IBM SOA
Policy Gateway Pattern

Contents

Chapter 1. SOA Policy overview

Policy management plays a key role in governing policies in a structured and consistent manner. Policies can be used to enable better governance in any service-oriented environment.

A policy is an independent element that can be applied to one or many resources, including different services. The assignment of the policy and any associated metadata, especially in a distributed environment, can take place at a variety of enforcement points and decision points.

The SOA Policy architecture

The SOA Policy architecture describes the interaction of the Policy Administration Point (PAP), Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Information Point (PIP), and the Policy Monitoring Point (PMP). In the pattern, the PAP is provided by WSRR, the PEP is provided by WebSphere® DataPower®, and the PMP through the DataPower monitoring component.

The organization of the basic policy architecture and definition of those key points:

- **Policy Administration Point**. Provides policy capabilities for authoring of a policy, management and governance of the policy and its assignment to resources, and administration of the policy results during run time. The PAP includes a repository to store policies. The PAP is provided by WSRR.
- **Policy Enforcement Point**. A Policy Enforcement Point is a functional point that runs on the middleware. It performs the following actions:
  - Enforces policies.
  - Receives enforcement policy updates and makes them ready or translates them for usage.
  - Provides enforcement metrics to the Policy Monitoring Point.
  - Provides enforcement policy results and analytics to the Policy Administration Point and Policy Monitoring Points.
  - Changes the places where policies are applied and enforced depending on the lifecycle stage:
    - During design time, WSRR itself is the point of enforcement.
    - During run time, policies are typically enforced by the underlying intermediary (middleware) system that connects service providers with consumers.

  In this pattern, the PEP is supplied by WebSphere DataPower.
- **Policy Decision Point**. A Policy Decision Point evaluates participant requests against relevant policies or contracts and attributes. The PDP renders an authorization, eligibility, or validation decision to provide calculated results.
- **Policy Information Point**. A Policy Information Point provides external information to the Policy Decision Point, such as LDAP attribute information, or the results from a database, with information that must be evaluated to make a policy decision.

- **Policy Monitoring Point**. A functional component that provides the detailed policy monitoring function for the overall architecture; for example, the overview of the policy in the distributed environment. It performs the following actions:
  - Receiving monitoring policy updates and making them ready or translating them for usage.
  - Capturing the real-time collection and statistics analysis for display.
  - Correlating, analyzing, and visualizing the data that is fed in by the various real-time collectors, including Policy Enforcement Points.
  - A management console that provides visibility into the management of the distributed network of policy enforcement points, and the status of these enforcements.
  - Logging, aggregating measurements, and highlighting significant events as specified by the monitoring policy.
  - Providing monitoring policy analytics to the Policy Administration Point and Policy Enforcement Points.

  In this pattern, the PMP is provided by the DataPower monitoring component.

The consumer and provider both interact with the middleware, which in turn interacts with the repository and any monitoring software.

How the SOA Policy architecture works together

The SOA Policy pattern flow is shown in Figure 1 on page 3.

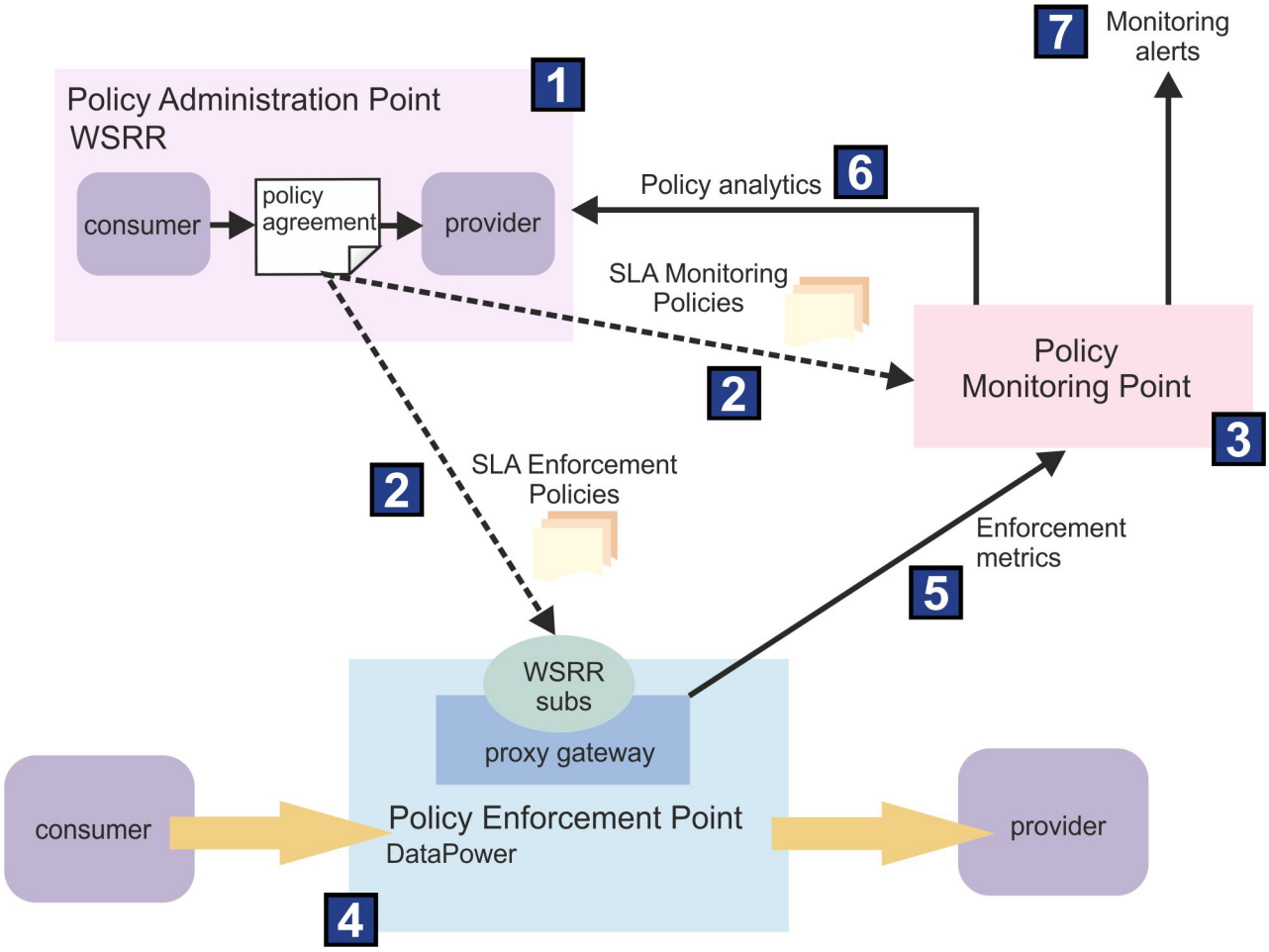


Figure 1. Service Level Agreement (SLA) Policy - the SOA deployment model

1 Policies are authored and then attached to services that require that policy. Typically has the following order:

1. The set of services are loaded or created in the service repository. This action is a part of the Policy Administration Point.
2. The set of policies that are required is created in the Policy Administration Point by using the policy lifecycle:
   - Policies are attached to the services that require those policies – at the service, operation, or endpoint level as required.

2 Automated publish/subscribe of policies from the Policy Administration Point to the Policy Enforcement Points and the Policy Monitoring Point:

1. As a part of the setup, the monitoring service subscribes to the monitoring policy from WSRR. This action occurs only once.
2. As a part of the setup, proxy gateways are created in each WebSphere DataPower appliance (or virtual appliance) that has service transactions with policy enforcement. This action occurs only once, and is added or changed as required.

3. As a part of the setup, each proxy gateway in the appliance subscribes to policies from WSRR for services that it is responsible for. This action occurs only once, and is added or changed as required.
4. As a part of the setup, WebSphere DataPower is configured so that policies can be shared by other appliances in a cluster. This action occurs only once, and is added or changed as required.
5. The Policy Monitoring Point downloads the monitoring policies as they are published.
6. The Policy Monitoring Point converts the policies into the internal representation called situation policies.
7. WebSphere DataPower downloads the WSDLs for services that it is responsible for transacting.
8. WebSphere DataPower downloads the policies for services that it is responsible for when notified by WSRR.
9. WebSphere DataPower converts the policies into internal WebSphere DataPower representation in the form of SLM objects.

Monitoring of SOA policies with reporting and notification of operations:

1. Monitoring policies are active in the Policy Monitoring Point Situation Policy.
2. The Policy Monitoring Point receives monitoring information and places that information in workspaces.

Enforcement of SOA Policies:

1. Enforcement policies are active in the various WebSphere DataPower appliances.
2. WebSphere DataPower receives service transactions and applies policies for that consumer service and provider service.

The Policy Enforcement Point sends SOA Policy Enforcement statistics to the Policy Monitoring Point.

The Policy Monitoring Point sends monitoring events to the Policy Administration Point:

1. Events are set up in the Policy Administration Point that requires monitoring from the Policy Monitoring Point. This action occurs only once, and is added or changed as required.
2. As situation policies evaluate to true, events are pushed to the Policy Authoring Point from the Policy Monitoring Point.

Monitoring of alerts:

- Situation policies run periodically and take operational action as specified in the policy. The default is every 5 minutes.

The SOA Policy lifecycle

Policies are governed by the SOA Policy lifecycle. The lifecycle takes the policy from being initially identified, through to being deployed in production, and, finally, to being deprecated when it is no longer required.

For more information about the lifecycle transitions and states in the SOA Policy lifecycle, see IBM® WebSphere Service Registry and Repository Version 8.0 Information Center - SOA policy lifecycle.

Policy standards

The web technical community groups, W3C and OASIS, created standards to define the policies applicable to web services.

- **WS-Policy**: The Web Services Mediation Policy 1.0 domain defines a set of policy assertions for describing mediation requirements for a service.
- **Web Services Policy 1.5 - Framework**: Defines a framework and a model for expressing policies that refer to domain-specific capabilities, requirements, and general characteristics of entities in a web services-based system.

Examples of specifications that define domain-specific policy assertions:

- WS-MediationPolicy
- WS-SecurityPolicy
- WS-ReliableMessaging and WS-ReliableMessagingPolicy
- WS-SecureConversation
- WS-Security
- WS-Transactions
- WS-Trust

For more information about WS-MediationPolicy, see ftp://public.dhe.ibm.com/software/solutions/soa/pdfs/WSMediationPolicy1.7-20130506.pdf.

The WS-Policy Data Model includes the following entities:

- **Policy**: An unordered collection of “policy alternatives”.
- **Policy Alternative**: A policy alternative is a collection of “policy assertions”.
- **Policy Assertion**: Represents an individual preference; for example, a requirement or a capability.
- **Policy Parameters**: The opaque payload of a “policy assertion”.
- **Policy Subject**: An entity that a policy expression can be bound to. This entity is used in a WS-PolicyAttachment document.

The following example, Figure 2 on page 6, shows a security policy expression that uses assertions defined in WS-Security and WS-SecurityPolicy:

```
(01) <wsp:Policy
   xmlns:sp=http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702
   xmlns:wsp=http://www.w3.org/ns/ws-policy
   xmlns:wsu=http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
   wsu:Id="SecureMessages">  <!-- policy expression -->
(02)   <wsp:ExactlyOne>
(03)     <wsp:All>    <!-- policy alternative #1 -->
(04)       <sp:SignedParts>;   <!-- policy assertion -->
(05)         <sp:Body>   <!-- policy assertion parameter -->
(06)       </sp:SignedParts>
(07)     </wsp:All>
(08)     <wsp:All>    <!-- policy alternative #2 -->
(09)       <sp:EncryptedParts> <!-- policy assertion -->
(10)         <sp:Body/>   <!-- policy assertion parameter -->
(11)       </sp:EncryptedParts>
(12)     </wsp:All>
(13)   </wsp:ExactlyOne>
(14) </wsp:Policy>
```

Lines (03-07) represent one policy alternative for signing a message body.

Lines (08-12) represent a second policy alternative for encrypting a message body.

Lines (02-13) show the `ExactlyOne` policy operator. Policy operators group policy assertions into policy alternatives. A valid interpretation of the policy is that an invocation of a web service either signs or encrypts the message body, but not both.

Figure 2. Use of Web Services Policy with security policy assertions.

Figure 3 shows a policy definition.

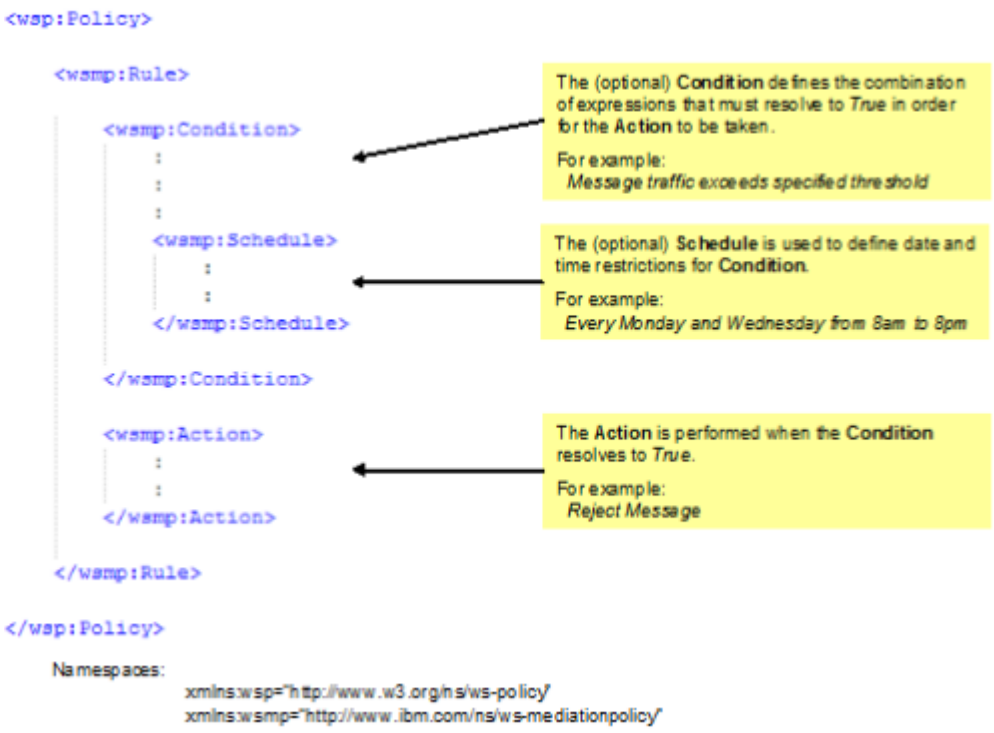


Figure 3. Overview of Policy structure

Policy Attachment

The Policy Attachment Document role is to associate a set of WS-Policy policies with a specific service attachment point for enforcement such as a Web Services attachment point.

For example, the Web Services platforms can support attachment points that are based on:

- WSDL Element URI 1.1 elements
- WS-Addressing elements

The syntax is defined in the WS-PolicyAttachment specification:

```
<wsp:PolicyAttachment>
    <wsp:AppliesTo>


    </wsp:AppliesTo>
    <wsp:Policy>


    </wsp:Policy>
</wsp:PolicyAttachment>
```

Figure 4. WS-PolicyAttachment specification

WSRR exposes REST interfaces to acquire the appropriate policy attachments in an SLA model. Information on the Consumer-Provider pair to which the policy applies is passed to the ESB in WS-PolicyAttachment format. The syntax is defined in the WS-PolicyAttachment: Message Content Filters specification.

The policy can be specified for a provider service only, for a specific consumer-provider pair, or for Anonymous consumers. Anonymous consumers provide a way of defining a default policy that applies only to consumers for which no other policies apply.

In Figure 4, the domain-specific policy subject to which the policy applies (the provider) is contained in the `<wsp:AppliesTo>` section. It is followed by the consumer-context filter to which the policy applies (consumer). Then, in the `<wsp:Policy>` section, the policy, or policies, are declared or referenced.

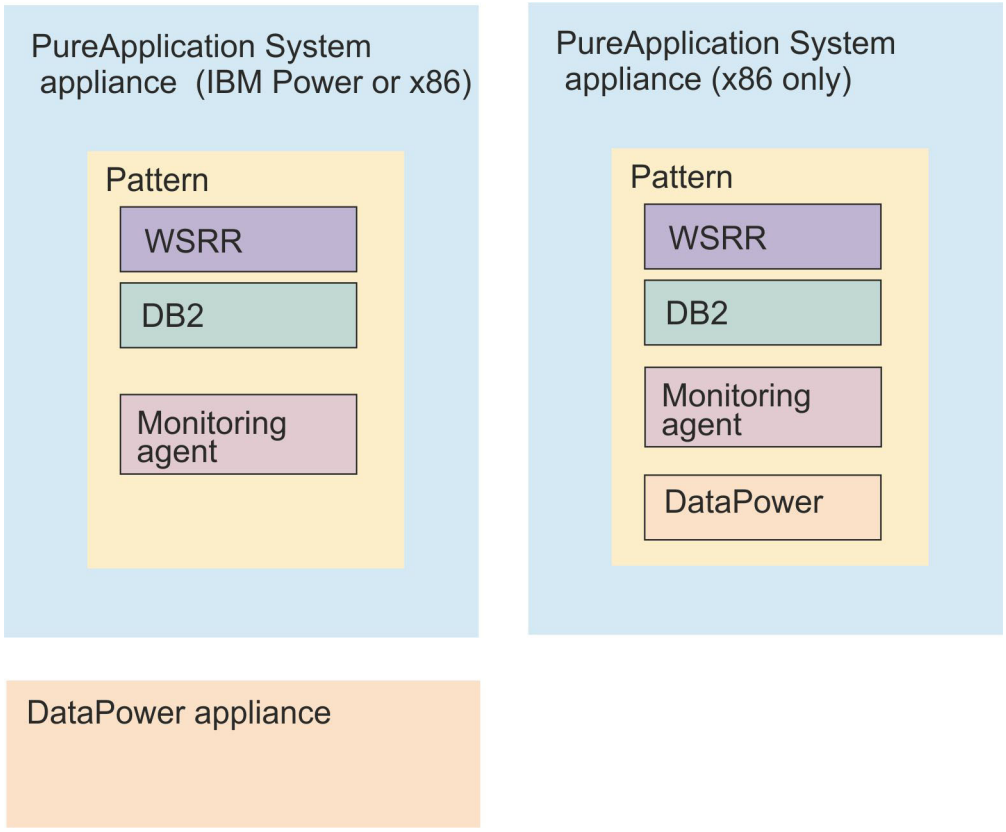
Chapter 2. Pattern overview

The IBM SOA Policy Gateway Pattern is set of virtual system patterns that provide a policy enforcement point, a policy administration point, and a policy monitoring point.

You can install the IBM SOA Policy Gateway Pattern on an IBM PureApplication® System appliance on IBM Power® or x86 architectures.

The policy administration point is provided by virtual system patterns that provision WSRR in a multitier architecture, delivering a production and staging environment. The policy enforcement point can be provided by a WebSphere DataPower appliance. Alternatively, on x86, your PureApplication System can deploy a virtual DataPower image. In either case, a domain is created during virtual system pattern deployment. The policy monitoring point is provided by a monitoring add-on to the PureApplication System monitoring service.

The following diagram illustrates the capabilities that are derived from IBM SOA Policy Gateway Pattern



There are examples of policy in many, if not all Service Oriented Architecture (SOA) environments. Service producers and consumers agree the capabilities, performance, and characteristics of the service during the design phase. To implement these agreements, you can use Service Level Definitions (SLDs) and Service Level Agreements (SLAs). Use the pattern to define policies for SLDs and

SLAs in an efficiently administered, defined, and governed way. Policy types that are used in this pattern include the following policies:

- **Mediation Policies** -
  - Rejection - Reject or throttle requests that arrive at a rate greater than defined.
  - Logging - Create a log message with the policy enforcement point when a service is called.
  - Transformation.
  - Validation - Validate the service call against the service definition.
  - Routing - Based on the message, route to a specific endpoint.
- **Security Policies**: The sample demonstrates enforcement of XACML access control security policies. These policies are not currently governed within the policy administration point.
- **Monitoring Polices**: You can define monitoring policies on PureApplication System deployments.

The IBM SOA Policy Gateway Pattern contains the following virtual system patterns:

- SOA Policy Gateway Basic Runtime Sample (x86 only)
- SOA Policy Gateway Governance Master
- SOA Policy Gateway Basic Runtime
- SOA Policy Gateway Basic Runtime External DataPower
- SOA Policy Gateway Advanced Runtime
- SOA Policy Gateway Advanced Runtime External DataPower
- System Monitoring for SOA Policy Gateway Pattern 2.5 (a shared service)

The virtual system patterns work together to provide a multi-stage services governance environment. The IBM SOA Policy Gateway Pattern also provides the capability to provision multiple DataPower domains that are configured to the governance environment during the pattern deployment.

For more information about SOA Policy, see Chapter 1, “SOA Policy overview,” on page 1.

Related concepts:

Chapter 1, “SOA Policy overview,” on page 1
Policy management plays a key role in governing policies in a structured and consistent manner. Policies can be used to enable better governance in any service-oriented environment.

“SOA Policy Gateway Basic Runtime External DataPower” on page 23
The SOA Policy Gateway Basic Runtime External DataPower pattern is the same as the Basic Runtime pattern, but requires that external DataPower appliances are specified on deployment.

“SOA Policy Gateway Basic Runtime Sample (x86)” on page 19
The SOA Policy Gateway Basic Runtime Sample provisions a basic runtime pattern with a sample interface and application that demonstrates the policies that are currently supported in this release.

“SOA Policy Gateway Governance Master” on page 20
The SOA Policy Gateway Governance Master pattern provides a clustered governance environment for authoring and managing services and policies. The environment is provisioned with the WSRR default Governance Enablement Profile configured. The default Governance Enablement Profile supports two promotion targets, Staging and Production.

"SOA Policy Gateway Advanced Runtime External DataPower" on page 27
The SOA Policy Gateway Advanced Runtime External DataPower is the same as the Advanced Runtime pattern, but requires that external DataPower appliances are specified on deployment.

"System Monitoring for SOA Policy Gateway" on page 29
The System Monitoring for SOA Policy Gateway shared service provides the monitoring components for the SOA Policy Gateway.

Chapter 3. Getting started with the IBM SOA Policy Gateway Pattern

This pattern uses WebSphere DataPower to control messages by using governed polices and service definitions in WSRR. Review the topics in this section to understand how to download and install the pattern, how to verify the pattern after installation, accept licenses, and the user roles involved.

Downloading and installing the patterns

The IBM SOA Policy Gateway Pattern for use with IBM PureApplication System is packaged for download from Passport Advantage®.

Before you begin

You download the IBM SOA Policy Gateway Pattern to an interim system, which can be a Linux or Microsoft Windows system. You then run the installer on the interim system to install the patterns on the IBM PureApplication System.

Ensure that there is 16 GB of space available for the `CIQ1LML.tar.gz` file (Power target) or `CIQ1VML.tar.gz` file (x86 target), and an extra 40 GB for the extracted files. Java™ Runtime Environment (JRE) Version 6 must also be installed before starting the pattern installation. You can download the JRE for Linux from the following address: http://www.ibm.com/developerworks/java/jdk/linux/download.html

About this task

The IBM SOA Policy Gateway Pattern is packaged in the `CIQ1LML.tar.gz` file for a Power target system, or the`CIQ1VML.tar.gz` file for an x86 target system. This archive contains the open virtual archive (OVA) files, script package files, and pattern definition files.

Procedure

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

1. Access the Passport Advantage website: Passport Advantage.
2. Download the archive file that contains the images, script packages, and patterns to use. The file is named `CIQ1LML.tar.gz` (Power target) or `CIQ1VML.tar.gz` (x86 target).
3. Open a terminal on Linux, or a command prompt window on Windows, and navigate to the directory where the archive file was downloaded.
4. Extract the contents of the archive file to your local file system. On Linux, the following extract command is used:

```
tar xvzf archive_file
```

   On Windows, use extra archive extraction software to extract the contents of the archive file.
5. Change to the `installer` directory:

```
cd installer
```

6. To install the IBM SOA Policy Gateway Pattern into the IBM PureApplication System, run the installer. 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 `<host>` is the IBM PureApplication System, and username and password are the Cloud Administrator credentials. For example:

```
./installer -h drivensnow.hillesden.ibm.com -u cbadmin -p cbadmin
```

7. When prompted, accept the IBM SOA Policy Gateway Pattern license.
   a. On Microsoft Windows: after accepting the license agreement, if a new line in the terminal displays >>>, type `quit()` and press the Enter key. Repeat step 7.
8. The patterns are imported. As each pattern is installed, a message is displayed in the installer to indicate it installed successfully. For example:

```
Importing pattern "SOA Policy Gateway 2.5.0.0 - Governance Master" ...
Import pattern "SOA Policy Gateway 2.5.0.0 - Governance Master" successfully.
```

Results

The patterns and scripts are loaded and the Virtual System patterns are created.

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

What to do next

Accept licenses in IBM PureApplication System, see .

To validate the installation, see "Verifying the installed pattern."

Verifying the installed pattern

You can verify that the pattern is successfully installed.

Before you begin

Ensure that all steps from "Downloading and installing the patterns" on page 13 are completed.

About this task

After you install the pattern, you can verify the pattern installation to ensure all parts are successfully installed.

Procedure

To verify the installation of the IBM SOA Policy Gateway Pattern, complete the following steps:

1. Open the Workload console on the appliance where the pattern was installed.
2. Verify the Virtual Images by navigating to **Catalog** > **Virtual Images** and locate the following items:
   - DB2® Enterprise 10.1.0.2
   - WebSphere Service Registry and Repository 8.0.0.2
   - WebSphere DataPower X152 Virtual Edition (x86 systems only)

3. Navigate to **Catalog** > **Script Packages**, and locate:
   - SOA Policy Gateway 2.5.0.0 - DataPower Domain
   - SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)
   - SOA Policy Gateway 2.5.0.0 - External DataPower Monitoring
   - SOA Policy Gateway 2.5.0.0 - Promotion
   - SOA Policy Gateway 2.5.0.0 - Sample (x86 only)
   - SOA Policy Gateway 2.5.0.0 - Security
   - SOA Policy Gateway 2.5.0.0 - Add_Named_Queries
   - SOA Policy Gateway 2.5.0.0 - Tear Down

   These script packages are all present in a successful installation.
4. Navigate to **Patterns** > **Virtual Systems**. On x86 systems, locate:
   - SOA Policy Gateway 2.5.0.0 - Advanced Runtime
   - SOA Policy Gateway 2.5.0.0 - Advanced Runtime External DataPower
   - SOA Policy Gateway 2.5.0.0 - Basic Runtime
   - SOA Policy Gateway 2.5.0.0 - Basic Runtime External DataPower
   - SOA Policy Gateway 2.5.0.0 - Basic Runtime Sample
   - SOA Policy Gateway 2.5.0.0 - Governance Master

   On Power Systems, locate:
   - SOA Policy Gateway 2.5.0.0 - Advanced Runtime
   - SOA Policy Gateway 2.5.0.0 - Basic Runtime
   - SOA Policy Gateway 2.5.0.0 - Governance Master

   These patterns are all present in a successful installation.
5. Navigate to **Cloud** > **Pattern Types** and locate the following item:
   - System Monitoring for SOA Policy Gateway Pattern 2.5.0.0

   This pattern is present in a successful installation.

Results

You verified the installation of the IBM SOA Policy Gateway Pattern.

What to do next

If you have a successful installation, you can go on to accept licenses, see "Accepting licenses." If your install was not successful, repeat step 7 onwards of the topic "Downloading and installing the patterns" on page 13.

Accepting licenses

You must accept licenses for newly-installed parts before you can work with the patterns.

Before you begin

Ensure that all steps from "Downloading and installing the patterns" on page 13 are completed.

About this task

Before any virtual image can be used, you must accept the required license for it.

Procedure

To accept licenses, complete the following steps:

1. Open the Workload console on the appliance where the pattern was installed.
2. Select **Catalog** > **Virtual Images**.
3. Locate the following images in the **Virtual Images** list and confirm that the license has been accepted in the details pane, if not click 'accept' to view and accept the license. For x86 systems:
   - WebSphere DataPower XI52 Virtual Edition, Version 6.0.0.0 - Image reference number: XI52.6.0.0.0231528 (2013/06/16 14:14:19)
   - WebSphere Service Registry and Repository 8.0.0.2 - Image reference number: 201309062038
   - DB2 Enterprise 10.1.0.2 - Image reference number: 39
   - IBM OS Image for Red Hat Linux Systems, version 2.0.0.3 - Image reference number: 136

   For Power systems:
   - WebSphere Service Registry and Repository 8.0.0.2 - Image reference number: 201309080001
   - DB2 Enterprise 10.1.0.2 - Image reference number: 50
   - IBM OS Image for AIX® Systems version 2.0.0.2 - Image reference number: 126
4. To accept a license, click the image to view its details. The status is displayed. Click **accept** for the License Agreement, and then click any of the licenses that must be accepted before the virtual image can be used. The status displays **Read-only** and the License agreement displays **Accepted** when complete. If a license is not accepted, the image icon contains a red box with a cross.

Results

You accepted the licenses for the IBM SOA Policy Gateway Pattern.

What to do next

If you have a successful installation, and have accepted all licenses, you can go on to work with the pattern, see Chapter 5, “Working with the IBM SOA Policy Gateway Pattern,” on page 45. If your installation was not successful, repeat step 7 onwards of the topic “Downloading and installing the patterns” on page 13.

Configuring user access

To enable users to access the images and patterns on the appliance, the appliance administrator must first allow 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 appliance administrator, can add other users to access and administer the patterns. They do this using the system console.

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 console.
     a. From the menu click **System** > **Users**.
     b. Click the **Add** icon.
     c. Provide a short user name as well as the actual name of the user, email address, and passwords and click **OK**.
     d. Select the user that 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 that 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 already added the virtual images, provide access for the users or group to the virtual images. Switch to the workload console and click **Patterns** > **Virtual systems** to open the Virtual system patterns window. Select an IBM SOA Policy Gateway Pattern virtual image to display its details. Add the users or group in the **Access granted to** field.

What to do next

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

Related information:

IBM PureApplication System: Managing users and groups

Chapter 4. Patterns, parts, and script packages

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

Patterns describe the function that is 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 WSRR part is put into a pattern, which is then deployed, the result is a virtual machine that has a running WSRR instance.

Patterns

When the virtual images are loaded into IBM PureApplication System, and the access is assigned to the users, users can begin to work with the patterns.

Patterns provide a repeatable topology that can be deployed to a cloud. Deployed patterns are virtual systems that run 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.

SOA Policy Gateway Basic Runtime Sample (x86)

The SOA Policy Gateway Basic Runtime Sample provisions a basic runtime pattern with a sample interface and application that demonstrates the policies that are currently supported in this release.

The SOA Policy Gateway Basic Runtime Sample pattern is only available on x86 systems.

The SOA Policy Gateway Basic Runtime Sample pattern has the following parts:

- WSRR Standalone server
- DB2 Enterprise
- DataPower

The SOA Policy Gateway Basic Runtime Sample pattern installs a sample application in the deployed environment. The pattern installs a sample domain within DataPower that implements a sample service, installs sample WSDL and attached policies in WSRR for the service, and provides a test application to demonstrate the enforced policies. For more information about the sample application, see “The sample application” on page 57. It installs a sample domain within DataPower, installs sample WSDL and Policies in WSRR, and demonstrates multiple policies against a service.

The following diagram shows the basic runtime sample.

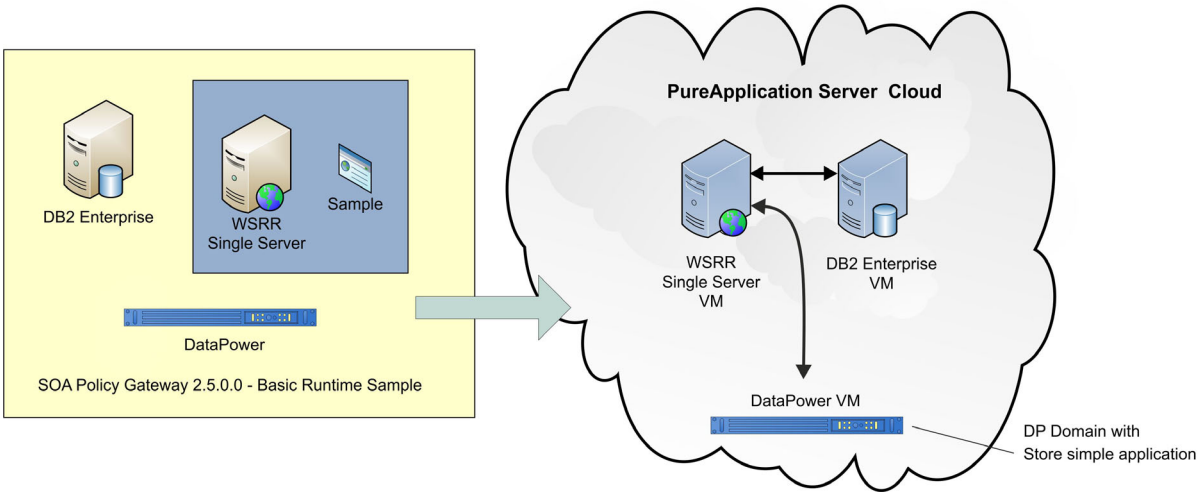


Figure 5. PureApplication Server configuration with DataPower VM (x86 only)

Policies that are implemented include:

Table 1. Policies included in the Basic Runtime with Sample pattern

Policy type	Description
Logging	Based on a requests context ID, logs the request in DataPower.
Routing	Based on a requests context ID, routes the request to a specified endpoint.
Validation	Validates the request against the service implementations WSDL.
Rejection	Controls requests to a service based on the message count with actions: reject, queue, and others.
Security AAA	Control access to the service by using XACML-based user authorization. The XACML is not stored in WSRR.
Security Redaction	Redacts parts of the response message that is based on XACML. The XACML is not stored in WSRR.

Scripts and advanced options

The pattern requires the following scripts.

On the WSRR Standalone server part:

- SOA Policy Gateway 2.5.0.0 - Sample

View the part and script parameters:

- "DB2 Enterprise part" on page 29
- "WSRR Standalone server part" on page 34
- "DataPower part" on page 37
- "Script: SOA Policy Gateway 2.5.0.0 - Sample" on page 40

SOA Policy Gateway Governance Master

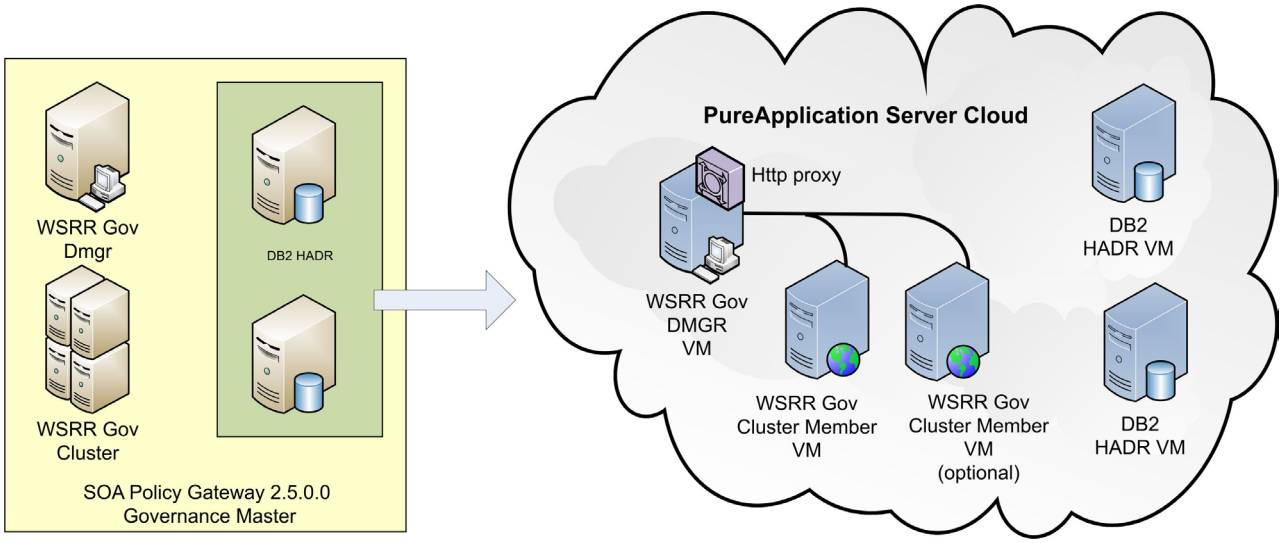
The SOA Policy Gateway Governance Master pattern provides a clustered governance environment for authoring and managing services and policies. The

environment is provisioned with the WSRR default Governance Enablement Profile configured. The default Governance Enablement Profile supports two promotion targets, Staging and Production.

The SOA Policy Gateway Governance Master pattern requires the following parts:

- DB2 HADR Primary
- DB2 HADR Standby
- WSRR Deployment manager
- WSRR Custom nodes

Note: The Governance Master pattern must be deployed before the runtime patterns are deployed. Parameters that are used to configure the Governance Master pattern are used by the runtime patterns to configure itself with the Governance Master.



Part parameters

View the part parameters:

- "DB2 Enterprise HADR Primary part" on page 31
- "DB2 Enterprise HADR Standby part" on page 33
- "WSRR Deployment manager part" on page 35
- "WSRR Custom nodes part" on page 36
- "Script: SOA Policy Gateway 2.5.0.0 - Security" on page 41
- "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39

Using the Governance pattern as a governance master

The SOA Policy Gateway Governance Master pattern is deployed with the default WSRR Governance Enablement Profile that includes two promotion stages, Staging and Production. For more information about the Governance Enablement Profile in WSRR, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile. The basic runtime or advanced runtime patterns can be deployed into this integration as promotion targets. For more information about how to configure promotion targets, see "Adding an additional runtime environment" on page 54.

Related information:

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

SOA Policy Gateway Basic Runtime

The SOA Policy Gateway Basic Runtime pattern is the simplest means to provide a SOA Policy Gateway runtime, it includes two DataPower instances (x86 only), a standalone WSRR instance, a standalone DB2 instance, and a Base OS instance (for hosting the DataPower monitoring agents).

Note: This topic describes the pattern available on x86. For the IBM Power pattern, see "SOA Policy Gateway Basic Runtime External DataPower" on page 23.

The SOA Policy Gateway Basic Runtime pattern requires the following parts:

- WSRR Standalone server
- DB2 Enterprise
- WebSphere DataPower X152 Virtual Edition
- SOA monitoring for DataPower (in a Core OS part)

The following diagram shows the configuration of the SOA Policy Gateway Basic Runtime pattern.

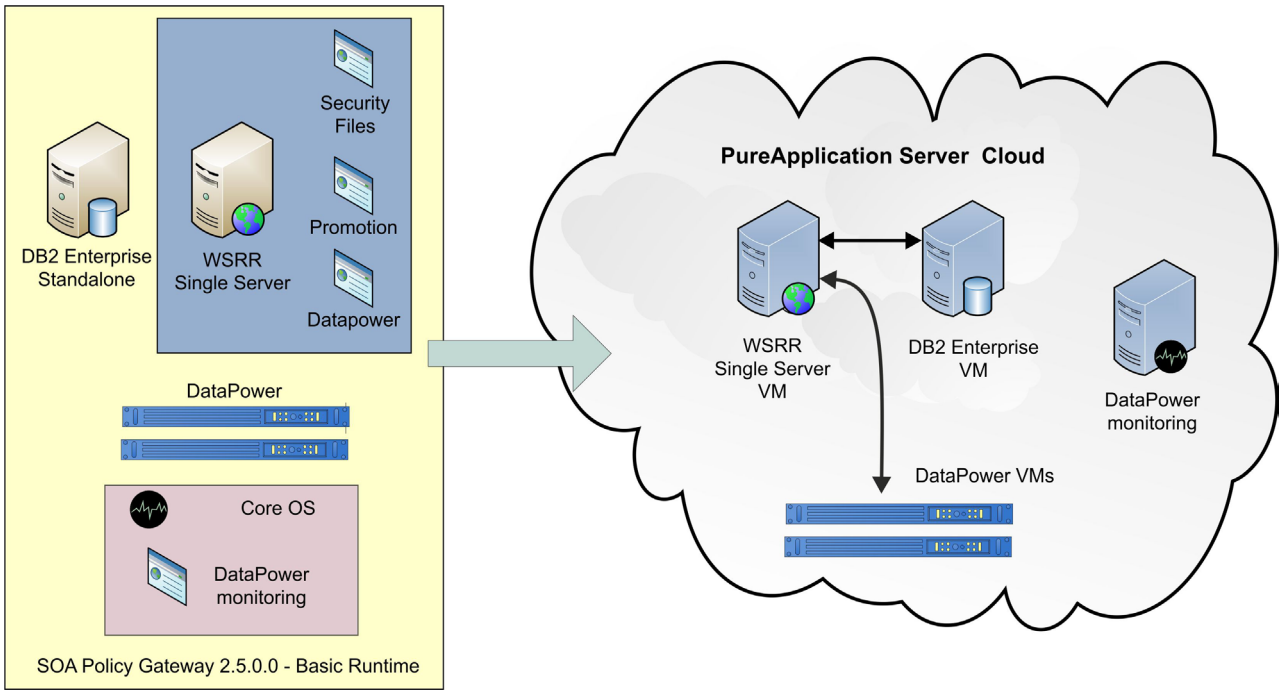


Figure 6. PureApplication Server configuration with DataPower VM

Scripts and advanced options

The pattern requires user input to the following scripts at deploy time.

On the WSRR Standalone server part:

- SOA Policy Gateway 2.5.0.0 - Security
- SOA Policy Gateway 2.5.0.0 - Promotion

- SOA Policy Gateway 2.5.0.0 - DataPower Domain

On the Core OS part:
- SOA Policy Gateway 2.5.0.0 - DataPower Monitoring

View the part and script parameters:
- "WSRR Standalone server part" on page 34
- "DB2 Enterprise part" on page 29
- "DataPower part" on page 37
- "Script: SOA Policy Gateway 2.5.0.0 - Security" on page 41
- "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain" on page 38
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)" on page 41

Configuring the basic runtime with a governance master

When a basic runtime pattern is configured with a governance master pattern the following occurs:
- Cross-cell security is configured
- The `promotion.xml` file on the governance master is updated with the deployment data for the basic runtime deployment.

To configure promotion, you must choose one of the following stage options:
- production
- staging

These options align with the levels provided by the Governance Enablement Profile in WSRR. For more information about the Governance Enablement Profile in WSRR, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile.

Note: You can use this pattern to provision a standalone system, with no governance master. To do this you specify the governance master parameters as "Unset" when deploying the pattern. These settings cause the promotion script to generate an error during deployment, and the deployment shows as **failed**, but you can ignore the error.

SOA Policy Gateway Basic Runtime External DataPower

The SOA Policy Gateway Basic Runtime External DataPower pattern is the same as the Basic Runtime pattern, but requires that external DataPower appliances are specified on deployment.

Note: This description applies to the pattern on IBM Power systems.

The SOA Policy Gateway Basic Runtime External DataPower pattern has the following parts:
- WSRR Standalone server
- DB2 Enterprise
- SOA monitoring for DataPower (in a Core OS part)

The following diagram shows the configuration of the SOA Policy Gateway Basic Runtime External DataPower pattern.

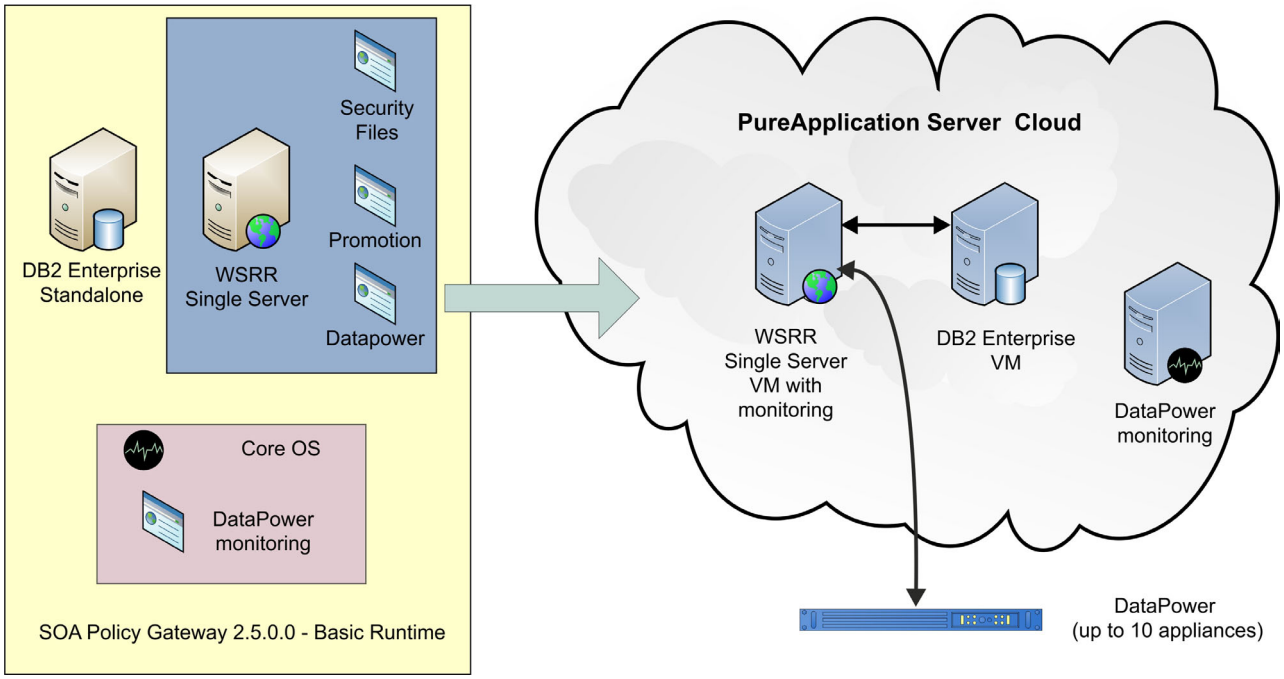


Figure 7. PureApplication Server configuration with DataPower appliance

Scripts and advanced options

The pattern requires user input to the following scripts at deploy time.

On the WSRR Standalone server part:

- SOA Policy Gateway 2.5.0.0 - Security
- SOA Policy Gateway 2.5.0.0 - Promotion
- SOA Policy Gateway 2.5.0.0 - DataPower Domain

On the Core OS part:

- SOA Policy Gateway 2.5.0.0 - DataPower Monitoring

View the part and script parameters:

- "WSRR Standalone server part" on page 34
- "DB2 Enterprise part" on page 29
- "Script: SOA Policy Gateway 2.5.0.0 - Security" on page 41
- "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain" on page 38
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)" on page 41

Configuring the basic runtime with a governance master

When a basic runtime pattern is configured with a governance master pattern the following occurs:

- Cross-cell security is configured
- The `promotion.xml` file on the governance master is updated with the deployment data for the basic runtime deployment.

To configure promotion, you must choose one of the following stage options:

- production
- staging

These options align with the levels provided by the Governance Enablement Profile in WSRR. If the governance profile differs, "other" is chosen when governance masters governance profile is changed. For more information about the Governance Enablement Profile in WSRR, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile.

Note: You can use this pattern to provision a standalone system, with no governance master. To do this you specify the governance master parameters as "Unset" when deploying the pattern. These settings cause the promotion script to generate an error during deployment, and the deployment shows as **failed**, but you can ignore the error.

SOA Policy Gateway Advanced Runtime

The SOA Policy Gateway Advanced Runtime includes two DB2 server instances in an HADR configuration, and a WSRR cluster with a single Deployment Manager and two Custom Nodes.

Note: This topic describes the pattern available on x86. For the IBM Power pattern, see "SOA Policy Gateway Advanced Runtime External DataPower" on page 27.

The pattern requires the following parts:

- WSRR Deployment manager
- WSRR Custom nodes
- DB2 HADR Primary
- DB2 HADR Standby
- WebSphere DataPower X152 Virtual Edition
- SOA monitoring for DataPower (in a Core OS part)

The following diagram shows the configuration of an advanced runtime system.

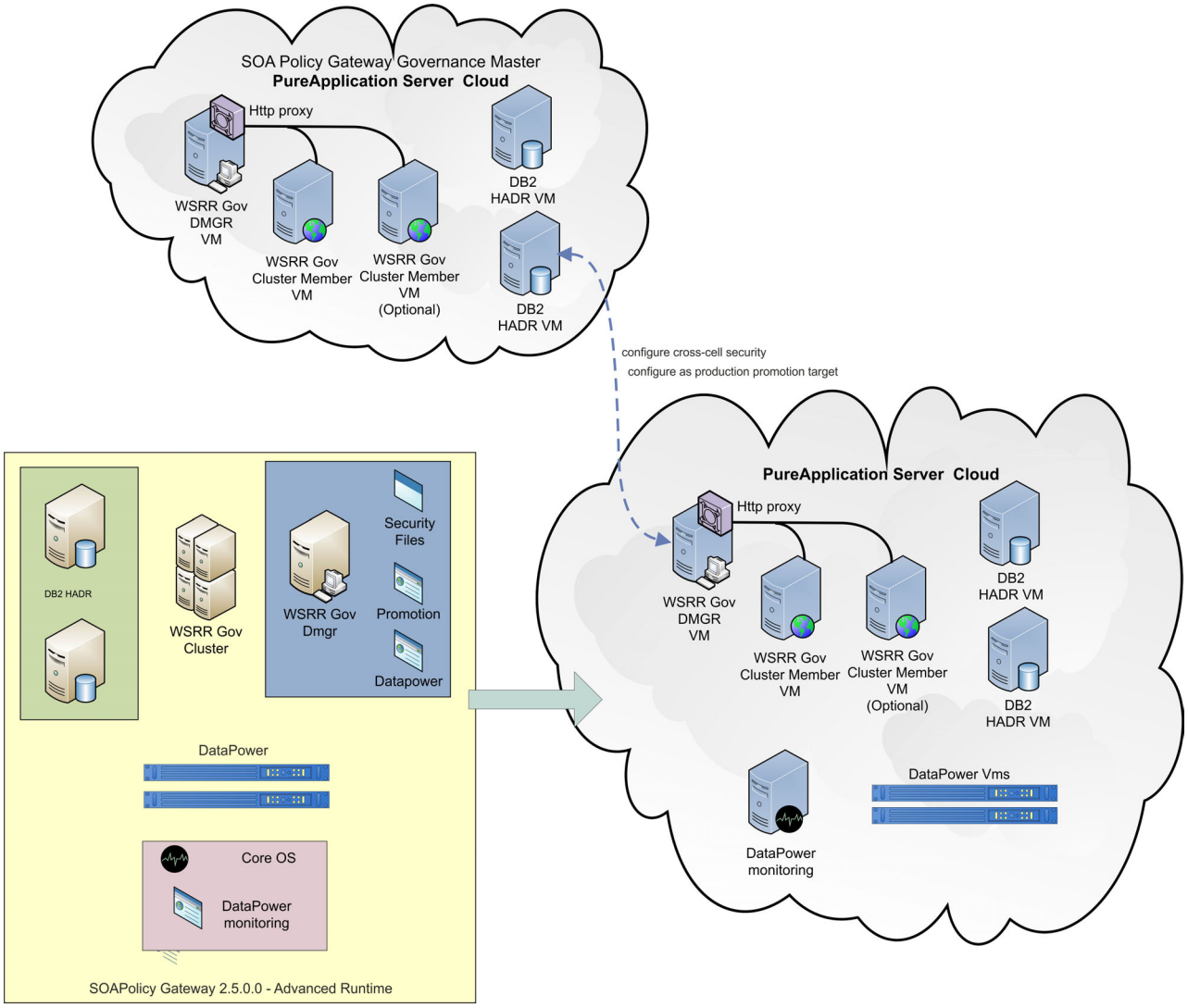


Figure 8. PureApplication Server configuration with DataPower VMs

Scripts and advanced options

The pattern requires user input to the following scripts at deploy time:

On the WSRR Deployment manager part:

- SOA Policy Gateway 2.5.0.0 - Security
- SOA Policy Gateway 2.5.0.0 - Promotion
- SOA Policy Gateway 2.5.0.0 - DataPower Domain

On the Core OS part:

- SOA Policy Gateway 2.5.0.0 - DataPower Monitoring

View the part and script parameters:

- "DB2 Enterprise HADR Primary part" on page 31
- "DB2 Enterprise HADR Standby part" on page 33
- "WSRR Deployment manager part" on page 35
- "WSRR Custom nodes part" on page 36

- "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain" on page 38
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)" on page 41

Configuring the advanced runtime with a governance master

When an advanced runtime pattern is configured with a governance master pattern, the following actions occur:

- Cross-cell security is configured
- The `promotion.xml` file on the governance master is updated with the data from the advanced runtime deployment.

To configure promotion, you must choose one of the following stage options:

- production
- staging

These options align with the levels provided by the Governance Enablement Profile in WSRR. For more information about the Governance Enablement Profile in WSRR, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile.

SOA Policy Gateway Advanced Runtime External DataPower

The SOA Policy Gateway Advanced Runtime External DataPower is the same as the Advanced Runtime pattern, but requires that external DataPower appliances are specified on deployment.

Note: This description applies to the SOA Policy Gateway Advanced Runtime pattern on IBM Power systems.

TheSOA Policy Gateway Advanced Runtime External DataPower pattern requires the following parts:

- WSRR Deployment manager
- WSRR Custom nodes
- DB2 HADR Primary
- DB2 HADR Standby
- SOA monitoring for DataPower (in a Core OS part)

The following diagram shows the configuration of an advanced runtime system.

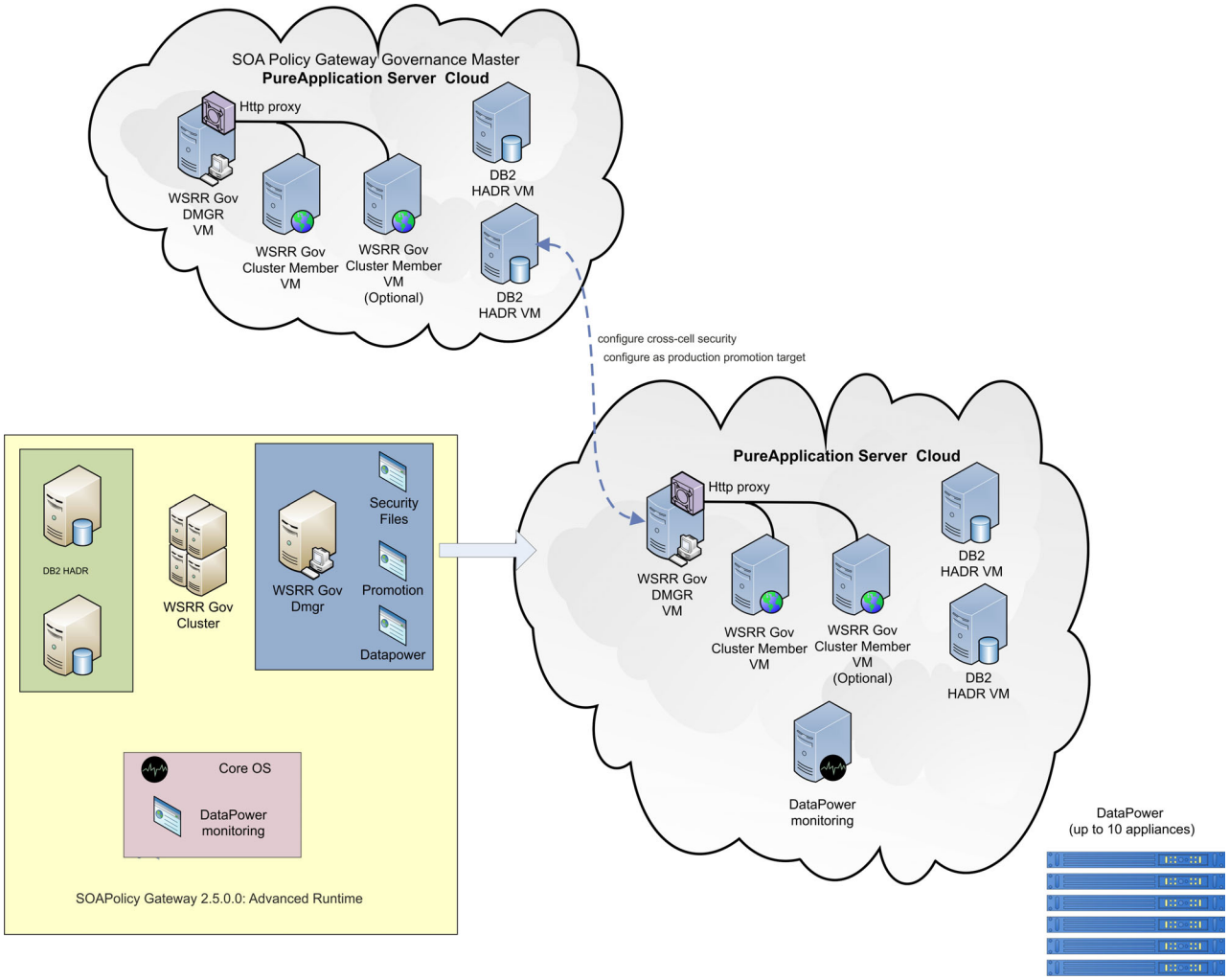


Figure 9. PureApplication Server configuration with DataPower appliances

Scripts and advanced options

The pattern requires user input to the following scripts at deploy time.

On the WSRR Deployment manager part:

- SOA Policy Gateway 2.5.0.0 - Security
- SOA Policy Gateway 2.5.0.0 - Promotion
- SOA Policy Gateway 2.5.0.0 - DataPower Domain

On the Core OS part:

- SOA Policy Gateway 2.5.0.0 - DataPower Monitoring

View the part and script parameters:

- "DB2 Enterprise HADR Primary part" on page 31
- "DB2 Enterprise HADR Standby part" on page 33
- "WSRR Deployment manager part" on page 35
- "WSRR Custom nodes part" on page 36
- "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39
- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain" on page 38

- "Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)" on page 41

Configuring the advanced runtime with a governance master

When an advanced runtime pattern is configured with a governance master Pattern the following occurs:

- Cross-cell security is configured
- The `promotion.xml` file on the Governance Master is updated with the data from the Advanced Runtime deployment.

To configure promotion, you must choose one of the following stage options:

- production
- staging

These options align with the levels provided by the Governance Enablement Profile in WSRR. For more information about the Governance Enablement Profile in WSRR, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile.

Shared Service

The pattern includes a shared service that is used by deployed patterns to provide monitoring.

System Monitoring for SOA Policy Gateway

The System Monitoring for SOA Policy Gateway shared service provides the monitoring components for the SOA Policy Gateway.

Monitoring in the basic and advanced runtime patterns is provided by the DataPower monitoring service running in a Core OS part. The monitoring service itself uses ITCAM for SOA components that are contained in the System Monitoring for SOA Policy Gateway Pattern. Monitoring of the WSRR instances also requires that the System Monitoring for WebSphere Application Server shared service is running.

Follow the related link for detailed ITCAM for SOA documentation.

Related information:

ITCAM for SOA 7.2.1 documentation (from Fix Central)

Parts

The following parts comprise the IBM SOA Policy Gateway Pattern.

DB2 Enterprise part

The DB2 Enterprise part provides some configuration options.

The configurable parameters of the DB2 Enterprise 10.1.0.2 virtual system image are described in the following table:

Table 2. Configurable parameters

Parameter name	Default value	Description
Virtual CPUs	1	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	2048	The amount of memory that is allocated to this virtual machine, in megabytes.
Instance owner group	db2iadm1	The group to which the DB2 instance owner belongs.
Instance owner	db2inst1	The ID of the DB2 instance owner. This user ID is used as the installation owner of the DB2 instance and as the owner of the databases and schemas.
Password (Instance owner)	password	The password for the user ID db2inst1 of the operating system.
Verify password	password	Verifies the instance owner password.
Fenced user group	db2fadm1	The group to which the DB2 fenced owner belongs.
Fenced user	db2fenc1	The ID of the DB2 fenced user. The fenced user ID is used to run user-defined functions (UDFs) and stored procedures outside the address space that is used by the DB2 database. The fenced user is a user under which "fenced" stored procedures can run with reduced operating system authority.
Password (db2fenc1)		The password for the fenced user ID
Verify password		Verifies the fenced user password.
DAS user group	dasadm1	The group to which the DB2 DAS owner belongs.
DAS user	dasusr1	The user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Password (DAS user)	password	The password for the DAS user.
Verify password	password	Verifies the dasusr1 password.
DB2 Service port	50000	The port is locked and cannot be changed.
Database creation	Create-new-database	This value is locked and cannot be changed.
Name for the new database	WSRR	This value is locked and cannot be changed.
Codeset for the new database	UTF-8	

Table 2. Configurable parameters (continued)

Parameter name	Default value	Description
Territory for the new database	US	
Collation for the new database	SYSTEM	
Pagesize for the new database	32768	This value is locked and cannot be changed.
DB2 compatibility mode	Default	This value is locked and cannot be changed.
Configure all raw disks for use by DB2	NO	
Password (root)		The password for the root user ID. This is the password for the operating system of the virtual machine that is represented by this part in the pattern.
Verify password		Verifies the root password.
Password (virtuser)		The password for the virtuser user ID of the operating system. This user ID is used as a non-root user ID for the virtual machine.
Verify password		Verifies the virtuser password.
Enable VNC	True	This value is locked and cannot be changed.

DB2 Enterprise HADR Primary part

The DB2 Enterprise HADR Primary part provides some configuration options.

The configurable parameters of the DB2 Enterprise HADR Primary part are described in the following table:

Table 3. Configurable parameters

Parameter name	Default value	Description
Virtual CPUs	1	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	2048	The amount of memory that is allocated to this virtual machine, in megabytes.
Instance owner group	db2iadm1	The group to which the DB2 instance owner belongs.
Instance owner	db2inst1	The ID of the DB2 instance owner. This user ID is used as the installation owner of the DB2 instance and as the owner of the databases and schemas.
Password (Instance owner)	password	The password for the user ID db2inst1 of the operating system.
Verify password	password	Verifies the instance owner password.

Table 3. Configurable parameters (continued)

Parameter name	Default value	Description
Fenced user group	db2fadm1	The group to which the DB2 fenced owner belongs.
Fenced user	db2fenc1	The ID of the DB2 fenced user. The fenced user ID is used to run user-defined functions (UDFs) and stored procedures outside the address space that is used by the DB2 database. The fenced user is a user under which "fenced" stored procedures can run with reduced operating system authority.
Password (db2fenc1)		The password for the fenced user ID
Verify password		Verifies the fenced user password.
DAS user group	dasadm1	The group to which the DB2 DAS owner belongs.
DAS user	dasusr1	The user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Password (DAS user)	password	The password for the DAS user.
Verify password	password	Verifies the dasusr1 password.
DB2 Service port	50000	The port is locked and cannot be changed.
Database creation	Create-new-database	This value is locked and cannot be changed.
Name for the new database	WSRR	This value is locked and cannot be changed.
Codeset for the new database	UTF-8	
Territory for the new database	US	
Collation for the new database	SYSTEM	
Pagesize for the new database	32768	This value is locked and cannot be changed.
DB2 compatibility mode	Default	This value is locked and cannot be changed.
Configure all raw disks for use by DB2	NO	
Password (root)		The password for the root user ID. This is the password for the operating system of the virtual machine that is represented by this part in the pattern.
Verify password		Verifies the root password.

Table 3. Configurable parameters (continued)

Parameter name	Default value	Description
Password (virtuser)		The password for the virtuser user ID of the operating system. This user ID is used as a non-root user ID for the virtual machine.
Verify password		Verifies the virtuser password.
Enable VNC	True	This value is locked and cannot be changed.

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

DB2 Enterprise HADR Standby part

The DB2 Enterprise HADR Standby part provides some configuration options.

Table 4. Configurable parameters

Parameter name	Default value	Description
Virtual CPUs	1	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	2048	The amount of memory that is allocated to this virtual machine, in megabytes.
Instance owner group	db2iadm1	The group to which the DB2 instance owner belongs.
Instance owner	db2inst1	The ID of the DB2 instance owner. This user ID is used as the installation owner of the DB2 instance and as the owner of the databases and schemas.
Password (Instance owner)	password	The password for the user ID db2inst1 of the operating system.
Verify password	password	Verifies the instance owner password.
Fenced user group	db2fadm1	The group to which the DB2 fenced owner belongs.
Fenced user	db2fenc1	The ID of the DB2 fenced user. The fenced user ID is used to run user-defined functions (UDFs) and stored procedures outside the address space that is used by the DB2 database. The fenced user is a user under which "fenced" stored procedures can run with reduced operating system authority.
Password (db2fenc1)		The password for the fenced user ID
Verify password		Verifies the fenced user password.
DAS user group	dasadm1	The group to which the DB2 DAS owner belongs.

Table 4. Configurable parameters (continued)

Parameter name	Default value	Description
DAS user	dasusr1	The user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Password (DAS user)	password	The password for the DAS user.
Verify password	password	Verifies the dasusr1 password.
DB2 Service port	50000	The port is locked and cannot be changed.
Database creation	Create-new-database	This value is locked and cannot be changed.
Name for the new database	WSRR	This value is locked and cannot be changed.
Codeset for the new database	UTF-8	
Territory for the new database	US	
Collation for the new database	SYSTEM	
Pagesize for the new database	32768	This value is locked and cannot be changed.
DB2 compatibility mode	Default	This value is locked and cannot be changed.
Configure all raw disks for use by DB2	NO	
Password (root)		The password for the root user ID. This is the password for the operating system of the virtual machine that is represented by this part in the pattern.
Verify password		Verifies the root password.
Password (virtuser)		The password for the virtuser user ID of the operating system. This user ID is used as a non-root user ID for the virtual machine.
Verify password		Verifies the virtuser password.
Enable VNC	True	This value is locked and cannot be changed.

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

WSRR Standalone server part

The WSRR Standalone server part provides some configuration options.

The configurable parameters of the WSRR Standalone server part are described in the following table:

Table 5. Configured parameters

Parameter name	Default value	Description
Virtual CPUs	1	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	4096	The amount of memory that is allocated to this virtual machine, in megabytes.
Cell name	Set to one of the following values: • SOAPolicySampleCell (basic runtime sample pattern) • SOAPolicyBasicCell (basic runtime pattern) • SOAPolicyBasicCell (basic runtime external DataPower pattern)	
Node name	Set to one of the following values: • SOAPolicySampleNode (basic runtime sample pattern) • SOAPolicyBasicNode (basic runtime pattern) • SOAPolicyBasicNode (basic runtime external DataPower pattern)	
Password (root)		The password for the root user ID. This is the password for the operating system of the virtual machine that is represented by this part in the pattern.
Verify password		Verifies user input for Password (root).
WebSphere administrative user name	virtuser	The WebSphere Application Server administrative user name. You must not change this value.
WebSphere administrative password		The WebSphere Application Server administrative user password.
Verify password		Verifies user input for WebSphere Application Server administrative password.
Enable VNC	True	This value is locked and cannot be changed.

WSRR Deployment manager part

The WSRR Deployment manager part provides some configuration options.

The configurable parameters of the WSRR Deployment manager part are described in the following table:

Table 6. Configurable parameters

Parameter name	Default value	Description
Virtual CPUs	1	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	2048	The amount of memory that is allocated to this virtual machine, in megabytes.
Cell name	SOAPolicyAdvancedCell	The cell name for the Advanced Runtime pattern.
Node name	SOAPolicyAdvancedNode	The node name for the node residing on the Deployment Manager virtual machine in Advanced Runtime pattern.
Password (root)		The password for the root user ID. This is the password for the operating system of the virtual machine that is represented by this part in the pattern.
Verify password		Verifies user input for Password (root).
WebSphere administrative user name	virtuser	The WebSphere Application Server admin user name. You must not change this value.
WebSphere administrative password		The WebSphere Application Server admin user password.
Verify password		Verifies user input for WebSphere Application Server administrative password.
Enable VNC	True	This value is locked and cannot be changed.

WSRR Custom nodes part

The WSRR Custom nodes part provides some configuration options.

The configurable parameters of the WSRR Custom nodes part are described in the following table:

Table 7. Configurable parameters

Parameter name		Description
Virtual CPUs	2	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	4096	The amount of memory that is allocated to this virtual machine, in megabytes.
Cell name	CloudBurstCell	The cell name value in the Custom node part configuration is ignored.
Node name	SOAPolicyAdvancedNode	The node name for the node residing on the Custom node virtual machine in Advanced Runtime pattern.

Table 7. Configurable parameters (continued)

Parameter name		Description
Password (root)		The password for the root user ID. This is the password for the operating system of the virtual machine that is represented by this part in the pattern.
Verify password		Verifies the user input for Password (root).
WebSphere administrative user name	virtuser	The WebSphere Application Server environment admin user name. You must not change this value.
WebSphere administrative password		The WebSphere Application Server environment admin user password.
Verify password		Verifies user input for WebSphere Application Server administrative password.
Enable VNC	True	This value is locked and cannot be changed.

DataPower part

The DataPower part has some configuration options.

The configurable parameters of the DataPower virtual system image are described in the following table:

Table 8. Configured parameters

Parameter name	Default value	Description
Virtual CPUs	4	The number of virtual processors that are allocated for the virtual machine that is represented by this part.
Memory size (MB)	4096	The amount of memory that is allocated to this virtual machine, in megabytes.
admin password		The password for the DataPower administrator.
Verify password		Verifies user input for admin password.
Enable SSH	True	Enables SSH (for using the DataPower command line interface).
SSH port	22	The port for SSH.
Enable XML Management interface	True	Enables the XML Management interface. When enabled, this interface allows administrators to send status and configuration requests to the DataPower appliance through a standard SOAP interface.
XML Management Interface port	5550	The port for the XML Management interface.
Enable Web Management Service	True	Enables the WebGUI for interacting with the DataPower appliance.

Table 8. Configured parameters (continued)

Parameter name	Default value	Description
Web Management Service port	9090	The port for the WebGUI.
RAID directory	raid0	The directory where you can access files in the DataPower auxiliary data storage.

Script packages

There are seven script packages that are provided with the IBM SOA Policy Gateway Pattern.

The following script packages that are included with this pattern:

- SOA Policy Gateway 2.5.0.0 - DataPower Domain
- SOA Policy Gateway 2.5.0.0 - Promotion
- SOA Policy Gateway 2.5.0.0 - Samples
- SOA Policy Gateway 2.5.0.0 - Security
- SOA Policy Gateway 2.5.0.0 - DataPower Domain
- SOA Policy Gateway 2.5.0.0 - Add Named Queries
- SOA Policy Gateway 2.5.0.0 - Tear Down

The Add Named Queries and Tear Down scripts contain no user-configurable parameters.

Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain

The DataPower Domain script provisions the DataPower domain during deployment. The script configures the connection between the WSRR run time and up to 10 DataPower (virtual) appliances.

Parameters

Table 9. Configurable parameters

Parameter name	Default value	Description
DataPower_hostname	*This value is locked and cannot be changed.*	The hostname for the DataPower instance or appliance to be monitored.
DataPower_admin_id	*This value is locked and cannot be changed.*	The administrator user ID for that instance or appliance.
DataPower_XML_mgmt_port	*This value is locked and cannot be changed.*	The port for communicating with the XML Management interface in the DataPower instance or appliance.
DataPower_admin_password	*This value is locked and cannot be changed.*	The password for the administrator user ID.
Verify password	*This value is locked and cannot be changed.*	Repeat the password for the administrator user ID.
DataPower2_hostname	*This value is locked and cannot be changed.*	
DataPower2_admin_id	*This value is locked and cannot be changed.*	

Table 9. Configurable parameters (continued)

Parameter name	Default value	Description
DataPower2_XML_mgmt_port	*This value is locked and cannot be changed.*	
DataPower2_admin_password	*This value is locked and cannot be changed.*	
Verify password	*This value is locked and cannot be changed.*	
...		...
DataPower10_hostname	*This value is locked and cannot be changed.*	
DataPower10_admin_id	*This value is locked and cannot be changed.*	
DataPower10_XML_mgmt_port	*This value is locked and cannot be changed.*	
DataPower10_admin_password	*This value is locked and cannot be changed.*	
Verify password	*This value is locked and cannot be changed.*	
New_DataPower_domain	The default value depends on the pattern type: • SOAPolicyAdvancedRuntime • SOAPolicyBasicRuntime	The new domain name to be created on each DataPower appliance or instance. It must not match any existing domain or the script package fails or exits. The value cannot contain any spaces.
Remove_security_files	True	For support use, you can ignore this setting.

Script: SOA Policy Gateway 2.5.0.0 - Promotion

The Promotion script enables a Basic Runtime or Advanced Runtime pattern to be integrated with a pre-deployed SOA Policy Gateway Governance Master pattern. It establishes cross-cell security between the Runtime and the Governance pattern, while optionally configuring WSRR promotion into the governance master.

Parameters

Table 10. Configurable parameters

Parameter name	Default value	Description
WSRR_GOV_DMGR_hostname		The host name of the Dmgr for the WSRR Cluster.
WSRR_GOV_DMGR_cellname	SOAPolicyGMCell	The Cell Name for the WSRR Cluster.
WSRR_GOV_admin_user	virtuser	The Admin Id for the WSRR Governance Cell.
WSRR_GOV_admin_password		The password for the Admin ID for the WSRR Governance Cell.
Verify password		Verifies user input for WSRR_GOV_admin_password.

Table 10. Configurable parameters (continued)

Parameter name	Default value	Description
Promotion_environment		Must be one of staging, production, or Unset. These values are case-sensitive and must match exactly.
LTPA_key_password		An LTPA Key is exported and used during the Script Package. The key is from the Governance Master and is used across all CELLS in the promotion environment. This is the password used when exporting that LTPA key.
Verify password		Verifies user input for LTPA_key_password.

Script: SOA Policy Gateway 2.5.0.0 - Sample

The Sample script configures the sample application parameters for use with the SOA Policy Gateway Basic Runtime Sample pattern.

Parameters

None of these parameters can be set by the user.

Table 11. Configurable parameters

Parameter name		Description
SCP_host	*This value is locked and cannot be changed.*	
SCP_user	*This value is locked and cannot be changed.*	
SCP_password	*This value is locked and cannot be changed.*	
Verify password	*This value is locked and cannot be changed.*	
SCP_zip_location	*This value is locked and cannot be changed.*	
CLIENT_PUBLIC_KEY_file	*This value is locked and cannot be changed.*	
CLIENT_PUBLIC _KEY_password	*This value is locked and cannot be changed.*	
Verify password		
CLIENT_PRIVATE_KEY_file	*This value is locked and cannot be changed.*	
CLIENT_PRIVATE_KEY_password	*This value is locked and cannot be changed.*	
Verify password		
CLI_FILE_file	*This value is locked and cannot be changed.*	
Verify password	*This value is locked and cannot be changed.*	

Table 11. Configurable parameters (continued)

Parameter name		Description
DataPower_hostname	*This value is locked and cannot be changed.*	The hostname of the DataPower instance.
DataPower_XML_mgmt_port	*This value is locked and cannot be changed.*	The port that is used for the DataPower XML Management Interface.
DataPower_admin_id	*This value is locked and cannot be changed.*	The administrator user ID with appropriate permissions to use the XML Management Interface.
DataPower_admin_password	*This value is locked and cannot be changed.*	The password for the DataPower_admin_id.
Verify password	*This value is locked and cannot be changed.*	Verifies user input for DataPower_admin_password.
SOAPolicySample_DataPower_domain	*This value is locked and cannot be changed.*	The sample domain name. It must not match any existing domain on the DataPower instance.
SamplePolicySample_starting_port	*This value is locked and cannot be changed.*	The application requires 5 free ports, which are used sequentially from this value. For example, if the value is 62000, ports 62000-62004 is used. The script does not check if the ports are free.
LDAP_hostname	*This value is locked and cannot be changed.*	The hostname of the WSRR standalone part, where an LDAP server is also hosted.
LDAP_port	*This value is locked and cannot be changed.*	The port for the LDAP server.
LDAP_password	*This value is locked and cannot be changed.*	The password that is used when binding with the LDAP_DN.
Verify password	*This value is locked and cannot be changed.*	Verifies user input for LDAP_password.
LDAP_DN	*This value is locked and cannot be changed.*	The distinguished name that is used to bind to the LDAP.

Script: SOA Policy Gateway 2.5.0.0 - Security

The Security script copies security information (certificates and so on) between the DataPower and WSRR systems in the pattern.

The configuration parameters for the security script files are for support use. You should leave them set to their default values.

Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)

The DataPower Monitoring script specifies the connection parameters for the DataPower monitoring shared service. The ITCAM DataPower data collectors and agent run in the Core OS part.

Parameters

The monitoring service can monitor up to 10 DataPower virtual appliances.

Table 12. Configurable parameters

Parameter name	Default value	Description
DataPower1_hostname		The hostname for the DataPower virtual appliance to be monitored.
DataPower1_admin_id	admin	The administrator user ID for that virtual appliance.
DataPower1_XML_mgmt_port	5550	The port for communicating with the XML Management interface in the DataPower virtual appliance.
DataPower1_admin_password		The password for the administrator user ID.
Verify password		Repeat the password for the administrator user ID.
DataPower2_hostname		
DataPower2_admin_id	admin	
DataPower2_XML_mgmt_port	5550	
DataPower2_admin_password		
Verify password		
...		...
DataPower10_hostname		
DataPower10_admin_id	admin	
DataPower10_XML_mgmt_port	5550	
DataPower10_admin_password		
Verify password		

Script: SOA Policy Gateway 2.5.0.0 - External DataPower Monitoring

The DataPower Monitoring script specifies the connection parameters for the DataPower monitoring shared service. The ITCAM DataPower data collectors and agent run in the Core OS part.

Parameters

The monitoring service can monitor up to 10 DataPower appliances.

Table 13. Configurable parameters

Parameter name	Default value	Description
DataPower1_hostname		The hostname for the DataPower appliance to be monitored.
DataPower1_admin_id	admin	The administrator user ID for that appliance.
DataPower1_XML_mgmt_port	5550	The port for communicating with the XML Management interface in the DataPower appliance.

Table 13. Configurable parameters (continued)

Parameter name	Default value	Description
DataPower1_admin_password		The password for the administrator user ID.
Verify password		Repeat the password for the administrator user ID.
DataPower2_hostname		
DataPower2_admin_id	admin	
DataPower2_XML_mgmt_port	5550	
DataPower2_admin_password		
Verify password		
...		...
DataPower10_hostname		
DataPower10_admin_id	admin	
DataPower10_XML_mgmt_port	5550	
DataPower10_admin_password		
Verify password		

Chapter 5. Working with the IBM SOA Policy Gateway Pattern

The IBM SOA Policy Gateway Pattern provides the pattern definitions for repeatable deployment. These topics describe how to deploy the patterns.

As part of the deployment process, configure the part parameters. For more information, see "Deploying patterns" on page 47. The patterns are described in Chapter 4, "Patterns, parts, and script packages," on page 19.

Related tasks:

Chapter 3, "Getting started with the IBM SOA Policy Gateway Pattern," on page 13
This pattern uses WebSphere DataPower to control messages by using governed polices and service definitions in WSRR. Review the topics in this section to understand how to download and install the pattern, how to verify the pattern after installation, accept licenses, and the user roles involved.

Planning the pattern configuration and pattern prerequisites

The IBM SOA Policy Gateway Pattern provides a means to quickly and reliably provision an environment for governing service definitions and policies, and enforcing those policies. Deployment of the pattern starts with the governance master, followed by the runtime pattern.

Preparing and deploying the IBM SOA Policy Gateway Pattern

- If you are using an external DataPower appliance, prepare the appliance for remote administration. For more information, see "Configuring a DataPower appliance for the IBM SOA Policy Gateway Patterns" on page 46.

Deploy the governance master pattern:

1. Deploy a SOA Policy Gateway Governance Master pattern. Wait for the deployment to complete before deploying runtime patterns. For more information, see "Deploying the governance master pattern" on page 50.

Deploy the runtime patterns:

1. Decide whether a basic runtime pattern with a standalone environment, or an advanced runtime pattern with a clustered environment is needed.
2. Determine how many DataPower instances or appliances your runtime patterns require.

   Patterns that include DataPower have two DataPower instances by default. You can configure up to 10 DataPower instances. For more information, see "Adding DataPower instances to a pattern" on page 55.

   Patterns with external DataPower can be configured to work with up to 10 DataPower appliances. See "Deploying the Basic and Advanced External DataPower Patterns" on page 56.

   Note: Extra DataPower instances and appliances cannot be added after this configuration is completed.
3. Configure the runtime pattern with the governance master pattern information. For more information, see "SOA Policy Gateway Governance Master deployment information" on page 50. You can omit governance master pattern information to deploy a standalone system, if required (although this will show an error on deployment, the error can be ignored).

4. Specify whether the runtime system is staging or production.
5. Deploy your pattern. For more information, see “Deploying an advanced runtime pattern” on page 52 or “Deploying a basic runtime pattern” on page 51.
6. Wait until fully deployed before you deploy another runtime.

When deployment of the runtime patterns is completed:

1. WSRR and WebSphere security can be updated from the default security configuration. For more information, see “Security for the IBM SOA Policy Gateway Pattern patterns.”
2. The DataPower domain is ready for gateway configuration. If using a virtual DataPower appliance, you must first apply the latest fix pack, see “Updating DataPower in the deployed instance” on page 53.

Configuring a DataPower appliance for the IBM SOA Policy Gateway Patterns

Complete the following DataPower configuration steps before you run the SOAPolicy scripts.

Procedure

1. Log in to the DataPower appliance WebGUI as an Administrator.
2. Search for `XML Management Interface`.
3. Make sure that its state is enabled.
4. Make sure that the following are active and secured correctly:
   - SOAP Management URI
   - SOAP Configuration Management
   - SOAP Configuration Management (v2004)
   - AMP Endpoint
   - SLM Endpoint
   - WS-Management Endpoint
   - WSDM Endpoint
   - UDDI Subscription
   - WSRR Subscription

Security for the IBM SOA Policy Gateway Pattern patterns

Mutual authentication occurs between the DataPower applications and the scripts in the Basic and Advanced Patterns. The scripts perform the necessary certificate exchange. Note that the default SSL certificates supplied with the pattern are attributed to the host that was used to create the pattern.

Increasing security

The WSRR images and the WebSphere Application Server images that are used in the patterns have only the default security in place. To produce a more secure environment, you can use standard WebSphere Application Server security techniques.

See the WebSphere Network Deployment Version 8.0 Information Center at the following links:

- WebSphere Application Server, Network Deployment (Distributed platforms and Windows), Version 8.0: IBM WebSphere Application Server, Network Deployment (Distributed platforms and Windows), Version 8.0 Information Center
- Application security: IBM WebSphere Application Server, Network Deployment (Distributed platforms and Windows), Version 8.0 Information Center - Securing applications and their environment
- End to end paths for security: IBM WebSphere Application Server, Network Deployment (Distributed platforms and Windows), Version 8.0 Information Center - Securing applications and their environment

Deploying patterns

Deploying patterns with IBM PureApplication System into the cloud provides a running SOA policy gateway environment. You can deploy the predefined patterns available with the IBM SOA Policy Gateway Pattern images, or deploy patterns that you created.

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. You require detail of the environment, cloud group, and IP group to deploy into from your PureAS system administrator.

About this task

You deploy the pattern by using the Workload console.

Procedure

To deploy the IBM SOA Policy Gateway Patterns to run in your private cloud, complete the following steps:

1. From the list of patterns in the Virtual System Patterns window, select the pattern to deploy.
2. Click the **Deploy** icon.
3. Complete the required fields to deploy the pattern. In the window, enter a name for the virtual system and enter any other required information. A check mark beside each item indicates that it does not require further configuration. You can change the parameters for configured parts, prior to deploying the pattern, by clicking the part name to open the editor for the part. Virtual machines are created, in the required order, and then started.

Results

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

What to do next

You can view the status of your instance, to see when deployment is complete and begin to administer it, from the Virtual System Instances window.

Related information:

IBM PureApplication System: Managing virtual system patterns

Deploying the system monitoring shared service

Deploying the System Monitoring for SOA Policy Gateway shared service provides the monitoring components for your virtual system.

Before you begin

The PureAS system administrator must start the System Monitoring shared service, and advise you of the cloud group and environment that they started it in. You must use the same cloud group and environment to deploy the SOA Policy Gateway system monitoring shared service and your runtime and governance patterns.

Monitoring the WSRR instances also requires that the System Monitoring for WebSphere Application Server shared service is started, so you must ensure that it is present on your PureAS system.

Procedure

Complete the following steps in the workload console:

1. Click **Instances** > **Shared Services**.
2. Verify that the System Monitoring service is running in the cloud group to which your patterns will deploy. If it is not running, contact your PureAS administrator to start it.
3. To enable the DataPower monitoring shared service:
   a. Click **Cloud** > **Pattern Types**.
   b. Select the **System Monitoring for SOA Policy Gateway Pattern 2.5.0.0** entry in the Pattern Types pane.
   c. Click **Enable** in the **Status** field, and wait until the status field changes to **Disable**.
4. To start the WebSphere Application Server monitoring shared service:
   a. Click **Instances** > **Shared Services**.
   b. Click the plus symbol in the Shared Service Instances pane to open the Deploy Shared Service window.
   c. Select **System Monitoring for WebSphere Application Server** and click **OK**.
   d. In the Configure and Deploy a Shared Service window, specify whether you want the service to start on previously-deployed patterns by selecting the bottom two check boxes. Click **OK**.
   e. In the Deploy Virtual Application window, specify the **Target cloud group**, **IP group**, and **Profile** as advised by your PureAS system administrator. These must be the same as those to which your Virtual Systems deploy.
5. To start the WebSphere DataPower monitoring shared service:
   a. Click **Instances** > **Shared Services** in the menu bar.
   b. Click the plus symbol in the Shared Service Instances pane to open the Deploy Shared Service window.
   c. Select **System Monitoring for WebSphere DataPower** from the list and click **OK**.

d. In the Configure and deploy a shared service window, specify whether you want monitoring to start on previously-deployed patterns by selecting the bottom two check boxes. Click **OK**.
e. In the Deploy Virtual Application window, specify the **Target cloud group**, **IP group**, and **Profile** as advised by your PureAS system administrator. These must be the same as those to which your Virtual Systems deploy.
f. Generate and save an SSH Key if you require debug access to the monitoring shared service.
g. Click **OK**.

Results

The System Monitoring for WebSphere DataPower shared service is shown as running. The System Monitoring for WebSphere Application Server shared service is shown as running.

What to do next

To verify the deployment, see "Verifying the deployment" on page 54.

Deploying the basic runtime sample pattern

Deploying the SOA Policy Gateway Basic Runtime Sample pattern creates a running virtual system instance of the pattern. This pattern is available only on x86 systems.

About this task

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

Procedure

To deploy the SOA Policy Gateway Basic Runtime Sample pattern, complete the following steps:

1. In the Workload Console, click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.5.0.0 - Basic Runtime Sample**.
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 unique name for the instance.
   b. Expand the **Choose Environment** section, and specify the **Profile** advised by your PureAS system administrator.
   c. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the parts and scripts. Specify the **Cloud group** and **IP group** as advised by your PureAS system administrator. See the following topics for details of the pattern-specific and script-specific configuration parameters.

      Note: All passwords for this pattern are defaulted to `password`.

      - "DataPower part" on page 37
      - "DB2 Enterprise part" on page 29.
      - "WSRR Standalone server part" on page 34

- "Script: SOA Policy Gateway 2.5.0.0 - Sample" on page 40

5. Click **OK** to deploy the pattern.

What to do next

To verify the deployment, see "Verifying the deployment" on page 54.

Deploying the governance master pattern

Deploying the SOA Policy Gateway Governance Master pattern creates a running virtual system instance of the pattern.

Procedure

To deploy the SOA Policy Gateway Governance Master pattern, complete the following steps:

1. In the Workload Console, click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.5.0.0 - Governance Master**.
3. Click the **Deploy** icon.
4. Complete the 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 unique name for the instance.
   b. Expand the **Choose Environment** section, and specify the **Profile** as advised by your PureAS system administrator.
   c. Configure the virtual patterns. Click **Configure virtual parts** then click the part name to open the editor for the parts and scripts. Specify the **Cloud group** and **IP group** as advised by your PureAS system administrator. See the following topics for details of the pattern-specific and script-specific configuration parameters.
      - "DB2 Enterprise HADR Primary part" on page 31
      - "WSRR Deployment manager part" on page 35
      - "WSRR Custom nodes part" on page 36
      - "DB2 Enterprise HADR Standby part" on page 33
5. Click **OK** to deploy the pattern.

What to do next

To verify the deployment, see "Verifying the deployment" on page 54.

SOA Policy Gateway Governance Master deployment information

The Governance Master must be deployed before the runtime patterns are deployed.

About this task

Deployment information from the Governance Master instance is required as input to deployment values for the runtime patterns.

Procedure

To find the required values from the Governance Master instance:

1. Navigate to **Instances** > **Virtual Systems**.

2. Select the deployment Governance Master instance.
3. Expand **Virtual machines**.
4. Expand the virtual machine named ***WSRRDMGR***.
5. Note the following points:
   - In the **Hardware and network** section, note the Hostname and IP address. The hostname is the **Network interface 0** value.
   - In the **WebSphere configuration** section, note the Cell name.

   The hostname or IP, cell name, and the WebSphere administrative username and password that are used during deployment of the Governance Master instance are required inputs to the following parameters in the runtime patterns:
   - WSRR_GOV_DMGR_hostname
   - WSRR_GOV_DMGR_cellname
   - WSRR_GOV_admin_user
   - WSRR_GOV_admin_password

   If you want to deploy a runtime pattern as a standalone system, you can set these parameters to "Unset". This setting causes the deployment to appear as **failed** in **Virtual System** > **Instances** because the promotion script package fails. The deployment is still usable, however.

Deploying a basic runtime pattern

Deploying a basic runtime pattern creates a running virtual system instance of the pattern.

Before you begin

Complete the following tasks before you deploy a basic runtime pattern:

- If you are deploying a basic runtime pattern with external DataPower, configure your DataPower appliances for the IBM SOA Policy Gateway Pattern; see "Configuring a DataPower appliance for the IBM SOA Policy Gateway Patterns" on page 46. On Power systems, only external DataPower is supported.
- Obtain the Governance Master deployment information; see "SOA Policy Gateway Governance Master deployment information" on page 50.

About this task

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

Note: If you are using the Governance Enablement Profile (GEP), you cannot deploy a staging and production environment concurrently in the runtime patterns. This limitation is because it can cause conflict during the promotion properties configuration process. Deploy the staging environment first, and then the production environment.

Procedure

To deploy a basic runtime pattern, complete the following steps:

1. Click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.5.0.0 - Basic Runtime External DataPower** or **SOA Policy Gateway 2.5.0.0 - Basic Runtime**.

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 unique name for the instance.
   b. Expand the **Choose Environment** section, and specify the **Profile** advised by your PureAS system administrator.
   c. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the parts and scripts. Specify the **Cloud group** and **IP group** as advised by your PureAS system administrator. See the following topics for details of the pattern-specific and script-specific configuration parameters.

      Note: If you want to deploy the pattern without a governance master, enter 'Unset' as the governance master hostname parameter. Be aware that this results in the promotion script package being reported as failing on deployment, but has no other consequences.
      - "DataPower part" on page 37
      - "DB2 Enterprise part" on page 29
      - "WSRR Standalone server part" on page 34
      - "Script: SOA Policy Gateway 2.5.0.0 - Security" on page 41
      - "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39
      - "Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain" on page 38
      - "Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)" on page 41
5. Click **OK** to deploy the pattern.

What to do next

To verify the deployment, see "Verifying the deployment" on page 54.

Deploying an advanced runtime pattern

Deploying an advanced runtime pattern creates a running virtual system instance of the pattern.

Before you begin

Complete the following tasks before you deploy the advanced runtime pattern:

- If you are deploying an advanced runtime pattern with external DataPower, configure your DataPower appliances to connect to the pattern. See "Configuring a DataPower appliance for the IBM SOA Policy Gateway Patterns" on page 46. On Power systems, only external DataPower is supported.
- Obtain the Governance Master deployment information; see "SOA Policy Gateway Governance Master deployment information" on page 50.

About this task

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

Note: If you are using the Governance Enablement Profile (GEP), you cannot deploy a staging and production environment concurrently in the runtime patterns.

This limitation is because it can cause conflict during the promotion properties configuration process. Deploy the staging environment first, and then the production environment.

Procedure

To deploy an advanced runtime pattern, complete the following steps:

1. Click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.5.0.0 - Advanced Runtime External DataPower** or **SOA Policy Gateway 2.5.0.0 - Advanced Runtime**.
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 unique name for the instance.
   b. Expand the **Choose Environment** section, and specify the **Profile** advised by your PureAS system administrator.
   c. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the parts and scripts. Specify the **Cloud group** and **IP group** as advised by your PureAS system administrator. See the following topics for details of the pattern-specific and script-specific configuration parameters.

      Note: If you want to deploy the pattern without a governance master, enter 'Unset' as the governance master hostname parameter. Be aware that this results in the promotion script package being reported as failing on deployment, but has no other consequences.

      - "DataPower part" on page 37
      - "DB2 Enterprise HADR Primary part" on page 31
      - "WSRR Deployment manager part" on page 35
      - "Script: SOA Policy Gateway 2.5.0.0 - Promotion" on page 39
      - "Script: SOA Policy Gateway 2.5.0.0 - DataPower Domain" on page 38
      - "WSRR Custom nodes part" on page 36
      - "DB2 Enterprise HADR Standby part" on page 33
      - "Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)" on page 41
5. Click **OK** to deploy.

What to do next

To verify the deployment, see "Verifying the deployment" on page 54.

Updating DataPower in the deployed instance

After deploying a pattern that includes a WebSphere DataPower component, you must update DataPower to the most recent fix pack.

About this task

You update DataPower by downloading the fix pack from Fix central, and applying it in the DataPower WebGUI.

Procedure

1. Download the update package from Fix Central:
   a. In Fix Central, search for WebSphere DataPower SOA Appliances.
   b. Select and download the package XI52-virtual-6.0.0.1-Firmware.
2. Connect to the WebGUI for the DataPower virtual machine in your deployed pattern, see "Connecting to the console of a virtual DataPower" on page 83.
3. From the control panel, select **System Control**.
4. Locate the **Boot Image** section.
5. Upload to the DataPower appliance the `xi6001.scrypt4` file from the downloaded fixpack. Use the File Manager on the DataPower WebGUI.
6. Select the uploaded scrypt from the **Firmware File** list.
7. Accept the license conditions, and click **Boot Image**.
8. Follow the prompts to install the fix pack.

Verifying the deployment

After you deploy 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 with deployment" on page 99.
2. Optional: If you deployed the SOA Policy Gateway Basic Runtime Sample, test the deployed instance by following the tutorial to send some sample messages by using the sample applications provided. See "Running the sample test cases" on page 60.

Adding an additional runtime environment

The Governance Enablement Profile comes with a pre-defined environment classification system that contains four distinct environments; Development, Test, Staging, and Production.

About this task

The Staging and Production environments are also codified in the SOA lifecycle that defines the lifecycle of Capability Versions, such as Service Versions. There are states and transitions that are specific to the Staging and Production environments, allowing for controlled promotion into these runtime environments by defining the target systems in the promotion configuration file. This procedure is appropriate if your organization defines environments in the same way, with Staging as a pre-Production environment that allows testing before the Capability Version is opened for general use. However, many organizations require more environments, so modifications are needed in the profile to accommodate these differences. This section describes one way that a new runtime environment can be added into the WSRR Governance Enablement Profile.

For more information about planning a deployment environment, see "Planning the pattern configuration and pattern prerequisites" on page 45.

Procedure

1. Deploy the predefined SOA Policy Gateway Governance Master. For more information, see "Deploying the governance master pattern" on page 50.

2. Optional: Modify the WSRR Governance Enablement Profile. For more information, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Tutorial: Customizing runtime environments.
3. Configure the basic runtime or advanced runtime patterns with the Governance Master details. For more information, see "SOA Policy Gateway Governance Master deployment information" on page 50.

   Note: The promotion environment value must be set to "Unset".
4. Deploy the predefined basic runtime or advanced runtime patterns. For more information, see "Deploying a basic runtime pattern" on page 51 and "Deploying an advanced runtime pattern" on page 52.

Adding DataPower instances to a pattern

Basic and advanced patterns with internal DataPower instances have two instances by default. Each pattern can have up to 10 DataPower instances in total.

About this task

The patterns themselves cannot be edited. You can add more DataPower instances to the basic runtime or advanced runtime patterns by making a copy of the pattern and editing it.

Procedure

1. Open the pattern in the Workload Console.
2. Click **Clone**, and specify a name for the copy of the pattern.
3. Click **Edit**.
4. Drag more DataPower parts from the parts list to add them to the pattern.
5. Click **Done editing**.

Deleting DataPower instances from a pattern

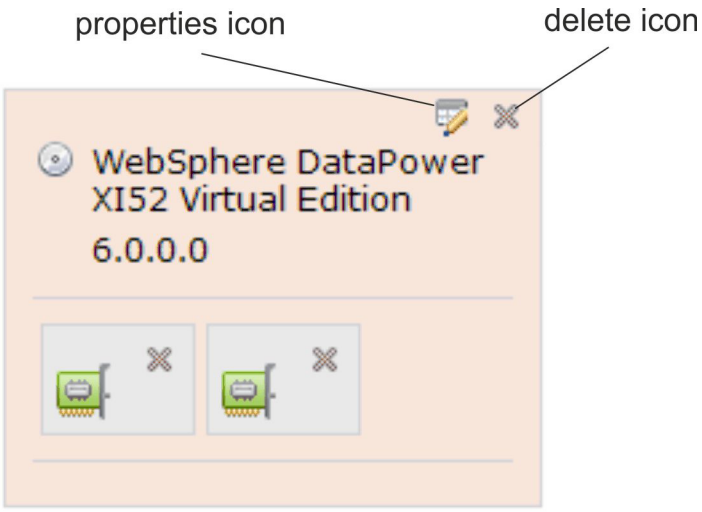
You can delete internal DataPower instances from a pattern if required.

About this task

The patterns themselves cannot be edited. You can delete DataPower instances from the basic runtime or advanced runtime patterns by making a copy of the pattern and editing it.

Procedure

1. Open the pattern in the Workload Console.
2. Click **Clone**, and specify a name for the copy of the pattern.
3. Click **Edit**.
4. Delete a DataPower instance by clicking the delete icon.



Note: The DataPower instances must be deleted in reverse numerical order. Each DataPower instance on the canvas has a number in its name field, which is visible by clicking the properties icon. The name is of the format: 'DataPower_XI52*x*' where *x* is the number (the first DataPower instance does not have a number at all, its name is: 'DataPower_XI52'). The highest numbered DataPower instances are usually at the top left of the canvas.

5. Click **Done editing**.

Deploying the Basic and Advanced External DataPower Patterns

The SOA Policy Gateway Basic Runtime External DataPower and SOA Policy Gateway Advanced Runtime External DataPower patterns can be deployed with up to 10 DataPower appliances.

About this task

For more information about deploying patterns, see “Deploying a basic runtime pattern” on page 51 or “Deploying an advanced runtime pattern” on page 52. For more information about the configuration parameters you must set values for, see “WSRR Standalone server part” on page 34, “WSRR Deployment manager part” on page 35, and “Script: SOA Policy Gateway 2.5.0.0 - DataPower Monitoring (x86 only)” on page 41.

Procedure

1. Deploy the pattern, and click **Configure virtual parts**.
2. For the WSRR standalone or WSRR deployment manager part, enter the following information for each appliance:
   - DataPower_hostname
   - DataPower_XML_mgmt_port
   - DataPower_admin_id
   - DataPower_admin_password
   - Verify password
   - New_DataPower_domain

The sample application

The sample application consists of a Web Service and a RESTful API both described and governed in WSRR. A DataPower domain is configured with WSRR to be a gateway and a sample Web Client is provided to exercise the services.

The basic scenario in the sample application is that of an inventory application for a store (Warehouse), and a RESTful service that duplicates one of the operations for mobile. The Store web service has three operations:

- purchase
- findInventory
- returnProduct

The last operation, findInventory, is also available as a RESTful service.

The Sample Web Service

The basic service level definition (SLD) has two mediation policies attached:

- Validation against Store.wsdl. The sample assumes that the DataPower Validation is turned off.
- Reject if there are more than 5 messages in 90 seconds. This threshold is low for ease of demonstration.

The consumer of the Store service is the StoreConsumer application, which has the consumer ID of “CEO”. This consumer has two Service Level Agreements (SLAs), Gold and Silver. If a request comes into DataPower with the consumer ID of “CEO”, and a Context ID of “Silver”, the request is allowed to pass through, because the Silver SLA is in place. If the consumer ID is “CEO”, and the context ID is “Gold”, the Gold SLA is matched. This SLA has a re-route policy attached to it, so the request is re-routed to the alternate endpoint stated in the policy.

If a request arrives with a consumer ID other than “CEO”, there is no Application Version with this consumer ID. There are therefore also no SLAs that could match, so this is a request from an anonymous consumer. As such, any policies attached to the anonymous SLA are applied. In this case, this causes a notification to appear in the logs. Note, the sample does not include a way to send a request with a consumer ID that is not “CEO”.

The scenario also performs authorization for the findInventory operation, which is based upon user group membership. An LDAP server is provided with the sample for mapping user credentials to the correct group.

The sample application flow diagram shows the flow of the application with each box representing a different DataPower gateway.

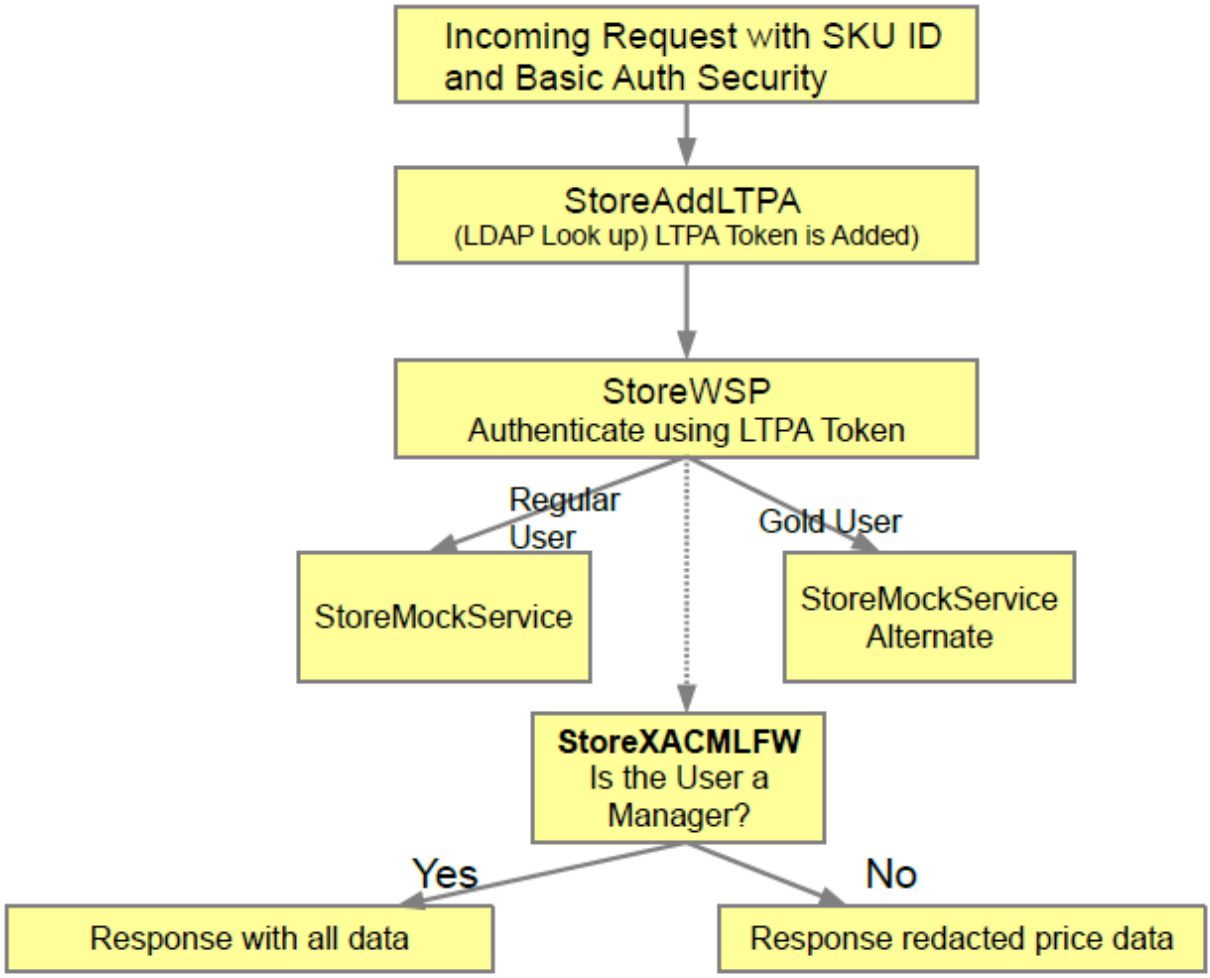


Figure 10. The sample application flow diagram

The Sample RESTful Service

The RESTful service is governed in a similar way to the web service, except in how policies are used. As with the web service there are two SLAs: one for Silver customers and one for Gold customers. For the REST service, however, there are no polices attached at the SLD level (applied to all requests). Instead, there is one policy attached to each of the SLAs. The Gold SLA has a policy that rejects messages after more then 5 requests are made in 9 seconds, and Silver allows 2 requests in 90 seconds before rejecting.

Overview of WSRR artifacts in the sample

The WSRR artifacts describing the Store Service are described here. The artifacts for the REST service follow a similar pattern.

Bob's Warehouse is the organization that owns both the providing Store service and the consuming StoreConsumer application.

The Warehouse Business Service is the object under which all of the versions of the Store service sit. The Store service version represents a particular version of Store

service. This version is the service being provided for re-use. The Store service level definition (SLD) has two policies attached; the first policy rejects messages after 5 messages in 90 seconds and the second policy does validation against the Store.wsdl schema. These policies mean that requests to Store service are validated, and a maximum of 5 requests are allowed through to the service in any 90 second period, regardless of who the request comes from. The SLD also has an anonymous service level agreement (SLA). Any policies attached to this SLA are applied when requests come in for which there is no matching SLA. An SLA matches if the following conditions are satisfied:

- There is a consuming Application Version that matches the consumer ID in the request.
- There is an SLA in place between this consuming application version and the SLD for the service being consumed, which matches the context ID in the request

The StoreConsumer business application represents the StoreConsumer Application, while the StoreConsumer Application Version is a particular version of this application. This application is the consumer: it is re-using the Store service. It has the consumer ID of “CEO”. There are two SLAs in place for this application, which constitute an agreement to allow this application to consume the Store service. One has the context ID of “Gold”, meaning it matches requests from the StoreConsumer application which have the context ID of “Gold” in the request, and one matches Silver. The Gold SLA has a policy attached to re-route requests, so any requests from the StoreConsumer application that have context ID set to Gold are rerouted to the endpoint specified in the policy. The Silver SLA has no policies attached, so its existence means that requests from the StoreConsumer application that have a context ID of Silver are allowed to pass through, though no policy is applied.

In this sample, there is a notify policy attached to the anonymous SLA.

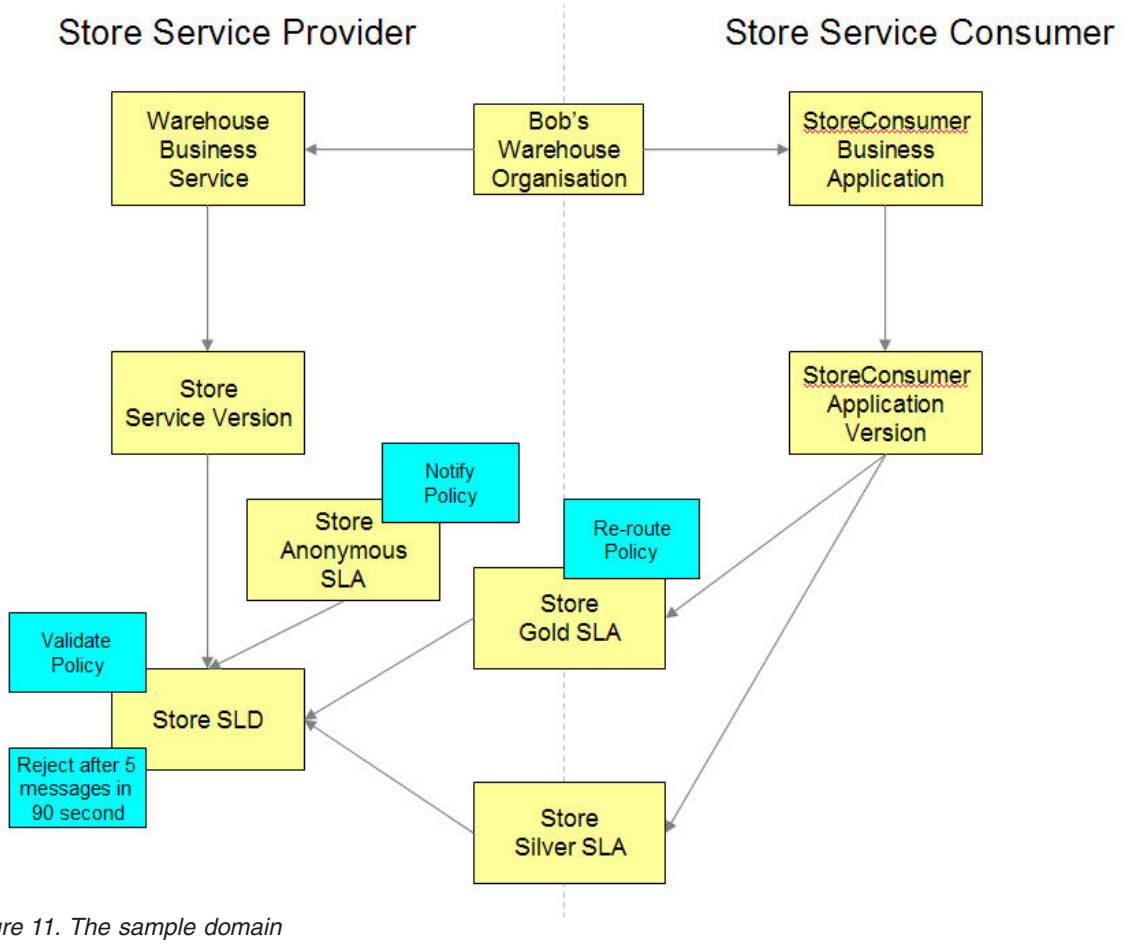


Figure 11. The sample domain

Running the sample test cases

You can use sample web application or the command line to test the Sample application on the deployed SOA Policy Gateway Basic Runtime Sample. Six command line test variations can be run on the sample application.

To deploy the Basic Sample Runtime, see "Deploying the basic runtime sample pattern" on page 49.

Running the sample web application test case

To run the web application test case:

1. Find the hostname of the deployed WSRR environment by opening the deployed Virtual System Instance. To find the hostname, 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.
2. Open the URL in a Web browser: `http://<wssrHostName>:9080/SoaPolicyTester`
3. The following options are available:
   - **Standard Request** - Sends a findInventory request to the store service. The context ID is Silver. The consumer ID is CEO. A successful result displays the text "Part: SKU10 Price: 401.73".

- **Routing Policy Test** - Same as Standard Request, but with Context ID of Gold. The request is routed to an alternate endpoint running the service. A successful result returns “Part: GOLDSKU10 Price: 401.73”.
- **Validation Policy Test** - Sends a request with an invalid payload. The validation policy requires DataPower to validate the request and reject those messages that are invalid. A successful result is a response message from DataPower `"Internal Error (from client)"`.
- **REST Gold** - Send request to the SKU RESTful service with Consumer ID CEO and Context ID Gold. Gold requests are subjected to a policy permitting only 5 messages in 90 seconds. A successful request displays the result “Part: SKU33 Price: 136.43”.
- **REST Silver** - Same as Rest GOLD, but with Silver Context ID. Silver requests are allowed a separate 3 requests in 90 seconds. A successful request displays the result “Part: SKU33 Price: 136.43”.
- **User ID** - The User ID option has two possible values; Full Content or Redacted Content. Each option results in requests originating from different users. The sample utilises an XACML policy, which allows only Managers to see the price. The value of Price in the response message is redacted unless Full Content is selected. A successful result for requests when Redacted Content is selected contains “Price: 0.0”. The RESTful service does not support redaction. The user selected has no effect.

4. Open the WSRR console and explore the service and policies. For more information, see “Connecting to WSRR - Business Space” on page 80.

The sample can also be exercised by using the command line. This is the only way to send traffic that uses the Anonymous SLA

Demonstrating XACML Permit/Deny with the Redaction scenario by using the command line

The following request XML can be sent to the DataPower StoreAddLTPA Service:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:stor="http://company.ibm.com/store">
  <soapenv:Header>
    <store:ConsumerIdentifier xmlns:store="http://store.com">CEO</store:ConsumerIdentifier>
    <store:ContextIdentifier xmlns:store="http://store.com">silver</store:ContextIdentifier>
  </soapenv:Header>
  <soapenv:Body>
  <stor:findInventory>
    <findInventoryReq>
      <sku>SKU10</sku>
    </findInventoryReq>
  </stor:findInventory>
  </soapenv:Body>
</soapenv:Envelope>
```

Assuming that the example request XML is contained in a file named `silver.xml`, enter the following curl command:

```
curl -k --data-bin @./silver.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd http://<yourDataPowerHostName>:62005/Store/Store
```

In this example, ConsumerX is a Manager so the full price information is visible in the response:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
```

```
<KD4NS:KD4SoapHeaderV2
xmlns:KD4NS="http://www.ibm.com/KD4Soap">AFIAAgAkZmExODgzNTQtY2Q1ZC0z
YjU0LWEyMzItZGM3MmEzNWY0MTAzACRmYWVjYjA1Mi1jMWUxLTMyODEtOWY3Ni0wY2IxN
mRhMDc4MjkAAw==</KD4NS:KD4SoapHeaderV2>
</soapenv:Header>
<soapenv:Body>
  <b:findInventoryResponse xmlns:a="http://company.ibm.com/"
     xmlns:b="http://company.ibm.com/store">
  <findInventoryRes>
<sku>SKU10</sku>
<price>461.73</price>
<inventory>460</inventory>
<msrp>923.46</msrp>
<supplierID>IBM</supplierID>
</findInventoryRes>
</b:findInventoryResponse>
</soapenv:Body></soapenv:Envelope>
```

Running the Redaction scenario by using the command line

ConsumerA is not a manager so sees a different response. Enter the curl command:

```
curl -k --data-bin @./silver.xml -H "Content-Type: text/xml"
-u ConsumerA:passw0rd http://<yourDataPowerHostName>:62005/Store/Store
```

Notice that the response has the price redacted. The price is displayed as 0.0:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header><KD4NS:KD4SoapHeaderV2
xmlns:KD4NS="http://www.ibm.com/KD4Soap">AFIAAgAkZmExODgzNTQtY2Q1ZC0zYjU0L
WEyMzItZGM3MmEzNWY0MTAzACRmYWVjYjA1Mi1jMWUxLTMyODEtOWY3Ni0wY2IxNm
RhMDc4MjkAAw==</KD4NS:KD4SoapHeaderV2>
</soapenv:Header>
<soapenv:Body>
<b:findInventoryResponse xmlns:a="http://company.ibm.com/"
xmlns:b="http://company.ibm.com/store">
<findInventoryRes>
<sku>SKU10</sku>
<price>0.0</price>
<inventory>460</inventory>
<msrp>923.46</msrp>
<supplierID>IBM</supplierID>
</findInventoryRes>
</b:findInventoryResponse>
</soapenv:Body></soapenv:Envelope>
```

Testing the routing policy by using the command line

For the routing policy attached to the gold SLA to be enforced, the context ID and consumer ID must be matched. In this case, the SLA for Gold Customers has the context ID of Gold, and the consuming service version has the consumer ID of CEO. Here is the content of a sample request (you can see that the context ID and consumer ID match as required):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:stor="http://company.ibm.com/store">
<soapenv:Header>
  <store:ConsumerIdentifier xmlns:store="http://store.com">CEO</store:ConsumerIdentifier>
  <store:ContextIdentifier xmlns:store="http://store.com">Gold</store:ContextIdentifier>
</soapenv:Header><soapenv:Body>
<stor:findInventory><findInventoryReq>
  <sku>SKU10</sku>
  </findInventoryReq>
</stor:findInventory>
</soapenv:Body></soapenv:Envelope>
```

Assuming that the example request XML is contained in a file named gold.xml, enter the following curl command:

```
curl -k --data-bin @./gold.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd http://<yourDataPowerHostName>:62005/Store/Store
```

The response is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
  <KD4NS:KD4SoapHeaderV2
  xmlns:KD4NS="http://www.ibm.com/KD4Soap">AFIAAgAkZmExODgzNTQtY2Q1ZC0zYjU0L
  WEyMzItZGM3MmEzNWY0MTAzACRmYWVjYjA1Mi1jMWUxLTMyODEtOWY3Ni0wY2IxNm
  RhMDc4MjkAAw==</KD4NS:KD4SoapHeaderV2>
</soapenv:Header><soapenv:Body>
<b:findInventoryResponse xmlns:a="http://company.ibm.com/"
xmlns:b="http://company.ibm.com/store">
<findInventoryRes>
  <sku>GOLDSKU10</sku>
  <price>461.73</price>
  <inventory>460</inventory>
  <msrp>923.46</msrp>
  <supplierID>IBM</supplierID>
</findInventoryRes></b:findInventoryResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Note the return response has a GOLDSKU for the SKU value, indicating that the gold endpoint was used.

Testing the validation of the schema by using the command line

The validation policy checks the schema of the request against the Store.wsdl and its associated Company.xsd.

The following XML, badvalid.xml, shows a request that is invalid because the body contains an element named <skubad> when it should be <sku>:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:stor="http://company.ibm.com/store">
<soapenv:Header>
<store:ConsumerIdentifier xmlns:store="http://store.com">CEO</store:ConsumerIdentifier>
<store:ContextIdentifier xmlns:store="http://store.com">silver</store:ContextIdentifier>
</soapenv:Header>
<soapenv:Body>
<stor:findInventory>
<findInventoryReq>
<skubad>SKU10</skubad>
</findInventoryReq>
</stor:findInventory>
</soapenv:Body></soapenv:Envelope>
```

If you enter the following curl request:

```
curl -k --data-bin @./badvalid.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd
http://<yourDataPowerHostName>:62005/Store/Store
```

The following error is displayed:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Body>
<env:Fault><faultcode>env:Client</faultcode>
```

```
<faultstring>Internal Error (from client)</faultstring>
</env:Fault>
</env:Body>
</env:Envelope>
```

Testing rejection in the mediation policy by using the command line

One of the mediation policies included in the sample tests rejection after the message count runs 5 times in 90 seconds. Run the following command 6 times:

```
curl -k --data-bin @./silver.xml -H "Content-Type: text/xml" -u ConsumerX:passw0rd
http://<yourDataPowerHostName>:62005/Store/Store
```

The sample request is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<KD4NS:KD4SoapHeaderV2
xmlns:KD4NS="http://www.ibm.com/KD4Soap">AFIAAgAkZmExODgzNTQtY2Q1ZC0z
YjU0LWEyMzItZGM3MmEzNWY0MTAzACRmYWVjYjA1Mi1jMWUxLTMyODEtOWY3Ni0wY2IxN
mRhMDc4MjkAAw==</KD4NS:KD4SoapHeaderV2>
</soapenv:Header>
<soapenv:Body>
<b:findInventoryResponse xmlns:a="http://company.ibm.com/"
xmlns:b="http://company.ibm.com/store">
<findInventoryRes>
```

In this case, ConsumerX is a Manager, therefore, the full price information is displayed as for the first five runs:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<KD4NS:KD4SoapHeaderV2
xmlns:KD4NS="http://www.ibm.com/KD4Soap">AFIAAgAkZmExODgzNTQtY2Q1ZC0z
YjU0LWEyMzItZGM3MmEzNWY0MTAzACRmYWVjYjA1Mi1jMWUxLTMyODEtOWY3Ni0wY2IxN
mRhMDc4MjkAAw==</KD4NS:KD4SoapHeaderV2>
</soapenv:Header>
<soapenv:Body>
<b:findInventoryResponse xmlns:a="http://company.ibm.com/"
xmlns:b="http://company.ibm.com/store">
<findInventoryRes>
<sku>SKU10</sku>
<price>461.73</price>
<inventory>460</inventory>
<msrp>923.46</msrp>
<supplierID>IBM</supplierID>
</findInventoryRes></b:findInventoryResponse>
</soapenv:Body>
</soapenv:Envelope>
```

On the sixth run the following error occurs:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Body>
<env:Fault>
<faultcode>env:Client</faultcode>
<faultstring>Rejected (from client)</faultstring>
</env:Fault>
</env:Body>
</env:Envelope>
```

Note: You might see this error sooner if you run other tests within the 90-second interval.

Testing notification in the mediation policy by using the command line

The notify policy is attached to the anonymous SLA. This is enforced when a request comes in from a consumer who does not have an SLA in place. In this sample, the only consumer that has SLAs in place is CEO, so a request containing the consumer ID set to anything else causes the policy on the anonymous SLA to be enforced. In this case ConsumerX is a Manager, so the full price information is displayed:

To test this functionality by using the command line, create a file named `anon.xml` that contains the following xml:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:stor="http://company.ibm.com/store">
<soapenv:Header>
  <store:ConsumerIdentifier xmlns:store="http://store.com">ABC</store:ConsumerIdentifier>
  <store:ContextIdentifier xmlns:store="http://store.com">Gold</store:ContextIdentifier>
</soapenv:Header><soapenv:Body>
<stor:findInventory><findInventoryReq>
  <sku>SKU10</sku>
  </findInventoryReq>
</stor:findInventory>
</soapenv:Body></soapenv:Envelope>
```

Then enter the following command:

```
curl -k --data-bin @./anon.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd http://<yourDataPowerHostName>:62005/Store/Store
```

The following message is output in the default log of the domain:

```
Notify action triggered ('operation_38_2_sla1-1-filter_1-notify') from source policy (
   'LogEveryTime_287d0790-83d9-11e1-a255-9187e20cddb0_05aec6ec-3674-4165-85de-a0f7be48a938'
```

Note: Logging must be set to “notice” to see this message. If it is not, click the **Troubleshooting** icon in the DataPower Web Console. In the Logging section, change the Log level value to “notice” and click **Set Log Level**. To find the log, return to the Control Panel and click the **View Logs** icon.

Testing the RESTful service by using the command line

You can also access the RESTful interface from the command line by using curl. As with the web client, a ContextID of Gold permits 5 messages per 90 seconds and Silver only 2 messages.

To test this functionality by using the command line, create a file named restRequest.xml that contains the following xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<a:WarehouseSKUPost xmlns:a="http://company.ibm.com/">
   <postRequest>
      <sku>SKU33</sku>
      <purchaseCost>136.43</purchaseCost>
      <inventory>429</inventory>
      <msrp>272.86</msrp>
      <returns>0</returns>
   </postRequest>
</a:WarehouseSKUPost>
```

Then enter the following command to test with contextID Gold:

```
curl -k --data-bin @./restRequest.xml -H "Content-Type: text/xml" -H "consumerID:CEO" -H "contextID:Gold"  http://<yourData
```

To test with the silver contextID use the same command, but replace Gold with Silver.

A successful response is:

```
<?xml version="1.0" encoding="UTF-8"?>
<a:WarehouseSKUGet xmlns:a="http://company.ibm.com/">
<getRequest>
<sku>SKU33</sku>
<purchaseCost>136.43</purchaseCost>
<inventory>429</inventory>
<msrp>272.86</msrp>
<returns>0</returns>
<supplierID>ABB</supplierID>
<purchaseID/>
</getRequest>
</a:WarehouseSKUGet>
```

After the threshold has been breached you receive the following message:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"><env:Body><env:Fault><faultcode>env:Client</faultcode><
```

To exercise the anonymous SLA for the RESTful service, which simply has a notify policy attached, use any ContextID and ConsumerID other than those registered. The notify appears in the DataPower log as described earlier for the Web Services example.

Related tasks:

"Deploying the basic runtime sample pattern" on page 49
Deploying the SOA Policy Gateway Basic Runtime Sample pattern creates a running virtual system instance of the pattern. This pattern is available only on x86 systems.

Extending the sample application

The sample application can be modified by modifying the Bindings style sheet and the XSL style sheets.

Modifications to the Bindings style sheet

The variable xacml-subjects has been added to the style sheet `apil-xacml-binding-new.xsl`. It encompasses the creation of the subjects section of the request. This variable is later accessed in `sendToPDP.xsl`.

```
<xsl:variable name="xacml-subjects">
  <xacml-context:Subject
  SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<!--
************************************************
Starting here, use the MC result as subject.
************************************************
```

sendToPDP.xsl

This style sheet calls the StoreXACMLFW by using url-open. The call is on box to another XML Firewall, so no SSL Proxy profile is used. To move the Policy Decision Point (PDP) to another DataPower box, an SSL Proxy profile could be created and used with the url-open call.

```
<xsl:param name="resource" />
<!--
<xsl:variable name="incoming_resource">
<xsl:value-of select="$resource" />
</xsl:variable>
<xsl:message dp:priority="debug">
********** ABOUT TO CALL PDP for RESOURCE equal *************
<xsl:value-of select="$incoming_resource" />
</xsl:message>
-->
- <!--
building the XACML request for masking
-->
<xsl:variable name="customized-request">
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header />
<soapenv:Body>
<xacml-context:Request xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:ws="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
- <!--
copy in the subjects saved from AAA request processing
-->
<xsl:copy-of select="dp:variable('var://context/snip/xacml/xacmlSubjects')/*" />
<xacml-context:Resource>
<xacml-context:ResourceContent>
<xsl:copy-of select="./soap:Envelope/soap:Body" />
</xacml-context:ResourceContent>
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>PriceInfo</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action>
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>View</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment>
<xacml-context:Attribute AttributeId="ContextId" DataType="http://www.w3.org/2001/XMLSchema#string"
Issuer="http://security.tivoli.ibm.com/policy/distribution">
<xacml-context:AttributeValue>StorePriceData</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Environment>
</xacml-context:Request>
</soapenv:Body>
</soapenv:Envelope>
</xsl:variable>
- <!--
Use set-variable so that it is visible in Probe, which is convenient
-->
<dp:set-variable name="'var://context/snip/xacml/BacksideXacmlRequest'" value="$customized-request" />
- <!--
Report the XACML-REQUEST to the debug log
-->
<xsl:message dp:priority="debug">
<XACML-REQUEST>
<xsl:copy-of select="$customized-request" />
</XACML-REQUEST>
</xsl:message>
<xsl:variable name="headers">
<header name="SOAPAction">xacml:authorization</header>
</xsl:variable>
- <!--
```

```
Call the XACML PDP for decision
-->
<xsl:variable name="rtss-response">
<xsl:variable name="StoreGWURL">
<xsl:value-of select="concat('http://','127.0.0.1',':',$StoreGWPort,'/rtss/authz/services/AuthzService')" />
</xsl:variable>
<dp:url-open target="{$StoreGWURL}" http-headers="$headers" response="responsecode">
<xsl:copy-of select="$customized-request" />
</dp:url-open>
</xsl:variable>
- <!--
Use set-variable so that it is visible in Probe, which is convenient
-->
<dp:set-variable name="'var://context/snip/xacml/BacksideXacmlResponse'" value="$rtss-response" />
- <!--
Report the XACML-RESPONSE to the debug log
-->
<xsl:message dp:priority="debug">
<XACML-RESPONSE>
<xsl:value-of select="$rtss-response" />
</XACML-RESPONSE>
</xsl:message>
</xsl:template>
</xsl:stylesheet>
```

Note the following points about the `sendToPDP.xsl` file:

1. The style sheet obtains the port for the XACMLFW from `soavars.xsl`.
2. The variable rtssResponse is expected to be of exactly the form Runtime Security Services would use, and in turn of the form that the DataPower on-box PDP can process.
3. The style sheet constructs a SOAP request. The subject information is constructed by the earlier `apil-binding.xsl` style sheet and is obtained by the following copy of select request:

```
<xsl:copy-of select="dp:variable('var://context/snip/xacml/xacmlSubjects')/*" />
```

4. The action is simply to view the action: `<xacml-context:AttributeValue>View</xacml-context:AttributeValue>`
5. The environment is the StorePriceData, known as an Application object in IBM Tivoli® Security Policy Manager or Runtime Security Services terminology.

StorePrivateDataXACML.xml

The following code shows the policy style sheet for redaction.

```
<PolicySet PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides"
PolicySetId="RPS:StorePrivateData:policy:dc703409-d408-49b3-acc1-16c89c844fce:rolec4a9f664-a0af-451b-b80b-
1cafdb9fd9f0:role:2884ab77-58d1-4b1d-8728-7d528169d608" Version="1.0">
<Target>
<Subjects>
<Subject>
<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">MANAGER</xacml:AttributeValue>
<SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#string"
SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:accesssubject"
/>
</SubjectMatch>
</Subject>
</Subjects>
</Target>
<Policy PolicyId="PPS:StorePrivateData:dc703409-d408-49b3-acc1-16c89c844fce:c4a9f664-a0af-451b-
b80b-1cafdb9fd9f0:c4a9f664-a0af-451b-b80b-1cafdb9fd9f0:pps"
```

```
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
Version="1.0">
<Target>
<Resources>
<Resource>
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">PriceInfo</xacml:AttributeValue>
<ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
</ResourceMatch>
</Resource>
</Resources>
<Actions>
<Action>
<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">View</xacml:AttributeValue>
<ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
</ActionMatch>
</Action>
</Actions>
</Target>
<Rule Effect="Permit" RuleId="PPS:StorePrivateData:dc703409-d408-49b3-acc1-16c89c844fce:c4a9f664-a0af-451b-
b80b-1cafdb9fd9f0:c4a9f664-a0af-451b-b80b-1cafdb9fd9f0:pps:rules:0">
<Target />
</Rule>
</Policy>
</PolicySet>
</PolicySet>
```

Note the following points:

- The Role must be Manager:

```
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">MANAGER</xacml:AttributeValue>
```

- The Resource must be PriceInfo:

```
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
 xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">PriceInfo</xacml:AttributeValue>
```

- The Action must be View:

```
<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
 xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">View</xacml:AttributeValue>
```

Modifying the sample XSL style sheets

You can modify the redaction style sheet, `noPriceInfo.xsl`

Procedure

Modify the Redaction style sheet.
The `noPriceInfo.xsl` style sheet contains the following code, which will replace any price values with zeroes. You can add other fields to the redaction logic, or add more complicated transformations that involve computation to determine values for fields.

```
<!-- private access only fields -->
<xsl:template match="price">
<price>0.0</price>
</xsl:template>
<xsl:template match="Price">
<Price>0.0</Price>
</xsl:template>
```

Later, the style sheet performs an identity transform on all other elements.

Further exploration of the sample

To learn more about the sample, you can configure the XACML Policy Decision Point (PDP) on DataPower and edit policy documents.

Altering the XACML PDP on DataPower

You can explore altering the XACML used for the security Policy Decision Point (PDP) in DataPower to learn more about access control with XACML.

Procedure

To change or add a PDP:

1. From the DataPower Control Panel, search for `XACML PDP`.
2. Either click an existing PDP or click **Add**.
3. Enter a URL, for example, `local:///storePrivateDataXACML.xml`.
4. Add any dependent or directory files that are required to support the policy.

   Note: If you edit an XACML policy file directly on the file system, you must go back to the PDP definition and reenter the URL, or anything you have changed, or restart the domain for your change to take effect.

Adding new or editing existing policy documents

Use the Business Space user interface to add new policy documents or edit existing ones.

Before you begin

Configure the SOA Governance space. For more information, see "Configuring Business Space for the first use" on page 81.

Procedure

1. Create a mediation policy with the conditions and actions you require; for example, a condition of Message Count > 5 messages in 5 minutes and an action of reject. For more information about creating a mediation policy, see "Authoring new mediation policies" on page 95.
2. Govern the mediation policy. For more information about governing a policy document, see "Managing the lifecycle of the policy" on page 97.
   a. Click the policy document in the Service Registry Navigator or search for it in the search widget. The actions are displayed in the Policy Document Editor.
   b. Click **Propose Specification**.
   c. Click **Approve Specification**.

   The policy is approved. You can redefine, supercede, or deprecate the policy to manage the lifecycle or edit an existing definition.
3. Attach the policy. In Business Space, find the SLD or SLA that you want to attach the policy to. There are four places you could do this in the sample:
   - Store SLD - attach your policy here if you want it to apply to any use of the Store service.
   - Gold SLA - attach your policy here if you want it to apply only to Gold requests from the CEO consumer.

- Silver SLA - attach your policy here if you want it to apply only to Silver requests from the CEO consumer.
- Anonymous SLA - attach your policy here if you want it to apply to any requests coming from consumers other than CEO.

Related tasks:

"Authoring new mediation policies" on page 95
You can create new mediation policies by using the Business Space user interface. When you author mediation policies, you specify the conditions and actions for the policy.

"Managing the lifecycle of the policy" on page 97
Policies can be transitioned between governance states by using the Business Space user interface. Polices must be in the Approved state to be enforced by DataPower.

Related information:

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

The DataPower sample domain

The pattern provides a sample DataPower domain, that enables you to start using the pattern. As a DataPower developer, you can use the existing gateways as a template for your own applications. The sample environment contains five gateways. There is one primary gateway for the Store service, and four supporting gateways that provide example back-ends for the Store Gateway to call, XACML support for a redaction scenario, and a front end to provide extra security functionality.

Store Web Service Proxy

The Store Web Service Proxy (WSP) is the primary gateway of the application domain. It receives a request with an LTPA token attached.

When requested, the processing rule for the request completes the following actions:

1. Validates the request, as requested by the Validation policy. For more information, see "Overview of WSRR artifacts in the sample" on page 58.
2. Routes the request to the alternate endpoint if the service level agreement (SLA) is "Gold".
3. Authenticates, completes authorization, and accounting (AAA) on the request. The authentication includes the following actions:
   a. Authenticates the user with an LTPA token.
   b. Maps the credentials against the LDAP server that provides information as to which groups the customer belongs. These groups include Manager, Clerk, and Customer.
   c. Transforms the provided inputs into a request object that the XACML policy decision point (PDP) can understand.
   d. Completes authorization by using an XACML PDP on the DataPower box, with an XACML policy document that can be created in IBM Tivoli Security Policy Manager. The criteria of the policy is that the user must be a Manager, Customer, or Clerk. For the findInventory operation, the returns require either Manager or Clerk, and purchases can be made by customers.
4. Sets the ConsumerID value by using an XSL script.
5. Removes the entire HTTP Security Header from the request.

6. Calls the Store service back end.

When the request is processed, the response processing rule completes the following actions:

1. Calls the StoreXACMLFW gateway, that acts as the PDP in the scenario.
2. Based on the response, the price info field is redacted (zeroed out) depending on if the user has the Manager role or not.

XML firewalls in the sample

The following XML firewalls are defined in the sample.

StoreAddLTPA XML firewall

The function of the StoreAdd LTPA XML firewall is to provide a front end with a port that users can call by using only Basic authentication (for example, no LTPA). The request processing rule:

1. Identifies with Basic authentication.
2. Authenticates with a simple LDAP lookup.
3. Adds an LTPA token as part of the post processing.
4. Forwards the request to the StoreWSP security policy with the LTPA information now attached.

StoreMockService XML firewall

The StoreMockService is an example service that uses an XML Firewall as an implementation. The findInventory, purchase, and return operations all are supported. The response values are static. This example service is created when it is not possible to include a WebSphere Application Server in the pattern. The three request rules of the policy use a matching action to determine the request operation and based on a match, respond with a static SOAP response. Static SOAP responses are provided based on the request operation instead of a full service implementation.

StoreMockServiceAlternate XML firewall

The StoreMockServiceAlternate is an example service that uses an XML Firewall as an implementation. The findInventory, purchase, and return operations all are supported. This service is used to demonstrate enforcement of the routing policy.

StoreXACMLFW firewall

This scenario performs redaction based on the result of an XACML-based permit/deny mechanism. In DataPower, there is no way to call an individual AAA action in the response flow. A separate gateway is created to contain the XACML Policy Decision Point (PDP). This PDP was encapsulated in an AAA action on the request rule of the StoreXACMLFW.

StoreXACMLFW is an XML firewall gateway in DataPower. This implementation is used because it is a simple way to provide the functionality. The StoreXML firewall uses the same WSDL interface as the Tivoli Runtime Security Services server. The StoreWSP gateway creates the request object and sends it, protected by SSL, to the StoreXMLFW gateway.

The request rule of the StoreXML firewall does the following tasks:

1. Performs AAA by using the SSL information for authentication.

2. Performs authorization by using an on-box XACML PDP. The policy that is used by the PDP is originally authored in IBM Tivoli Security Policy Manager but can be recreated by using a standard editor, and the schema is defined in the XACML specification.
3. No transformation of the request is necessary in this authorization processing.
4. If the XACML request is valid, the request processing rule does a fetch of a Permit response and returns to the client. Otherwise, an exception occurs that is handled by the exception processing rule and returns a Deny response to the client.

Note: The Permit/Deny/Indeterminate is an example-level response only. Additional error information could be included in a customer-specific flow.

XACML security policy

This topic describes how XACML documents are created.

The XACML documents that are used in the sample were created by the IBM Tivoli Security Policy Manager policy editor, but you can use any text or XML editor to create such documents. To construct or modify existing XACML policies, see the OASIS specifications: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

The XACML security policy that is used in the sample is contained in `storeSWPXACML.xml` and `storePrivateDataXACML.xml`. These policies are used to evaluate the request coming in to the policy decision point (PDP). The request is made up of four key elements:

1. The Subjects section - Contains the details of the Distinguished Name of the request caller, as well as the groups that the caller belongs to.
2. The resource section - Contains the documents that the caller wants to have access to. Two types of resource are used in the sample. The first type is the operation on the web service and the second type is the authorization to the data on the response, in this case the priceInfo resource.
3. The Environment section - Contains information about the environment of the request.
4. The action - What the user wants to do with the authorized material. In the redaction scenario, the action is simply to view the priceInfo data.

StoreWSP security policy

The security policy in the `storeSWPXACML.xml` file maps groups to Web Service Operations.

An example security policy is as follows:

```
<PolicySet PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:denyoverrides"
PolicySetId="RPS:Store:policy:aed2df4e-4159-4df0-ada2-f148d9b56cef:roled200c213-27f9-
4d17-8305-b0d3ca8fcf54:role:09b60522-76b8-4280-9c1a-31d026441164" Version="1.0">
<Target>
<Subjects>
<Subject>
<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:x500Name-equal">
<xacml:AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">CN=MANAGER, CN=groups,
DC=ibm.com</xacml:AttributeValue>
<SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:group-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name"
SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" />
</SubjectMatch>
</Subject>
```

```
<Subject>
<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">MANAGER</xacml:AttributeValue>
<SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#string"
SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" />
</SubjectMatch>
</Subject>
</Subjects>
</Target>
<Policy PolicyId="PPS:StoreSOAP:findInventory:aed2df4e-4159-4df0-ada2-f148d9b56cef:d200c213-27f9-
4d17-8305-b0d3ca8fcf54:d200c213-27f9-4d17-8305-b0d3ca8fcf54:pps"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
Version="1.0">
<Target>
<Resources>
<Resource>
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">{http://company.ibm.com/store}findInventory</xa
cml:AttributeValue>
<ResourceAttributeDesignator AttributeId="urn:ibm:xacml:profiles:web-services:1.0:wsdl:1.1:operation"
DataType="http://www.w3.org/2001/XMLSchema#string" />
</ResourceMatch>
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">{http://company.ibm.com/store}StoreSOAP</xac
ml:AttributeValue>
<ResourceAttributeDesignator AttributeId="urn:ibm:xacml:profiles:web-services:1.0:wsdl:1.1:port"
DataType="http://www.w3.org/2001/XMLSchema#string" />
</ResourceMatch>
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">{http://company.ibm.com/store}Store</xacml:Attr
ibuteValue>
<ResourceAttributeDesignator AttributeId="urn:ibm:xacml:profiles:web-services:1.0:wsdl:1.1:service"
DataType="http://www.w3.org/2001/XMLSchema#string" />
</ResourceMatch>
</Resource>
</Resources>
</Target>
```

Note: In the subjects section, a match occurs on the x500 name or the subject role of Manager. If you examine the entire policy `.xml` file, you can see that there are similar mappings for Customer and Clerk. You can see that the findInventory operation is authorized to use all three groups while the returnProduce and purchase operations are limited to only certain groups.

The Redaction Gateway

Details about the storeCallPDP.xsl style sheet.

Examine the storeCallPDP.xsl style sheet, and note the following points:

1. The inclusion of the storeSendToPDP.xsl style sheet. This style sheet contains the logic to call storeXAMLFW.
2. The call to the template call_PDP inside storeSendToPDP.
3. The extraction of the decision from the response of the call, for example, “Permit”.
4. The setting of the `var:/context/response/displayfilter` value to either the allData.xsl or noPriceInfo.xsl style sheets.

5. The structure in the XACML for the Reaction, storePrivateDataXACML.xml, is nearly identical to the structure used in the StoreWSP scenario. The difference is that only the Manager role has access.

storeCallPDP.xsl

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:dp="http://www.datapower.com/extensions"
extension-element-prefixes="dp" exclude-result-prefixes="dp">
  <xsl:include href="storeSendToPDP.xsl" />
    <xsl:template match="/">
      <xsl:call-template name="call_PDP">
        <xsl:with-param name="resource" select="'StorePrivateData'" />
      </xsl:call-template>
      <xsl:variable name="decision">
        <xsl:copy-of select="dp:variable('var://context/snip/xacml/BacksideXacmlResponse')/
    *[local-name()='url-open']/*[localname()='response']/*[local-name()='Envelope']/*[local-name()='Body']/
    *[local-name()='Response']/*[local-name()='Result']/*[localname()='Decision']" />
      </xsl:variable>
      <xsl:message dp:priority="debug">
        <DECISION-FROM-RTSS>
          <xsl:value-of select="$decision" />
        </DECISION-FROM-RTSS>
      </xsl:message>
      <xsl:choose>
      <xsl:when test="$decision = 'Permit'">
        <xsl:message dp:priority="debug">****** SETTING THE PRIVATE FILTER ********</xsl:message>
        <dp:set-variable name="'var://context/response/displayFilter'" value="'local:///allData.xsl'" />
      </xsl:when>
      <xsl:otherwise>
        I<dp:set-variable name="'var://context/response/displayFilter'" value="'local:///noPriceInfo.xsl'" />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

WSRR artifacts created in the SOA Policy Gateway Basic Runtime Sample

The WSRR artifacts created in the SOA Policy Gateway Basic Runtime Sample pattern, and how the sample uses them.

Table 14. WSRR artifacts created for the SOA Policy Gateway Basic Runtime Sample pattern

Object	Description
Organization	Bob's Warehouse. This is the area of the business that owns the Store service
Business Capability	Warehouse. This represents all versions of the Store service, and is owned by the Bob's Warehouse organization.
Service Version	Store. This represents version 1.0 of the Store service.
WSDL	Store.wsdl
XSD	Company.xsd
Policy	• Validate.xml • RouteForGold.xml • LogEveryTime.xml • RejectAfter5MsgIn90Seconds.xml
SLD	Store SLD. Any policies attached here apply to any request for this service.

Table 14. WSRR artifacts created for the SOA Policy Gateway Basic Runtime Sample pattern (continued)

Object	Description
Gold SLA	Gold SLA. The existence of this SLA means gold requests from the consumer CEO are not be counted as anonymous. Any policies attached here are enforced on gold requests from the consumer CEO.
Silver SLA	Silver SLA. The existence of this SLA means silver requests from the consumer CEO are not be counted as anonymous. With no policies attached, the request is allowed through.
Anonymous SLA	Anonymous Users. Policies attached here are enforced on any requests that do not have a matching SLA in place. In this sample, any request from a consumer other than CEO, or any request from CEO that is not Gold or Silver, have the Anonymous SLA policies enforced on it.

DataPower artifacts created in the SOA Policy Gateway Basic Runtime Sample

The DataPower artifacts created in the SOA Policy Gateway Basic Runtime Sample pattern.

Table 15. DataPower artifacts created for theSOA Policy Gateway Basic Runtime Sample pattern

Type	Name	Purpose
WebService Proxy	StoreWSP	The principal service.
XML Firewalls	StoreAddLTPA StoreMockService StoreAlternateMockService StoreXACMLFW	Authenticates and adds the LTPA Token. The service provider for non-Gold customers The service provider for Gold customers Checks the access to PriceInfo.
WSRR Server	WSRRSVR	The connection to WSRR.
WSRR Subscription	StoreSub	Provides search information for the WSRR namespace, object, and so on.
AAA Policy	StoreAddLTPA	Basic authentication and identification for LDAP. Looks-up authentication. Adds the LTPA token to the request.
AAA Policy	StoreWSDLAAA	LTPA identification and authentication. Group mapping for the authorization. XACML authorization.
AAA Policy	StoreXACMLFWAZ	XACML authorization for PriceInfo.
SSL Proxy Profile	WSRRPP	SSL proxy profile for the WSRR Server.
Crypto Profile	WSRRCP	Crypto profile for the WSRR Server.

Table 15. DataPower artifacts created for theSOA Policy Gateway Basic Runtime Sample pattern (continued)

Type	Name	Purpose
Validation Credentials	WSRRVC	Validation credentials contain the Crypto certificate WSRRCERT. All other settings are default.
Crypto Certificate	WSRRCERT	WSRRCERT uses the signer certificate. This certificate was either extracted from the NodeDefaultKeyStore, default certificate for a single server, or from the CMSKeyStore default certificate in the case of an ND environment where an IBM HTTP Server was present.

The StoreWSP Web Service Proxy processing rules

The central gateway of the sample is StoreWSP. The Policy for the gateway contains a request and response rule.

Request rule

The primary policy action of the StoreWSP_default_request-rule is called AAA. In the AAA action, the LTPA Token is validated, the users groups are retrieved, and an authorization is performed to see if the user is in the Manager, Clerk, or Customer LDAP group. This validation is performed when the AAA AZ step calls the StoreWSDLPDP Policy Decision Point (PDP), on the DataPower appliance. This PDP uses the storeWSPXACML.xml XACML policy.

Response rule

In the response rule, StoreWSP_default_response-rule, the transform calls the StoreXACMLFW XML firewall service.

This transform determines whether the user is authorized to access the price information based on whether the user is a member of the Manager group. If they are, the *var:///context/response/displayFilter* variable is set to `local:///allData.xsl`. If they are not a member of the Manager LDAP group, the *var:///context/response/displayFilter* variable is set to `local:///noPriceInfo.xsl`.

The transform then performs the style sheet actions on the response.

StoreXAMLFW Processing Rules

The custom style sheet storeSendToPDP.xsl makes a call to the local XML FW StoreXACMLFW. There are two processing rules used in this firewall. The StoreXACMLFW_request contains a single AAA policy action that uses the allData.xsl transform. This AAA action, StoreXACMLFWAZ, in turn calls the XACML PDP StorePDP action. Using the storePrivateDataXACML.xml XACML policy, a determination is made whether the user is authorized to the price information.

The sample XSL style sheets

The sample application contains the following style sheets ending in `.xsl`, which are located in the local directory of the installed domain.

Table 16. Style sheets in the sample application

Style sheet	Purpose
allData.xsl	An Identity style sheet that copies all of the data from the source to the target. It is used both for the Redaction function and for the call to the XACML XML Gateway.
apil-xacml-binding-new.xsl	Uses the credential mapping information to create a SOAP request that can be processed by the DataPower appliance Policy Decision Point (PDP). This style sheet is a modification of the tspm-xacml-binding-sample.xsl style sheet that is provided in the store directory of the DataPower appliance. The key functionality that is provided by this adapted script is to add an externally accessible variable that makes the subject information of the XACML request available to the redaction style sheet.
noPriceInfo.xsl	This style sheet sets the price element to a value of 0.0.
rgxacml.xsl	This style sheet is a customization of the tspm-retrieve-groups.xsl style sheet in the store directory of the DataPower appliance. The primary purpose of this style sheet is to provide the LDAP DN, hostname, password, port, and so on, so that the incoming user can be looked up and their group information retrieved.
soavars.xsl	This style sheet is an example only style sheet that defines the LDAP information in variables used by the rgxacml.xsl style sheet. In the example the password is unencrypted, which is not a production practice.
storeCallPDP.xsl	This style sheet has the code to call the XACML Gateway, handles the Permit/Deny decision, and sets the filter variable to run either allData.xsl or noPriceInfo.xsl.
storeSendToPDP.xsl	This style sheet constructs a SOAP Request that is sent to the XACML Gateway. It includes the subject information that is obtained in the apil-xacml-binding-new.xsl style sheet, the resource information, the action information, and the environment information.

DataPower objects that use the XSL style sheets

The DataPower objects use some of the XSL style sheets that are provided with the sample application.

Table 17. DataPower objects that use the XSL style sheets

Style sheet	Purpose
allData.xsl	Used internally in the storeCallPDP.xsl style sheet. The style sheet is used as the custom transform in AAA policy StoreXACMLFWAZ.
apil-xacml-binding-new.xsl	Used as the custom style sheet in StoreWSDLAAA AAA policy AZ step.
noPriceInfo.xsl	Used internally in the storeCallPDP.xsl style sheet.
soavars.xsl	Used internally in the rgxacml.xsl style sheet.
storeCallPDP.xsl	Called as a transform in the Store_default-response rule.
storeSendToPDP.xsl	Used internally in the storeCallPDP.xsl style sheet.

Chapter 6. Working with the deployed instance

After one of the IBM SOA Policy Gateway Patterns is deployed, you can view the deployed instance by clicking **Instances** > **Virtual systems** in the workload console.

Viewing the instance details

You can see the details of a deployed instance by selecting it from the list of instances in the Virtual System Instances window. The virtual system instance details are displayed. The details include a list of virtual machines that are provisioned on the cloud infrastructure for that deployment, the IP address, and virtual machine status.

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

To see the status of the virtual machines and scripts during provisioning, 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. The script logs are accessible in the **Script Packages** section. You can connect to any consoles available by using the links in the **Consoles** section.

Accessing deployed instances

After deploying a virtual system pattern, you can view the virtual system instance that was created to see your IBM SOA Policy Gateway Pattern environment, and access its component parts.

Before you begin

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

About this task

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

Procedure

To administer the IBM SOA Policy Gateway 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, you can log in to the components of the virtual system from the console links in the virtual system view. The components that are available depend on the pattern that you created. They can include:
   - WebSphere Application Server administrative console
   - WSRR Web UI
   - WSRR Business Space
   - DataPower WebGUI

Connecting to WSRR - Business Space

Use the Business Space user interface to work with WSRR..

About this task

Business Space is one of two graphical interfaces that you can use to work with WSRR. A full description of using Business Space with WSRR is in the WSRR Information Center (see the related link).

You can connect to a WSRR instance Business Space in your deployed pattern by clicking a link in the workload console, or by entering the URL in a web browser.

Procedure

1. To connect from the workload console:
   a. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
   b. From the list of instances in the Virtual System Instances window, select your deployed system.
   c. Click **Virtual machines** in the detail view of the deployed system to expand the list.
   d. Locate WSRR in the list of virtual machines and click the plus sign to view details.
   e. Under the **Consoles** section, click **WSRR_Business_Space**.
   f. Enter the WSRR administrative user ID and password.
2. To connect from a web browser:
   a. Open a web browser.
   b. Find the host name and port numbers for WSRR. View details of your deployment as described in step 1. Expand the **Virtual machines** section and select the virtual machine for the WSRR Server to see the virtual machine details. In the **Hardware and network** section, the host name is the **Network interface 0** value.
   c. Enter the WSRR Web UI URL: http://*hostname*:9443/BusinessSpace, where *hostname* is the host name of the WSRR server.
   d. Enter the WSRR administrative user ID and password.

Results

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

What to do next

If you are using Business Space on the WSRR system for the first time, see "Configuring Business Space for the first use" and follow the steps to create the SOA Governance space.

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 SOA Governance space must be created.

Before you begin

For information about accessing Business Space, see "Connecting to WSRR - Business Space" on page 80.

About this task

To use the Business Space widgets, you must create a Space. Spaces are defined for specific roles. Policy authoring is best suited for working with in the SOA Governance space. If a SOA Governance space does not yet exist, you must create it. To create a space that is based on the Service Registry for SOA Governance template, complete these steps:

Procedure

1. Click **Manage Spaces** 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, `SOA Governance`. Optionally, enter a description.
4. Select **Service Registry for SOA Governance** 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 to it.

Results

The SOA Governance space is created. To open the SOA Governance space:

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

What to do next

You can add more actions to the Service Registry Actions widget:

1. In Business Space, click **Edit Page**.
2. In the Service Registry Actions widget, click **Edit Settings**.
3. Select the following actions to display:
   - Create a Service Level Definition

- Create a Service Version
- Create a Service Level Agreement
- Create a Business Capability

4. In the Service Registry Actions widget, click **Save and Close**.
5. Click **Finish Editing**.

Connecting to WSRR - WSRR Web UI

Use the WSRR Web UI to work with WSRR.

About this task

The WSRR Web UI is one of two graphical interfaces that you can use to work with WSRR. A full description of using the WSRR web UI is in the WSRR Information Center (see the related link). In most cases you might prefer to use the Business Space interface, but there are some tasks (such as creating monitoring policies) that must be completed in the WSRR Web UI.

You can connect to the WSRR Web UI of a WSRR instance in your deployed pattern by clicking a link in the workload console, or by entering the URL in a web browser.

Procedure

1. To connect from the workload console:
   a. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
   b. From the list of instances in the Virtual System Instances window, select your deployed system.
   c. Click **Virtual machines** in the detail view of the deployed system to expand the list.
   d. Locate WSRR in the list of virtual machines and click the plus sign to view details.
   e. Under the **Consoles** section, click **WSRR_Web_UI**.
   f. Enter the WSRR administrative user ID and password.
2. To connect from a web browser:
   a. Open a web browser.
   b. Find the host name and port numbers for WSRR. View details of your deployment as described in step 1. Expand the **Virtual machines** section and select the virtual machine for the WSRR Server to see the virtual machine details. In the **Hardware and network** section, the host name is the **Network interface 0** value.
   c. Enter the WSRR Web UI URL: http://*hostname*:9443/ServiceRegistry, where *hostname* is the host name of the WSRR server.
   d. Enter the WSRR administrative user ID and password.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

Connecting to WebSphere Application Server administrative console

Use the WebSphere Application Server administrative console to fine-tune security settings and complete other administrative tasks..

About this task

Full details of working with the WebSphere Application Server administrative console are in the Information center. Follow the related link.

You can connect to the WebSphere Application Server administrative console in your deployed pattern by clicking a link in the workload console, or by entering the URL in a web browser.

Procedure

1. To connect from the workload console:
   a. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
   b. From the list of instances in the Virtual System Instances window, select your deployed system.
   c. Click **Virtual machines** in the detail view of the deployed system to expand the list.
   d. Locate WSRR in the list of virtual machines and click the plus sign to view details.
   e. Under the **Consoles** section, click **WebSphere**.
   f. Enter the WSRR administrative user ID and password.
2. To connect from a web browser:
   a. Open a web browser.
   b. Find the host name and port numbers for WSRR. View details of your deployment as described in step 1. Expand the **Virtual machines** section and select the virtual machine for the WSRR Server to see the virtual machine details. In the **Hardware and network** section, the host name is the **Network interface 0** value.
   c. Enter the WSRR Web UI URL: http://*hostname*:9043/ibm/console, where *hostname* is the host name of the WSRR server.
   d. Enter the WSRR administrative user ID and password.

Related information:

WebSphere Application Server V8.0 Information Center

Connecting to the console of a virtual DataPower

Use the DataPower console to configure the Policy Enforcement Point.

About this task

Full details of configuring your gateway are in the WebSphere DataPower Information Center. Follow the related link.

You connect to the console by using a web browser. You retrieve connection details by viewing details of your deployed pattern in the workload console.

Procedure

1. Retrieve the details you need by using the workload console:
   a. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
   b. From the list of instances in the Virtual System Instances window, select your deployed system.
   c. In the detail view, expand the **Virtual machines** section and select the virtual machine for the DataPower appliance to see the virtual machine details. In the **Hardware and network** section, the host name is the **Network interface 0** value.
2. Open a web browser and enter the URL `https://hostname:9090/dp`, where *hostname* is the host name of your virtual appliance.

Related information:

WebSphere DataPower V6.0 Information Center

Connecting to the monitoring console

Use the monitoring console to view monitoring information.

About this task

Access the monitoring console from the Virtual System Instances window.

The monitoring functionality is provided by ITCAM for SOA. Download the documentation from the related link for more information, and search for information on DataPower installations.

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 and select the virtual machine you want to monitor.
4. Under **General information**, locate **Monitoring**, and click the **Click to open** link.

Related information:

ITCAM for SOA 7.2.1 documentation (from Fix Central)

Stopping and starting the deployed instance

You can stop and start the deployed instance from the workload console. You can also stop and start individual virtual machines in the pattern.

To stop a running deployed instance:

1. Select **Instances** > **Virtual Systems** and select the instance from the **Virtual System Instances** list.
2. Click the **Stop** icon in the instance title bar.

To start a stopped deployed instance:

1. Select **Instances** > **Virtual Systems** and select the instance from the **Virtual System Instances** list.

2. Click the **Start** icon in the instance title bar.

Note: A known defect in DB2 10.1.0.2 results in the DB2 processes not always restarting when the instance stops and restarts. In this case, you must start DB2 process manually, by logging into the DB2 node as db2inst1 and running **db2start**. You might also need to restart the WSRR processes on the WSRR nodes.

To stop individual virtual machines.

1. Expand the **Virtual Machines** section of the instance view.
2. Select the **Manage** link for the machine that you want to stop.
3. Click the stop icon in the manage bar.

To start individual virtual machines.

1. Expand the **Virtual Machines** section of the instance view.
2. Select the **Manage** link for the machine that you want to start.
3. Click the start icon in the manage bar.

You can also stop and start WSRR and DB2 from the command line. Click the **Login** link to connect by using the SSH console.

You stop and start WSRR by stopping and starting the WebSphere Application Server profile. See Managing profiles using commands in the WebSphere Application Server Information Center.

In the Advanced Pattern, after the DMGR and Custom Nodes are restarted, the WSRR cluster needs starting. To do this, open the WebSphere Application Server administrative console and select **Servers** > **Clusters** > **WebSphere Application Server Clusters**. Select **WSRRCluster_1**, then click **Start**.

You can stop and start DB2 by using system commands. See System Commands in the DB2 Information Center.

Post-deployment pattern configuration

After deploying the patterns, you must configure security and other settings.

Configuring the Policy Enforcement Point

The DataPower appliance or instance is the Policy Enforcement Point (PEP) of the IBM SOA Policy Gateway Pattern. When the Application Domain is deployed, it is possible to create the content of that domain.

Procedure

When setting up your configurations, ensure that different domain names are used on each DataPower appliance, otherwise the ITCAM for SOA topology workspaces does not display the correct data .
Create a Web Service Proxy (WSP):

1. From the DataPower Control Panel, click **Web Service Proxy**.
2. Click **Add** and enter a name for the Proxy.
3. Open the **WSRR Subscription** tab. In the WSRR Server list, click **WSRRSVR**.

4. Provide the other information that is required, such as the Front Side Handler, the namespace, the object name, and so on, to create the configuration of the Web Service Proxy.

Create policies for the WSP:

5. Open the **Policy** tab for the WSP Editor.
6. Click **Processing Rules** at the appropriate level. You can either create a new rule or edit the default rule provided. The key policy action to add is the **AAA Action**. This handles the Identification, Authentication, and Authorization that are key to the pattern.

   Key things that you must specify for the AAA action include the Input and Output, as well as the AAA Policy. You can create the policy while creating the AAA Policy Action, or you might create it before this by using the AAA editor.

   - Identification is the step where the user is Identified. In the sample, there are two forms of identification used. In the StoreAddLTPA XML firewall, the identification used basic authentication. In the StoreWSP firewall, identification was provided by LTPA token.
   - Authentication is the step where the user is proved to be a user who is known to the system. There are many options to choose from. In the sample, there are two examples; the first where the user was looked up using LDAP, and the second that accepted a valid LTPA Token.
   - Authorization is the step where the user is authorized to the resource, in this case the web service operations. The following key elements must be specified to use XACML on-box PDP authorization:
     - The Method: **Use XACML Authorization**.
     - The XACML Version; for example, 2.0.
     - PDP Type; for example, deny based PDP.
     - Use On box PDP: **On**
     - The name of the PDP, which has the XACML specified.
     - Configure the PDP. For more information, see "Altering the XACML PDP on DataPower" on page 70.
     - The custom XSL style sheet to bind AAA and XACML: use `apil-xacml-bindingnew.xsl` as a starting point.

To configure the gateway to use Redaction:

7. Modify the XACML `.xml` file to match the particular security policies you want to enforce for the redaction.
8. Create an XML Firewall with an AAA action that follows the redaction sample.
9. Modify the PDP used by the above AAA action to point to the style sheet you are using to enforce redaction.
10. Copy and modify the `storeCallPDP.xsl` style sheet, that creates the SOAP payload for the XACML service. In particular, make sure that the Action and Resource match your requirements for the XACML policy document you created.
11. Make sure that your modified style sheet calls the correct port for your new XACML XML Firewall.

DataPower objects created in the basic runtime and advanced runtime patterns

An overview of the DataPower objects that are created in the basic runtime and advanced runtime patterns and their function.

Table 18. DataPower pattern objects

Object	Description
Domain	A Domain that can be used for the users application.
WSRR Server	Named WSRRSVR. The SOAP URL, user name, and password are configured, as well as a SSL Proxy Profile with Validation Credentials.
SSL Proxy Profile	Named WSRRPP, it is a forward (client) profile. It uses the Crypto Profile WSRRCP. All other defaults are used.
Crypto Profile	WSRRCP contains a validation credentials object WSRRVC, that contains the Signer Certificate that was uploaded as part of the pattern scripts.
Validation Credentials	WSRR Validation Credentials contain the Crypto Certificate WSRRCERT. All other settings are default.
Crypto Certificate	WSRRCERT uses the signer cert. This certificate was either extracted from the NodeDefaultKeyStore, default Cert for a single server or the CMSKeyStore Default certificate in the case of an ND environment where an IBM HTTP Server was present.

Example use of the WSRR Server Definition in a Web Service Proxy:

1. From the DataPower Control Panel, click **Web Service Proxy**.
2. Click **Add** and provide a **Name** for the Proxy.
3. Next, select the **WSRR Subscription** tab
4. Select WSRR Server in the menu. The WSRRSVR object is available.
5. Provide the other information required such as the Front Side Handler, the namespace, the object name, and so one, to create the configuration of the Web Service Proxy.

Certificate DN values for DataPower certificates

When SSL is used with the provided IBM SOA Policy Gateway Patterns, the DN host verification is more strict than the default WebSphere Application Server security. (This topic applies to external DataPower appliances.)

DN host verification is not enabled in WebSphere Application Server by default. However, in the script packages that are used by the IBM SOA Policy Gateway Patterns, DN host verification is turned on and cannot be disabled. A specific certificate that works between the default WebSphere Application Server and DataPower might not work for the "SOA Policy Gateway 2.5.0.0 - Security" script package or the "SOA Policy Gateway 2.5.0.0 - Sample" script package that is used with the IBM SOA Policy Gateway Pattern. For example, a DN of `myserver.yourcompany.com` might be accepted by the WebSphere Application Server defaults, but not by the script packages. To add or remove the DataPower certificates that are used with the deployment, see "Removing or Adding DataPower Certificates to the WSRR Truststore."

Removing or Adding DataPower Certificates to the WSRR Truststore

This task describes how to add or remove DataPower certificates. This topic applies to deployed patterns with external DataPower appliances.

About this task

The DataPower certificates are uploaded to the WSRR truststore to simplify sync update between WSRR and DataPower for policy updates. If this capability is not needed, you can remove DataPower Certificates. You can also add new DataPower Certificates if the certificates need to be changed.

Procedure

1. To remove certificates:
   a. Log in to the WebSphere Application Server administrative console at `https://hostname:9043/ibm/console`, where *hostname* is the host name of the WSRR system. Enter the administrative user name and password.
   b. Navigate to **Security, SSL certficates and key management**.
   c. Click **Key Stores and Certificates**.
   d. Click **NodeDefaultTrustStore** if your deployment is based on a basic runtime pattern, or **CellDefaultTruststore** if you deployed an advanced runtime pattern.
   e. Click **Signer Certificates**.
   f. Select the check boxes of any certificates you want to remove.
   g. Click **Delete**.
   h. Click **Save**.
2. To add new DataPower Certificates, click **Add** to add the new certificate.
   a. Log in to the WebSphere Application Server administrative console at `https://hostname:9043/ibm/console`, where *hostname* is the host name of the WSRR system. Enter the administrative user name and password.
   b. Navigate to **Security, SSL certficates and key management**.
   c. Click **Key Stores and Certificates**.
   d. Click **NodeDefaultTrustStore** if your deployment is based on a basic runtime pattern, or **CellDefaultTruststore** if you deployed an advanced runtime pattern.
   e. Click **Signer Certificates**.
   f. Click **Add** and specify the new certificates.
   g. Click **Save**.

Changing the LTPA Keys

This procedure describes how to change the LTPA key. The LTPA key is shared among all cells in the patterns. It is not used in the SOA Policy Gateway Basic Runtime Sample pattern. The LTPA Key is exported from the Governance Master and imported into runtime environments, such as staging or production.

About this task

You complete these actions in the WebSphere Application Server administrative console. For more information, follow the related link.

Procedure

1. Export the new LTPA Key from the Governance Master WSRR Dmgr.
2. Import the LTPA Key into the Runtime WSRR instances, which are Dmgr or Stand Alone.

3. If the Runtime instance is based on an advanced runtime pattern, complete the following in order:
   a. Synchronize all nodes.
   b. Stop the WSRR Cluster.
   c. Stop the node agents.
   d. Stop the Dmgr.
4. If the WSRR system is based on an advanced runtime pattern, it must be restarted in reverse order:
   a. Start the Dmgr.
   b. Start the node agents.
   c. Start the WSRR Cluster.
5. If the WSRR is a Standalone Server (based on a basic runtime pattern), it must be stopped and restarted for the LTPA Key change to take effect.

Related information:

WebSphere Application Server V8.0 Information Center

Service creation and governance

Use the WSRR Business Space user interface to create and govern business services and their associated objects.

The SOA Governance space must be created in Business Space before policies can be created. If the SOA Governance space does not exist, see "Configuring Business Space for the first use" on page 81 and follow the steps to create the space.

For more information about creating a new governed service, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Tutorial: Governing a new service.

For more information about governing an existing service, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Tutorial: Governing an existing service.

Related tasks:

"Connecting to WSRR - Business Space" on page 80
Use the Business Space user interface to work with WSRR..

Policies

Implementation details for using WSRR as the Policy Authoring Point and WebSphere DataPower as the Policy Enforcement Point when you create mediation policies.

Policies in WSRR

You can use WSRR to create all of the SOA policies, including SLA (Service Level Agreement) policies, mediation policies, monitoring policies, and custom policies. Using the Business Space user interface, you can create, update, or delete a policy document in WSRR. The policy document can contain a policy expression that specifies a number of policies for a particular policy domain. Alternatively you can create a policy document that assembles existing policies from other documents. Individual policies are referred to using policy identifiers, which you specify when

you add policies to your document. A policy expression represents the declaration of a policy and is equivalent to a <wsp:Policy> element in a WS-Policy document.

To create a mediation policy in Business Space, see "Authoring new mediation policies" on page 95.

Mediation policy assertions

Service Level Agreements (SLAs) originate from a business requirement that the quality of service that is provided by a service meets a specified standard. As a service is designed, functional requirements are created to guide the logic of what the service does. Non-functional requirements are specified in parallel as part of the analysis and design of that service to designate the quality of service that the service is expected to provide. For example, the business might have a service that supplies information in response to a customer internet query. The target is to return the response within 3 seconds. As part of the engineering of the end-to-end transaction, it is determined that this service must return its information within 2 seconds to meet the business non-functional requirements.

You can write a policy that implements runtime checks on the performance of the service and acts when requirements are met to guarantee that the service meets its SLA. For example, you might have a service primary endpoint that is normally (95% of the time) able to provide service response within two seconds. The SOA architect creates a secondary endpoint on another server that can be used as a hot standby for primary endpoint outages, but is also authorized to be used for overflow traffic when the primary endpoint is not able to keep up with the transaction load. You can write a policy that checks the service response time and reroutes traffic when necessary to meet the SLA.

Another example where SLAs are maintained through runtime policy is a situation where a service is responding to transactions that have various consumers, each with a different level of priority. A simple example might have "gold" and "bronze" customers, where the business guarantees only a specific quality of service for the "gold" customers. In this example, you can check whether the consumer is "gold" and reroute to the secondary endpoint, leaving the "bronze" customer to deal with a slower response time. The business decided because "bronze" customers provide insufficient incremental revenue to justify the expense of engineering a response time to meet the SLA of the "gold" customers.

In a third example, you might have a situation where a service does the best it can, but when it determines that it is under load, queues or even rejects messages from low priority consumer services. One example is when a batch routine floods the system with consumer requests at an unexpected time. To protect the quality of service, you can create a runtime policy that is in effect during business hours only, and that rejects all batch requests during this period.

More generically, mediation policy allows for validation and transformation on the incoming message from the client (consumer) before presentation to the server (provider).

Policies support this type of message validation and transformation. Policies can be specified for a provider service only, for a specific consumer-provider pair, or for Anonymous consumers for a provider service. Policies for Anonymous customers provide a way of defining a default policy that applies only to consumers for which no other policies apply. Using this feature allows policies to be specified for rogue consumers that do not identify themselves. Such consumer services could

then have their transactions rejected. This can be useful to prevent denial of service attacks from consumer hackers attempting to flood the system with transactions meant to bring down a provider service.

Mediation policy conditions

Mediation assertions can be made that allow runtime policy to control the SLA of the service, transformation of messages from consumer to provider, or to validate the message schema of the consumer message.

SLA policy conditions, a special type of mediation policy, effectively allows for a classic if-then-else construct with a condition and then a set of actions to be performed depending on how the condition evaluates. Specifying a condition is optional. If no condition is specified, it is equivalent to the logical condition that evaluates to True, and any actions that are specified are enforced accordingly.

The condition, if specified, must consist of a Boolean expression or a schedule specification, or the condition can include both.

Schedule

The schedule, if specified, identifies when the policy is in effect. The date and time are evaluated by the local Policy Enforcement Point and the time zone that is used is that of the Policy Enforcement Point. If no schedule is specified, the policy starts as soon as it is downloaded from the Policy Authoring Point to the Policy Enforcement Point, and continues indefinitely.

The schedule defines an optional start date and an optional stop date, an optional daily timeframe, and an optional list of weekdays. For example, a schedule can be defined as being effective from October 1st 2012 to October 30th 2012, from 8 a.m. to 5 p.m. on Wednesdays and Sundays.

The parameters for the schedule that can be specified are as follows:

- **StartDate** - This optional attribute specifies in xs:date format the date at which the schedule becomes effective. StartDate is inclusive and if this attribute is not present, the schedule becomes effective immediately today. (Click the xs:time hyperlink to understand this industry standard).
- **StopDate** - This optional attribute specifies in xs:date format the date at which the schedule stops being effective. StopDate is exclusive and the specified date must be after the start date. When the stop date is before or the same as the start date, the schedule is never effective. If this attribute is not present, the schedule is effective indefinitely.
- **Daily** - This optional element specifies the daily timeframe during which the schedule is effective. If this element is not present, the schedule is effective all day.
  - **StartTime** – If Daily is specified, then this attribute is required. It specifies in xs:time format the time at which the schedule starts daily. (Click the xs:time hyperlink to understand this industry standard).
  - **StopTime** - If Daily is specified, then this attribute is required. It specifies in xs:time format the time at which the schedule stops daily. StopTime is exclusive and if the specified time is earlier than or the same as the daily start time, the schedule stops at the specified stop time on the next day.
- **Weekdays** - This optional element specifies the days of the week included in the schedule. If this element is not present, every day of the week is included in the

schedule. This element affects only the start of the daily timeframe, as schedules are allowed to run past midnight. For example, if a schedule is set to start at 11 p.m. and run for 2 hours on Wednesdays, the schedule effectively ends on Thursday at 1 a.m.

– **Days** - If Weekdays is specified, then this attribute is required. It lists the weekdays included in the schedule, as a list of names separated with the plus sign ('+'); for example, “Monday+Tuesday+Wednesday+Thursday+Friday+Saturday+Sunday”.

Mediation policy condition expression

The condition expression, if specified, is a non-repeating element that specifies a Boolean expression.

The expression comprises three parameters: Attribute, Operator, and Value, plus optional Interval and Limit parameters. If the application of the Operator on the Attribute and the Value, plus the Interval and Limit when appropriate, evaluates to True, the expression evaluates to True. The Limit element is only used with the HighLow and TokenBucket operators. If not specified, the value of Limit is 0. If Interval is not specified, the default is 60 seconds.

The parameters for Expression that can be specified are as follows:

- **Attribute** - The following table summarizes the defined attributes and their type.

Table 19. Defined attributes

Attribute	Description and Type
ErrorCount	The number of faults that are observed during this monitoring interval.
MessageCount	The number of actual messages that are intercepted during the monitoring interval.
InternalLatency	The internal latency (processing time) in seconds.
BackendLatency	The appliance-to-server latency in seconds.
TotalLatency	The sum of back-end and internal latency in seconds.

- **Operator** - The following table summarizes the available operators and their meaning:

Table 20. Operators

Operator	Meaning
GreaterThan	A simple numeric algorithm that evaluates to True when the Attribute is greater than the defined Value.
LessThan	A simple numeric algorithm that evaluates to True when Attribute is less than the defined Value.

Table 20. Operators (continued)

Operator	Meaning
TokenBucket	A rate-based algorithm that allows bursting. The algorithm consists of a bucket with a maximum capacity of Limit tokens. The bucket refills at a constant rate of Value tokens per Interval, while for each unit of Attribute a token is removed. This algorithm evaluates to True when there are no tokens in the bucket, and evaluates to False otherwise. Here is an example to help explain the algorithm: Assume Limit=100, Value=5, Interval=1 second, and the Attribute=MessageCount. 1. The bucket starts full with a maximum capacity of 100 tokens 2. When a message arrives, the algorithm checks whether the bucket holds any tokens: a. If it does, the algorithm evaluates to False and one token is removed from the bucket b. If it does not, the algorithm evaluates to True. 3. All the while, every second, the algorithm adds 5 tokens back to the bucket as room permits.
HighLow	An algorithm that evaluates to True when Attribute reaches the high threshold specified as the Value and then continues to evaluate to True until Attribute reaches the low threshold specified as the Limit.

- **Value** – This is a positive integer element. "0" is valid.
- **Interval** - This optional element defines in xs:duration format the time interval, used as a sliding window, to measure the wsme:Attribute when evaluating the expression,. If not specified, the interval used is 60 seconds. If specified, a reasonable value must be specified, taking into account the configured capabilities of the Policy Enforcement Point. That is, the higher this value, the more memory is needed by the Policy Enforcement Point to keep track of the attribute. (Click the xs:duration hyperlink to understand this industry standard.)
- **Limit** - This optional integer element defines the additional Limit argument required when wsme:Operator is TokenBucket or HighLow. The unit depends on the wsme:Operator specified.

  When wsme:Operator is HighLow it defines the low threshold while wsme:Value defines the high threshold. The specified threshold must be lower than that of wsme:Value. When not specified the default Limit is 0.

  When wsme:Operator is TokenBucket it defines the maximum size of the burst, or maximum number of tokens in the bucket, while Value specifies the rate at which the bucket is refilled, in number of tokens per Interval. When not specified the default Limit is 0 and TokenBucket is then equivalent to a GreaterThan operation.

Mediation policy actions

The Mediation Action element specifies the actions to be taken. Although the syntax allows many combinations, not all of them make sense and when conflicting actions are specified, such as asking for a message to be both queued and rejected, the behavior is rejected by the Policy Authoring Point. The mediation policy actions allowed are:

- **QueueMessage** – This action specifies that transactions are queued when the logical condition is met. Message processing does not recommence until the

logical condition is no longer met. The queue methodology and any associated timeouts are as defined by the Policy Enforcement Point, in this case WebSphere DataPower. When several actions are specified within a single Action element, QueueMessage must be the first action.

- **RejectMessage** – This action specifies that transactions are rejected when the logical condition is met. Transactions continue to be rejected until the logical condition is no longer met. When transactions are rejected, a SOAP fault is returned to the client (consumer) service. When several actions are specified within a single Action element, RejectMessage must be the first action. QueueMessage and RejectMessage are mutually exclusive.
- **Notify** - This optional element specifies that a notification is produced when the logical condition is met. For DataPower, a message is written to the DataPower system log.
- **RouteMessage** - This optional element specifies that messages is routed to specified endpoint destination when the logical condition is met. Messages continue to be routed to the specified endpoint until the logical condition is no longer met.
  - **EndPoint** – This parameter is required when an action of RouteMessage is specified. The endpoint value supported can be an IP address, hostname, or virtual host; such as load balancer group.
- **ValidateMessage** - This optional element specifies that messages is validated against the specified grammars. Messages are rejected when validation fails. Either XSD or WSDL must be specified as a subparameter if ValidateMessage is specified. SCOPE is optional, and if not specified, SOAPBody is used for the validation.
  - **XSD** - Specifies that messages are validated against the XML schema identified by the URI it contains.
  - **WSDL** - Specifies that messages are validated against the Web services description (WSDL) identified by the URI it contains.
  - **SCOPE** – Specifies what part of the message is validated. The following table lists the possible values and what they mean:

Table 21. ValidateMessage elements

Value	Description
SOAPBody	The contents of the SOAP Body element, without special processing for SOAP faults. (Default)
SOAPBodyOrDetails	The contents of the detail element for SOAP faults, and the contents of the Body otherwise.
SOAPEnvelope	The entire SOAP message, including the envelope.
SOAPIgnoreFaults	No validation if the message is a SOAP fault, the contents of the SOAP Body otherwise.

- **ExecuteXSL** - Specifies that an XSL transform is performed with the specified style sheet and parameters. Transactions are rejected when the execution fails. Style sheet information must be specified, while parameters are optional, and must be specified as needed by the particular style sheet specified.
  - **Stylesheet** - Specifies that the transform operation uses the stylesheet specified by the contained URI. The style sheet MUST be an XSLT file.
  - **Parameter** - This optional, repeating element specifies a style sheet parameter to be used for the ExecuteXSL operation.
    - **Name** – This attribute is required for each corresponding Parameter parameter and specifies the name of the parameter.

- **Value** - This attribute is required for each corresponding Name parameter and specifies the value of the parameter.

Authoring new mediation policies

You can create new mediation policies by using the Business Space user interface. When you author mediation policies, you specify the conditions and actions for the policy.

Before you begin

For information about accessing Business Space, see "Connecting to WSRR - Business Space" on page 80.

The SOA Governance space must be created before policies can be created. If the SOA Governance space does not exist, see "Configuring Business Space for the first use" on page 81 and follow the steps to create the space.

You must also configure Business Space to create WS-MediationPolicy 1.7 mediation policies from the Actions widget. See , Service Registry Actions widget

About this task

Author new policies by using the SOA Governance space.

Procedure

1. Open the SOA Governance space:
   a. Click **Go To Spaces**. The Go To Spaces dialog is displayed.
   b. Click the space for SOA Governance users. The specific name depends 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. Add conditions and actions as required. For more information about the conditions and actions, see "Policies" on page 89 and IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Creating a mediation policy.
5. Click **Finish**.

Results

The policy is created and stored in WSRR. To view the policy document for the policy you created, select the policy document in the Service Registry Navigator widget. 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.

Related concepts:

"Policies" on page 89
Implementation details for using WSRR as the Policy Authoring Point and WebSphere DataPower as the Policy Enforcement Point when you create mediation policies.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Creating a mediation policy

Authoring new monitoring policies

You can create new monitoring policies by using the WSRR web UI. When you author monitoring policies, specify the conditions and actions for the policy.

Before you begin

For information about accessing the WSRR web UI, see "Connecting to WSRR - WSRR Web UI" on page 82.

Procedure

1. Open the WSRR web UI.
2. Click **View** > **Service Documents** > **Policy Documents** and in the collection view click **New**.
3. From the list of available Policy Frameworks select **Monitoring**. Click **Next**. This creates a policy document with a root policy expression in it.
4. Enter a meaningful name, and an optional description.
5. Click the Policy tab, click **Edit policy document**, and add conditions and actions as required. For more information about the conditions and actions, follow the related links.
6. Click **Publish**.

Results

The policy is created and stored in WSRR. You can view the policy document for the policy in Business Space, select the policy document in the Service Registry Navigator widget. 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.

Related concepts:

"Policies" on page 89
Implementation details for using WSRR as the Policy Authoring Point and WebSphere DataPower as the Policy Enforcement Point when you create mediation policies.

Related information:

Policy authoring tasks

Working with the policy authoring tool

Managing policies

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

Before you begin

Configure the SOA Governance space. For more information, see "Configuring Business Space for the first use" on page 81.

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 policy details:

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 any conditions or actions, these are displayed. Create and modify the conditions and actions as required.
c. Click **Finish** to save and close the policy editor. The Service Registry Detail widget refreshes to show the changes that are made.

3. To delete the policy:
   a. Transition the policy to a governance state that allows for editing or deletion of the policy document. For more information about transitioning a policy through the SOA Policy Lifecycle, see "Managing the lifecycle of the policy."
   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.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Policies in the governance enablement profile

Managing the lifecycle of the policy

Policies can be transitioned between governance states by using the Business Space user interface. Polices must be in the Approved state to be enforced by DataPower.

About this task

For more information about governance, see "The SOA Policy lifecycle" on page 5.

Procedure

To transition a policy to a different lifecycle state, complete the following steps. Repeat these steps as many times as required to reach the wanted lifecycle state:

1. In Business Space, open the policy document for the policy by selecting the policy document in the Service Registry Navigator widget. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget. The **Governance state** property displays the current governance state for the profile.
2. Click **Action**. A list of possible lifecycle transitions is displayed along with other possible operations.
3. Select the required lifecycle transition to move the policy to the required state. The **Governance state** property of the policy is updated to show the new lifecycle state.

Related concepts:

"The SOA Policy lifecycle" on page 5
Policies are governed by the SOA Policy lifecycle. The lifecycle takes the policy from being initially identified, through to being deployed in production, and, finally, to being deprecated when it is no longer required.

Related information:

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

Policies attached to a service

Policies can be attached to a service by using WSRR.

For more information, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Policy attachment tasks.

Chapter 7. Troubleshooting

Get assistance with diagnosing problems that you might have before, during, and after deployment of the pattern.

Use the links to find topics relevant to a problem with the patterns.

Troubleshooting problems with deployment

You can troubleshoot common problems that you encounter when you deploy the patterns in the IBM SOA Policy Gateway Pattern.

Failure to connect to external DataPower appliance during deployment

Try the following solutions:

- Check with the DataPower Administrator that the user and password are valid:
  - In the DataPower, Web GUI validate that the user exists by going to **Control Panel** > **Manage User Accounts**.
  - Check that the account exists.
  - Check that user is privileged to use the XML Management Interface; for example, the system administrator.
  - The DataPower Administrator might have to check whether the user account is enabled in the user agent settings; for example, the Basic Authentication Settings.
- Check that the DataPower host name is correct
- Check that the DataPower XML Management Interface is enabled.

Troubleshooting an error for the domain already existing

Try the following solution:

- On the DataPower Control Panel, open the Application Domains. Check if the Domain already exists.

Troubleshooting a port overlap error for the sample application

If one of the sample services is unavailable, check if the ports in your domain conflict with other domains.

Try the following solutions:

- LOG in to DataPower and switch to the sample domain. Then, open the Control Panel and click the XML Firewall icon. Check that the XML Firewalls are all in Up state.
- Search for HTTP Front Side Handler. Check that the single HTTP Front Side handler is in Up state.

Troubleshooting promotion failure

Many problems might arise during promotion, including failure to connect to Governance Master during deployment.

Try the following solutions:

- Check the parameters:
  - Check the user of the Governance Master WSRRCELL.
  - Check the password for the user of the Governance Master WSRR Cell.
  - Check the host name of the WSRR Governance Master Cell.
  - Check the CELL name of the WSRR Governance Master Cell.
- Check the signer certificate exchange:
  - Go to the Cell Default Trust Store of the Governance Master cell and make sure that there is a certificate entry for the Dmgr or the Standalone server of the runtime environment.
  - Go to each Runtime Environment and check the CellDefaultTrust store (for the ND environment case) or the NodeDefaultTrustStore (for WSRR Standalone servers) to make sure that there is a certificate for the Dmgr of the Governance Master.
  - Export the LTPA keys from both cells by using the same password, and check that they are the same (for example, the bytes).
- Make sure that the promotion properties file contains server sections with the appropriate host and port, and user and password information. This information can be found in the ServiceRegistry console for the Governance Master:
  - Go to the GovernanceMasterDMgrHost or ServiceRegistry and switch to the Configurations perspective. In the Actions section, find **Promotion** and open the promotion properties file. For each environment, there should be XML elements for each server in the staging WSRR node or cluster. If a production cluster or node exists, there should be server:port entries for each, and in addition there should be user and password information.
- Check that the Service Version and SOAP Endpoint both have Classification for staging and Production.
  - In the Service Registry Console, select the SOA Governance perspective. Open the Service Version, and select the Classifications tab. Staging and Production must be enabled.

Troubleshooting customized CLI failures

Try the following solutions:

- Check the defaultLog for error messages in the DataPower Domain.
- Enable the CLI debugging and check those logs before any additional runs of the CLI.

Troubleshooting problems in the deployed instance

You can troubleshoot common problems in the deployed instance.

Failed connections to the LDAP Server or the DataPower StoreWSP port

You might have an issue with the Domain settings if the DataPower logs show a connection error to either LDAP or the StoreWSP gateway and if you are using the host alias name; for example, `xyz` instead of the fully qualified host `xyz.company.com` name for one of the following parameters in the script package:

- The DataPower host name
- The LDAP host name

Try the following solution:

1. In the DataPower Administration Console, switch to the default domain.
2. Search for `Configure DNS Settings`.
3. Click the Search Domains tab.
4. Make sure that your domain; for example, `company.com`, is in the list. If it is not in the list, click `Add` and add it to the list.

Problems with monitoring

If monitoring is not available on the deployed nodes, you must verify that the required shared services are running. Navigate to **Instances** > **Shared Services**

Verify that System Monitoring and System Monitoring for WebSphere DataPower are running in the same cloud group as your deployed instances. For WSRR monitoring, also verify that System Monitoring for WebSphere Application Server is running in your cloud group.

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 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 packages have errors, click the log links for **remote_std_out.log** and **remote_std_err.log** next to the script package names.
   b. Log in to 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
3. For DataPower:
   a. Retrieve the **default.log** file for the domain that is created by the pattern.
   b. Retrieve the **default.log** file for the default domain.
4. For monitoring problems, collect these logs from the Base OS and WSRR nodes (excluding WSRR Custom Nodes):
   - `/0config/0config.log`
   - `/opt/IBM/maestro/ITCAMSOADP/lx8266/d4/KD4/logs/*` (x86)
   - `/opt/IBM/maestro/ITCAMSOADP/aix523/d4/KD4/logs/*` (Power)

Chapter 8. Maintenance and support

You can perform maintenance functions such as applying emergency fixes.

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 complete 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 an 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 by 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.

Note: You must shut down all WSRR and WAS processes before installing an emergency fix. Log in by using SSH to all WSRR nodes, and shutdown the processes with the **stopServer.sh** and **stopNode.sh** (Custom Nodes only) commands.

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 9. 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.