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. Service Orientated Architecture (SOA) practices help businesses to identify and focus on the key services of the business. By adding policies, we add points of control and agility for business and information technology. The result is that SOA more consumable, improving time-to-value for business users with reduced costs for their projects, and accelerates the adoption of SOA solutions.

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 Authoring Point (PAP), Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Information Point (PIP), and the Policy Monitoring Point (PMP). In this pattern, the PAP is achieved using WSRR, and the PEP is achieved using WebSphere® DataPower®.

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

- **Policy Authoring 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 runtime. Includes a repository to store policies. In this pattern, this is achieved using WSRR.
- **Policy Enforcement Point** - A Policy Enforcement Point is a functional point that runs on the middleware that:
  - 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 actually applied and enforced depending on the lifecycle stage:
    - During design time, the service registry and repository 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, this is achieved using WebSphere DataPower.
- **Policy Decision Point** - A Policy Decision Point evaluates participant requests against relevant policies or contracts and attributes. It 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. This includes:
  - Receiving monitoring policy updates and making them ready or translates them for usage.
  - Capturing the real time collection and statistics analysis for display.
  - Correlating, analyzing, and visualizing the data 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.

  Note: Monitoring is not included in this pattern.

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 actionable pattern flow is shown in Figure 1 on page 3 and described below.

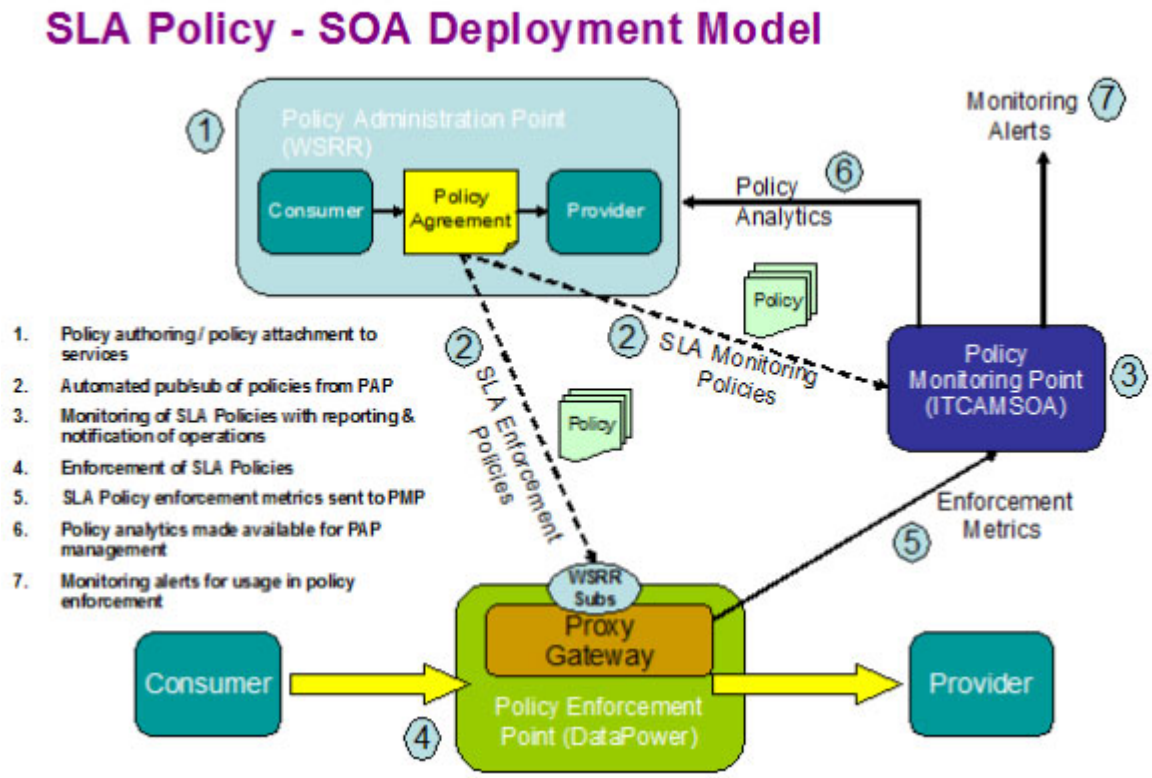


Figure 1. Service Level Agreement (SLA) Policy - the SOA deployment model

1. Policies are authored and then attached to services that require that policy. Typically this follows the following order:
   a. The set of services are loaded or created in the service repository. This is a part of the Policy Authoring Point.
   b. The set of policies required are created in the Policy Authoring Point using the policy lifecycle:
      1) Policies are attached to the services that require those policies – at the service, operation, or endpoint level as required.
2. Automated pub/sub of policies from the Policy Authoring Point to the Policy Enforcement Points and the Policy Monitoring Point:

   Note: Monitoring using ITCAM for SOA is not included in this pattern.
   a. As a part of the setup, ITCAM for SOA subscribes to the monitoring policy from WSRR. This occurs only once.
   b. As a part of the setup, proxy gateways are created in each WebSphere Data Power® appliance that has service transactions with policy enforcement. This occurs only once, and is added or changed as required.
   c. 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 occurs only once, and is added or changed as required.
   d. As a part of the setup, WebSphere DataPower is configured so that policies can be shared by other appliances in a cluster. This occurs only once, and is added or changed as required.
   e. ITCAM for SOA downloads the monitoring policies as they are published.
   f. ITCAM for SOA converts the policies into the internal representation called situation policies.

g. WebSphere DataPower downloads the WSDLs for services that it is responsible for transacting.
h. WebSphere DataPower downloads the policies for services that it is responsible for when notified by WSRR.
i. WebSphere DataPower converts the policies into internal WebSphere DataPower representation in the form of SLM objects.

3. Monitoring of SOA policies with reporting and notification of operations:
   a. Monitoring policies are active in the ITCAM for the SOA Situation Policy.
   b. ITCAM for SOA receives monitoring information and places that information in workspaces.

   Note: Monitoring is not provided in this pattern.
4. Enforcement of SOA Policies:
   a. Enforcement policies are active in the various WebSphere DataPower appliances.
   b. WebSphere DataPower receives service transactions and applies policies for that consumer service and provider service.
5. The Policy Enforcement Point sends SOA Policy Enforcement statistics to the Policy Monitoring Point.

   Note: Monitoring is not included in this pattern.
6. The Policy Monitoring Point sends monitoring events to the Policy Authoring Point:
   a. Events are set up in the Policy Authoring Point that need to be monitored from the Policy Monitoring Point. This occurs only once, and is added or changed as required.
   b. As situation policies evaluate to true, events are pushed to the Policy Authoring Point from the Policy Monitoring Point.

   Note: Monitoring is not included in this pattern.
7. Monitoring of alerts:
   a. Situation policies run periodically and take operational action as specified in the policy. The default is every 5 minutes.

The SOA Policy lifecycle

Mediation policies are governed using the SOA Policy lifecycle. This takes the policy from being initially identified, through to being deployed in production, and, finally, 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, have created standards to service the requirement to define the policy 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/WSMediationPolicy1.6-20120124.pdf.

The WS-Policy Data Model includes:

- **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 is used in a WS-PolicyAttachment document.

The following example, Figure 2, shows a security policy expression using 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 above would be that an invocation of a Web service will either sign or encrypt the message body, but not both.

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

Figure 3 shows a policy definition.

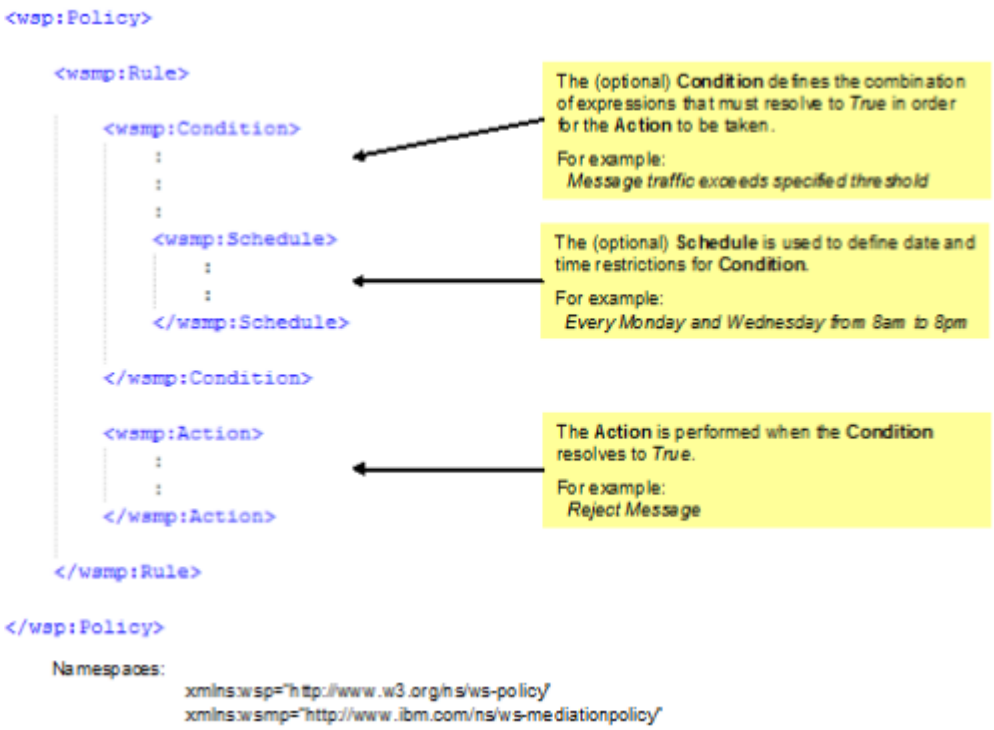


Figure 3. Overview of Policy structure

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 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 only applies to consumers for which no other policies apply.

In Figure 4 on page 6, the domain-specific policy subject to which the policy applies (the provider) is contained in the `<wsp:AppliesTo>` section 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 providing a policy enforcement point and a policy administration point. The policy administration point is provided by virtual system patterns that provision WSRR in a multi-tier architecture, delivering a production and staging environment. The policy enforcement point is provided by the WebSphere DataPower appliance in which a domain is created during virtual system pattern deployment.

There are examples of policy in many, if not all Service Orientated Architecture (SOA) environments. Service producers and consumers agree the capabilities, performance, and characteristics of the service during the design phase. To do this, you can use Service Level Definitions (SLDs) and Service Level Agreements (SLAs). This pattern allows you to define policies for SLDs and SLAs in an efficiently administered, defined, governed, and utilized way. Policy types used in this pattern include the following:

- **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**: In the sample we demonstrate the means to enforce XACML access control security policies. These are not governed within the policy administration point at this time.

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

- SOA Policy Gateway Basic Runtime Sample
- SOA Policy Gateway Governance Master
- SOA Policy Gateway Basic Runtime
- SOA Policy Gateway Advanced Runtime

The four 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 configured to the governance environment during the pattern deployment. Combined, the following deployment topologies are provided:

- Standalone deployment
- Pilot deployment
- Full production deployment

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

It is possible to manually configure the deployed virtual system pattern to include monitoring with ITCAM for SOA Version 7. This provides the basic monitoring of events and expands policy support to include monitoring policies. Monitoring

policies allow event situations to be defined within the Policy Authoring Point (PAP) and be attached to a service definition, allowing the monitor to act when the event situation occurs.

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. Service Orientated Architecture (SOA) practices help businesses to identify and focus on the key services of the business. By adding policies, we add points of control and agility for business and information technology. The result is that SOA more consumable, improving time-to-value for business users with reduced costs for their projects, and accelerates the adoption of SOA solutions.

“SOA Policy Gateway Basic Runtime” on page 21
The SOA Policy Gateway Basic Runtime provides a simple means to provide a runtime that can be used stand-alone or integrated with a deployed SOA Policy Gateway Governance Master pattern. The SOA Policy Gateway Basic Runtime pattern supports the deployment of a DataPower domain that is configured to communicate with the WSRR runtime server provisioned within the pattern.

“SOA Policy Gateway Basic Runtime Sample” on page 18
The SOA Policy Gateway Basic Runtime Sample provisions a SOA Policy Gateway Basic Runtime with a sample interface and application that demonstrates the policies 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” on page 23
The SOA Policy Gateway Advanced Runtime includes more high availability options and must be used with the SOA Policy Gateway Governance Master.

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

This pattern uses WebSphere DataPower to control messages using governed polices and service definitions in WSRR. Review the topics in this section to understand what is covered in this scenario, the reasons why a business might want to follow the scenario, the user roles involved, and an overview of the capability delivered with the product.

Before you begin

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

Procedure

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

1. Download and install the IBM SOA Policy Gateway Pattern. For more information about downloading the packages from Passport Advantage®, see "Downloading and installing the patterns" on page 12.
2. Optional: Configure user access. For more information, see "Configuring user access" on page 14.
3. Configure and deploy the pattern
   a. Accept the imported virtual system image licenses for WSRR.
   b. Accept all license agreements on the DB2® Enterprise.
   c. Deploy the pattern:
      1) Decide on the deployment topology. For more information, see IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Deployment topologies.
      2) If using a standalone deployment topology, deploy a single Basic Runtime pattern without promotion configured.
      3) For other topologies, first deploy the SOA Policy Gateway Governance Master pattern. This provides a governance environment for services and policies.
      4) After the Governance Master pattern is successfully deployed, choose the type of runtime environment that you need. For a testing or staging environment, a Basic Runtime will typically suffice. For a production environment, choose the Advanced Runtime environment. The runtimes can be registered with the governance enablement profile promotion configuration for Governance Master. Promotion options include production, staging, or for no promotion for manual promotion configuration.

      For more information, see "Deploying patterns" on page 61.
   d. Verify the deployment. See "Verifying the deployment" on page 66.
   e. Secure the WSRR environment. For more information on planning and configuring WSRR security, see the IBM WebSphere Service Registry and Repository Version 8.0 Information Center.
   f. Configure the provisioned DataPower domain. For more information, see "Security management" on page 54.

4. Use the deployed instance. For more information, see Chapter 6, "Working with the deployed instance," on page 91.

Downloading and installing the patterns

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

Before you begin

Ensure that there is 10 GB of space available for the `CI9G9ML.tar.gz` file and an additional 10 - 14 GB for the extracted files.

The `CI9G9ML.tar.gz` file must be downloaded to a system running Linux or Microsoft Windows. Java™ Runtime Environment (JRE) Version 6 must also be installed prior to starting the pattern installation. You can download this version 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 `CI9G9ML.tar.gz` file. 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 web site: Passport Advantage.
2. Download the archive file containing the images, script packages, and patterns to use. The file is named `CI9G9ML.tar.gz`.
3. Open a terminal on Linux, or a command prompt window on Windows, and navigate to the directory where the `CI9G9ML.tar.gz` file was downloaded.
4. Extract the contents of the `CI9G9ML.tar.gz` file to your local file system. On Linux, the extract command is: On Linux, the extract command is:

```
tar xvzf CI9G9ML.tar.gz
```

   On Windows, use additional archive extraction software to extract the contents of `CI9G9ML.tar.gz`.
5. Ensure that the following extracted files have execute permission on Linux systems:
   - `chmod a+x installer/installer`
   - `chmod a+x installer/deployer.cli/bin/deployer`
   - `chmod a+x installer/deployer.cli/bin/3.1.0.2-20120531075842/deployer`
6. Change to the `installer` directory:

```
cd installer
```

7. To install the IBM SOA Policy Gateway Pattern into the Cloud appliance, 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 Cloud Appliance, and username and password are the Cloud Administrator credentials. For example:

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

8. 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.
9. The patterns are imported. As each pattern is installed, a message is displayed in the installer to indicate it has been installed successfully. For example:

```
Importing pattern "SOA Policy Gateway 2.0.0.0 - Governance Master" ...
Import pattern "SOA Policy Gateway 2.0.0.0 - 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 used in the IBM SOA Policy Gateway Pattern already exists in the catalog, it is not overwritten.

What to do next

Accept licenses in the IBM Workload Deployer appliance or in IBM PureApplication System.

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

Verify the installed pattern

You can verify the pattern is successfully installed, and accept any required licenses to use the pattern.

Before you begin

Ensure that all steps from "Downloading and installing the patterns" on page 12 are completed.

About this task

After installing the pattern, you can verify the pattern installation. Before any virtual image can be used, you must accept the required license for it.

Procedure

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

1. Log in to the IPAS console or the IWD console on the host where the pattern was installed.
2. Verify the Virtual Images by navigating to `Catalog -> Virtual Images` and locate: DB2 9.7.5.0 and WebSphere Service Registry and Repository 8.0.0.1. If a license is not accepted, the image icon will contain a red box with a cross.
   a. To accept a license, click the image to view its details. The current status is displayed. Click **accept** for the License Agreement, and then click any of the

licenses which must be accepted before the virtual image can be used. The current status will display Read-only and the License agreement will display Accepted when complete.

3. `Navigate to Catalog -> Script Packages`, and locate:
   - SOA Policy Gateway 2.0.0.0 - DataPower Domain
   - SOA Policy Gateway 2.0.0.0 - Promotion
   - SOA Policy Gateway 2.0.0.0 - Sample
   - SOA Policy Gateway 2.0.0.0 - Security

   These script packages are all present in a successful installation.
4. `Navigate to Patterns -> Virtual Systems`, and locate:
   - SOA Policy Gateway 2.0.0.0 - Advanced Runtime
   - SOA Policy Gateway 2.0.0.0 - Basic Runtime
   - SOA Policy Gateway 2.0.0.0 - Basic Runtime Sample
   - SOA Policy Gateway 2.0.0.0 - Governance Master

   These patterns are all present in a successful installation.

Results

You have 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: Chapter 5, "Working with the IBM SOA Policy Gateway Pattern," on page 51. If your install was not successful, repeat step 7 onwards of the topic"Downloading and installing the patterns" on page 12.

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.

Procedure

To configure user access, complete the following steps:

1. Choose one of the following options to configure users and, optionally, user groups:
   - Add and configure a user from the Users window of the interface.
     a. From the menu click **System** > **Users**.
     b. Click the **Add** icon.
     c. Provide a short user name as well as the user's actual name, email address, and passwords and click **OK**.
     d. Select the user you added in the Users panel to configure access. Configure the access and actions of the user you selected.

e. Add the user to one or more user groups in the **User groups** field.

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

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

What to do next

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

Related information:

IBM PureApplication System: Managing users and groups

IBM Workload Deployer: Managing users and groups

Chapter 4. Patterns, parts, and script packages

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

Patterns describe the function provided by each virtual machine in a virtual system. Each function is identified as a part in the pattern. Patterns take on the characteristics of their associated parts. For example, when a WSRR part is put into a pattern, which is then deployed, the result is a virtual machine that has a running WSRR instance.

Parts

Parts describe the components that are configured on a virtual machine. Each part has a set of properties (parameters) that are used during deployment to help define the overall configuration of the virtual system. When you load the IBM SOA Policy Gateway Pattern images onto IBM Workload Deployer, the parts are included.

Patterns

The IBM SOA Policy Gateway Pattern pattern contains four patterns:

- SOA Policy Gateway Basic Runtime
- SOA Policy Gateway Basic Runtime Sample
- SOA Policy Gateway Advanced Runtime
- SOA Policy Gateway Governance Master

For detailed information about using the IBM Workload Deployer to access existing patterns or create custom pattern, see http://publib.boulder.ibm.com/infocenter/worlodep/v3r0m0/topic/com.ibm.worlodep.doc/welcome.html.

Patterns

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

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

SOA Policy Gateway Basic Runtime

The SOA Policy Gateway Basic Runtime contains the following required parts:

- DB2 Enterprise
- WSRR Standalone server

SOA Policy Gateway Basic Runtime Sample

The SOA Policy Gateway Basic Runtime Sample contains the following required parts:

- DB2 Enterprise
- WSRR Standalone server

SOA Policy Gateway Advanced Runtime

The SOA Policy Gateway Advanced Runtime contains the following required parts:

- WSRR Deployment manager
- DB2 Enterprise HADR Primary
- DB2 Enterprise HADR Standby
- WSRR Custom Node

SOA Policy Gateway Governance Master

The SOA Policy Gateway Governance Master contains the following required parts:

- WSRR Deployment manager
- DB2 Enterprise HADR Primary
- DB2 Enterprise HADR Standby
- WSRR Custom Node

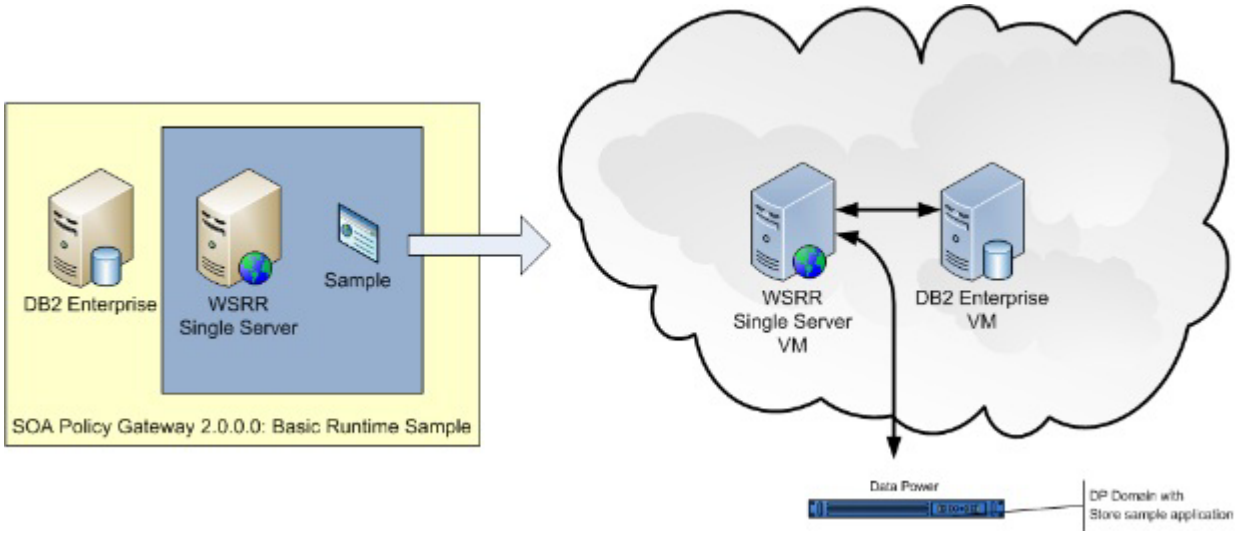
SOA Policy Gateway Basic Runtime Sample

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

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

- WSRR Standalone server
- DB2 Enterprise

The SOA Policy Gateway Basic Runtime Sample pattern installs a sample application in the deployed environment. It 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 69. It installs a sample domain within DataPower, installs sample WSDL and Policies in WSRR, and demonstrates multiple policies against a service.



Policies 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 using XACML-based user authorization. The XACML is not stored in WSRR.
Security Redaction	Redacts parts of the response message based on XACML. The XACML is not stored in WSRR.

Scripts and advanced options

The SOA Policy Gateway Basic Runtime pattern requires the following scripts.

On the WSRR Standalone server part:

- SOA Policy Gateway 2.0.0.0 - Sample

View the part and script parameters:

- "DB2 Enterprise part configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern" on page 27
- "WSRR Standalone server part configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern" on page 34
- "SOA Policy Gateway 2.0.0.0 - Sample script configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern" on page 44

Related concepts:

"DB2 Enterprise part" on page 25
The DB2 Enterprise part provides some configuration options.

"WSRR Standalone server part" on page 33
The WSRR Standalone server part provides some configuration options.
"Script: SOA Policy Gateway 2.0.0.0 - Sample" on page 43
The Sample script configures the sample application parameters for use with the SOA Policy Gateway Basic Runtime Sample pattern.
"The sample application" on page 69
The sample application is a configurable DataPower Domain and a set of WSRR Artifacts that can be used to demonstrate the capabilities of the pattern.

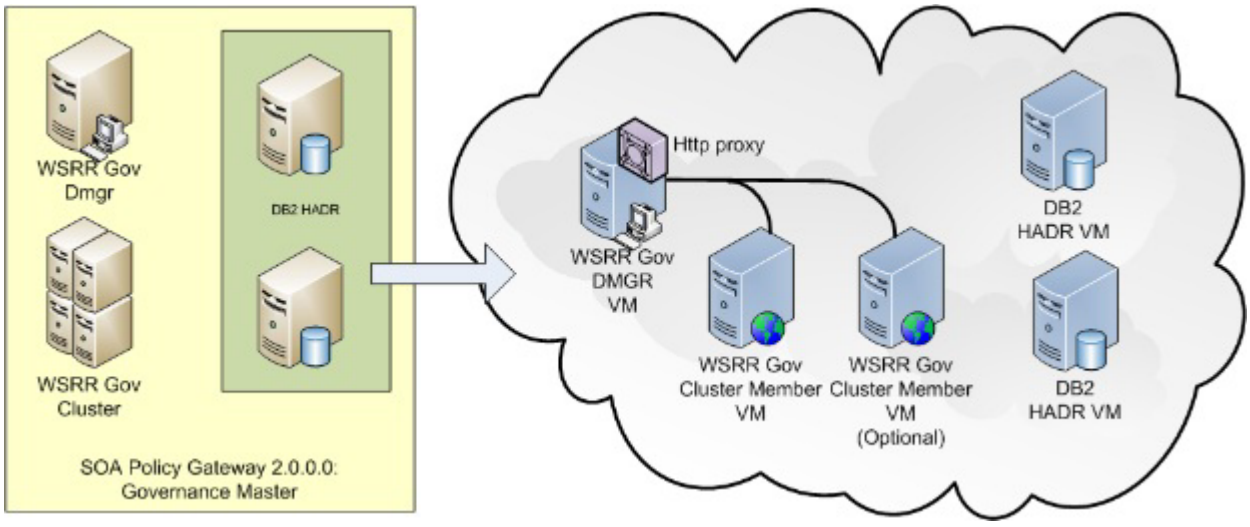
SOA Policy Gateway Governance Master

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 used to configure the Governance Master pattern are used by the runtime patterns to configure itself with the Governance Master. Only the SOA Policy Gateway Basic Runtime pattern or SOA Policy Gateway Advanced Runtime can be configured into the Governance Master.



Scripts and advanced options

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

- SOA Policy Gateway 2.0.0.0 - Security
- SOA Policy Gateway 2.0.0.0 - Promotion
- SOA Policy Gateway 2.0.0.0 - DataPower Domain

View the part and script parameters:

- "DB2 Enterprise HADR Primary part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 30
- "DB2 Enterprise HADR Standby part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 32
- "WSRR Deployment manager part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 36
- "WSRR Custom nodes part configuration parameters for the SOA Policy Gateway Governance Master pattern" 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 which 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 SOA Policy Gateway Basic Runtime and SOA Policy Gateway Advanced Runtime patterns can be deployed into this integration as promotion targets. For more information about how to configure this, see "Scenario: Adding an additional runtime to the pattern" on page 66.

Related concepts:

"DB2 Enterprise HADR Primary part" on page 28
The DB2 Enterprise HADR Primary part provides some configuration options.

"DB2 Enterprise HADR Standby part" on page 30
The DB2 Enterprise HADR Standby part provides some configuration options.

"WSRR Deployment manager part" on page 35
The WSRR Deployment manager part provides some configuration options.

"WSRR Custom nodes part" on page 37
The WSRR Custom nodes part provides some configuration options.

Related information:

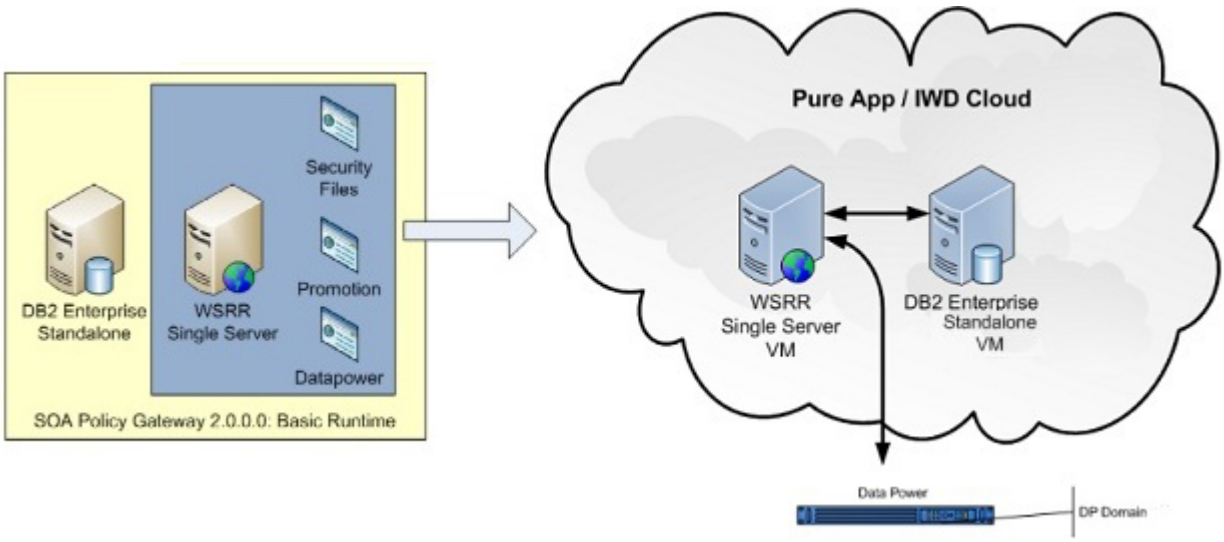
IBM WebSphere Service Registry and Repository Version 8.0 Information Center - Governance Enablement Profile

SOA Policy Gateway Basic Runtime

The SOA Policy Gateway Basic Runtime provides a simple means to provide a runtime that can be used stand-alone or integrated with a deployed SOA Policy Gateway Governance Master pattern. The SOA Policy Gateway Basic Runtime pattern supports the deployment of a DataPower domain that is configured to communicate with the WSRR runtime server provisioned within the pattern.

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

- WSRR Standalone server
- DB2 Enterprise



Scripts and advanced options

The SOA Policy Gateway Basic Runtime pattern requires the following scripts.

On the WSRR Standalone server part:

- SOA Policy Gateway 2.0.0.0 - Security
- SOA Policy Gateway 2.0.0.0 - Promotion
- SOA Policy Gateway 2.0.0.0 - DataPower Domain

View the part and script parameters:

- "WSRR Standalone server part configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 33
- "DB2 Enterprise part configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 26
- "SOA Policy Gateway 2.0.0.0 - Security script configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 47
- "SOA Policy Gateway 2.0.0.0 - Promotion script configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 42
- "SOA Policy Gateway 2.0.0.0 - DataPower Domain script configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 40

Promoting the SOA Policy Gateway Basic Runtime into a Governance Runtime

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
- other or Unset

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.

Related concepts:

"The sample application" on page 69
The sample application is a configurable DataPower Domain and a set of WSRR Artifacts that can be used to demonstrate the capabilities of the pattern.

"DB2 Enterprise part" on page 25
The DB2 Enterprise part provides some configuration options.

"WSRR Standalone server part" on page 33
The WSRR Standalone server part provides some configuration options.

"Script: SOA Policy Gateway 2.0.0.0 - Security" on page 46
The Security script copies security information, contained in a ZIP file, required for communicating with a DataPower appliance onto the Dmgr or WSRR machine from an external file server that supports Linux secure copy program (SCP).

"Script: SOA Policy Gateway 2.0.0.0 - Promotion" on page 41
The Promotion script enables a SOA Policy Gateway Basic Runtime or SOA Policy Gateway 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, whilst optionally configuring WSRR promotion into the governance master.

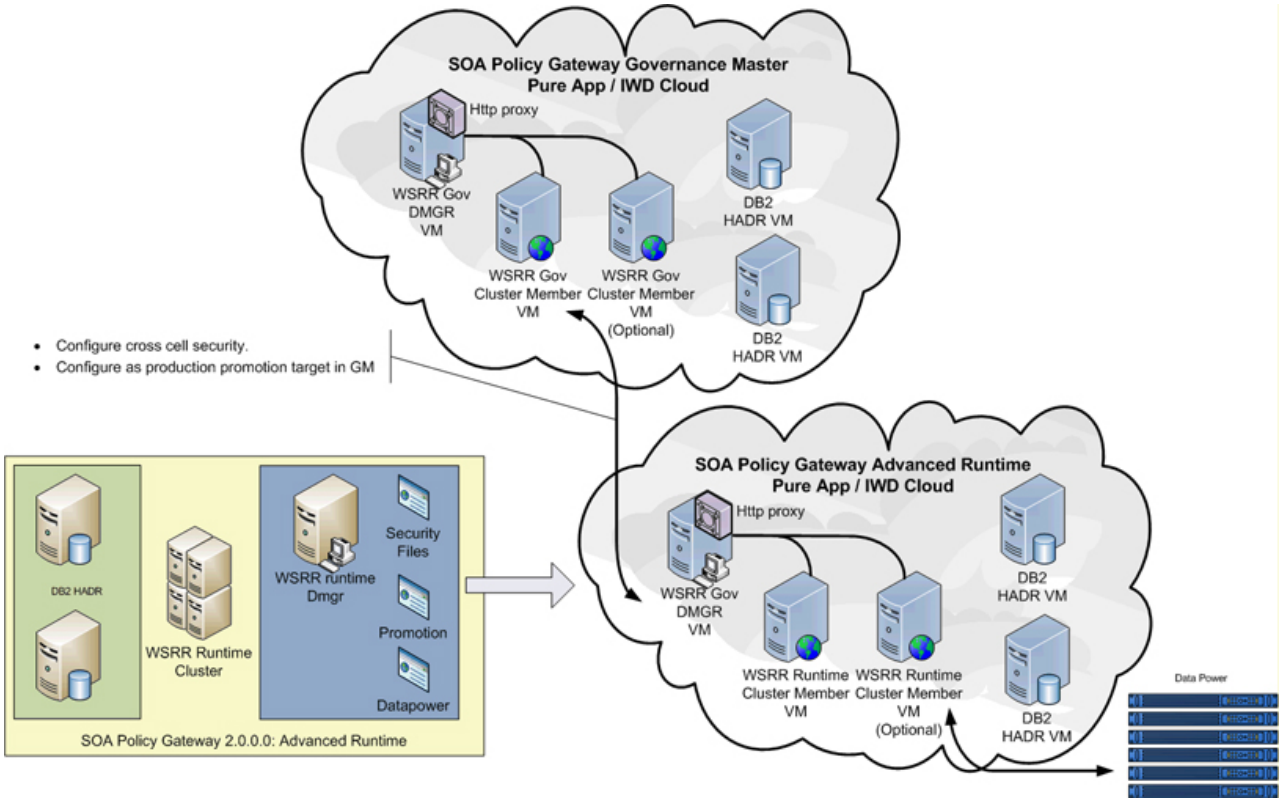
"Script: SOA Policy Gateway 2.0.0.0 - DataPower Domain" on page 40
The DataPower Domain script provisions the DataPower domain during deployment. The script configures the connection between a single DataPower domain and the WSRR runtime. A separate DataPower Domain script is required for each DataPower domain that is connected to the WSRR runtime.

SOA Policy Gateway Advanced Runtime

The SOA Policy Gateway Advanced Runtime includes more high availability options and must be used with the SOA Policy Gateway Governance Master.

TheSOA Policy Gateway Advanced Runtime pattern requires the following parts:

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



Scripts and advanced options

The SOA Policy Gateway Governance Master pattern requires the following scripts on the WSRR Deployment manager part:

- SOA Policy Gateway 2.0.0.0 - Security
- SOA Policy Gateway 2.0.0.0 - Promotion
- SOA Policy Gateway 2.0.0.0 - DataPower Domain (one per DataPower domain)

View the part and script parameters:

- "DB2 Enterprise HADR Primary part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 29
- "DB2 Enterprise HADR Standby part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 31
- "WSRR Deployment manager part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 35
- "WSRR Custom nodes part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 38
- "SOA Policy Gateway 2.0.0.0 - Security script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 48
- "SOA Policy Gateway 2.0.0.0 - Promotion script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 42
- "SOA Policy Gateway 2.0.0.0 - DataPower Domain script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 41

Promoting the SOA Policy Gateway Advanced Runtime into a Governance Runtime

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
- other or "Unset"

These options align with the levels provided by the Governance Enablement Profile in WSRR. If the governance profile on the Governance Master has been altered, use "other" as the promotion level. 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.

Related concepts:

"DB2 Enterprise HADR Primary part" on page 28
The DB2 Enterprise HADR Primary part provides some configuration options.

"DB2 Enterprise HADR Standby part" on page 30
The DB2 Enterprise HADR Standby part provides some configuration options.

"WSRR Deployment manager part" on page 35
The WSRR Deployment manager part provides some configuration options.

"WSRR Custom nodes part" on page 37
The WSRR Custom nodes part provides some configuration options.

"Script: SOA Policy Gateway 2.0.0.0 - Security" on page 46
The Security script copies security information, contained in a ZIP file, required for communicating with a DataPower appliance onto the Dmgr or WSRR machine from an external file server that supports Linux secure copy program (SCP).

"Script: SOA Policy Gateway 2.0.0.0 - Promotion" on page 41
The Promotion script enables a SOA Policy Gateway Basic Runtime or SOA Policy Gateway 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, whilst optionally configuring WSRR promotion into the governance master.

"Script: SOA Policy Gateway 2.0.0.0 - DataPower Domain" on page 40
The DataPower Domain script provisions the DataPower domain during deployment. The script configures the connection between a single DataPower domain and the WSRR runtime. A separate DataPower Domain script is required for each DataPower domain that is connected to the WSRR runtime.

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 9.7.5 virtual system image are described in the following table:

Table 2. Configurable parameters

Parameter name	Description
Virtual CPUs	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Verifies the db2inst1 password.
Password (db2fenc1)	The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Verifies the db2fenc1 password.
Password (dasusr1)	The user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Verifies the dasusr1 password.
Password (root)	The password for the root user ID. This is the password for the operating system of the virtual machine 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.

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

DB2 Enterprise part configuration parameters for the SOA Policy Gateway Basic Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 3. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.

Table 3. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Password (db2inst1)	Yes		The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Yes		Verifies the db2inst1 password.
Password (db2fenc1)	Yes		The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Yes		Verifies the db2fenc1 password.
Password (dasusr1)	Yes		The user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Yes		Verifies the dasusr1 password.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the root password.
Password (virtuser)	Yes		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	Yes		Verifies the virtuser password.

DB2 Enterprise part configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern

In the SOA Policy Gateway Basic Runtime Sample, default values are pre-configured for all parameters.

Table 4. Configured parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	Yes	password	The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Yes	password	Verifies the db2inst1 password.
Password (db2fenc1)	Yes	password	The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Yes	password	Verifies the db2fenc1 password.

Table 4. Configured parameters (continued)

Parameter name	Required	Default value	Description
Password (dasusr1)	Yes	password	The user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Yes	password	Verifies the dasusr1 password.
Password (root)	Yes	password	The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes	password	Verifies the root password.
Password (virtuser)	Yes	password	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	Yes	password	Verifies the virtuser password.

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 5. Configurable parameters

Parameter name	Description
Virtual CPUs	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Verifies the db2inst1 password.
Password (db2fenc1)	The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Verifies the db2fenc1 password.
Password (dasusr1)	The password for the user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Verifies the dasusr1 password.
Password (root)	The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Verifies the root password.

Table 5. Configurable parameters (continued)

Parameter name	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.

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

DB2 Enterprise HADR Primary part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 6. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	Yes		The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Yes		Verifies the db2inst1 password.
Password (db2fenc1)	Yes		The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Yes		Verifies the db2fenc1 password.
Password (dasusr1)	Yes		The password for the user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Yes		Verifies the dasusr1 password.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the root password.
Password (virtuser)	Yes		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	Yes		Verifies the virtuser password.

DB2 Enterprise HADR Primary part configuration parameters for the SOA Policy Gateway Governance Master pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 7. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	Yes		The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Yes		Verifies the db2inst1 password.
Password (db2fenc1)	Yes		The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Yes		Verifies the db2fenc1 password.
Password (dasusr1)	Yes		The password for the user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Yes		Verifies the dasusr1 password.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the root password.
Password (virtuser)	Yes		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	Yes		Verifies the virtuser password.

DB2 Enterprise HADR Standby part

The DB2 Enterprise HADR Standby part provides some configuration options.

Table 8. Configurable parameters

Parameter name	Description
Virtual CPUs	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.

Table 8. Configurable parameters (continued)

Parameter name	Description
Verify password	Verifies the db2inst1 password.
Password (db2fenc1)	The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Verifies the db2fenc1 password.
Password (dasusr1)	The password for the user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Verifies the dasusr1 password.
Password (root)	The password for the root user ID. This is the password for the operating system of the virtual machine 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.

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

DB2 Enterprise HADR Standby part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 9. Configurable parameters

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

Table 9. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Password (db2fenc1)	Yes		The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Yes		Verifies the db2fenc1 password.
Password (dasusr1)	Yes		The password for the user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Yes		Verifies the dasusr1 password.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the root password.
Password (virtuser)	Yes		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	Yes		Verifies the virtuser password.

DB2 Enterprise HADR Standby part configuration parameters for the SOA Policy Gateway Governance Master pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 10. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.
Password (db2inst1)	Yes		The password for the user ID db2inst1 of the operating system. This user ID is used as the install owner of the DB2 instance and as the owner of the databases and schemas.
Verify password	Yes		Verifies the db2inst1 password.
Password (db2fenc1)	Yes		The password for the user ID used to run user-defined functions (UDFs) and stored procedures outside the address space used by the DB2 database. The fenced user is a user under which some stored procedures ("fenced" stored procedures) can run with reduced operating system authority. This can help prevent fenced stored procedures from overwriting instance files because the operating system will prevent it.
Verify password	Yes		Verifies the db2fenc1 password.

Table 10. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Password (dasusr1)	Yes		The password for the user ID for the DB2 administration server user that is used to run the DB2 administration server on your system. The default user is dasusr1 and the default group is dasadm1. This user ID is also used by the DB2 GUI tools to perform administration tasks against the local server database instances and databases.
Verify password	Yes		Verifies the dasusr1 password.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the root password.
Password (virtuser)	Yes		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	Yes		Verifies the virtuser password.

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 11. Configured parameters

Parameter name	Description
Virtual CPUs	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	The amount of memory allocated to this virtual machine, in megabytes.
Password (root)	The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Verifies user input for Password (root).
WebSphere administrative user name	The WebSphere environment admin user name.
WebSphere administrative password	The WebSphere environment admin user password.
Verify password	Verifies user input for WebSphere administrative password.
Reserve physical memory	The physical memory reserved for exclusive use by this virtual machine.

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

WSRR Standalone server part configuration parameters for the SOA Policy Gateway Basic Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 12. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	4096	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical memory	Yes	False	The physical memory reserved for exclusive use by this virtual machine.
Cell name	Yes	SOAPolicyBasicCell	The WebSphere cell name on the virtual machine in Basic Runtime pattern.
Node name	Yes	SOAPolicyBasicNode	The WebSphere node name on the virtual machine in Basic Runtime pattern.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies user input for Password (root).
WebSphere administrative user name	Yes	virtuser	The WebSphere environment admin user name.
WebSphere administrative password	Yes		The WebSphere environment admin user password.
Verify password	Yes		Verifies user input for WebSphere administrative password.

WSRR Standalone server part configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern

In the SOA Policy Gateway Basic Runtime Sample, default values are pre-configured for all parameters.

Table 13. Configured parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	4096	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical memory	Yes	False	The physical memory reserved for exclusive use by this virtual machine.
Password (root)	Yes	password	The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes	password	Verifies user input for Password (root).
WebSphere administrative user name	Yes	virtuser	The WebSphere environment admin user name.
WebSphere administrative password	Yes	password	The WebSphere environment admin user password.

Table 13. Configured parameters (continued)

Parameter name	Required	Default value	Description
Verify password	Yes	password	Verifies user input for WebSphere administrative password.

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 14. Configurable parameters

Parameter name	Description
Virtual CPUs	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical CPUs	The physical CPUs reserved for exclusive use by this virtual machine.
Reserve physical memory	The physical memory reserved for exclusive use by this virtual machine.
Cell name	The WebSphere cell name for the Advanced Runtime pattern.
Node name	The node name for the WebSphere 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 represented by this part in the pattern.
Verify password	Verifies user input for Password (root).
WebSphere administrative user name	The WebSphere environment admin user name.
WebSphere administrative password	The WebSphere environment admin user password.
Verify password	Verifies user input for WebSphere administrative password.

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

WSRR Deployment manager part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 15. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.

Table 15. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Reserve physical CPUs	Yes	False	The physical CPUs reserved for exclusive use by this virtual machine.
Reserve physical memory	Yes	False	The physical memory reserved for exclusive use by this virtual machine.
Cell name	Yes	SOAPolicyAdvancedCell	The WebSphere cell name for the Advanced Runtime pattern.
Node name	Yes	SOAPolicyAdvancedNode	The node name for the WebSphere node residing on the Deployment Manager virtual machine in Advanced Runtime pattern.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies user input for Password (root).
WebSphere administrative user name	Yes	virtuser	The WebSphere environment admin user name.
WebSphere administrative password	Yes		The WebSphere environment admin user password.
Verify password	Yes		Verifies user input for WebSphere administrative password.

WSRR Deployment manager part configuration parameters for the SOA Policy Gateway Governance Master pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 16. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	1	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	2048	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical CPUs	Yes	False	The physical CPUs reserved for exclusive use by this virtual machine.

Table 16. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Reserve physical memory	Yes	False	The physical memory reserved for exclusive use by this virtual machine.
Cell name	Yes	SOAPolicyGMCell	The WebSphere cell name for the Advanced Runtime pattern.
Node name	Yes	SOAPolicyGMNode	The node name for the WebSphere node residing on the Deployment Manager virtual machine in Advanced Runtime pattern.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies user input for Password (root).
WebSphere administrative user name	Yes	virtuser	The WebSphere environment admin user name.
WebSphere administrative password	Yes		The WebSphere environment admin user password.
Verify password	Yes		Verifies user input for WebSphere administrative password.

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 17. Configurable parameters

Parameter name	Description
Virtual CPUs	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical CPUs	The physical CPUs reserved for exclusive use by this virtual machine.
Reserve physical memory	The physical memory reserved for exclusive use by this virtual machine.
Cell name	The cell name value in the Custom node part configuration is ignored. Cell name specified in the Deployment manager part configuration is used.

Table 17. Configurable parameters (continued)

Parameter name	Description
Node name	The node name for the WebSphere node residing on the Custom node 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 represented by this part in the pattern.
Verify password	Verifies the user input for Password (root).
WebSphere administrative user name	The WebSphere environment admin user name.
WebSphere administrative password	The WebSphere environment admin user password.
Verify password	Verifies user input for WebSphere administrative password.

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

WSRR Custom nodes part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 18. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	2	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	4096	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical CPUs	Yes	False	The physical CPUs reserved for exclusive use by this virtual machine.
Reserve physical memory	Yes	False	The physical memory reserved for exclusive use by this virtual machine.
Node name	Yes	SOAPolicyAdvancedNode	The node name for the WebSphere node residing on the Custom node virtual machine in Advanced Runtime pattern.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the user input for Password (root).
WebSphere administrative user name	Yes	virtuser	The WebSphere environment admin user name.
WebSphere administrative password	Yes		The WebSphere environment admin user password.

Table 18. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Verify password	Yes		Verifies user input for WebSphere administrative password.

WSRR Custom nodes part configuration parameters for the SOA Policy Gateway Governance Master pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 19. Configurable parameters

Parameter name	Required	Default value	Description
Virtual CPUs	Yes	2	The number of virtual processors allocated for the virtual machine represented by this part.
Memory size (MB)	Yes	4096	The amount of memory allocated to this virtual machine, in megabytes.
Reserve physical CPUs	Yes	False	The physical CPUs reserved for exclusive use by this virtual machine.
Reserve physical memory	Yes	False	The physical memory reserved for exclusive use by this virtual machine.
Node name	Yes	SOAPolicyGMNode	The node name for the WebSphere node residing on the Custom node virtual machine in Advanced Runtime pattern.
Password (root)	Yes		The password for the root user ID. This is the password for the operating system of the virtual machine represented by this part in the pattern.
Verify password	Yes		Verifies the user input for Password (root).
WebSphere administrative user name	Yes	virtuser	The WebSphere environment admin user name.
WebSphere administrative password	Yes		The WebSphere environment admin user password.
Verify password	Yes		Verifies user input for WebSphere administrative password.

Script packages

There are 4 script packages provided with the IBM SOA Policy Gateway Pattern.

The script packages included with this pattern are:

- SOA Policy Gateway 2.0.0.0 - DataPower Domain
- SOA Policy Gateway 2.0.0.0 - Promotion
- SOA Policy Gateway 2.0.0.0 - Samples
- SOA Policy Gateway 2.0.0.0 - Security

Script: SOA Policy Gateway 2.0.0.0 - DataPower Domain

The DataPower Domain script provisions the DataPower domain during deployment. The script configures the connection between a single DataPower domain and the WSRR runtime. A separate DataPower Domain script is required for each DataPower domain that is connected to the WSRR runtime.

Parameters

Table 20. Configurable parameters

Parameter name	Description
DataPower_hostname	The hostname of the DataPower appliance where the sample application will be installed.
DataPower_XML_mgmt_port	The port used for the DataPower XML Management Interface, typically 5550.
Datapower_admin_id	The administrator user ID with appropriate permissions to use the XML Management Interface.
DataPower_admin_password	The password for the DataPower_admin_id.
Verify password	Verifies user input for DataPower_admin_password.
New_DataPower_domain	The new domain name to be created on the DataPower appliance. It must not match any existing domain or the script package will fail or exit. The value can not contain any spaces.
securityFileCleanUp	Determines if the `DomainZipFile.zip` file and the WSRR Certificate uploaded to DataPower are deleted from the WSRR instance where the script packages are run. If this file is not removed, it would be a security exposure if the certificates remained on the instance.

SOA Policy Gateway 2.0.0.0 - DataPower Domain script configuration parameters for the SOA Policy Gateway Basic Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 21. Configurable parameters

Parameter name	Required	Default value	Description
DataPower_hostname	Yes		The hostname of the DataPower appliance where the sample application will be installed.
DataPower_XML_mgmt_port	Yes	5550	The port used for the DataPower XML Management Interface, typically 5550.
Datapower_admin_id	Yes		The administrator user ID with appropriate permissions to use the XML Management Interface.
DataPower_admin_password	Yes		The password for the DataPower_admin_id.
Verify password	Yes		Verifies user input for DataPower_admin_password.
New_DataPower_domain	Yes		The new domain name to be created on the DataPower appliance. It must not match any existing domain or the script package will fail or exit. The value can not contain any spaces.

Table 21. Configurable parameters (continued)

Parameter name	Required	Default value	Description
Remove_security_files	Yes	true	Determines if the `DomainZipFile.zip` file and the WSRR Certificate uploaded to DataPower are deleted from the WSRR instance where the script packages are run. If this file is not removed, it would be a security exposure if the certificates remained on the instance.

SOA Policy Gateway 2.0.0.0 - DataPower Domain script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 22. Configurable parameters

Parameter name	Required	Default value	Description
DataPower_hostname	Yes		The hostname of the DataPower appliance where the sample application will be installed.
DataPower_XML_mgmt_port	Yes	5550	The port used for the DataPower XML Management Interface, typically 5550.
Datapower_admin_id	Yes		The administrator user ID with appropriate permissions to use the XML Management Interface.
DataPower_admin_password	Yes		The password for the DataPower_admin_id.
Verify password	Yes		Verifies user input for DataPower_admin_password.
New_DataPower_domain	Yes		The new domain name to be created on the DataPower appliance. It must not match any existing domain or the script package will fail or exit. The value can not contain any spaces.
Remove_security_files	Yes	true	Determines if the `DomainZipFile.zip` file and the WSRR Certificate uploaded to DataPower are deleted from the WSRR instance where the script packages are run. If this file is not removed, it would be a security exposure if the certificates remained on the instance.

Script: SOA Policy Gateway 2.0.0.0 - Promotion

The Promotion script enables a SOA Policy Gateway Basic Runtime or SOA Policy Gateway 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, whilst optionally configuring WSRR promotion into the governance master.

Parameters

Table 23. Configurable parameters

Parameter name	Description
WSRR_GOV_DMGR_hostname	The host name of the Dmgr for the WSRR Cluster.
WSRR_GOV_DMGR_cellname	The WebSphere Cell Name for the WSRR Cluster.
WSRR_GOV_admin_user	The Admin Id for the WebSphere WSRR Governance Cell.

Table 23. Configurable parameters (continued)

Parameter name	Description
WSRR_GOV_admin_password	The password for the Admin ID for the WebSphere WSRR Governance Cell.
Verify password	Verifies user input for WSRR_GOV_admin_password.
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 which 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.

SOA Policy Gateway 2.0.0.0 - Promotion script configuration parameters for the SOA Policy Gateway Basic Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 24. Configurable parameters

Parameter name	Required	Default value	Description
WSRR_GOV_DMGR_hostname	Yes		The host name of the Dmgr for the WSRR Cluster.
WSRR_GOV_DMGR_cellname	Yes		The WebSphere Cell Name for the WSRR Cluster.
WSRR_GOV_admin_user	Yes		The Admin Id for the WebSphere WSRR Governance Cell.
WSRR_GOV_admin_password	Yes		The password for the Admin ID for the WebSphere WSRR Governance Cell.
Verify password	Yes		Verifies user input for WSRR_GOV_admin_password.
Promotion_environment	Yes		Must be one of staging, production, or Unset. These values are case sensitive and must match exactly.
LTPA_key_password	Yes		An LTPA Key is exported and used during the Script Package which 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	Yes		Verifies user input for LTPA_key_password.

SOA Policy Gateway 2.0.0.0 - Promotion script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 25. Configurable parameters

Parameter name	Required	Default value	Description
WSRR_GOV_DMGR_hostname	Yes		The host name of the Dmgr for the WSRR Cluster.

Table 25. Configurable parameters (continued)

Parameter name	Required	Default value	Description
WSRR_GOV_DMGR_cellname	Yes		The WebSphere Cell Name for the WSRR Cluster.
WSRR_GOV_admin_user	Yes		The Admin Id for the WebSphere WSRR Governance Cell.
WSRR_GOV_admin_password	Yes		The password for the Admin ID for the WebSphere WSRR Governance Cell.
Verify password	Yes		Verifies user input for WSRR_GOV_admin_password.
Promotion_environment	Yes		Must be one of staging, production, or Unset. These values are case sensitive and must match exactly.
LTPA_key_password	Yes		An LTPA Key is exported and used during the Script Package which 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	Yes		Verifies user input for LTPA_key_password.

Script: SOA Policy Gateway 2.0.0.0 - Sample

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

Parameters

Note: Any parameter that requires the value Unset is case sensitive.

Table 26. Configurable parameters

Parameter name	Description
SCP_host	The host name of the SCP Server containing the `DomainZipFile.zip`.
SCP_user	The user name to use to connect to the SCP Server.
SCP_password	The password to use to log in to the SCP Server.
Verify password	Verifies user input for SCP_password.
SCP_zip_location	The URI location of the `DomainZipFile.zip`. For example, `/files/DomainZipFile.zip`.
CLIENT_PUBLIC_KEY_file	The name of the PEM Certificate File used to connect to the DataPower Appliances XML Management Interface port. Use the “Unset” value for Server Authentication only and for not using SSL.
CLIENT_PUBLIC _KEY_password	The password for the Public Certificate used to connect to the DataPower Appliances XML Management Interface port. The value is “Unset” if no password is used.
Verify password	Verifies user input for CLIENT_PUBLIC_KEY_password.

Table 26. Configurable parameters (continued)

Parameter name	Description
CLIENT_PRIVATE_KEY_file	The name of the PEM Key File used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. Use the "Unset" value for Server Authentication only and for not using SSL.
CLIENT_PRIVATE_KEY_password	The password for the key file used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. The value is "Unset" if no password is used.
Verify password	Verifies user input for CLIENT_PRIVATE_KEY_password.
CLI_FILE_file	The name of the CLI file contained in the `DomainZipFile.zip` file. This CLI is executed at the end of Domain install and WSRR Server Configuration.
Verify password	Verifies user input for LTPA_KEY_password.
DataPower_hostname	The hostname of the DataPower appliance where the sample application will be installed.
DataPower_XML_mgmt_port	The port used for the DataPower XML Management Interface.
DataPower_admin_id	The administrator user ID with appropriate permissions to use the XML Management Interface.
DataPower_admin_password	The password for the DataPower_admin_id.
Verify password	Verifies user input for DataPower_admin_password.
SOAPolicySample_DataPower_domain	The sample domain name. It must not match any existing domain on the DataPower appliance.
SamplePolicySample_starting_port	The application requires 5 free ports, which will be sequentially used from this value. For example, if the value is 62000, ports 62000-62004 will be used. No checking is done as to whether the ports are free by the script.
LDAP_hostname	The sample uses an LDAP server, this is the hostname of that server.
LDAP_port	The non-secure port of the LDAP server. Typically 389.
LDAP_password	The password used when binding with the LDAP_DN.
Verify password	Verifies user input for LDAP_password.
LDAP_DN	The distinguished name used to bind to the LDAP. For example, cn=root,dc=ibm.com.

SOA Policy Gateway 2.0.0.0 - Sample script configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Note: Any parameter that requires the value Unset is case sensitive.

Table 27. Configurable parameters

Parameter name	Required	Default value	Description
SCP_host	Yes		The host name of the SCP Server containing the `DomainZipFile.zip`.
SCP_user	Yes		The user name to use to connect to the SCP Server.
SCP_password	Yes		The password to use to log in to the SCP Server.
Verify password	Yes		Verifies user input for SCP_password.
SCP_zip_location	Yes		The URI location of the `DomainZipFile.zip`. For example, `/files/DomainZipFile.zip`.
CLIENT_PUBLIC_KEY_file	Yes		The name of the PEM Certificate File used to connect to the DataPower Appliances XML Management Interface port. Use the "Unset" value for Server Authentication only and for not using SSL.
CLIENT_PUBLIC _KEY_password	Yes		The password for the Public Certificate used to connect to the DataPower Appliances XML Management Interface port. The value is "Unset" if no password is used.
Verify password	Yes		Verifies user input for CLIENT_PUBLIC_KEY_password.
CLIENT_PRIVATE_KEY_file	Yes		The name of the PEM Key File used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. Use the "Unset" value for Server Authentication only and for not using SSL.
CLIENT_PRIVATE_KEY_password	Yes		The password for the key file used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. The value is "Unset" if no password is used.
Verify password	Yes		Verifies user input for CLIENT_PRIVATE_KEY_password.
DataPower_hostname	Yes		The hostname of the DataPower appliance where the sample application will be installed.
DataPower_XML_mgmt_port	Yes	5550	The port used for the DataPower XML Management Interface.
DataPower_admin_id	Yes		The administrator user ID with appropriate permissions to use the XML Management Interface.
DataPower_admin_password	Yes		The password for the DataPower_admin_id.
Verify password	Yes		Verifies user input for DataPower_admin_password.

Table 27. Configurable parameters (continued)

Parameter name	Required	Default value	Description
SOAPolicySample_DataPower_domain	Yes	SOAPolicySample	The sample domain name. It must not match any existing domain on the DataPower appliance.
SOAPolicySample_starting_port	Yes	62001	The application requires 5 free ports, which will be sequentially used from this value. For example, if the value is 62000, ports 62000-62004 will be used. No checking is done as to whether the ports are free by the script.
LDAP_hostname	Yes		The sample uses an LDAP server, this is the hostname of that server.
LDAP_port	Yes	389	The non-secure port of the LDAP server. Typically 389.
LDAP_password	Yes		The password used when binding with the LDAP_DN.
Verify password	Yes		Verifies user input for LDAP_password.
LDAP_DN	Yes		The distinguished name used to bind to the LDAP. For example, cn=root,dc=ibm.com.

Script: SOA Policy Gateway 2.0.0.0 - Security

The Security script copies security information, contained in a ZIP file, required for communicating with a DataPower appliance onto the Dmgr or WSRR machine from an external file server that supports Linux secure copy program (SCP).

The security file that is copied contains the following:

- DPC Access Certificate
- DPC Access Public Certificate
- DPC Private Key
- DP CLI Script
- Folder of certificate chain

The command line interface (CLI) script for DataPower allows you to configure a deployed domain during the pattern deployment phase.

Note: Confidential security certificates should be deleted from the external file server after deployment.

Parameters

Table 28. Configurable parameters

Parameter name	Description
SCP_host	The host name of the SCP Server containing the `DomainZipFile.zip` file.
SCP_user	The user name to use to connect to the SCP Server.
SCP_password	The password to use to log in to the SCP Server.
Verify password	Verifies user input for SCP_password.

Table 28. Configurable parameters (continued)

Parameter name	Description
SCP_zip_location	The URI location of the `DomainZipFile.zip` file; for example, `/files/DomainZipFile.zip`.
CLIENT_PUBLIC_KEY_file	The name of the PEM Certificate File used to connect to the DataPower Appliances XML Management Interface port.
CLIENT_PUBLIC_KEY_password	The password for the client certificate used to connect to the DataPower Appliances XML Management Interface port. This is required if available for Mutual Authentication. This value can be "Unset" if no password is used.
CLIENT_PRIVATE_KEY_file	The name of the PEM Key File used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication.
CLIENT_PRIVATE_KEY_password	The password for the key file used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. This value can be "Unset" if no password is used.
CLI_file	The name of the CLI file contained in the `DomainZipFile.zip`. This CLI is run at the end of Domain install and WSRR Server Configuration.

SOA Policy Gateway 2.0.0.0 - Security script configuration parameters for the SOA Policy Gateway Basic Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 29. Configurable parameters

Parameter name	Required	Default value	Description
SCP_host	Yes		The host name of the SCP Server containing the `DomainZipFile.zip` file.
SCP_user	Yes		The user name to use to connect to the SCP Server.
SCP_password	Yes		The password to use to log in to the SCP Server.
Verify password	Yes		Verifies user input for SCP_password.
SCP_zip_location	Yes		The URI location of the `DomainZipFile.zip` file; for example, `/files/DomainZipFile.zip`.
CLIENT_PUBLIC_KEY_file	Yes		The name of the PEM Certificate File used to connect to the DataPower Appliances XML Management Interface port.
CLIENT_PUBLIC_KEY_password	Yes		The password for the client certificate used to connect to the DataPower Appliances XML Management Interface port. This is required if available for Mutual Authentication. This value can be "Unset" if no password is used.

Table 29. Configurable parameters (continued)

Parameter name	Required	Default value	Description
CLIENT_PRIVATE_KEY_file	Yes		The name of the PEM Key File used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication.
CLIENT_PRIVATE_KEY_password	Yes		The password for the key file used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. This value can be "Unset" if no password is used.
CLI_file	Yes	Unset	The name of the CLI file contained in the `DomainZipFile.zip`. This CLI is run at the end of Domain install and WSRR Server Configuration.

SOA Policy Gateway 2.0.0.0 - Security script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern

Required parameters without a default value must be configured before the pattern can be deployed.

Table 30. Configurable parameters

Parameter name	Required	Default value	Description
SCP_zip_location	Yes		The URI location of the `DomainZipFile.zip` file; for example, `/files/DomainZipFile.zip`.
SCP_host	Yes		The host name of the SCP Server containing the `DomainZipFile.zip` file.
SCP_user	Yes		The user name to use to connect to the SCP Server.
SCP_password	Yes		The password to use to log in to the SCP Server.
Verify password	Yes		Verifies user input for SCP_password.
CLIENT_PUBLIC_KEY_file	Yes		The name of the PEM Certificate File used to connect to the DataPower Appliances XML Management Interface port.
CLIENT_PUBLIC_KEY_password	Yes		The password for the client certificate used to connect to the DataPower Appliances XML Management Interface port. This is required if available for Mutual Authentication. This value can be "Unset" if no password is used.
CLIENT_PRIVATE_KEY_file	Yes		The name of the PEM Key File used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication.

Table 30. Configurable parameters (continued)

Parameter name	Required	Default value	Description
CLIENT_PRIVATE_KEY_password	Yes		The password for the key file used to connect to the DataPower Appliances XML Management Interface port. This is required for Mutual Authentication. This value can be "Unset" if no password is used.
CLI_file	Yes	Unset	The name of the CLI file contained in the `DomainZipFile.zip`. This CLI is run at the end of Domain install and WSRR Server Configuration.

Chapter 5. Working with the IBM SOA Policy Gateway Pattern

The IBM SOA Policy Gateway Pattern provides a pattern definition for repeatable deployment of the topology that makes the product. Each pattern provides a specific function within the IBM SOA Policy Gateway Pattern and contains multiple images to support each pattern. The patterns must be configured before deployment based on the business needs.

As part of the deployment process, configure the part parameters. For more information, see "Deploying patterns" on page 61.

Related tasks:

Chapter 3, "Getting started with the IBM SOA Policy Gateway Pattern," on page 11
This pattern uses WebSphere DataPower to control messages using governed polices and service definitions in WSRR. Review the topics in this section to understand what is covered in this scenario, the reasons why a business might want to follow the scenario, the user roles involved, and an overview of the capability delivered with the product.

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. Determine the governance requirements and resources required.

In order to deploy the environment, prepare the DataPower appliance for remote administration and collect the assets required to securely communicate with the appliance. Testing the environment can be accomplished by deploying the SOA Policy Gateway Basic Runtime Sample, this confirms the environment is correctly configured for deployment and demonstrates the enforcement of the policies. After validation of the environment, the desired IBM SOA Policy Gateway Pattern governance and runtime configuration is decided using WSRR best practices. Deployment of the pattern starts with the Governance Master, followed by the Runtime patterns matched to desired configuration.

Preparing and deploying the IBM SOA Policy Gateway Pattern

Prepare DataPower and collect the security files:

1. Prepare the DataPower appliance for remote administration. For more information, see "Configuring DataPower for the IBM SOA Policy Gateway Patterns" on page 53.
2. If the DataPower appliance is secured, read the security section for DataPower, then collect the DataPower security files needed to communicate with it.
3. Confirm that a system DataPower in the cloud environment can communicate with the appliance and that the appliance can communicate with a deployed system.

The SOA Policy Gateway Basic Runtime Sample can be used to demonstrate the capabilities of the pattern before you create a production deployment. If the use of the Basic Runtime Sample is required, complete the following steps:

1. Provide an SCP server on Linux accessible from a deployed system within the cloud. SCP is the secure copy command. The SCP server provides a means to host the security files external to the pattern so the pattern will not need to be altered for every security configuration.
2. Provide an LDAP server to host the security IDs used by the sample application implemented in DataPower. For more information, see "Configuring the LDAP for the sample" on page 59.
3. Deploy the SOA Policy Gateway Basic Runtime Sample pattern to validate the infrastructure. For more information, see "Deploying the SOA Policy Gateway Basic Runtime Sample pattern" on page 62.
4. When use of the sample is complete, the LDAP server is not needed.

Prepare for production deployment:

1. Decide the scale needed for the deployment. Decide the cluster sizes for the Governance Master and the runtimes deployments.

   Note: When a cluster is deployed it cannot be extended with another cluster member.
2. Define the cell name and administrative user ID and password of the Governance Master.
3. Host the DataPower security `DomainZipFile.zip` file on an SCP server. For more information, see "Creating the Security `DomainZipFile.zip`" on page 54.

Deploy the Governance Master for the production environment:

1. Deploy a SOA Policy Gateway Governance Master pattern. Wait for the deployment to complete before deploying production environment runtime patterns. For more information, see "Deploying the SOA Policy Gateway Governance Master pattern" on page 63.

Deploy the production environment runtime patterns:

1. Decide whether a clustered or standalone environment is needed.
2. If more than one DataPower domain is required, clone the Basic Runtime pattern or Advanced Runtime pattern and add DataPower script packages to the clone for each domain needed.

   Note: Additional DataPower domains can not be added after this configuration has been completed.
   For more information, see "Deploying with multiple DataPower domains" on page 68.
3. Configure the runtime pattern with the Governance Master pattern information. For more information, see "SOA Policy Gateway Governance Master deployment information" on page 63.
4. Decide whether the runtime will be staging, production, or other.
5. Deploy the Basic Runtime or Advanced Runtime pattern. For more information, see "Deploying the SOA Policy Gateway Advanced Runtime pattern" on page 65 or "Deploying the SOA Policy Gateway Basic Runtime pattern" on page 64.
6. Wait until fully deployed before deploying another runtime

.

When deployment of the runtimes is completed:

1. The SCP file server is no longer required.

2. WSRR and WebSphere security can be updated from the default security configuration. For more information, see “Security management” on page 54.
3. The DataPower domain is ready for gateway configuration.

Configuring DataPower for the IBM SOA Policy Gateway Patterns

Complete the following DataPower configuration steps before running the SOAPolicy scripts.

Procedure

1. Log in to the supported DataPower appliance as an Administrator.
2. Search for `XML Management Interface`.
3. Make sure 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

Customers require different levels of security between WSRR and DataPower, particularly in the area of SSL. The IBM SOA Policy Gateway Pattern supports 3 levels of SSL communication between the configuration scripts and DataPower when using theSOA Policy Gateway Basic Runtime, SOA Policy Gateway Basic Runtime Sample, and SOA Policy Gateway Advanced Runtime patterns.

If SSL is not required

If you do not require SSL to be used, the public key and private keys for the curl client are not provided and left as “Unset”.

Note: If no SSL is used, all data sent to DataPower is unencrypted, including user and password information. This presents a security vulnerability. Passwords used in SOMA calls to DataPower do not support encryption, and are therefore are transported to the DataPower appliance unencrypted. Therefore, use server side authentication is used at a minimum to ensure security.

Mutual authentication between the DataPower applications and the scripts in the Basic and Advanced Patterns

If you require that mutual authentication occur between the DataPower applications and the scripts in the Basic and Advanced Patterns:

- The public key and private keys for the curl client must be provided.

Security management

The WSRR images and the WebSphere Application Server images used in the patterns have only the default security in place. In order to produce a truly secure environment, you need to secure them with standard WebSphere 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

Creating the Security `DomainZipFile.zip`

Create the Security `DomainZipFile.zip` for the SOA Policy Gateway Basic Runtime pattern, the SOA Policy Gateway Advanced Runtime pattern, and SOA Policy Gateway Basic Runtime Sample.

Procedure

Create the `DomainZipFile.zip` using the following rules:

1. The Structure of the `DomainZipFile.zip` must be as follows:

   Note: Only the directory structure is required, the individual file names can follow naming of your choice. However, all certificate and key files must be in PEM format.

   Note: The use of the DataPower Host Name in the path allows for different certificates to be used for different DataPower appliances.

Table 31. Files required for the Basic and Advanced patterns

File name, location relative to the root directory	Notes®
`CurlClientPublicKeyFile.crt`	Only required if Mutual Authentication is used. PEM format only.
`CurlClientPrivateKeyFile.key`	Only required if Mutual Authentication is used.
/dataPowerHostName/`certificate1.crt`	The DataPower certificates to be uploaded to WSRR. It requires that the entire Certificate Chain is in PEM format. DataPower certificates to be uploaded to WSRR. It must include only the following content: `------BEGINCERTIFICATE---- to -----END CERTIFICATE-----` The file extension must be either `.crt` or `.pem`.
/dataPowerHostName/`certificate2.crt`	The file extension must be either `.crt` or `.pem`
/dataPowerHostName/`certificate3.crt`	The file extension must be either `.crt` or `.pem`

2. For the SOA Policy Gateway Advanced Runtime pattern only, add the `cli` file to be run (optional):

Table 32. Additional files required for the Advanced pattern

File name, location relative to the root directory	Notes
`/cli.cli`	A single CLI file that will be run at the end of the DataPower Domain Configuration

3. Place `DomainZipFile.zip` on your SCP server location. Because of the sensitive nature of the files, it is recommended that you delete the file after configuration. The pattern configuration scripts will delete any files obtained from the `DomainZipFile.zip` as well as the copy of the `DomainZipFile.zip` that is created using SCP from your SCP environment.
4. Note the following SCP Server information:
   - The SCP Host Name
   - The SCP path to the `DomainZipFile.zip`
   - The SCP User and Password

Using the `DomainZipFile` file

Use cases of the `DomainZipFile` file for different levels of security in patterns.

The `DomainZipFile.zip` file can be used in the Basic Runtime, Basic Runtime Sample, and Advanced Runtime patterns.

SSL is not required to connect the pattern script packages to the DataPower appliance. If you do not use SSL, you do not have to create a `DomainZipFile.zip` file, unless you require a cli script to customize the DataPower domain created by the pattern. In this case, if you do not use server authentication as a minimum, the data will not be encrypted. This is a security risk because user and password information is passed to DataPower during the scripting client over a http connection, and this is protected by the certificates in the `DomainZipFile.zip` file.

If the DataPower host is not configured to validate the client certificate, you do not have to use Mutual Authentication between the script client and the DataPower appliance. It is recommended that you use Server Authentication at a minimum.

The case scenarios in this topic describe different levels of security.

The product supports the following case scenarios:

Case 1: No SSL is required

Case 2: No SSL is required but a cli script is needed to customize the domain

Case 3: Server authentication of the DataPower Certificate by the Script client is required

Case 4: Mutual Authentication with the DataPower Appliance is Required

Case 1: No SSL is required

It is recommended for the security reasons outlined that this option only be used for development scenarios. If you do not wish to use any SSL:

1. Set the parameters for SCP_host to “Unset”. If you are using the Basic Runtime or Advanced Runtime Patterns, SCP_host is in the SOA Policy Gateway 2.0.0.0 - Security Package Script. If you are using the Basic Runtime Sample pattern, SCP_host is in the SOA Policy Gateway 2.0.0.0 script. This sets the script in the pattern so that it does not retrieve the `DomainZipFile.zip` file using SCP.
2. Set the following parameters to “Unset” in the same script packages from step 1:

- CLIENT_PUBLIC_KEY_file
- CLIENT_PUBLIC_KEY_password
- Verify password
- CLIENT_PRIVATE_KEY_file
- CLIENT_PRIVATE_KEY_password
- Verify password

Case 2: No SSL is required but a cli script is needed to customize the domain

It is recommended for the security reasons outlined that this option only be used for development scenarios. If you do not want to use SSL but require a cli script:

1. Set the parameters for SCP_host to "Unset". If you are using the Basic or Advanced Runtime Patterns, SCP_host is in the SOA Policy Gateway 2.0.0.0 - Security Package Script. If you are using the Basic Runtime Sample pattern, SCP_host is in the SOA Policy Gateway 2.0.0.0 script. This sets the script in the pattern so that it does not retrieve the `DomainZipFile.zip` file using SCP.
2. Set the following parameters to Unset in the same script packages from step 1:
   - CLIENT_PUBLIC_KEY_file
   - CLIENT_PUBLIC_KEY_password
   - Verify password
   - CLIENT_PRIVATE_KEY_file
   - CLIENT_PRIVATE_KEY_password
   - Verify password

   Note: If SCP_host is "Unset" you do not require a `DomainZipFile.zip` file, unless you have a cli script you which want to run in the Basic Runtime and Advanced Runtime patterns.
3. Put the cli script file you want to use in the root of the `DomainZipFile.zip` file. An example structure of the `DomainZipFile.zip` file is as follows:

   ```
   /cli.cli
   ```

   This file is run at the end of the DataPower Domain script package. `cli.cli` is an example file name. The file name must not contain any spaces.

Case 3: Server authentication of the DataPower Certificate by the Script client is required

You must provide all the Certificates of the DataPower Certificate chain that protects the XML Management Interface. To locate these, complete the following steps:

1. Examine the SSL proxy profile for the XML Management Interface, and locate the CryptoProfile. The Crypto Profile will contain the identification credentials that contain the certificates used to protect the XML Management Interface.
2. Add these certificates to the `DomainZipFile.zip` file.

   The format is:
   - dataPowerHostName/certificateChainMember1.crt
   - dataPowerHostName/certificateChainMember2.pem
   - dataPowerHostName/certificateChainMember(n).crt

If you are using the multiple-domain scenario, the file can have two different `dataPowerHostName` directories, with the following files for each DataPower Certificate Chain:

- clientCertificate.crt clientKeyFile.key
- dataPowerHostName/certificateChainMember1.crt
- dataPowerHostName/certificateChainMember2.pem
- dataPowerHostName/certificateChainMember(n).crt
- dataPowerHostName2/certificateChainMember1a.crt
- dataPowerHostName2/certificateChainMember2a.pem
- dataPowerHostName2/certificateChainMember2(n).crt

Note: The DataPower Certificate chain files must be of type `.crt` or `.pem` and must only contain the certificate itself. The `.crt` or `.pem` file names used here are examples. The file name must not contain any spaces.

3. If you only require Server Authentication for the SOA Policy Gateway 2.0.0.0 - Security Package Script used by the Basic Runtime and Advanced Runtime Patterns, or the SOA Policy Gateway 2.0.0.0 - Sample script in the Basic Runtime Sample pattern, use "Unset" as the value for the following parameters in those scripts:
   - CLIENT_PUBLIC_KEY_file
   - CLIENT_PUBLIC_KEY_password
   - Verify password
   - CLIENT_PRIVATE_KEY_file
   - CLIENT_PRIVATE_KEY_password
   - Verify password
4. Optional: If a cli script is required:

   Put the cli script file you want to use in the root of the `DomainZipFile.zip` file. An example structure of the `DomainZipFile.zip` file is as follows:

   `/cli.cli`

   This file is run at the end of the DataPower Domain script package. `cli.cli` is an example file name. The file name must not contain any spaces.

Case 4: Mutual Authentication with the DataPower Appliance is Required

In this case, client and DataPower Server require validation of the other's certificates. This is only needed if the DataPower Host is configured in the SSL Proxy Profile for the XML Management Interface to validate the clients' certificates.

1. Add these certificates to the `DomainZipFile.zip` file.

   The format is:

   - clientCertificate.crt
   - clientKeyFile.key
   - dataPowerHostName/certificateChainMember1.crt
   - dataPowerHostName/certificateChainMember2.pem
   - dataPowerHostName/certificateChainMember(n).crt
   - dataPowerHostName2/certificateChainMember1a.crt
   - dataPowerHostName2/certificateChainMember2a.pem
   - dataPowerHostName2/certificateChainMember2(n).crt

Note: The DataPower Certificate chain files must be of type `.crt` or `.pem` and must only contain the certificate itself. The `.crt` or `.pem` file names used here are examples. The file name must not contain any spaces.

The client certificate and client key file can contain the data in the certificate or key file before the line in the file that reads: `-----BEGIN CERTIFICATE-----`.

2. If you require Mutual Authentication for the SOA Policy Gateway 2.0.0.0 - Security Package Script, used by the Basic and Advanced Runtime Patterns, or in the Basic Runtime Sample pattern SOA Policy Gateway 2.0.0.0 script, you must specify a value for the following parameters in these script packages:
   - CLIENT_PUBLIC_KEY_file
   - CLIENT_PRIVATE_KEY_file
   - CLIENT_PRIVATE_KEY_password
   - Verify password
3. If there is no password for the Public Key file, the value of the following can be "Unset":
   - CLIENT_PUBLIC_KEY_password
   - Verify password
4. The curl commands used by the script packages assume that the file type is `.pem`, so that the **--key-type** and **--cert-type** are set to PEM by default. The certificate and key files may contain this content before `-----BEGIN CERTIFICATE-----` in the particular certificate or key file.
5. Optional: If a cli script is required, using the Basic Runtime or Advanced Runtime patterns:

   Put the cli script file you want to use in the root of the `DomainZipFile.zip` file. An example structure of the `DomainZipFile.zip` file is as follows:

   `/cli.cli`

   This file is run at the end of the DataPower Domain script package. `cli.cli` is an example file name. The file name must not contain any spaces.

By selecting a case, you have configured the appropriate level of security, with or without using the `DomainZipFile.zip` file.

DataPower certificates to be uploaded to WSRR

You can provide a directory of certificates in the `dataPowerHostName` directory of the `DomainZipFile.zip` file. This can be uploaded to the WSRR Dmgr server or WSRR Standalone server.

Providing your own mechanism to download the `DomainZipFile.zip` file

You can provide your own `DomainZipFile.zip` without using the SCP server in the Security Script Package.

Procedure

To use other means to put the file into the environment, you must do the following:

1. The **`SCP_host`** parameter must be set to Unset.
2. You must create a custom script package to create the `DomainZipFile.zip` in the `/tmp` directory prior to running any of the SOA Gateway Pattern Scripts.
3. For Advanced Patterns, create the `DomainZipFile.zip` file in the `/tmp/security/RetrieveDomainFiles` directory.

4. For Basic with Sample Patterns, create the `DomainZipFile.zip` file in the `/installSample/Retrieve_Domain_Files` directory.

   Note: If the `DomainZipFile.zip` file is not present, the script might fail if the parameters indicate that certificates or keys are used.

CN values in certificates

The Certificates provided as part of the `DomainZipFile.zip` file must consider the CN value in the certificate.

HostName Verification is always active when you elect to use SSL, so you need to factor in the following when the Certificate is used in the Script Package:

- For Client Certificates (Public and Private/Key), you have no way of knowing the exact host that the WSRR Server or WSRR Dmgr that runs the script will be on. Therefore, the CN value must be generic enough to run on any potential client host in the IBM Workload Deployer environment; for example, `*clientname*.yourcompany.com`.
- The certificates for the DataPower machines are in individual directories in the `DomainZipFile.zip` file; for example:

```
dpHost1/cert1.crt
dpHost2/certb.crt
dpHost2/certbc.pem
```

- The CN value for the certificate (the final certificate in the chain for the DataPower host) must be valid for that host name; for example, `dp1.yourcompany.com` or `*dp*.yourcompany.com`.

Configuring the LDAP for the sample

The sample requires an Lightweight Directory Access Protocol (LDAP) with some specific entries.

About this task

The elements and properties must be defined when configuring the LDAP.

Note: Do not change these passwords.

As an alternative to the manual configuration steps, extract the content of the following `.zip` file, containing two LDIF files containing the configuration details provided in this task, and use these files to update the LDAP server: soaSamples.zip.

Procedure

Create an LDAP with the following elements:

1. Define the suffix:

```
dc=ibm.com
```

2. Define the domain `dc=ibm.com` with the following properties:

```
dn: dc=ibm.com
dc: ibm.com
objectclass: domain
objectclass: top
```

3. Define the containers:
   a. Define the container groups:

```
dn: cn=groups,dc=ibm.com
objectclass: container
objectclass: top
cn: groups
```

b. Define the container users:

```
dn: cn=users,dc=ibm.com
objectclass: container
objectclass: top
cn: users
```

4. Define the following users:
   a. User ConsumerA with the following properties:

```
dn: uid=ConsumerA,cn=users,dc=ibm.com
uid: ConsumerA
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: ConsumerA
cn: ConsumerA
userpassword: passw0rd
```

   b. User ConsumerB with the following properties:

```
dn: uid=ConsumerB,cn=users,dc=ibm.com
uid: ConsumerB
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: ConsumerB
cn: ConsumerB
userpassword: passw0rd
```

   c. User ConsumerX with the following properties:

```
dn: uid=ConsumerX,cn=users,dc=ibm.com
uid: ConsumerX
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: ConsumerX
cn: ConsumerX
userpassword: passw0rd
```

5. Define the following groups:
   a. Define the Group MANAGER with the following properties:

```
dn: cn=MANAGER,cn=groups,dc=ibm.com
objectclass: groupOfNames
objectclass: top
cn: MANAGER
member: uid=ConsumerX,cn=users,dc=ibm.com
```

   b. Define the Group Clerk with the following properties:

```
dn: cn=Clerk,cn=groups,dc=ibm.com
objectclass: groupOfNames
objectclass: top
cn: Clerk
member: uid=ConsumerA,cn=users,dc=ibm.com
```

   c. Define the Group Customer with the following properties:

```
dn: cn=Customer,cn=groups,dc=ibm.com
objectclass: groupOfNames
objectclass: top
cn: Customer
member: uid=ConsumerB,cn=users,dc=ibm.com
```

6. Make sure to collect the following information about the LDAP prior to running the sample:
   - The distinguished name (DN); for example `cn=root`.
   - The password; for example, `passw0rd`.
   - The non-secure port; for example, 389.
   - The LDAP Host name; for example, `ldap.customer.com`.

Deploying patterns

Deploying patterns with IBM Workload Deployer 3.1.0.2 or IBM SOA Policy Gateway Pattern into the cloud provides a running IBM PureApplication System environment. You can deploy the predefined patterns available with the IBM SOA Policy Gateway Pattern images, or deploy patterns that you have 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.

About this task

Deploying a pattern creates a virtual system, or a newly provisioned IBM SOA Policy Gateway Pattern runtime environment, that is running in the cloud.

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 defined and provides links to required consoles. The time for the deployment depends on the complexity of the pattern being deployed. A deployed pattern is a virtual system, or a newly 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 Workload Deployer: Managing virtual system patterns

IBM PureApplication System: Managing virtual system patterns

Deploying the SOA Policy Gateway Basic Runtime Sample pattern

Deploying the SOA Policy Gateway Basic Runtime Sample pattern creates a running virtual system instance of the pattern.

Before you begin

These prerequisites must be completed before deploying the pattern:

- Configure DataPower for the sample; see "Configuring DataPower for the IBM SOA Policy Gateway Patterns" on page 53.
- Configure Security for the sample; see "Security for the IBM SOA Policy Gateway Pattern patterns" on page 53.
- Setup the SCP server to host security files.
- Configure the LDAP for the sample; see "Configuring the LDAP for the sample" on page 59.

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. Click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.0.0.0 - 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. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the parts and script:

      Note: All passwords for this pattern, except the DataPower_admin_id parameter, are defaulted to `password`.
      - "DB2 Enterprise part configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern" on page 27.
      - "WSRR Standalone server part configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern" on page 34
      - "SOA Policy Gateway 2.0.0.0 - Sample script configuration parameters for the SOA Policy Gateway Basic Runtime Sample pattern" on page 44
5. Click **OK** to deploy the pattern.

What to do next

To verify the deployment, see "Verifying the deployment" on page 66.

Deploying the SOA Policy Gateway Governance Master pattern

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

About this task

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

Procedure

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

1. Click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.0.0.0 - Governance Master**.
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. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the part.
      - "DB2 Enterprise HADR Primary part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 30
      - "WSRR Deployment manager part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 36
      - "WSRR Custom nodes part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 39
      - "DB2 Enterprise HADR Standby part configuration parameters for the SOA Policy Gateway Governance Master pattern" on page 32
5. Click **OK** to deploy the pattern.

What to do next

To verify the deployment, see "Verifying the deployment" on page 66.

SOA Policy Gateway Governance Master deployment information

The Governance Master must be deployed before the SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced 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:
   - 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.

   Note: The hostname or IP, cell name, and the WebSphere administrative username and password used during deployment of the Governance Master instance are required inputs to the following parameters in the SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced Runtime patterns:
   - WSRR_GOV_DMGR_hostname
   - WSRR_GOV_DMGR_cellname
   - WSRR_GOV_admin_user
   - WSRR_GOV_admin_password

Deploying the SOA Policy Gateway Basic Runtime pattern

Deploying the SOA Policy Gateway Basic Runtime pattern creates a running virtual system instance of the pattern.

Before you begin

Complete the following before deploying the Basic Runtime pattern:

- Configure DataPower for the IBM SOA Policy Gateway Pattern; see "Configuring DataPower for the IBM SOA Policy Gateway Patterns" on page 53.
- Configure Security for the IBM SOA Policy Gateway Pattern; see "Security for the IBM SOA Policy Gateway Pattern patterns" on page 53.
- Setup the SCP server to host security files.
- Obtain the Governance Master deployment information; see "SOA Policy Gateway Governance Master deployment information" on page 63.

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 SOA Policy Gateway Basic Runtime pattern or SOA Policy Gateway Advanced Runtime pattern. This 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 the SOA Policy Gateway Basic Runtime pattern, complete the following steps:

1. Click **Patterns** > **Virtual Systems**.
2. From the Virtual System Patterns list, select **SOA Policy Gateway Basic Runtime 2.0.0.0**.
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. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the parts and scripts:
   - "DB2 Enterprise part configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 26
   - "WSRR Standalone server part configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 33
   - "SOA Policy Gateway 2.0.0.0 - Security script configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 47
   - "SOA Policy Gateway 2.0.0.0 - Promotion script configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 42
   - "SOA Policy Gateway 2.0.0.0 - DataPower Domain script configuration parameters for the SOA Policy Gateway Basic Runtime pattern" on page 40
5. Click **OK** to deploy the pattern.

What to do next

To verify the deployment, see "Verifying the deployment" on page 66.

Deploying the SOA Policy Gateway Advanced Runtime pattern

Deploying the SOA Policy Gateway Advanced Runtime pattern creates a running virtual system instance of the pattern.

Before you begin

Complete the following before deploying the Advanced Runtime pattern:

- Configure DataPower for the IBM SOA Policy Gateway Pattern; see "Configuring DataPower for the IBM SOA Policy Gateway Patterns" on page 53.
- Configure Security for the IBM SOA Policy Gateway Pattern; see "Security for the IBM SOA Policy Gateway Pattern patterns" on page 53.
- Setup the SCP server to host security files.
- Obtain the Governance Master deployment information; see "SOA Policy Gateway Governance Master deployment information" on page 63.

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 SOA Policy Gateway Basic Runtime pattern or SOA Policy Gateway Advanced Runtime pattern. This 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 the SOA Policy Gateway Advanced Runtime pattern, complete the following steps:

1. Click **Patterns** > **Virtual Systems**.

2. From the Virtual System Patterns list, select **SOA Policy Gateway 2.0.0.0 - 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. Optional: Choose the environment and schedule the deployment.
   c. Configure the virtual patterns. Click **Configure virtual parts**, then click the part name to open the editor for the parts and scripts:
      - "DB2 Enterprise HADR Primary part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 29
      - "WSRR Deployment manager part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 35
      - "SOA Policy Gateway 2.0.0.0 - Security script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 48
      - "SOA Policy Gateway 2.0.0.0 - Promotion script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 42
      - "SOA Policy Gateway 2.0.0.0 - DataPower Domain script configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 41
      - "WSRR Custom nodes part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 38
      - "DB2 Enterprise HADR Standby part configuration parameters for the SOA Policy Gateway Advanced Runtime pattern" on page 31
5. Click **OK** to deploy.

What to do next

To verify the deployment, see "Verifying the deployment."

Verifying the deployment

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

Procedure

1. Check the deployment logs for any failure in the virtual system deployment history. For more information, see "Troubleshooting problems with deployment" on page 109.
2. Optional: If you have deployed the SOA Policy Gateway Basic Runtime Sample, test the deployed instance by following the tutorial to send some sample messages using the sample applications provided. See "Running the sample test cases" on page 71.

Scenario: Adding an additional runtime to the pattern

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. This

means that there are states and transitions that are specific to the Staging and Production environments, thus allowing for controlled promotion into these runtimes by defining the target systems in the promotion configuration file. This is appropriate if your organization defines environments in the same way, with Staging as a pre-Production environment that allows testing before allowing the Capability Version to be opened to generalized use. However, many organizations require additional 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 51.

Procedure

1. Deploy the predefined SOA Policy Gateway Governance Master. For more information, see "Deploying the SOA Policy Gateway Governance Master pattern" on page 63.
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 SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced Runtime patterns with the Governance Master details. For more information, see "SOA Policy Gateway Governance Master deployment information" on page 63.

   Note: The promotion environment value must be set to "Unset".
4. Deploy the predefined SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced Runtime. For more information, see "Deploying the SOA Policy Gateway Basic Runtime pattern" on page 64 and "Deploying the SOA Policy Gateway Advanced Runtime pattern" on page 65.

Cloning and customizing the IBM SOA Policy Gateway Pattern

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

About this task

You can customize the patterns in the following ways:

- Adding additional DataPower domains. For more information, see "Deploying with multiple DataPower domains" on page 68.
- Increasing the default cluster size. For more information, see the IBM Workload Deployer, Version 3.1 Information Center.

  Note: When expanding the cluster size, increase the memory size of the WSRR Deployment Manager too.
- Allow you to choose the way to get the compressed security file on the server. For more information, see "Security management" on page 54.
- Allow you to define and lock your own default values; for example, the DataPower administrator ID. For more information about locking parameters, see the IBM Workload Deployer, Version 3.1 Information Center.

- Allow you to use your own mechanism to download the `DomainZipFile.zip` file. For more information, see "Providing your own mechanism to download the `DomainZipFile.zip` file" on page 58.

Procedure

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

1. From the left panel of the Pattern window, select the pattern to clone.
2. Click the Clone icon and provide a name for the new pattern. You can also provide additional information, like a description.
3. Select the new pattern and click the Edit icon to change the configuration. You can add and remove parts and configure them, increase or decrease the number of some parts, or change the order in which some parts are deployed.

What to do next

Ensure that you have all required parts properly configured for the type of pattern you created. You can deploy the pattern when your configuration is complete.

Related information:

IBM Workload Deployer: Managing virtual system patterns

Deploying with multiple DataPower domains

The SOA Policy Gateway Basic Runtime and SOA Policy Gateway Advanced Runtime patterns can be cloned and customized to include multiple DataPower domains.

Procedure

1. Clone the SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced Runtime pattern. For more information, see "Cloning and customizing the IBM SOA Policy Gateway Pattern" on page 67.
2. To edit the pattern, click **Edit**.
3. Expand the **Scripts** section.
4. For each additional domain to be added, drag and drop the **SOA Policy Gateway 2.0.0.0 DataPower Domain** script package onto the WSRR Deployment manager part for the Advanced Runtime pattern, or onto the WSRR Standalone part for the Basic Runtime pattern.
5. Click **Done editing**.
6. Deploy the pattern, entering the following information for each domain added:
   - DataPower_hostname
   - DataPower_XML_mgmt_port
   - DataPower_admin_id
   - DataPower_admin_password
   - Verify password
   - New_DataPower_domain
   - securityFileCleanUp

     Note: When using multiple domains, the last domain must have the securityFileCleanUp value set to **`true`**, and all other domains must have the value set to **`false`**.

For more information about deploying the patterns, see “Deploying the SOA Policy Gateway Basic Runtime pattern” on page 64 or “Deploying the SOA Policy Gateway Advanced Runtime pattern” on page 65.

The sample application

The sample application is a configurable DataPower Domain and a set of WSRR Artifacts that can be used to demonstrate the capabilities of the pattern.

The basic scenario in the sample application is an inventory application for a store (Warehouse). There is a Store web service that has three operations:

- purchase
- findInventory
- returnProduct

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

- Validation against Store.wsdl. This assumes that the DataPower Validation is turned off.
- Reject if there are more than 5 messages in 90 seconds. This is a low threshold for easy demos.

The consumers of this service currently have two Service Level Agreements (SLAs), Gold and Anonymous. If the customer context in the HTTP header is Gold, they are routed to the Alternate endpoint immediately. If they are anonymous, that is currently not gold, they go to the Store Mock Service endpoint, which has a different price value for the item.

The scenario also performs authorization for the findInventory operation, based upon user group membership. Figure 5 on page 70 shows the flow of the application with each box representing a different DataPower gateway.

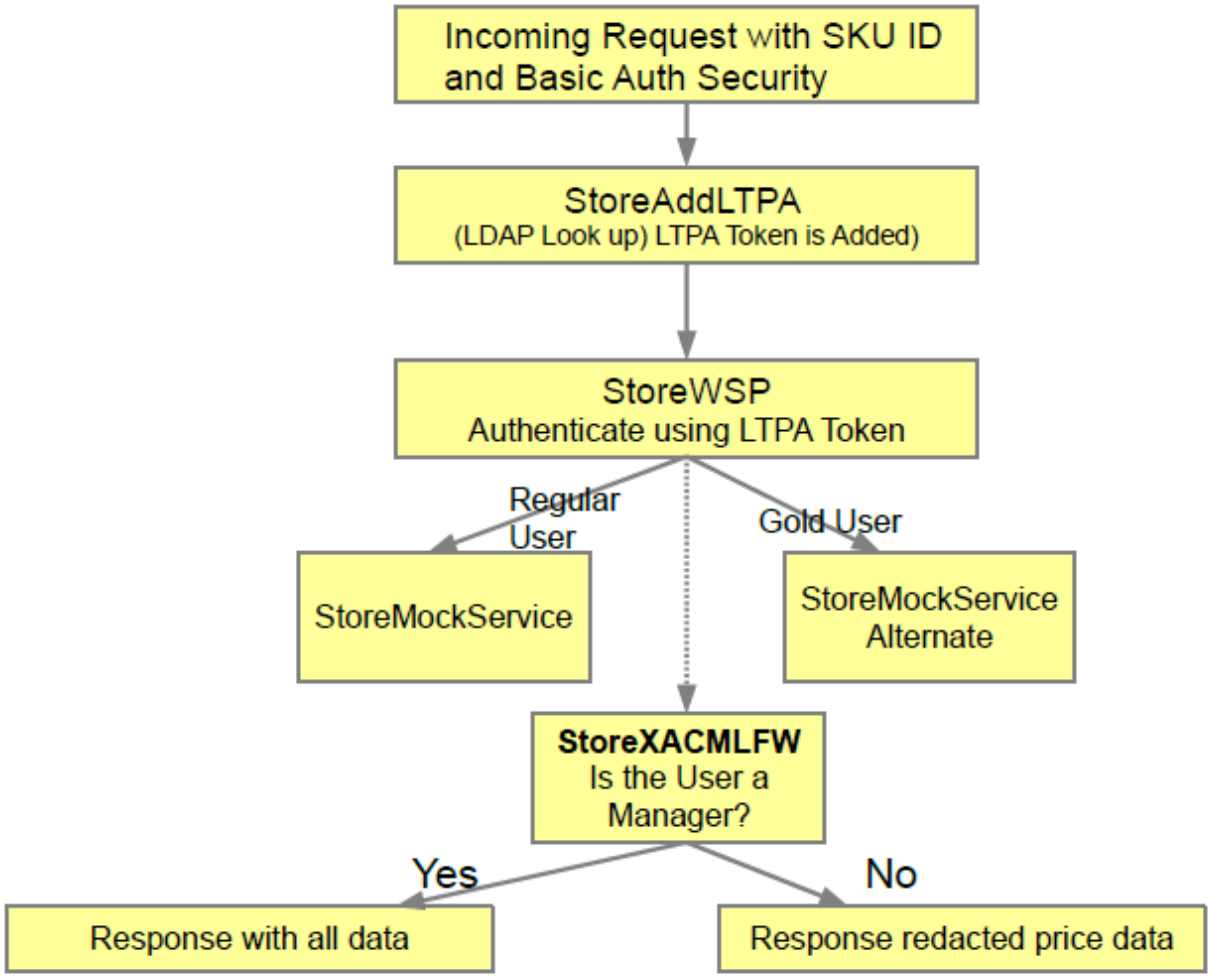


Figure 5. The sample application flow diagram

Related tasks:

"Cloning and customizing the IBM SOA Policy Gateway Pattern" on page 67
The IBM SOA Policy Gateway Pattern cannot be edited. If the topology provided in the IBM SOA Policy Gateway Pattern virtual system patterns do not provide the function you need, the pattern can be cloned and then edited to create new patterns.

Overview of WSRR artifacts in the sample

The WSRR artifacts describe the warehousing operation.

There are basic business capabilities for Warehouse, that is part of the larger Bob's Warehouse Organization. The service version, Store V1.0, represents the Store service. The Store service level definition (SLD) has two service level agreements (SLAs); one for Gold users that routes them to an alternate preferred service, and the Anonymous Users SLA that is for all other users and simply logs a notification on DataPower that the request was made. The Store SLD also has two other sample policies attached; the first policy rejects messages after 5 messages in 90 seconds and the second policy does validation against the Store.wsdl schema.

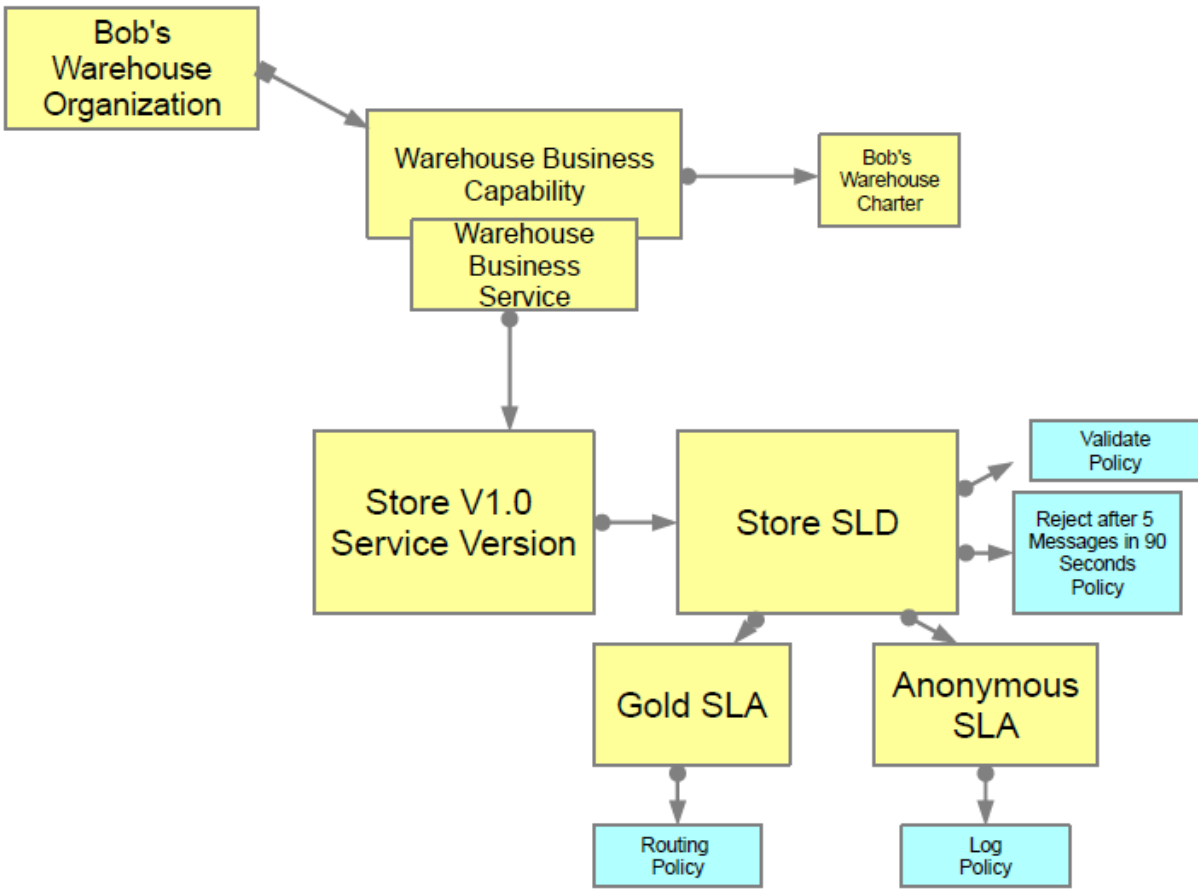


Figure 6. The sample domain

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. There are six command line test variations that can be run on the sample application.

To deploy the Basic Sample Runtime, see "Deploying the SOA Policy Gateway Basic Runtime Sample pattern" on page 62.

Note: The value of `SamplePolicySample_starting_port` used in the following XML samples is found in the logs for the SOA Policy Gateway Basic Runtime Sample.

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 do this, 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 testing screen for the sample application implemented in DataPower is displayed.
4. The options are:
   - **Send Standard** - Sends a findInventory request to the store service. The context ID is a "Silver" user. A successful result is `Part: SKU10 Price: 461.73`.
   - **Send Routed** - Sends a findInventory request to the store service. The context ID is a "Gold" user, so the request is routed to a Gold implementation of the service. A successful result is `Part: GOLDSKU10 Price: 461.73`.
   - **Send Invalid** - Sends a request with an invalid payload. The validation policy requires DataPower to validate the request and a successful result will be a response message from DataPower `"Internal Error (from client)"`.
   - **User ID = ConsumerA** - For calls with a UserID of ConsumerA, the XACML policy is enforced so that only Managers can see the price. The value of Price in the response message will be redacted. A successful result contains `Price: 0.0`.
   - **Many Standard Requests** - If more than 5 requests are performed within 90 seconds, the rejection policy is enforced. A successful response demonstrating the policy being enforced is: `Rejected: "Rejected (from client)"`.
5. Open the WSRR console and explore the service and policies. For more information, see "Connecting to WSRR - Business Space" on page 92.

To run the sample application test cases using the command line:

Demonstrating XACML Permit/Deny with the Redaction scenario 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 above is contained in a file named `silver.xml`, run the following curl command:

```
curl -k --data-bin @./silver.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd http://<yourDataPowerHostName>.com:<SamplePolicySample_starting_port+4>/Store/Store
```

In this example, ConsumerX is a Manager so we will see the full price information as 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 using the command line

ConsumerA is not a manager so will see a different response. Run the curl command:

```
curl -k --data-bin @./silver.xml -H "Content-Type: text/xml"
-u ConsumerA:passw0rd http://<yourDataPowerHostName>.com:<SamplePolicySample_starting_port+4>/Store/Store
```

Notice that the response has price redacted and is 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 using the command line

The SLA ContextId is used to trigger the Routing Policy. In this case, the SLA for Gold Customers has the value of “Gold” in the SLA. Here is the content of a sample request with Gold as the contextIdentifier:

```
<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 above is contained in a file named `gold.xml`, run the following curl command:

```
curl -k --data-bin @./gold.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd http://<yourDataPowerHostName>.com:<SamplePolicySample_starting_port+4>/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 using the command line

The validation policy checks the schema of the request against the Store.wsl 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 we run the following curl request:

```
curl -k --data-bin @./badvalid.xml -H "Content-Type: text/xml"
-u ConsumerX:passw0rd
http://<yourDataPowerHostName>.com:<SamplePolicySample_starting_port+4>/Store/Store
```

This produces the following error:

```
<?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 using the command line

One of the mediation policies included in the sample tests rejection after the message count has run 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>.com:<SamplePolicySample_starting_port+4>/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 will be displayed as below 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 you will see the following error:

```
<?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 have run other tests within the 90 second interval.

Testing notification in the mediation policy using the command line

In the case where the contextId is not "Gold", there is no SLA mapped and the Anonymous SLA is utilized. The mediation policy for the Anonymous SLA is to log or notify. This requires that Debug Mode be enabled for the Sample domain. Run the following command:

```
curl -k --data-bin @./silver.xml -H "Content-Type: text/xml" -u ConsumerX:passw0rd
http://<yourDataPowerHostName>.com:<SamplePolicySample_starting_port+4>/Store/Store
```

In this case ConsumerX is a Manager, so we will see the full price information 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:fin
dInventoryResponse 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>
```

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 debug 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 "debug" and click **Set Log Level**.

To find the log, select **Files** and **File Administration** > **File Management**. The log is located in the `logtemp` folder and named `default-log`. Because of the wrapping of the log, you might need to put the log file in a web browser window prior to running the test, and refresh the tab in the browser after running the test.

Related tasks:

"Deploying the SOA Policy Gateway Basic Runtime Sample pattern" on page 62
Deploying the SOA Policy Gateway Basic Runtime Sample pattern creates a running virtual system instance of the pattern.

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 using url-open. The call is on box to another XML Firewall, so no SSL Proxy profile is used. Had it been desired to move the Policy Decision Point (PDP) to another DataPower box, an SSL Proxy profile could have been 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>
```

If we examine the `sendToPDP.xsl` file we should note the following items:

1. The stylesheet 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.

Let us examine the policy style sheet for redaction.

StorePrivateDataXACML.xml

```
<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:

- 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

There are several points at which you can modify the `.xsl` scripts used in the application.

Procedure

To modify the sample XSL style sheets, you can:

1. Modify the credential mapping for AZ.

   Open the `rgxacml.xsl` style sheet and complete the following XSL statements:

```
<!-- Specify your LDAP Server -->
<xsl:variable name="server"><xsl:copy-of select="$LDAPHost"/></xsl:variable>
<xsl:variable name="bindDN"><xsl:copy-of select="$LDAPCN"/></xsl:variable>
<xsl:variable name="bindPassword"><xsl:copy-of
select="$LDAPPassword"/></xsl:variable>
<xsl:variable name="port"><xsl:copy-of select="$LDAPPort"/></xsl:variable>
```

   The following variables are defined in the `soavars.xsl` style sheet:

```
<xsl:variable name="LDAPHost" select='"yourldap.something.com"' />
<xsl:variable name="LDAPPort" select='"389"' />
<xsl:variable name="LDAPCN" select='"cn=root"' />
<xsl:variable name="LDAPPassword" select='"passw0rd"' />
<xsl:variable name="StoreGWHost" select='"yourDatapowerName"' />
<xsl:variable name="StoreGWPort" select='"62151"' />
```

   The sample contains an unencrypted password to the LDAP Server, it could be that you want to customize the provided style sheet to decrypt an encrypted password.

```
<!-- Specify base DN to begin search -->
<xsl:variable name="baseDN">dc=ibm.com</xsl:variable>
```

   The `baseDN` is hard coded as `dc=ibm.com`. If you have configured your LDAP with a different Suffix, baseDN, change this line to customize the sample.

2. Modify the Redaction style sheet.

   The `noPriceInfo.xsl` style sheet contains the following code, which will zero out any price values. 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 on an existing PDP or click **Add**.
3. Enter a URL; for example `local:///storePrivateDataXACML.xml`.
4. Add any dependent or directory files needed 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 re-enter the URL, or anything you've changed, or restart the domain for your change to take effect.

Editing policy documents

Use the Business Space user interface to edit policy documents.

Before you begin

Configure the SOA Governance space. For more information, see "Configuring Business Space for the first use" on page 93.

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 policies" on page 105.
2. Click **Finish**. The Browse view is displayed
3. Govern the mediation policy. For more information about governing a policy document, see "Managing the lifecycle of the policy" on page 107.
   a. Click on 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.

Related tasks:

"Authoring new policies" on page 105
When authoring mediation policies in the Business Space user interface, specify the conditions and actions for the policy.

"Managing the lifecycle of the policy" on page 107
Policies can be transitioned between governance states using the Business Space user interface.

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 templates for your own applications. The sample environment contains five gateways. There is one primary gateway for the Store service, and four supporting gateways provide example back-ends for the Store Gateway to call, XACML support for a redaction scenario, and a front end to provide additional 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 70.
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. This 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 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 performed by customers.
4. Sets the ConsumerID value 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 than users can call using only Basic authentication (for example, no LTPA or similar). The request processing rule:

1. Identifies with Basic authentication.
2. Authenticates with a very 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 using 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, responds 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 using an XML Firewall as an implementation. The findInventory, purchase, and return operations all are supported. This service is used to demonstrate the routing policy being enforced.

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 provided 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 using SSL, to the StoreXMLFW gateway.

The request rule of the StoreXML firewall does the following:

1. Performs AAA using the SSL information for authentication.
2. Performs authorization using an on-box XACML PDP. The policy used by the PDP is originally authored in IBM Tivoli Security Policy Manager but can be recreated 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 is thrown that is handled by the exception processing rule and returns a Deny response to the client.

Note: This 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 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 by hand. 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 used in the sample is contained in `storeSWPXACML.xml` and `storePrivateDataXACML.xml`. These policies are utilized 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 is the operation on the web service and the second 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 will see that there are similar mappings for Customer and Clerk. You will see that the findInventory operation is authorized to use all three groups whilst the returnProduce and purchase operations are limited to only certain groups.

The Redaction Gateway

Details about the storeCallPDP.xsl style sheet.

If you examine the storeCallPDP.xsl style sheet you will note these things:

1. The inclusion of the storeSendToPDP.xsl style sheet. This is the style sheet with the logic to call storeXAMLFW.
2. The call to the templete 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. Examining the XACML for the Reaction, storePrivateDataXACML.xml, the structure 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 33. WSRR artifacts created for the SOA Policy Gateway Basic Runtime Sample pattern

Object	Description
Organization	Bob's Warehouse.
Business Capability	Warehouse, owned by the Bob's Warehouse organization.
Service Version	Store 1.0 uses the Store Web Service, the Store Service Level Definition (SLD), and Warehouse Business Capability.
WSDL	Store.wsdl
XSD	Company.xsd
Policy	• Validate.xml • RouteForGold.xml • LogEveryTime.xml • RejectAfter5MsgIn90Seconds.xml
Policy Attachments	• Anonymous Users_GenericObject_Anonymous Users_LogEveryTime.xml - Attaches the LogEveryTime policy to the Anonymous Users Service Level Agreement (SLA). • Gold SLA_GenericObject_Gold SLA_RouteForGold.xml - Attaches the RouteForGold policy to the Gold SLA. • Store_GenericObject_Store_urn :RejectAfter5MsgIn90Seconds.xml - Attaches the RejectAfter5MsgIn90Seconds policy to the Store SLD. • Store_GenericObject_Store_urn:Validate.xml - Attaches the Validate policy to the Store SLD.
SLD	Store SLD - Used by the Store 1.0 Service Version.
SLA	Gold SLA - Routes to the Gold endpoint if the ContextId is “Gold”.

Table 33. WSRR artifacts created for the SOA Policy Gateway Basic Runtime Sample pattern (continued)

Object	Description
Anonymous SLA	Anonymous Users - Uses the LogEveryTime policy notification and is performed if the ContextId is not "Gold".

Sample application use of WSRR artifacts

The StoreWSP uses a WSRR Subscription to retrieve WSDL and policy artifacts. Whenever a request is processed through StoreWSP, the following actions are taken:

1. The Store 1.0 service version is connected to the Store SLD, that has two direct policies attached, Validate and RejectAfter5MsgIn90Seconds. The order the policies are run is indeterminate.
   a. If 5 requests have occurred in the last 90 seconds the request is rejected.
   b. The request is validated against Store.wsdl with its associated Company.xsd.
2. The Store 1.0 service uses the Store SLD, that has two SLAs; the Gold SLA for use with Gold users and the Anonymous Users SLA for all other users. If the ContextId attribute is "Gold" the request is routed to the StoreMockServiceAlternate XML Firewall, otherwise if it is "Silver" or any other value, the Anonymous Users SLA takes over and the LogEveryTime policy is run. This puts a notification in the `default.log` of the Sample domain. This notification can only be seen if the debug mode is enabled on the domain. The message is then routed to the StoreMockService XML firewall.

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 34. 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.

Table 34. DataPower artifacts created for theSOA Policy Gateway Basic Runtime Sample pattern (continued)

Type	Name	Purpose
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.
Validation Credentials	WSRRVC	Validation credentials contains 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 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 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 which 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 as to 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`, and are located in the local directory of the installed domain.

Table 35. 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 which 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 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 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 36. 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

When the IBM SOA Policy Gateway Pattern image has been deployed, you can register your own service definitions and attach your own policies to the definitions. You can also view and manage your deployed systems. To view the list of deployed instances, click **Instances** > **Virtual system**.

Viewing the instance details

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

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

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. Expand a running virtual machine to see the script logs in the **Script Packages** section and links to accessing the virtual machine using in the **Consoles** section.

Administering deployed instances

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

Before you begin

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

About this task

Deploying a pattern creates a virtual system instance, or a newly provisioned IBM SOA Policy 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 into the components of the virtual system from the console links in the virtual system view. The components available depends on the pattern that you created. For example, you could:
   - Launch and log in to the administrative console for the deployment manager and then see the clusters created.
   - Launch the process center and then download the process designer to author process applications.
   - Set up IBM Integration Designer and connect to the process center for process authoring.

Connecting to WSRR - Business Space

Use the Business Space user interface to administer policies.

About this task

Access the Business Space user interface using the host address of the WSRR system.

Procedure

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

        where *<hostname>* and *port* are the hostname and port value of the WSRR server.

Results

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

What to do next

If using Business Space on the WSRR system for the first time, see "Configuring Business Space for the first use" on page 93 and follow the steps to create the Operations space.

Related information:

IBM WebSphere Service Registry and Repository Version 8.0 Information Center

Connecting to WSRR - Service Registry Console

Use the Service Registry Console to classify service versions.

About this task

Access the Service Registry Console user interface using the host address of the WSRR system.

Procedure

1. Click **Instances** > **Virtual Systems** to access the Virtual System Instances window.
2. From the list of instances in the Virtual System Instances window, select the instance that was deployed. The instance details are displayed.
3. Access the WSRR system:
   - In the **Consoles** section, click **WSRR_Web_UI** to connect to the Business Space running on the WSRR system.
   - Alternatively, in an external Web browser:
     a. Find the hostname and port numbers for WSRR. Expand the **Virtual machines** section and select the virtual machine for the WSRR Standalone Server to see the virtual machine details. In the **Hardware and network** section, the hostname is the **Network interface 0** value.
     b. Enter the Service Registry Console URL: http://*hostname*/ServiceRegistry where *hostname* is the host name of the WSRR server.

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 92.

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 has not been created yet, you must create it. To create a space 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 on the space for SOA Governance users. The specific name will depend on what was specified when the space was created.

What to do next

You can add additional 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**.

Post-deployment pattern configuration

After deploying the patterns, security and other settings must be configured.

LDAP settings changes for the sample application

If you are using the SOA Policy Gateway Basic Runtime Sample and you have need to change the security settings for your LDAP server; for example, the password or the user name, you need to change these values in two places.

The places to make the changes are:

- The AAA Policy Authentication Section for the AAA policy StoreAddLTPA - To find this policy, use the search window of the DataPower Administration web user interface and search for `AAA`. Select the correct AAA Policy and change the value in the Authentication tab.
- The `soavars.xsl` file - Use the File Management Section in the DataPower Web Admin User Interface. Open the domain created by the SOA Policy Gateway Basic Runtime Sample pattern on the DataPower appliance and browse in the local directory for the `soavars.xsl` file. Change the LDAPHost, LDAPPort, LDAPCN, LDAPPassword variables as necessary.

  Note: You might need to restart the domain for these changes to take effect.

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.

DN host verification is not enabled in WebSphere Application Server by default. However, in the script packages used by the IBM SOA Policy Gateway Patterns, DN host verification is turned on and can not disabled. A very specific certificate that works between the default WebSphere Application Server and DataPower might not work for the "SOA Policy Gateway 2.0.0.0 - Security" script package or the "SOA Policy Gateway 2.0.0.0 - Sample" script package 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 used with the deployment, see "Removing or Adding DataPower Certificates to the WSRR Truststore."

Changing the LTPA Keys

This procedure describes how to change the LTPA key. The LTPA key is shared amongst all cells in Basic. 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, production, or Unset.

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 an Advanced ND environment, 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 environment is Advanced, 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, it must be stopped and restarted for the LTPA Key change to take effect.

Removing or Adding DataPower Certificates to the WSRR Truststore

This task describes how to add or remove DataPower certificates. A benefit of performing this task is that it simplifies future setup of the sync update capability between WSRR and DataPower for policy updates.

About this task

The DataPower certificates as part of the patterns used by the curl tool. DataPower Calls are uploaded into the Node or Cell Default Truststore. This simplifies setting up future uses of the sync update capability between WSRR and DataPower for policy updates. If this capability is not needed, this procedure describes how to

remove DataPower Certificates. This procedure also describes how to add new DataPower Certificates if the certificates need to be changed.

Procedure

1. Log in to the Dmgr or Stand Alone WSRR at `http://hostname:9060/admin`. Enter the user and password.
2. Navigate to **Security, SSL certficates and key management**.
3. Click **Key Stores and Certificates**.
4. Click **NodeDefaultTrustStore** if you chose a basic pattern, or **CellDefaultTruststore** if you chose an advanced pattern.
5. Click **Signer Certificates**.
6. Tick the check boxes of any certificates you want to remove.
7. Click **Delete**.
8. Click **Save**.
9. Optional: if you need to add new DataPower Certificates, click **Add** to add the new certificate.

Configuring the Policy Enforcement Point

The DataPower appliance 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

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 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 you must specify for the AAA action include the Input and Output, as well as the AAA Policy. You can create the policy whilst in the process of creating the AAA Policy Action, or you might have created it prior to this using the AAA editor.

   - Identification is the step where the user is Identified. In our sample, there were two forms of identification used. In the StoreAddLTPA XML firewall, the identification was performed with basic authentication. In the StoreWSP firewall, identification was provided by LTPA token.
   - Authentication is the step where the user is proven to be a user who is known to the system. There are many options to choose from. In the sample, we showed you 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 need to 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 81.
  - 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 wish 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` stylesheet, 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.

What to do next

In addition to creating a Domain and setting up a WSRR Server Configuration in the SOA Policy Gateway Advanced Runtime and SOA Policy Gateway Basic Runtime patterns, it is possible to extend the domain by running a custom CLI script. The CLI script should be in the root of the `DomainZipFile.zip` structure; for example, `/cli.cli`. The CLI can run any standard CLI commands, but all artifacts that the CLI refers to must exist or be accessible by the DataPower Domain created by the pattern. When you deploy an instance of the SOA Policy Gateway Advanced Runtime or SOA Policy Gateway Basic Runtime patterns you will be prompted for the CLI file name in the Security package parameters.

Working with the SOA Policy Gateway Basic Runtime pattern

TheSOA Policy Gateway Basic Runtime pattern consists of three major pieces of functionality; the files needed for security between the DataPower and WSRR pattern scripts are retrieved, a domain is configured on DataPower, and finally promotion is configured.

When completed the following actions will have occurred:

1. The new domain exists on the DataPower appliance specified.
2. A WSRR Server Definition exists in the domain.
3. The custom CLI script has been run against the DataPower domain.
4. A WSRR Server is configured.

5. Any DataPower signer certificates provided by the customer have been uploaded to the NodeDefaultTruststore of the WSRR cell.
6. Promotion between the SOA Policy Gateway Basic Runtime pattern WSRR cell and the SOA Policy Gateway Governance Master cell has been configured.
7. Signer Certificates have been exchanged. The Signer Certificate of the Governance Dmgr is placed in the NodeDefaultTrustStore of the Basic cell, and the Signer Certificate of the Basic cell Dmgr is placed in the CellDefaultTrustStore of the Governance cell.
8. LTPA Keys have been exchanged. The LTPA Key of the Governance cell is imported into the Basic cell.
9. Each host of the Governance Master WSRR cluster is added to the trusted realms of the Basic cell. Each host of the Basic cell WSRR cluster is added to the trusted realms of the Governance Master.
10. The promotion properties file is configured if the cell was designated as either staging or production environment in the given inputs.

Whilst you will need to take other steps to complete a fully secure production environment, the configuration performed at this moment allows you to do the following:

1. Create services and policies, and govern them through the SOA Policy lifecycle on WSRR (when staging and production environments have been provided), using the default GEP.
2. Create Web Service Proxies that can use the pre-created WSRR Server definition to build subscriptions.

Working with the SOA Policy Gateway Advanced Runtime pattern

The SOA Policy Gateway Advanced Runtime pattern consists of three major pieces of functionality; the files needed for security between the DataPower and WSRR pattern scripts are retrieved, a domain is configured on DataPower, and finally promotion is configured.

When completed the following actions will have occurred:

1. A new domain exists on the DataPower appliance specified.
2. A WSRR Server Definition exists in the domain.
3. The custom CLI script has been run against the DataPower domain.
4. A WSRR Clustered environment with 'n' nodes will have been created and configured.
5. Any DataPower signer certificates provided by the customer will have been uploaded to the CellDefaultTruststore of the WSRR cell.
6. Promotion between the SOA Policy Gateway Advanced Runtime pattern WSRR cell and the SOA Policy Gateway Governance Master cell has been configured:
   a. Signer Certificates have been exchanged. The Signer Certificate of the Governance Dmgr is placed in the CellDefaultTrustStore of the Advanced cell, and the Signer Certificate of the Advanced cell Dmgr is placed in the CellDefaultTrustStore of the Governance cell.
   b. LTPA Keys will have been exchanged. The LTPA Key of the Governance cell is imported into the Advanced cell.
   c. Each host of the Governance Master WSRR cluster is added to the trusted realms of the Advanced cell. Each host of the Advanced cell WSRR cluster is added to the trusted realms of the Governance Master.

d. The promotion properties file is configured if the cell was designated as either staging or production environment in the given inputs.

The current configuration enables you to do the following:

1. Create services and policies, and govern them through the SOA policy lifecycle on WSRR (when staging and production environments have been provided), using the default Governance Enablement Profile (GEP).
2. Create Web Service Proxies that can use the pre-created WSRR Server definition to build subscriptions.

Next, you must take additional steps to complete a fully secure production environment. For more information, see "Security for the IBM SOA Policy Gateway Pattern patterns" on page 53.

DataPower objects created in the Basic Runtime and Advanced Runtime patterns

An overview of the DataPower objects created in the SOA Policy Gateway Basic Runtime and SOA Policy Gateway Advanced Runtime patterns and their function.

Table 37. DataPower pattern objects

Object	Description
Domain	A Domain which can be used for the users application.
WSRR Server	Named WSRRSVR. The SOAP URL, User, 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 conttains a validation credentials object WSRRVC, that contains the Signer Certificate that was uploaded as part of the pattern scripts.
Validation Credentials	WSRR Validation Credentials contains the Crypto Certificate WSRRCERT. All other settings are default.
Crypto Certificate	WSRRCERT utilizes 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.

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 has not been created, see “Configuring Business Space for the first use” on page 93 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 92
Use the Business Space user interface to administer policies.

Policies

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

Policies in WSRR

WSRR can be used to author all of the SOA policies, including SLA (Service Level Agreement) policies, mediation policies, monitoring policies, custom policies, and other policy domains that shall be supported in the future. 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 which 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 adding 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 policies” on page 105.

Mediation policy assertions

Service Level Agreement’s (SLAs) should originate from a requirement of the business that the quality of service provided by a service must meet a specified standard. As a service is being designed, functional requirements are created to guide the logic of what the service does. Non-functional requirements should be specified in parallel as part of the analysis and design of that service to designate the quality of service that 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 in order to meet the business non-functional requirements.

We can write a policy that implements runtime checks on the performance of the service and take action when the SLA is not being met so as to guarantee that the service meets its SLA. For example, we might have a service primary endpoint that

is normally able to (95% of the time) provide service response within 2 seconds. The SOA architect has created a secondary endpoint on another server that is normally 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. We 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 a variety of consumers, each with a different level of priority. A simple example might have "gold" and "bronze" customers, where we only guarantee a specific quality of service for our "gold" customers. In this example, we can check if the consumer is "gold" and reroute to our secondary endpoint, leaving the "bronze" customer to deal with a slower response time. The business made the decision 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, we might have a situation where a service will do the best it can, but when it determines that it is under load, it will queue or even reject messages from low priority consumer services. One example of this is when a batch routine floods the system with consumer requests at an unexpected time. In order to protect the quality of service, we can create a runtime policy that is in effect during business hours only, and that will reject all batch requests during this period.

More generically, mediation policy allows for validation and transformation on the incoming message from the client (consumer) prior to 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 which only applies to consumers for which no other policies apply. Using this feature allows policies to be specified for rogue consumers that don't 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 evaluating to True and any actions 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 specified are evaluated by the local Policy Enforcement Point and the time zone 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 the date at which the schedule becomes effective in xs:date format. StartDate is inclusive and if this attribute is not present, the schedule becomes effective immediately today.

  Note: Click the xs:date hyperlink to understand this industry standard.
- **StopDate** - This optional attribute specifies the date at which the schedule stops being effective in xs:date format. StopDate is exclusive and the specified date should 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 the time at which the schedule starts daily in xs:time format. (Note: click on the xs:time hyperlink to understand this industry standard).
  - **StopTime** - If Daily is specified, then this attribute is required. It specifies the time at which the schedule stops daily in xs:time format. 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 only affects the start of the daily timeframe as schedules are allowed to run passed midnight. For example, if a schedule is set to start at 11 p.m. and run for 2 hours on Wednesdays, the schedule will effectively end 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 a required three parameters consisting of Attribute, Operator, and Value, plus an optional Interval and Limit. 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 may be specified are as follows:

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

Table 38. Defined attributes

Attribute	Description and Type
ErrorCount	The number of faults observed during this monitoring interval.
MessageCount	The number of actual messages 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 39. 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.
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 the time interval, used as a sliding window, to measure the wsme:Attribute when evaluating the expression, in xs:duration format. If not specified, the interval used is 60 seconds. If specified, a reasonable value should 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.

Note: 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 this defines the low threshold while wsme:Value defines the high threshold. The specified threshold should be lower than that of wsme:Value. When not specified the default Limit is 0.

  When wsme:Operator is TokenBucket this 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 will be rejected by the Policy Authoring Point. The mediation policy actions allowed are:

- **QueueMessage** – This action specifies that transactions will be queued when the logical condition is met. Message processing will not re-commence 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 should be the first action.
- **RejectMessage** – This action specifies that transactions will be rejected when the logical condition is met. Transactions will continue to be rejected until the logical condition is no longer met. When transactions are rejected, a SOAP fault will be returned to the client (consumer) service. When several actions are specified within a single Action element, RejectMessage should be the first action. QueueMessage and RejectMessage are mutually exclusive.
- **Notify** - This optional element specifies that a notification will be produced when the logical condition is met. For WebSphere DataPower, a message will be written to the DataPower system log.
- **RouteMessage** - This optional element specifies that messages will be routed to specified endpoint destination when the logical condition is met. Messages will 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 shall be validated against the specified grammars. Messages shall be rejected when validation fails. Either XSD or WSDL must be specified as a sub-parameter if ValidateMessage is specified. SCOPE is optional, and if not specified, SOAPBody is used for the validation.
  - **XSD** - Specifies that messages will be validated against the XML schema identified by the URI it contains.
  - **WSDL** - Specifies that messages will be validated against the Web services description (WSDL) identified by the URI it contains.
  - **SCOPE** – Specifies what part of the message will be validated. The following table lists the possible values and what they mean:

Table 40. 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 will be performed with the specified Stylesheet and Parameters. Transactions will be rejected when the execution fails. Stylesheet information must be specified, while Parameters are optional, and should be specified as needed by the particular stylesheet specified.
  - **Stylesheet** - Specifies the transform operation will use the stylesheet specified by the contained URI. The stylesheet MUST be an XSLT file.
  - **Parameter** - This optional, repeating element specifies a stylesheet 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 policies

When authoring mediation policies in the Business Space user interface, 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 92.

The SOA Governance space must be created before policies can be created. If the SOA Governance space has not been created, see "Configuring Business Space for the first use" on page 93 and follow the steps to create the space.

About this task

Author new policies 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 on 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 100 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 just created, select the policy document in the Service Registry Navigator Widget in the lower-left of the screen. Alternatively, search for the name you specified, including `.xml` on the end. The policy document is displayed in the Service Registry Detail widget on the right.

Related concepts:

“Policies” on page 100
Implementation details for using WSRR as the Policy Authoring Point and WebSphere DataPower as the Policy Enforcement Point when creating mediation policies.

Related information:

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

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 93.

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” on page 107.
   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 using the Business Space user interface.

About this task

For more information about governance, see "The SOA Policy lifecycle" on page 4.

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 desired lifecycle state:

1. In Business Space, open the policy document for the policy by selecting 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. 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 4
Mediation policies are governed using the SOA Policy lifecycle. This takes the policy from being initially identified, through to being deployed in production, and, finally, 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 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 you may 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 when deploying the patterns in the IBM SOA Policy Gateway Pattern.

Failure to connect to DataPower during deployment

Try the following solutions:

- Check with the DataPower Administrator that the user and password are valid:
  - In DataPower, validate 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 admin.
  - The DataPower Administrator might need to check if 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.
- Review the SSL Connection Failure steps below to validate that the Certificates are correctly installed both in the `DomainZipFile.zip` and on the DataPower appliance.

Troubleshooting failure of Mutual Authentication client authentication

Try the following solutions:

- Check that the correct certificates were in the `DomainZipFile.zip`.
- Check that the Crytpo Profile on the XML Management Interface Port has Validation Credentials with all the certificates in the Chain.
- Check that the passwords for the Client Public Key and Client Public Certificate are correct.

Troubleshooting failure of server authentication

Try the following solutions:

- Check that all of the certificates in the chain are present in the *`yourDataPowerHostName`* directory of the `DominZipFile.zip` file you are using.
- Check that the SSL Proxy Profile has a reverse crypto profile that contains the Identification Credentials with the Certificate Chain.

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:

- Sign into 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 the failure to connect to an SCP

Try the following solutions:

- Check that the SCP Host name is correct.
- Check that the SCP user is correct.
- Check that the SCP password is correct.
- Manually test the SCP from a Node in the IBM Workload Deployer or IBM PureApplication System environment with the supplied information.

Troubleshooting the failure to retrieve the `DomainZipFile.zip` file from SCP or debug missing artifacts

Try the following solutions:

- Check that the `DomainZipFile.zip` exists in the URI.
- Check that the file mentioned in the log failure exists in the correct location in the `DomainZipFile.zip` file. In particular, ensure that the certificates required are located in the correct directory.

Troubleshooting promotion failure

There are many problems that can arise in a 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, SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced Runtime, exist.

  - Go to each Runtime Environment, SOA Policy Gateway Basic Runtime or SOA Policy Gateway Advanced Runtime, 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 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 prior to any additional runs of the CLI.

Troubleshooting SSL failures due to missing DataPower certificates

If the correct hostname for your DataPower Certificates directory was not provided in the `DomainZipFile.zip` file, the script packages will fail to connect to the WSRR Server if Mutual or Server Authentication is enabled on the DataPower host.

Troubleshooting WSRR/DataPower connection issues

If you see that the status of the WSDL in a Web Service Proxy is in Down or Synchronizing state that never changes to Okay, check the following:

1. Check that the Crypto Certificate is valid for the WSRR Server (WSRRSVR).
2. Check that DataPower has the correct DNS set up to recognize the Hostname of the WSRR Server or Dmgr.
3. If the DNS is incorrect, a temporary work around is to change the URL in the WSRR Server definition to point directly to the IP by substituting the IP for the HostName in the URL.
4. Go to the WSRR Subscription and do a manual synchronize:
   a. Check the `default.log` for errors related to the connectivity of the WSRR Server.
5. Make sure that the certificates required match those in the Identification Credentials for the Crypto Profile of the DataPower Appliances XMLManagement Interface SSL Proxy Profile.

Troubleshooting problems in the deployed instance

You can troubleshoot common problems in the deployed instance.

Failure to connect to LDAP

To diagnose LDAP Failures in the sample, try the following solutions:

- In DataPower Control Panel Troubleshooting, ensure that the trace is in debug mode.
- Go to StoreAddLTPA, open the Probe details and enable the probe.
- Run a client test.
- View the logs in the probe. Look for LDAP Bind failure messages.
- Check the LDAP Host Name.
- Check the LDAP DN; for example, cn=root,dc=ibm.com.
- Check the LDAP password; for example, passw0rd.
- Check that the LDAP port is 389 and non-secure.
- Check that the entry passwords for ConsumerX, ConsumerA, ConsumerB are all passw0rd. Make sure that the LDIF file import has transcribed the correct passwords.

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 Admin 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.

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 package have errors, click the log links for **remote_std_out.log** and **remote_std_err.log** next to the script package names.
b. Log into the WSRR instance and check the server errors.
c. Refer to the WSRR troubleshooting guides: http://pic.dhe.ibm.com/infocenter/sr/v8r0/topic/com.ibm.sr.doc/cwsr_troubleshootingandsupport.html

3. For DataPower:
   a. Retrieve the **default.log** file for the domain created by the pattern.
   b. Retrieve the **default.log** file for the default domain.

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

Procedure

Complete the following steps to apply an interim fix.

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

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

Chapter 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.