listener port	Yes	Yes	2414	Allows the user to set the password for the db2fenc1 user.
Password (root)	Yes	Yes	password	Allows the user to set the password for the root user.
Administrative password (virtuser)	Yes	Yes	password	Allows the user to set the password for the virtual user for the WebSphere Message Broker system that is provisioned.

Note: Do not change the value of the WSRR_HOST_IPDDR parameter. This value contains the IP address of the WSRR instance.
Other parameters are inherited from the base virtual system pattern and are locked.

Draft comment

If you manually loaded the virtual image on the IBM Workload Deployer instead of using Image Loader, the predefined patterns are not available.

WSRR Standalone server part

The WSRR Standalone server part provides some configuration options.

The configurable parameters of the WebSphere Service Registry and Repository 8.0.0.1 virtual system image are described in the following table:

Table 2. Configurable parameters

Parameter name	Required	Configurable	Default value	Description
Password (root)	Yes	Yes	password	Allows the user to set the password for the root user.
WebSphere administrative user name	Yes	Yes	password	Allows the user to set the user ID used to login to WAS admin console.
WebSphere administrative password	Yes	Yes	password	Allows the user to set the password used to login to WAS admin console.

Other parameters are inherited from the base virtual system pattern and are locked.

DB2 Enterprise part

The DB2 Enterprise part provides some configuration options.

The configurable parameters of the DB2 Enterprise 9.7.5 virtual system image are described in the following table:

Table 3. Configurable parameters

Parameter name	Required	Configurable	Default value	Description
Password (db2inst1)	Yes	Yes	password	Allows the user to set the password for the db2inst1 user.
Password (db2fenc1)	Yes	Yes	password	Allows the user to set the password for the db2fenc1 user.
Password (dasusr1)	Yes	Yes	password	Allows the user to set the password for the dasusr1 user.
Password (root)	Yes	Yes	password	Allows the user to set the password for the root user.
Password (virtuser)	Yes	Yes	password	Allows the user to set the password for the virtuser user.

Other parameters are inherited from the base virtual system pattern and are locked.

Cloning the IBM SOA Policy Pattern

The IBM SOA Policy Pattern cannot be edited. If the topology provided in the IBM SOA Policy Pattern virtual system patterns do not provide the function you need, the pattern can be cloned and then edited to create new patterns.

About this task

Note: Customization of the IBM SOA Policy Pattern other than removing the sample scripts is not supported.

Procedure

To copy the patterns to edit them and create new patterns, complete the following steps:

1. From the left panel of the Pattern window, select the pattern to copy.
2. Click the Clone icon and enter a name for the new pattern.
3. Select the new pattern and click the Edit icon to change the configuration. You can add and remove parts and configure them, increase or decrease the number of some parts, or change the order in which some parts are deployed. For more information, see "Customizing the pattern."

What to do next

Ensure that you have all required parts properly configured for the type of pattern you created, and that the order of part and script deployments is valid. You can deploy the pattern when your configuration is complete.

Related tasks:

"Customizing the pattern"
To customize the pattern, clone the pattern and edit the cloned version.

"Deploying instances from the IBM SOA Policy Pattern" on page 18
Deploying the IBM SOA Policy Pattern creates a running virtual system instance of the pattern.

Related information:

IBM Workload Deployer: Managing virtual system patterns

IBM PureApplication System: Managing virtual system patterns

Customizing the pattern

To customize the pattern, clone the pattern and edit the cloned version.

About this task

The only supported customization of the pattern topology is the removal of the sample scripts. Do not make any other customizations to the IBM SOA Policy Pattern.

To customize policies and JMS endpoints, deploy the instance and make changes to the policies in Business Space and JMS endpoints in WebSphere MQ after deployment.

Procedure

1. Clone the pattern and click **Edit** on the new pattern. For more information about cloning a pattern, see "Cloning the IBM SOA Policy Pattern" on page 16.
2. To change parameters on the part such as initial passwords, click the Edit icon for the part. For more information about the default values on each part, see "Parts" on page 15.

3. To remove the sample scripts, click the Remove icon for the "SOA Policy Pattern: Create Sample Policies" script on the WSRR Standalone server part and "SOA Policy Pattern: Create Sample JNDI Bindings" script on the WebSphere Message Broker Basic part.
4. To add scripts from the Pattern Editor, drag and drop the script onto the relevant part. Script order of scripts in the IBM SOA Policy Pattern is important. For more information about the order of scripts, see "IBM SOA Policy Pattern" on page 10. To change the order of scripts, click **Ordering** and drag and drop the scripts into the correct order. The order of the scripts on the part in the Topology view is updated to show these changes.
5. Click **Done editing** to save the changes to the pattern.

What to do next

When the pattern has been created, you can deploy an instance of the pattern.

Related tasks:

"Cloning the IBM SOA Policy Pattern" on page 16
The IBM SOA Policy Pattern cannot be edited. If the topology provided in the IBM SOA Policy Pattern virtual system patterns do not provide the function you need, the pattern can be cloned and then edited to create new patterns.

"Deploying instances from the IBM SOA Policy Pattern"
Deploying the IBM SOA Policy Pattern creates a running virtual system instance of the pattern.

"Connecting to WSRR" on page 27
Use the Business Space user interface to administer policies.

"Connecting to the WebSphere MQ system" on page 26
Use the VNC console to access the WebSphere MQ system to add, edit, or remove JMS destinations.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

IBM WebSphere Message Broker Version 8.0.0.0 Information Center

Deploying instances from the IBM SOA Policy Pattern

Deploying the IBM SOA Policy Pattern creates a running virtual system instance of the pattern.

Before you begin

To deploy a pattern you must first have either a predefined pattern or a new pattern that is complete, with all required parts configured.

About this task

Deploying a pattern creates a virtual system instance that is running in the cloud.

Procedure

To deploy the IBM SOA Policy Pattern, complete the following steps:

1. Click **Patterns** > **Virtual Systems**
2. From the Virtual System Patterns list, select the pattern to deploy.

3. Click the Deploy icon.
4. Complete the required fields to deploy the pattern. A check mark beside each item indicates that it does not require further configuration.
   a. In the **Virtual system name** box, enter a name for the instance.
   b. Optional: To change the parameters for configured parts, click **Configure virtual parts** then click the part name to open the editor for the part.

      Note: Usernames and passwords are preconfigured with default values within the configuration settings. For more information, see the part details for each part "Parts" on page 15.
5. Click **OK** to deploy the pattern.

Results

The deployment process creates and starts virtual machines for the parts defined and provides links to required consoles. The time for the deployment depends on the complexity of the pattern being deployed. A deployed pattern is a virtual system, or a newly provisionedIBM SOA Policy Pattern runtime environment.

What to do next

To view the status of your instance, to see when deployment is complete and to administer it, click **Instances** > **Virtual system** and select the instance from the Virtual System Instances list. For more information about viewing the details of an instance or using it, see Chapter 5, "Working with the deployed instance," on page 25.

To verify the success of the deployment, see "Verify the deployment."

To test the deployment by sending some sample messages, see Chapter 4, "Tutorial: Working with the sample application," on page 21.

Related concepts:

Chapter 5, "Working with the deployed instance," on page 25
When the IBM SOA Policy Pattern image has been deployed, you can configure your policies and JMS destinations for the deployed instance. To view the list of deployed instances, click **Instances** > **Virtual system**.

Related tasks:

"Verify the deployment"
When you have deployed the pattern, verify that the deployment was successful.

Chapter 4, "Tutorial: Working with the sample application," on page 21
Complete the tasks in this tutorial to verify that the pattern has been configured by viewing the artefacts created in WebSphere Message Broker, WebSphere MQ, and WSRR. Then, running the sample application sends some messages that are routed to different queues based on the policies provided with the sample.

Related information:

IBM Workload Deployer: Managing virtual system patterns

IBM PureApplication System: Managing virtual system patterns

Verify the deployment

When you have deployed the pattern, verify that the deployment was successful.

Procedure

1. Check the deployment logs for any failure in the virtual system deployment history. For more information, see "Troubleshooting problems in the deployed instance" on page 43.
2. Optional: Test the deployed instance by following the tutorial to send some sample messages using the sample applications provided. See Chapter 4, "Tutorial: Working with the sample application," on page 21.

Chapter 4. Tutorial: Working with the sample application

Complete the tasks in this tutorial to verify that the pattern has been configured by viewing the artefacts created in WebSphere Message Broker, WebSphere MQ, and WSRR. Then, running the sample application sends some messages that are routed to different queues based on the policies provided with the sample.

Before you begin

This tutorial requires that the IBM SOA Policy Pattern has been installed and deployed. See "Downloading and installing the pattern" on page 5.

About this task

The sample policies and applications that are provided with the IBM SOA Policy Pattern can be used to send some sample messages that are routed using the two sample policies to two sample JMS destinations based on the PolicySelector value in the message. This tutorial describes how to examine the policies created in WSRR using the Business Space user interface. The tutorial also describes sending sample messages with the PolicySelector value of GID007, and viewing the messages on the queue for the JMS destination in WebSphere MQ before and after the messages are received.

Procedure

1. View the deployed system in the appliance:
   a. Click **Instances** > **Virtual Systems**.
   b. From the list of instances in the Virtual System Instances window, select the instance that was deployed. The instance details are displayed.
   c. To see the virtual machines that are deployed as part of the instance, expand the **Virtual machines** section in the instance details pane.
2. View the sample policies in WSRR:
   a. In the instance details pane, expand the **Consoles** section.
   b. To connect to Business Space, click **WSRR Business Space**. For more information about connecting to Business Space, see "Connecting to WSRR" on page 27.
   c. Log in to Business Space with the WebSphere administrative username and password. For more information about the default passwords that were created when the pattern was deployed, see "WSRR Standalone server part" on page 15.
   d. Open the Operations space:
      1) Click **Go To Spaces** at the top of the page. The Go To Spaces dialog is displayed.
      2) Click on the space for Operations users. The specific name will depend on what was specified when the space was created.
   e. On the Overview tab, enter `SampleRoutingSchedule01` into the search box.
   f. In the search types list, select **Policy Document** and click **Search**. The Collection widget lists the SampleRoutingSchedule01 policy.
   g. Select the SampleRoutingSchedule01 policy. The policy details are shown in the Detail widget. Note that the PolicySelector value is `GID007`. This is the

PolicySelector value that your sent messages must contain to be routed by using the SampleRoutingSchedule01 policy.

h. Click the Edit icon to view more policy details. In the **Actions** section, note the JMS endpoint destination that messages are routed to.

3. Send some sample JMS messages with a PolicySelector value of GID007. These messages will be routed by using the SampleRoutingSchedule01 policy:
   a. Open a command prompt.
   b. In the installation directory for the SOA Policy pattern, navigate to `/opt/ibm/mqsi/8.0.0.1/sample/JMSSendReceive`.
   c. To send a sample message, enter the following command as virtuser:

```
java com.ibm.jms.SendJMSMessage file:///home//virtuser//JNDI-DIR
'<data><msg>11</msg></data>' GID007
```

   For more information about the sample application, see "Samples" on page 13. Repeat this command to send as many messages as you would like.

4. Connect to the WebSphere MQ system and view the queues:
   a. In the instance details pane, expand the **Consoles** section.
   b. To connect to the WebSphere MQ system using VNC, click **VNC**. For more information about connecting to WebSphere MQ, see "Connecting to the WebSphere MQ system" on page 26.
   c. Authenticate using the virtuser credentials. For more information about the default passwords created when the pattern was deployed, see the part details "WebSphere Message Broker Basic part" on page 15.
   d. To see various queues created by this pattern, including the depth of the queue, you can run the following command in the shell prompt:

```
runmqsc <MB8QMGR>
```

   For example, the DESTINATION1_OUTQ queue contains 14 messages in the following output:

```
1 : DIS QL(DES*) CURDEPTH
AMQ8409: Display Queue details.
   QUEUE(DESTINATION1_OUTQ)                TYPE(QLOCAL)
   CURDEPTH(14)
AMQ8409: Display Queue details.
   QUEUE(DESTINATION2_OUTQ)                TYPE(QLOCAL)
   CURDEPTH(0)
```

5. Receive the sample messages:
   a. Open a command prompt.
   b. In the installation directory, navigate to `/opt/ibm/mqsi/8.0.0.1/sample/JMSSendReceive`.
   c. To receive all of the sample JMS messages from the JMS endpoint destination specified in the policy document for the SampleRoutingSchedule01 policy, enter the following command as virtuser:

```
java com.ibm.jms.ReceiveJMSMessage file:///home//virtuser//JNDI-DIR DESTINATION1_OUT
```

   For more information about the sample application, see "Samples" on page 13.

   d. To access the stdout log to see the output message, click **Instances** > **Virtual system** , expand the **Virtual machines** section, and click **remote_std_out.log**.

6. Optional: Repeat step 4 to verify that the messages are no longer on the queue.

7. Optional: Repeat step 3 and send some messages with a PropertySelector value of GID008. This will route the messages to a different JMS endpoint destination.[1]

Related concepts:

"Samples" on page 13
Sample policies and applications are provided with the default pattern that configure sample policies and sample JNDI destinations.

1.

Draft comment
Related Links below - are there any more related links?

Chapter 5. Working with the deployed instance

When the IBM SOA Policy Pattern image has been deployed, you can configure your policies and JMS destinations for the deployed instance. To view the list of deployed instances, click **Instances** > **Virtual system**.

Viewing the instance details

The details of a deployed instance can be seen by selecting an instance in the list of instances in the Virtual System Instances window. The virtual system instance details are then displayed in a window with the title of that instance. The details include a list of virtual machines provisioned on the cloud infrastructure for that deployment, the IP address, virtual machine status, and role status. Role is a unit of function that is performed by the virtual application middleware on a virtual machine. You can also view the virtual machine role health status information. For example, a red check mark is on the green status arrow when the CPU status is critical on the virtual machine.

To see the provisioning and deployment status of the instance, see the **Current status** value in the details view.

During provisioning, to see the status of the virtual machines and scripts, expand the **History** section in the details view.

To see the details of the virtual machines and script logs, expand the **Virtual machines** section in the details view. The host and IP address of the system is the **Network interface 0** value in the **Hardware and network** section. Expand a running virtual machine to see the script logs in the **Script Packages** section and links to accessing the virtual machine using in the **Consoles** section.

Administering the IBM SOA Policy Pattern instances

After deploying a virtual system pattern into the cloud, you can view and administer the virtual system instance that was created to see your IBM SOA Policy Pattern environment.

Before you begin

To view a virtual system instance, you must first have deployed a virtual system pattern.

About this task

Deploying a pattern creates a virtual system instance, or a newly provisioned IBM SOA Policy Pattern runtime environment. When deployment is complete, the virtual system instance is running in the cloud.

Procedure

To administer the IBM SOA Policy Pattern virtual system instances, complete the following steps:

1. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
2. From the list of instances in the Virtual System Instances window, select the instance that was deployed.
3. If the instance is running in the cloud, you can log into the components of the virtual system from the console links in the virtual system view. The components available depends on the pattern that you created. For example, you could:
   - Launch and log in to the Business Space user interface in WSRR to administer policies.
   - Launch the VNC console for WebSphere MQ to administer JMS endpoints and queues.

Managing JMS providers

The IBM SOA Policy Pattern supports WebSphere MQ as the JMS Provider. WebSphere MQ provides the JMS administration tool to create JNDI bindings to manage the JMS administered object. The JMS client can use the JNDI bindings to retrieve the administered objects.

For more information about using the WebSphere MQ JMS administration tool, see the IBM WebSphere MQ 7.0 Information Center.

You might need to create new the JNDI destination using the WebSphere MQ JMS administration tool. To create JMS destinations, see "Managing JMS destinations" on page 37.

WebSphere MQ Explorer is used to administer the JMS Destination queues, and can also be used to manage JMS Administered objects. You can browse the messages in the MQ JMS queues to perform various administration tasks. For example, and for more information about connecting to the MQ system, see "Connecting to the WebSphere MQ system."

Related information:

IBM WebSphere MQ 7.0 Information Center - System Administration Guide

Connecting to the WebSphere MQ system

Use the VNC console to access the WebSphere MQ system to add, edit, or remove JMS destinations.

About this task

Access the WebSphere MQ system using the console link in the virtual machine details to the VNC console.

Procedure

1. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
2. From the list of instances in the Virtual System Instances window, select the instance that was deployed. The instance details are displayed.
3. Expand the **Virtual machines** section.
4. In the **Consoles** section, click **VNC** to connect to the WebSphere MQ system.

Results

The WebSphere MQ system is displayed. To administer JMS destinations, see "Managing JMS destinations" on page 37.

Connecting to WSRR

Use the Business Space user interface to administer policies.

About this task

Access the Business Space user interface by using the console link.

Procedure

1. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
2. From the list of instances in the Virtual System Instances window, select the instance that was deployed. The instance details are displayed.
3. Access the WSRR system by using the Business Space user interface:
   - In the **Consoles** section, click **WSRR Business Space** to connect to the Business Space running on the WSRR system.
   - Alternatively, in an external web browser:
     a. Find the hostname and port numbers for WSRR. Expand the **Virtual machines** section and select the virtual machine for the WSRR Standalone Server to see the virtual machine details. In the **Hardware and network** section, the hostname is the **Network interface 0** value.
     b. Enter the Business Space URL:
        – For the WSRR Standalone server with security enabled: `https://hostname:9443/BusinessSpace`
        – For the cluster: `http://hostname/BusinessSpace`

        where *hostname* and *port* are the host name and port value of the WSRR server.

Results

Business Space is displayed, and can be used to add, edit, or remove policies.

What to do next

If using Business Space on the WSRR system for the first time, see "Configuring Business Space for the first use" on page 28 and follow the steps to create the Operations space. To administer policies, see "Policy management" on page 29.

Related concepts:

"Policy usage in the IBM SOA Policy Pattern" on page 29
Policies in WSRR are administered using the Business Space user interface. Policies can be added, edited, or removed at any time. Policies are selected based on their PolicySelector property value and governance state. Policies are valid if they are in the Approved, Deprecated, or Superseded governance states, and policies in other governance states are discarded during schedule validation. The Schedule condition is the only condition that this pattern accepts, and the routing action is the only action that is accepted by this pattern.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

Configuring Business Space for the first use

Before the Business Space user interface can be used to create policies, the Operations space must be created.

Before you begin

Log in to the Business Space user interface by using the WSRR administrative username and password. For information about accessing Business Space, see "Connecting to WSRR" on page 27. For more information about the default passwords created with this pattern, see "WSRR Standalone server part" on page 15.

About this task

If an Operations space has not been created, you must create one. Spaces in Business Space are defined for specific roles. Policy authoring is best performed in the Operations space because it contains widgets for administering policies.

Procedure

To create a space that is based on the Service Registry for Operations template, complete the following steps:

1. Click **Manage Spaces** from the selection of space management links at the top of the page. The Space Manager dialog is displayed.
2. Click **Create Space**. The Create Space dialog is displayed.
3. Enter a name in the **Space name** field; for example, `Operations Space`. Optional: enter a description.
4. Select **Service Registry for Operations** from the **Create a new space using a template** list, and then click **Save**.
5. The new space is displayed in the **Space manager** list. Click the new space to open it.

Results

The Operations space is created. To open the Operations space:

1. Click **Go To Spaces** at the top of the page. The Go To Spaces dialog is displayed.
2. Click on the space for Operations users. The specific name will depend on what was specified when the space was created.

Managing the routing behavior of the SOA Policy pattern

JMS destinations and policies can be added, edited, or removed at any point. Policies are active if they are in the Approved, Deprecated, or Superseded governance states. Policies in other governance states are discarded during schedule validation.

About this task

When an instance has been deployed, you can manage policies in the registry and change the JMS destinations to customize the message flow.

Procedure

To make changes to the routing behavior:

- To change the policies, administer policies using the Business Space user interface. For more information, see "Policy management."
- To change the JMS destinations, administer the JMS endpoints and bindings. For more information, see "Managing JMS destinations" on page 37.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - SOA policy lifecycle

Policy management

Policy management plays a key role in enabling policy and governance in any environment, including a service-oriented architecture (SOA). In WSRR, policies are administered using the Business Space user interface.

SOA practices help businesses identify and focus on optimizing the value of its key resources such as services, processes, and information. By adding policies to the SOA, we add points of control and agility for business and IT. This makes SOA more consumable and accelerates the usefulness and adoption of SOA solutions. Policy management is only applicable when policies are abstracted from the resources and enforcement points that they are eventually applied to. Where policies are implemented, enforced, and tightly bound to the resource itself, the agility and flexibility within SOA is limited. Any change to the tightly bound policy requires the resource to also be updated and not just the policy.

A separately authored and maintained policy has the advantage that the context to which it can be applied is not limited; for example, "a transaction must complete in 2 seconds or less". The benefits are:

- The policy can be applied to a variety of transactions, such as a credit card transaction or a price lookup transaction.
- There is the ability to change the policy only once centrally and have this change applied to multiple resources. This is not possible with tightly bound policies.
- The policy says nothing about how or where it gets enforced. This can be configured later if the testing or production environment is subject to change.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - SOA policy lifecycle

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile

Policy usage in the IBM SOA Policy Pattern

Policies in WSRR are administered using the Business Space user interface. Policies can be added, edited, or removed at any time. Policies are selected based on their

PolicySelector property value and governance state. Policies are valid if they are in the Approved, Deprecated, or Superseded governance states, and policies in other governance states are discarded during schedule validation. The Schedule condition is the only condition that this pattern accepts, and the routing action is the only action that is accepted by this pattern.

Accessing Business Space

To access the Business Space user interface to administer policies, see "Connecting to WSRR" on page 27. For more information about the Business Space user interface and managing policies, see WebSphere Service Registry and Repository Version 8.0 Information Center - Using the Business Space user interface.

The PolicySelector property value

Policies are identified in WSRR based on the PolicySelector property value. The PolicySelector property is a customizable string. For example, in the sample data provided with this pattern there are two policies, each with a different PolicySelector value, GID007 and GID008.

Policies with the same PolicySelector value are considered to be different versions of the same policy. If multiple policies exist with the same PolicySelector property value and matching date and time conditions, the policy selected to use is based on the governance state of the policy in the following order of precedence:

1. Approved governance state
2. Superseded governance state
3. Deprecated governance state

Where multiple policies have the same PolicySelector value and same valid governance state, the most recently updated policy is selected.

To assign the PolicySelector value to a new policy, see "Assigning the PolicySelector property for new policies" on page 34.

The Schedule specification

The Schedule element describes the schedule requirements for the days and times when the policy is valid. Here is an example of the policy document schema for the schedule specification:

```
<xs:element name="Schedule">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Daily" maxOccurs="1" minOccurs="0">
     <xs:complexType>
      <xs:attribute name="StartTime" type="xs:time" />
      <xs:attribute name="StopTime" type="xs:time" />
     </xs:complexType>
    </xs:element>
    <xs:element name="WeekDays" maxOccurs="1" minOccurs="0">
     <xs:complexType>
      <xs:attribute name="Days" type="xs:string" />
     </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attribute name="StartDate" type="xs:date" use="optional" />
   <xs:attribute name="StopDate" type="xs:date" use="optional" />
  </xs:complexType>
 </xs:element>
```

The `Schedule` element is the root element of the schedule specification in the policy document. If the schedule element is missing, the schedule starts immediately and continues indefinitely so the policy always applies. If the `Schedule` element is present, the following elements are used for policy validation:

- **Daily** - Specifies the start time, specified in StartTime attribute, and stop time, specified in the StopTime attribute, that the policy applies. If the Daily element is not specified, the policy applies all day beginning at midnight. If the stop time is before or equal to the start time, the condition is considered to span midnight and will still be valid until the stop time that the following morning, even if the following day is entered as the stop date or is not one of the valid WeekDays days.
- **WeekDays** - A String containing the days, from Sunday to Saturday, that the policy can start to apply. The week days listed specify the start time of the policy because schedules can run past midnight. If the WeekDays element is not specified, the policy applies every day of the week.
- **StartDate** - Specifies the date that the policy starts to apply. The date is inclusive; for example, if today is the StartDate date, the policy applies today. If the StartDate element is not specified, the current day is used as the StartDate date.
- **StopDate** - Specifies the date that the policy stops applying. This element contains the date up to which the policy applies. The date is exclusive; for example, if today is the StopDate date, the policy does not apply today. If the stop date is before the start date, the policy never applies. If there is a StartDate element but the StopDate element is not specified, the policy applies indefinitely after the StartDate date.

For more information about policies that span midnight, see the "Policies that span midnight" section.

Policies that span midnight

The policy spans a midnight boundary if the policy StopTime time is earlier than, or equal to, the StartTime time. This means that the policy still applies until the stop time the following day, even if that day is equal to the StopDate date or not one of the specified valid WeekDays days. For example, if a schedule is set to start at 11 p.m. and to run for 2 hours on Wednesdays, the policy will effectively end on Thursday at 1 a.m.

The following examples are some schedules that span midnight:

1. If a schedule contains `<WeekDays Days="Monday"/>` and `<Daily StartTime="22:00:00" StopTime="02:00:00"/>`, this describes one interval that starts on Monday night and ends on Tuesday morning because Monday was specified as the day the policy starts to apply. This will repeat weekly unless dates are specified.
2. The following schedule applies for the last 2 hours of April 1, and the first 2 hours of April 2 because the StartDate and StopDate dates have specified that the policy starts on April 1 and stops on April 2:

```
<Schedule StartDate="2012-04-01" StopDate="2012-04-02">
    <Daily StartTime="22:00:00" StopTime="02:00:00"/>
 </Schedule>
```

Related concepts:

"Samples" on page 13
Sample policies and applications are provided with the default pattern that configure sample policies and sample JNDI destinations.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - SOA policy lifecycle

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Using the Business Space user interface

The SOA policy lifecycle

The SOA policy lifecycle is used to govern a policy from being initially identified, through to being deployed in production, and, finally, deprecated when it is no longer required.

When a policy has been created in WSRR, the policy gets initiated into the SOA policy lifecycle by default and is placed in the Identified governance state. For more information about the governance lifecycle states, including a diagram of the lifecycle and the transitions that moves the policy forward to each state, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - SOA policy lifecycle. The policy can also be transitioned back to a previous governance state to allow for revision of the policy.

A policy can be in one of the following states:

- Identified
- Specification
- Review
- Approved
- Superceded
- Deprecated
- Retired

Even though all these governance states are valid states, when it comes to the IBM SOA Policy Pattern, the following are the valid governance states in which a policy is enforced:

1. Approved
2. Superceded
3. Deprecated

Selection rules for determining which policy gets enforced

Any policy that is not in one of the valid states (Approved, Superceded, Deprecated) won't get enforced by the WebSphere Message Broker message flow. If multiple valid policies are retrieved from WSRR for a particular schedule condition, the following selection rules are applied:

1. The governance state has the following order of precedence:
   - Approved
   - Superceded
   - Deprecated

2. If more than one valid policy has the same highest precedence based on governance state, the policies are sorted in ascending order of policy names and the first policy is selected.
3. If more than one valid policy share the same name and the same governance state, the policy that was updated most recently gets enforced.

Authoring new policies

When authoring policies in the Business Space user interface, enter the schedule conditions and an endpoint to route the message to.

Before you begin

Access the Business Space. For information about accessing Business Space, see "Connecting to WSRR" on page 27.

The Operations space must be created before policies can be created. If the Operations space has not been created, see "Configuring Business Space for the first use" on page 28 and follow the steps to create the space.

About this task

Author new policies using the Operations space. When you have finished authoring the new policies, the PolicySelector property value must be entered for each policy.

Procedure

1. Open the Operations space:
   a. Click **Go To Spaces**. The Go To Spaces dialog is displayed.
   b. Click on the space for Operations users. The specific name will depend on what was specified when the space was created.
2. On the Overview tab, click **Create a Mediation Policy**.
3. Enter a meaningful name, and an optional description.
4. Click **Add Schedule condition**. You can choose to specify one or more of the schedule condition options:
   - The start date.
   - The end date.
   - Specific days of the week.
   - Specific hours of the day.

   Use the calendar and clock icons to specify dates and times. For more information about the Schedule condition in this pattern, see the Schedule specification section in "Policy usage in the IBM SOA Policy Pattern" on page 29.

   Note: Conditions other than the scheduling are not supported in this pattern.
5. Specify the routing action if the conditions are true:
   a. Under Actions If All Conditions are True, click **Add Action**.
   b. Select **Route Message**, and click **Add**.

      Note: Actions other than message routing are not supported in this pattern.
   c. Specify an endpoint. This is the target JMS endpoint you wish a message to go to, if this policy is selected and schedule condition applies.
6. Click **Finish**.

Results

The policy is created and is stored in WSRR. To view the policy document for the policy you just created, select the policy document in the Service Registry Navigator Widget in the lower-left of the screen. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget on the right.

What to do next

When you have finished creating your policies, assign the PolicySelector property a value for each policy. For more information, see "Assigning the PolicySelector property for new policies."

Assigning the PolicySelector property for new policies

The PolicySelector property value in a policy document is used to determine which policy to apply to a message. This value must be manually specified for each new policy.

About this task

To determine which policies should be evaluated for a particular message, a property called PolicySelector exists on all policy documents. By setting this property to a value that matches the value within the message itself, one or more policies can be associated with a message. For all policies that have the PolicySelector property set to the value in the message, the schedule condition in the policies is evaluated to determine which policy should be enforced. For more details about the scheduling condition within this pattern, see "Policy usage in the IBM SOA Policy Pattern" on page 29.

Procedure

1. Open the policy document for the policy. To do this, select the policy document in the Service Registry Navigator Widget in the lower-left of the screen. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget on the right.
2. To edit the policy, click the **Edit** icon in the Service Registry Detail widget.
3. Enter a value in the text box for **PolicySelector**. This is the value that is matched against the value in the message in order to select which policies to apply to it.
4. Click **Finish**.

Results

Messages containing the PolicySelector value you entered for this policy can now have this policy applied when this policy is in a valid governance state.

Editing policies

If you want to change a policy, you can edit an existing one. Policies can be edited or removed using the Business Space user interface.

Before you begin

Open the policy document for the policy. To open the policy document, select the policy document in the Service Registry Navigator Widget, in the lower-left of the

screen. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget.

Note: The Identified governance states is the only state that allows the policy to be deleted. If the policy is not in the Identified governance state, it must be set to this state. See "Editing and deleting policies" on page 36.

Procedure

To change the schedule or the routing destination of a policy in the Identified governance state:

1. Click the **Edit** icon in this widget to edit the policy document. A window is displayed with options to edit the policy details.
   a. If the policy has a schedule condition, the schedule condition is displayed. You can add, edit, or remove the date, day, and time values.

      Note: Conditions other than the scheduling are not supported in the IBM SOA Policy Pattern.
   b. The message routing action is displayed, and the **Route Message** box has an Endpoint value. You can add a new endpoint or edit the existing endpoint. The value for the endpoint cannot be blank and must be a valid endpoint location.

      Note: Actions other than message routing are not supported in the SOA Policy pattern.
2. Click **Finish** to save and close the policy editor.

Results

The Service Registry Detail widget refreshes to show the changes that you made.

"Editing and deleting policies" on page 36
Policies can be edited or removed using the Business Space user interface.

"Deleting policies"
If you want to remove a policy, you can delete it. Policies can be edited or removed using the Business Space user interface.

Deleting policies

If you want to remove a policy, you can delete it. Policies can be edited or removed using the Business Space user interface.

Before you begin

Open the policy document for the policy. To open the policy document, select the policy document in the Service Registry Navigator Widget, in the lower-left of the screen. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget.

Note: The Identified or Retired governance states are the only states that allow the policy to be deleted. If the policy is not in the Identified or Retired governance state, it must be set to one of these states. See "Editing and deleting policies" on page 36.

Procedure

1. Click **Action** > **Delete**. The Delete option is listed in the menu.
2. Select **Delete** to delete the policy.

Related tasks:

"Editing and deleting policies"
Policies can be edited or removed using the Business Space user interface.

"Editing policies" on page 34
If you want to change a policy, you can edit an existing one. Policies can be edited or removed using the Business Space user interface.

Editing and deleting policies

Policies can be edited or removed using the Business Space user interface.

Draft comment

Editors suggest splitting this into 2 x topics: Editing and Deleting

Procedure

1. To open the policy document for the policy, select the policy document in the Service Registry Navigator Widget in the lower-left of the screen. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget on the right.
2. To change the schedule or the routing destination of a policy in the Identified governance state:

   Note: The Identified governance state is the only state that allows for the policy to be edited. If the policy is not in the Identified governance state, it must be set to this state. See "Editing and deleting policies."

   a. Click the **Edit** icon in this widget to edit the policy document. A window is displayed with options to edit the policy details.
   b. If the policy has a schedule condition, the schedule condition is displayed. You can add, edit, or remove the date, day, and time values.

      Note: Conditions other than the scheduling are not supported in the SOA Policy pattern.

   c. The message routing action is displayed, and the **Route Message** box has an Endpoint value. You can add a new endpoint or edit the existing endpoint. The value for the endpoint cannot be blank and must be a valid endpoint location.

      Note: Actions other than message routing are not supported in the SOA Policy pattern.

   d. Click **Finish** to save and close the policy editor. The Service Registry Detail widget refreshes to show the changes that you made.
3. To delete the policy:
   a. If the policy is not in the Identified or Retired governance states, transition it to one of these states. For more information about transitioning a policy through the SOA Policy Lifecycle, see "Editing and deleting policies."
   b. Click **Action** > **Delete**. The Delete option is listed in the menu.
   c. Select **Delete** to delete the policy.
   d. Click **Yes** to confirm the deletion.

Managing JMS destinations

The policy administrator can define additional routing JMS endpoints to be used by new policies, but the JMS route message endpoint mentioned in the policies must also be defined on the WebSphere Message Broker system.

When a new policy is created in WSRR, the new JNDI destination binding definitions must be created for the Route Message Endpoint value set in the policy document. The new JMS destination details must be merged with the existing JMS destinations details in the JMS definition file, `JMS.def`, that was created when the pattern was instantiated. This means that administrators must take the existing JMS definition file, add the new JMS destination definitions, and regenerate the JNDI bindings file, `.bindings`, used by Message Broker to connect to the JMS provider.

Creating JMS destinations

You can create new JNDI destination definitions for the new Routing message endpoint JMS destination with the "jndi://<*DESTINATION*>" format.

Before you begin

Connect to the WebSphere MQ system. For more information, see "Connecting to the WebSphere MQ system" on page 26

Procedure

To create a new JNDI destination, complete the following steps:

1. Create the necessary WebSphere MQ destination queues for the new Routing message Endpoint JMS Destination. Run the WebSphere MW runmqsc command to define a local queue, for example:

```
$runmqsc MB8QMGR
DEFINE QL(<Queue Name>)
END
```

   Where `MB8QMGR` is the queue manager name used in this pattern, and *<Queue Name>* is the MQ destination queue name.

2. Add the JNDI definition for new Routing message Endpoint destination.
   a. Edit the `JMS.def` file located in `/home/virtuser/soapolicyjmsdef` by adding the new JNDI definition. In the following example, a new JNDI definition is added for the JMS Routing message Endpoint "jndi://<*DESTINATION*>" along with existing mandatory JNDI definitions:

```
$vi /home/virtuser/soapolicyjmsdef/JMS.def

# Define a QueueConnectionFactory
# Only parameters being overridden from their default values
# are specified.
# This sets up a MQ client binding.

DEF QCF(QCF) +
TRANSPORT(CLIENT) +
QMANAGER(MB8QMGR) +
HOSTNAME(127.0.0.1) +
PORT(2414)

#

DEF Q(REQUEST_IN) +
QUEUE(REQUEST_INQ) +
QMANAGER(MB8QMGR)
```

```
DEF Q(REPLY_OUT) +
QUEUE(REPLY_OUTQ) +
QMANAGER(MB8QMGR)

DEF Q(BACKOUT) +
QUEUE(SYSTEM.DEAD.LETTER.QUEUE) +
QMANAGER(MB8QMGR)

# Add new JNDI definition for Route Message Endpoint value
# jndi://<DESTINATION> set in the new policy doc
# Replacing <DESTINATION> and <Destination MQ QueueName>
# values with their actual values.

DEF Q(<DESTINATION>) +
QUEUE(<Destination MQ QueueName>) +
QMANAGER(MB8QMGR)

END
```

b. Save and close the JMS.def file.

c. Run the following command to create the bindings definition:

```
$/opt/mqm/java/bin/JMSAdmin < /home/virtuser/soapolicyjmsdef/JMS.def
```

This creates the JNDI bindings definition file in /home/virtuser/JNDI-DIR/.bindings.

What to do next

Share the JMS connection information with external clients. For more information, see "Sharing JMS connection information with external clients"

Sharing JMS connection information with external clients

After you have modified the JMS definitions file, JMS.def, the bindings file, .bindings, must be regenerated following a change and distributed to external clients.

1. Open the JMS.def file located in /home/virtuser/soapolicyjmsdef/ and update the default (127.0.0.1) text, written as <*Broker System Hostname/IPAddress*> in the following example, with the hostname of the Message Broker system:

```
$vi JMS.def

# Define a QueueConnectionFactory
# Only parameters being overridden from their default values
# are specified.
# This sets up a MQ client binding.

DEF QCF(QCF) +
TRANSPORT(CLIENT) +
QMANAGER(MB8QMGR) +
HOSTNAME(<Broker System Hostname/IPAddress>) +
PORT(2414)

#

DEF Q(REQUEST_IN) +
QUEUE(REQUEST_INQ) +
QMANAGER(MB8QMGR)

DEF Q(REPLY_OUT) +
QUEUE(REPLY_OUTQ) +
QMANAGER(MB8QMGR)

DEF Q(BACKOUT) +
QUEUE(SYSTEM.DEAD.LETTER.QUEUE) +
```

```
QMANAGER(MB8QMGR)

#Add new JNDI definition for jndi://<DESTINATION> routing message
# endpoint by replacing <DESTINATION>

<Destination MQ QueueName> values with actual values.
DEF Q(<DESTINATION>) +
QUEUE(<Destination MQ QueueName>) +
QMANAGER(MB8QMGR)
END
```

2. Run the following command to create the bindings definition file, located in `/home/virtuser/JNDI-DIR/.bindings`, that is used by the external remote MQ JMS client:

```
$cd /home/virtuser/soapolicyjmsdef
$/opt/mqm/java/bin/JMSAdmin < /home/virtuser/soapolicyjmsdef/JMS.def
```

3. The generated `/home/virtuser/JNDI-DIR/.bindings` definition file is used by the remote JMS client to connect to the MQ JMS provider hosted on the Message Broker system.

Chapter 6. Troubleshooting

The troubleshooting process, in general, requires that you isolate and identify a problem, then seek a resolution. Administrators can perform troubleshooting of problems caused during the pattern deployment or in the instance.

Collecting diagnostic information

You can use logs to help to find and resolve problems. Logs are stored on the appliance and can be viewed from the user interface, or they can be downloaded to your local file system.

Procedure

To collect diagnostic information, complete the following steps:

1. View the virtual instances:
   a. Click **Instances** > **Virtual system**.
   b. Select the instance in the list of instances in the Virtual System Instances window.
2. For the WebSphere Message Broker virtual machine:
   a. In the **Virtual machines** section, expand the WebSphere Message Broker virtual machine and inspect for any errors in the **Script Packages** section. If any of the script package have errors, click the log links for **remote_std_out.log** and **remote_std_err.log** next to the script package names.
   b. Log into the WebSphere Message Broker instance and check the WebSphere MQ logs and MQ errors.
   c. Refer the product troubleshooting guides: http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/topic/com.ibm.etools.mft.doc/bu03830_.htm
3. For the WSRR virtual machine:
   a. In the **Virtual machines** section, expand the WSRR virtual machine and inspect for any errors in the **Script Packages** section. If any of the script package have errors, click the log links for **remote_std_out.log** and **remote_std_err.log** next to the script package names.
   b. Log into the WSRR instance and check the server errors.
   c. Refer to the WSRR troubleshooting guides: http://pic.dhe.ibm.com/infocenter/sr/v8r0/topic/com.ibm.sr.doc/cwsr_troubleshootingandsupport.html

Troubleshooting problems with pattern installation

Common failures can occur when installing the pattern.

Procedure

Check the following problems when troubleshooting the pattern installation:

1. Problem: Unable to communicate with the appliance. This happens when the appliance location is not valid. Solution: Check that the appliance name that is specified as part of the -h parameter of the loader is valid.

2. Problem: Invalid username or password. This occurs when the username or password that was supplied to the loader is unable to access the appliance. Solution: Check the username and password provided with the -u and -p parameters respectively.
3. Problem: The loader fails while trying to load and image. Solution: There are multiple reasons for this:
   a. Check that the username and password provided have permission to load the image.
   b. Check that the image is already present in the appliance and that the user has access to it. In both of these cases, contact the administrator of the appliance and request additional privileges on the appliance or request access to the existing images.
4. Problem: The loader fails to connect to the IBM Workload Deployer or IBM PureApplication System due to BSO authentication error. Solution: Authenticate with the IBM Workload Deployer or PureApplication System.
5. Problem: The loader imports the existing WebSphere Message Broker, WSRR, or DB2 builds and then the import fails. Solution: Make sure that the username specified in the loader command -u option has access permission to the already existing imported images on IBM Workload Deployer or IBM PureApplication System.

Troubleshooting problems with deployment

Deployment-time problems could be related to the under lying IBM Workload Deployer or IBM PureApplication System environment; for example, resources being unavailable.

About this task

Most of the configurable parameters have a default value assigned. If any of these default values have been removed, the pattern can't be deployed. The final dialog where the user is prompted for the instance name will have the deploy button disabled if any of the mandatory fields have been left as blank.

Reference the Troubleshooting sections in the IBM Workload Deployer Information Center or IBM PureApplication System Information Center for other problems during deployment.

Procedure

1. Problem: You are unable to deploy the IBM SOA Policy Pattern due to the license not being accepted. Solution: Ensure that the license agreement for the imported WebSphere Message Broker 8.0.0.1 and its related components has been accepted. Ensure that the license agreement of the imported WebSphere Service Registry and Repository 8.0.0.0 and its related components has been accepted. Ensure that the license agreement of the imported DB2 Enterprise 9.7.5.0 and its related components has been accepted.
2. Problem: You are unable to deploy the IBM SOA Policy Pattern due to mandatory parameters being missing. Solution: Ensure that the mandatory parameters that are modified or changed don't have a blank or null value.

Related information:

IBM Workload Deployer Version 3.1 Information Center

IBM PureApplication System Information Center

Troubleshooting problems in the deployed instance

If the troubleshooting problems that you suspect are related to the virtual system deployment, you need to review all the information within the virtual system entry of interest.

Procedure

Complete the following steps to troubleshoot problems in the deployed instance:

1. Click **Instances** > **Virtual system**. Select the instance in the list of instances in the Virtual System Instances window.
2. View the details of the instance:
   a. Check the status of the deployed instance in the **Current status** section.
   b. Check the history of the deployed instance in the **History** section. The history lists the actions that were performed during the deployment of the virtual system, each with a date and timestamp. You can look for errors within the history to assist with problem determination. In addition, the time stamps give you a good idea of how long individual actions took to complete.
   c. In the **Virtual machines** section, expand each of the virtual machines and check for any errors in the script packages. If any of the script package have errors, view the logs, **remote_std_out.log** and **remote_std_err.log**, next to the script package names.
3. Log into each of the deployed instance and manually verify that the required services have been started. If any of the installed products or services have trouble starting or report errors, refer to the individual product troubleshooting guides.

Chapter 7. Maintenance and support

You can perform maintenance functions such as applying interim fixes or updating your licenses.

Adding an emergency fix to the catalog

Interim fixes and fix packs are applied to virtual system instances as emergency fixes. You can add emergency fixes to your catalog to be applied to your virtual images.

Before you begin

You must be assigned the *Create new catalog content* permission or the IBM Workload Deployer Appliance *Administrator* role with full permissions to perform these steps.

About this task

Fixes are provided by IBM or an image provider and must be downloaded. New fixes are downloaded from IBM Fix Central. The fixes are then uploaded to the catalog and can be applied to all the applicable virtual system instances.

Procedure

Complete the following steps to add an emergency fix to your catalog.

1. Locate and download the emergency fix (or fixes) from Fix Central.
2. Optional: You can add multiple interim fixes at once. To add multiple fixes at once, download the compressed files from Fix Central and package them into a single compressed file.
3. From the menu, select **Catalog** > **Emergency Fixes**.
4. Click the add icon in the left panel.
5. Enter a name for the fix to add. Optionally, you can also add a description of the fix you are adding. The fix is shown in the left panel of the Emergency Fixes window and information for the fix is shown in the right panel.
6. Browse to the location where you stored the fix and click **Upload**. For security, only .zip, tgz, and pak files can be uploaded. Red Hat RPM is also supported.
7. Complete the information about the fix. You can grant access to users and supply a severity rating. Use the **Applicable to** field to specify the virtual image or virtual images to which this fix applies.

Results

The emergency fix is in the catalog and available to be applied to virtual system images.

Applying an emergency fix

Interim fixes and fix packs are applied to virtual system instances as emergency fixes. You can apply emergency fixes to your virtual system images.

Before you begin

You must be assigned the all access to the virtual system instance or be assigned the Appliance administration role with full permissions to perform these steps. The virtual system instance must be started for service to be scheduled or applied. The emergency fix must be added to the catalog before it can be applied to a virtual system.

About this task

When you add a new emergency fix, you define the virtual images for which the fix is applicable. The list of fixes available when you schedule a service request is constructed using all the fixes applicable to the virtual image used to create your virtual system instance. If a fix has already been applied to your virtual system, you can see it in the **History** listing and it is not included in the list of available fixes.

Procedure

Complete the following steps to apply an interim fix.

1. Select a virtual system instance to which to apply the fix from the Virtual System Instances window.
2. Click the “Apply service” icon.
3. Optional: Schedule a service request. By default, the fix is applied immediately. To schedule it to be applied at a later time, click **Schedule service** and provide the necessary information.
4. Click **Select service level or fixes**.
5. Click **Apply emergency fixes** to see and select the fix to apply. The emergency fix is applied to all virtual machines in the virtual system instance. The status of the virtual system instance shows that the service has been applied on the virtual system.
6. Check for errors. Check the following files to ensure that no errors occurred during the process of applying the emergency fixes:
   - Remote_std_out.log
   - Remote_std_err.log

   You can access the log files from the Virtual System Instances window.

Chapter 8. Appendices

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
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
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 information 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.

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 for use with this program.

Required cleanup

This book contains information on intended programming interfaces that allow the customer to write programs to obtain the services of the product.

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.

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

Trademarks

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 (www.ibm.com/legal/copytrade.shtml).

Required cleanup

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

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

  User Technologies Department (MP095)
  IBM United Kingdom Laboratories
  Hursley Park
  WINCHESTER,
  Hampshire
  SO21 2JN
  United Kingdom
- By fax:
  - From outside the U.K., after your international access code use 44-1962-816151
  - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.