SOA governance using WebSphere DataPower and WebSphere Service Registry and Repository, Part 3: Enforcing Service Level Agreements in Web Services

Leveraging new capabilities in WSRR V8.0 and DataPower V5.0 to model and enforce SLAs within the enterprise

Mario E. De Armas (drhoek@us.ibm.com)
Senior Technical Staff Member
IBM

Steve Groeger (groeges@uk.ibm.com)
Senior Systems Engineer
IBM

Phil Schaadt (pschaadt@haddonhillgroup.com)
Chief Technical Officer
Haddon Hill Group Inc.

Richard Duran (rduran@haddonhillgroup.com)
Principal Consultant
Haddon Hill Group Inc.

Deborah Zippel (dzippel@haddonhillgroup.com)
Lead WebSphere Analyst and Business Process Consultant
Haddon Hill Group Inc.

Starting with the releases of IBM® WebSphere® DataPower firmware V5.0 and WebSphere Service Registry and Repository V8.0, IT service developers and policy administrators can now attach and affect Enterprise Service Bus (ESB) behavior related to Service Level Management routing. Part 3 of this article series describes how to use the new WS-MediationPolicy capabilities in WSRR to author specific Service Level Agreements, which you can enforce on the DataPower appliance without manually creating the XSLT or processing rules and actions.

View more content in this series
Introduction

This article is the third in a series dedicated to discussing the new integration capabilities in WebSphere Service Registry and Repository (WSRR) V8.0 and WebSphere DataPower V5.0.

Part 1 describes the WS-MediationPolicy capabilities from the point of view of the DataPower SOA appliance (hereafter called the "appliance"), including sample XML, tips on creating your own policy using either WSRR or IBM Integration Designer, as well as how to attach it to a Web service proxied by the appliance.

Part 2 explains how to extend these capabilities even further by enabling the use of custom policy vocabularies to deploy specific proxy processing patterns not covered by the built-in policy domains. The first two articles were written with the assumption that you are familiar with XSLT, XML, WS-Policy, and DataPower configuration.

This article will demonstrate an example that uses the new WS-MediationPolicy capabilities in WSRR to author specific Service Level Agreements (SLAs), which you can enforce on the DataPower appliance without a manual creation of the XSLT or processing rules and actions.

Business value

This section describes how SOA Policy Gateway pattern implementation with WSRR and DataPower can give IT organizations better capability for SLA management by defining and managing Quality of Service (QoS) policies. Using the two products together, the IT organization can offer and enforce different SLA and QoS behaviors for different consumers of the same service, add new behaviors to existing services, or change the parameters of existing SLAs. Therefore, the enterprise can gain business value by increasing business agility and saving money on the IT infrastructure.

Consumer to provider SLAs

Many sophisticated IT organizations today cannot proactively protect, control, and monitor their services using SLAs because they still lack the capability to model, enforce, and report on quantified SLA parameters. This adversely affects the availability of business services.

Service Level Agreements (SLAs) are contracts that formally and quantifiably define the qualities of service. At the enterprise level, the SLA is usually a negotiated agreement between two parties, where one is the service consumer and the other is the service provider.

The formal SLA records a definitive understanding about services, priorities, responsibilities, guarantees, and warranties, and may specify quantified levels of availability, serviceability, performance, operation, or other attributes of the service.

SLA Management

SLA Management as an enterprise process includes:

- SLA contract definition (defining the relevant quality of service parameters) between the service consumer and service provider
• SLA negotiation (specifying the specific values for each of the quality of service parameters)
• Recording the quantified agreement as to services, priorities, responsibilities, guarantees, and warranties
• Specifying (usually) levels of availability, serviceability, performance, operational metrics, and other attributes of the service
• SLA enforcement in the runtime according to defined policies quantified by the service parameters above
• SLA monitoring and automated action when the agreed to parameters are violated

The SOA Policy Gateway pattern provides this capability to IT organizations. When implemented with the combination of WSRR as the Policy Administration and Authoring Point (PAP), and DataPower appliances as a Policy Enforcement Point (PEP), the IT organization has a complete design-time as well as run-time environment to define and manage Quality of Service (QoS) policies.

Using these two products together, the enterprise can save money on IT infrastructure and increase its business agility by offering and enforcing different SLA and QoS behaviors for different consumers of the same service, as well as add new behaviors to existing services or change the parameters of existing SLAs.

Therefore, the IT organization can provide business value through this increased agility within the IT infrastructure while ensuring consistent control and execution of services on the Web Service proxy within the DataPower appliance, including the ability to:

• Eliminate "rogue" services and rogue service consumers, thereby controlling and monitoring the use of all enterprise policies to only specific consumers and providers.
• Route traffic from specific consumers to a specific version of the service within the runtime, allowing for "permissive versioning" within the runtime, as well as time-of-day routing, thus ensuring automatic and specific availability schedules for each service.
• Monitor and enforce quantified SLAs for different service consumers and easily implement automatic changes in runtime behavior for situations where it makes sense to throttle, queue, or re-route transactions to meet a specific SLA, security requirements, or channel specific QoS.
• Report on service errors and take automated action to recover from service errors.
• Change policies in one place, enabling changes to be implemented and deployed more quickly by IT at lower cost with improved time to value.
• Reuse common Services policies across multiple services, decreasing cost and improving consistency and reliability.
• Audit and trace a specific policy enforcement to a specific consumer and a specific service invocation.

Implementing a scenario for a Service Policy Gateway

This article will guide you through the steps to implement a scenario for a Service Policy Gateway by defining a services policy within WSRR and then attaching these policies to either a Service Level Definition (SLD) or an SLA for subsequent enforcement using DataPower. The article also shows an example of testing and troubleshooting the scenario.
We will cover implementation considerations, configuration steps, and best practices. All configuration examples and integration functionality discussed in the article are available from the Download section of the article.

The "WSRR to Model Services and Policies The Governance Enablement Profile" API, which ships with the WSRR product, provides a robust and extensible object model to model all aspects of your services. The WSRR APIs allow for automated subscriptions from DataPower to instances of the WSRR service model.

**Modeling services**

WSRR objects, along with the properties ("metadata") recorded in the objects, create complete design-time time and run-time models of the services and applications in your business. Table 1 shows the main object types that are available.

**Table 1. Main object types in the Governance Enablement Profile (GEP)**

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>In any governance process, there must be ownership. The organization object embodies this ownership. Every other object must be owned by an organization, either directly or indirectly.</td>
</tr>
<tr>
<td>Business Capability</td>
<td>A business capability object expresses a business view of a capability that is required within your enterprise. The object describes the business needs and concerns, but not the implementation. The capability can be viewed from a generalized business perspective. There are a number of types of business capability. For example, a capability might be used with other capabilities to build a wider capability. In this case, the capability is considered and represented as a business service. Business service, business application, and business process are specific types of business capability.</td>
</tr>
<tr>
<td>Capability Version</td>
<td>A capability version is the realization of a business capability. The capability version indicates how the business capability is provided, for example, whether it is a Web service, whether it requires messaging, and so on. Aspects of the implementation, such as web service definitions or SCA modules, are referenced from the capability version. In the same way that business capabilities have specific types, so do capability versions. For example, capabilities that are built from other capabilities, or are used to build other capabilities, can be considered as service versions. Service versions, application versions, and process versions are all specific types of capability version. One of the key aspects of capability version objects is that they have version numbers. There might be numerous versions of a capability version that realize the same business capability.</td>
</tr>
<tr>
<td>Service level Definition</td>
<td>The service level definition (SLD) describes non-functional, or quality-of-service, characteristics of a capability. These include characteristics such as how many hours a day the capability is available, or the types of security needed for interactions with the capability. Consider a capability version deployed onto a number of different servers, servicing different types of customer, or different geographies. Each server might not be able to provide the same qualities of service and require unique SLDs. However, each consumer interacts with the capability in the same way.</td>
</tr>
<tr>
<td>Service Level Agreement</td>
<td>If a Capability version is offered for reuse in your enterprise, the SLA must be created to describe the details of the agreement between the capability consumer and provider. The SLA is usually a subset of...</td>
</tr>
</tbody>
</table>
the qualities of service offered by the SLD of the providing capability version. For example, the SLD might state that the service is available from Monday to Friday inclusive, but the SLA might define that the service is only required on Wednesday.

**Service Endpoint**

When a capability version is deployed on a runtime system, it has one or more network-addressable endpoints. These endpoints are represented by service endpoints. Service endpoint objects inform the user of the service where the service is available on the network.

**Policy Attachment**

Policies specify the requirements that must be met so that a service can be consumed by a client. For example, a Web service may require that all messages are digitally signed and encrypted; or that it is only available at certain times; or that it can only handle a certain number of messages. The Policy Object includes the Policy Document, the Policy Expression, and the Policy Attachment.

The example shown in Figure 1 illustrates how the WSRR objects relate to one another. A Credit Check business service is owned by the Common Services organization. This business service has a real implementation in the form of the Credit Check service version, which has been deployed into three geographical locations. Each geographical location hosts the service on a different server and so there is a different SLD for each installation. Each SLD can offer a distinct set of service endpoints for the service.

**Figure 1. Example of WSRR relationships**

![Diagram illustrating WSRR relationships]

**Modeling consumers**

After a service has been implemented, it can be offered for reuse. Reuse requires an SLA, and introduces the concept of service providers and service consumers.

The Commercial Division is defining and constructing their own capability called Eligibility Service. They decide to use the Credit Check capability from the Common Services organization as part of their own service. The Commercial Division is based in the United Kingdom (UK). They negotiate
the conditions of the service interaction and capture the conditions in an SLA named "SLA – Credit Check".

Figure 2 shows the capability version, Credit Check, in a consumer and provider relationship with another capability version, Eligibility Service. Figure 1 shows the final structure of the main objects in this governance example and shows the relationship between the eligibility service and the credit check service.

**Figure 2. Service consumer and provider relationship**

Lifecycles states
When certain WSRR objects are created, they are automatically placed into a lifecycle. There are three object types in WSRR that need to be in a specific lifecycle state for any policies attached to them to be determined to be valid, and hence, to be enforced by DataPower. Table 2 shows these WSRR objects, the lifecycle they use, and the actual lifecycle state that they need to be in for DataPower to enforce the policies attached to the WSRR objects.

**Table 2. WSRR lifecycle states for WSRR objects**

<table>
<thead>
<tr>
<th>WSRR object</th>
<th>WSRR lifecycle</th>
<th>Lifecycle state required for Policy enforcement by DataPower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Level Agreement</td>
<td>SLA Lifecycle</td>
<td>SLA Active</td>
</tr>
<tr>
<td>Service Level Definition</td>
<td>SLD Lifecycle</td>
<td>SLD Subscribable</td>
</tr>
<tr>
<td>Policy</td>
<td>SOA Policy Lifecycle</td>
<td>Approved</td>
</tr>
</tbody>
</table>

SLAs and SLDs
In order for DataPower to know which consumer's SLA to use for an incoming message, messages must contain a Consumer ID and a Context ID. The Consumer ID is for a given consumer. For example, a company (or a department, if it is an internal consumer), should have a unique identifier that can always be correlated back to a modeled consumer in the WSRR
registry. Depending on who or what is the consumer of the service, the Context ID allows different messages from each consumer to be given different levels of service. As a real-world example: using a travel reservation service, an airline, as the consumer, can qualify a request being made by a frequent flier. This flier is regarded as a gold customer and given a better level of service than normal customers.

The provider SLD describes how a service should be used. It references, for a Web service, the WSDL port and binding. It also describes any policies that apply to the service for all consumers.

For use with consumer SLA enforcement, the provider SLD is where the location of the Consumer ID and Context ID values are specified. The Context ID Location Information on the SLA and SLD and the Consumer ID Location Information properties on the SLD are required to be in a specific format. The format of these properties is: `<type> (<value>)`, where `<type>` is defined as shown in Table 3.

### Table 3. Values for Consumer ID and Context ID location information

<table>
<thead>
<tr>
<th>Type</th>
<th>Short Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>http://www.w3.org/TR/1999/REC-xpath-19991116</code></td>
<td>xpath10</td>
<td>The Context ID or Consumer ID is found in the XML message content using the <code>&lt;value&gt;</code>, which is defined in XPath 1.0 expression format. This is the default value.</td>
</tr>
<tr>
<td><code>http://www.ietf.org/rfc/rfc2616</code></td>
<td>http11header</td>
<td>The Context ID or Consumer ID is found as a HTTP Header definition by using the <code>&lt;value&gt;</code>, which is defined in the HTTP/1.1 Header Field Definition format.</td>
</tr>
</tbody>
</table>

1. To set these locations, ensure you are logged into the **WSRR Business Space UI** and on the **Operation space**. From the Watch List, navigate to the **details view of the SLD** and click the **pencil icon** to edit. Expand **Additional Properties** to see these properties. Enter the Context ID value as shown in Table 3 and Figure 3.
Figure 3. SLD Details view showing Location Information attributes

2. The Context Identifier is also shown as the first field under the Description of the SLA (see Figure 4). To set the value, navigate to the Details view of the SLA and click the pencil icon to edit.
3. The Consumer Identifier is also shown on the **Capability Version**, immediately below the **Version Number** (see Figure 5). To set the value, navigate to the **Details** view of the **Capability Version** and click the **pencil icon** to edit.
Figure 5. Capability Version Details view showing the Consumer Identifier attribute

Attaching policies (mediation policies)

For a policy to be consumed by DataPower, it must be attached in WSRR to the SLA or SLD. This section shows how to attach completed policies to SLAs or SLDs within WSRR.

Using the Business Space UI widget

Part 1 in this series showed how to use the WSRR Business Space UI to author a WS-Mediation policy. The completed policy is shown in Figure 6.
Figure 6. A completed WS-Mediation policy

Now that you have a completed policy, you need to attach this policy to an SLA or SLD within WSRR so that this policy can later be enforced by the DataPower appliance. In WSRR, there are two ways to attach a policy to an object:

- The first is from the policy itself.
- The second is from a specific item.

We will look at each method separately, starting with attaching from the policy.

Policies may be attached to SLAs in the same manner as to SLDs, so the "before" process will only be shown for the SLDs. When a policy is attached to an SLD or an SLA, it will show in the Service Registry Navigator on the Browse tab, as shown in Figure 7.
Once policies are attached to SLDs and the SLAs that consume the services they represent, they can be viewed from the **Graphical Explorer** as shown in Figure 8.

**Figure 8. Service Registry Graphical Explorer**

In Figure 9, the Graphical Explorer view shows the interactions of the Capability Version, SLD, and SLA with the associated policies.
Attaching to items from a policy expression

1. To attach the policy after it has been saved, click on the **Actions** drop down in the top left corner of the **detail widget** for the **Policy Expression**, then select the **Manage Policy Attachments** item from the drop down. When you click on the item, a pop-up displays as shown in Figure 10, which is used to attach this policy to various objects within WSRR.

**Figure 10. The Manage Policy attachments pop-up**

The "Manage Policy Attachments" pop-up shows any objects to which this policy is currently attached. Figure 10 shows no objects. The dialog only shows the link that can attach the
policy to specific items (autosuggestions) and another link that uses a query to attach the policy to multiple items.

2. Select the **Attach to Specific Items** link to select the specific items to be attached (Figure 11).

**Figure 11. Attach to Specific Items dialog**

![Attach to Specific Items dialog](image1)

3. From the **Specific Items** dialog, you can select the type of object within WSRR to which you are looking to attach the policy. You can also type part of the name of the object, including wild cards, to limit the search. If the auto-suggest capability has been enabled within WSRR when you type into the name field, some suggestions matching your selection are displayed, as shown in Figure 12, one of which you can select.

**Figure 12. Attach to Specific Items dialog with autosuggestions**

![Attach to Specific Items dialog with autosuggestions](image2)

4. If you want to attach to more than one object of the specified type, click the **Find** button and a dialog is displayed showing all the matching objects within WSRR, as shown in Figure 13. You can then select multiple items by selecting the check box beside each item and then clicking the **Finish** button.
5. To select the specific items to attach, go to the Manage Policy Attachments dialog, select the Attach to Items Using a Query link, which shows a dialog as shown in Figure 14.

**Figure 14. Policy Attachment Query Settings dialog**

6. On this dialog, you need to enter a Name for the attachment query and an optional description. Then click the Add Query link to define the query to use. See Figure 15.

**Figure 15. Policy Query Settings dialog**

7. Use the Query Type drop-down to select whether the query you will be using for the attachment will be a "Type Query" (which uses a simple wizard to help you generate the...
query), or a "Custom Query" (which requires you to type a complete XPath expression as the query). The use of the "Custom Query" is not recommended unless you find that you cannot create the results required with the "Type Query" wizard. The use of the "Custom Query" to search the model requires that you have a reasonable knowledge of XPath and WSRR's model and its usage of XPath.

8. Select the **Type Query** option and then select the **base object type** to which you want the policy attached by using the Type drop-down. This drop-down lists all the base object types that a policy can be attached to in WSRR. Once the type has been selected (see Figure 16), click the **Add** button.

**Figure 16. Policy Attachment Query Setting dialog with query specified**

9. Once you have added the initial query, you can include additional information to filter the items that the query will return. The additional parameters that you can filter on are Lifecycle states, Classifications, and Properties.

10. To add one of these filter types, click on the green plus icon next to the type you want to add (see Figure 16). Once selected, a pop-up appears showing the items that you can select. Figures 17, 18, and 19 show the different pop-ups when you select the different options.
11. Once you add the relevant additional filter information, you see a screen similar to the one shown in Figure 20. This shows that the query is for a Service Level Definition, with a lifecycle

**Figure 20. Policy Attachment Query Setting with completed query**

![Policy Attachment Query Settings](image1)

12. Add additional queries by selecting the **Add Query** link again. When you have added all queries, click the **OK** button.

13. The **Manage Policy Attachments** dialog now shows the attachment query that has just been created, as shown in Figure 21. If an attachment is made by using the **Specific Item**, the **Manage Policy Attachments** dialog shows an attachment to the specific item, as shown in Figure 22.

**Figure 21. Manage Policy Attachments with Query**

![Manage Policy Attachments](image2)
Figure 22. Manage Policy Attachments with a Specific Item

At this stage, no actual attachments have been made, so if the Cancel button is clicked, any new attachments will not be saved.

14. When you have created all the attachments required for this policy, click the Finish button to save them.

15. From the Manage Policy Attachments dialog, you can also detach the policy from an object it is attached to. This dialog displays all the objects (or queries) that the policy is attached to.

16. Select the red cross to the right of the item to remove the item from the list, and click the Finish button to detach the policy from that item.

17. After you click the Finish button and the attachments or detachments are performed, the Detail View for the policy is displayed again. In Figure 23, the "Attached To" section shows all the objects and queries that are attached to the policy.
18. If a policy was attached using a query, the "Attached To" section shows the name of the query that was used, but it does not show you the names of the objects that this policy is attached to. To see these, click on the link for the query, such as **All Subscribable Development SLDs belonging to IBM namespace**. This takes you to the detail view for the attachment query, as shown in Figure 24.

**Figure 23. Policy Expression details view showing items attached**

![Policy Expression details view](image)
**Figure 24. Policy Query Attachment Details view**

![Policy Attachment Details View](image)

**Attaching a policy from a specific item**

The second way to attach a policy to an item is from the specific item:

1. Navigate to the detail view of the object to which you wish to attach the policy. In this case, you are going to select an **Extended SLD** representing Version 1.0 of the Eligibility Service, which is called "SLD - Eligibility Service", as shown in Figure 25.
2. In order to attach a policy to this object, you need to edit the object. To do this, click the pencil icon in the top right corner. This then displays the edit panel for the object, as shown in Figure 26.
3. On this pop-up, there is the "Attached Policies" section, which shows all the policies that are currently attached to this object and the "Add Policy" link. Select the **Add Policy** link to display a new section on the pop-up, as shown in Figure 27.

**Figure 27. Add Policy section**

4. From this section, you can select the type of policy that you want to attach to this object. You can also type part of the name of the policy, including wild cards (asterisk *), to direct the search. If the autosuggest capability has been enabled within WSRR, when you type into the name field, some suggestions matching your selection are displayed, as shown in Figure 28, one of which you can select.
Figure 28. Policy Selection with autosuggestions

5. When you select a policy using the suggestions, it is displayed as "attached" to the object (see Figure 29). The policy is not actually attached until you click the Finish button.
Figure 29. Policy Attachment from an item showing an attached policy

Viewing policies in the WSRR Business Space

The Service Registry Collection on the Browse tab, as shown in Figure 30, provides a convenient place to view all the policies as well as their states.
Figure 30. Policy Attachment Collection View

![Policy Attachment Collection View](image)

Importing policies

Policies may also be created outside WSRR and then imported, as described in Part 2 of this series. Use the Load Documents command from the Service Registry Actions widget on the Overview tab to search your network for the appropriate document, or to load it from a remote location using a URL. WSRR automatically detects the document type, or you may select it from the options shown in Figure 31.

Figure 31. Load Documents window showing document types

![Load Documents window](image)
Checking that objects are in the appropriate state

As stated before, the SLDs, SLAs, and the policies attached to them must be in certain lifecycle states to have the policies enforceable by DataPower. The lifecycle states are also known as governance states and are commonly shown together with the objects' names in the collection views in WSRR, such as the Watch List on the Overview tab of the Business Space UI (Figure 32). SLDs must be in the Subscribable state, SLAs in the Active state, and Policies in the Approved state to be enforceable by DataPower.

Figure 32. Watch List – Service Registry Collection view showing objects and governance states

Configuring DataPower to enforce policies

This section shows how these policies, which are managed from within WSRR, can be consumed and enforced by a DataPower Web Service Proxy.

Creating a Web Service Proxy

1. First, create a Web Service proxy and then a subscription to WSRR (Figure 33).
2. Create a Web Service Proxy by clicking on the **Web Service Proxy** icon in the top left of the control panel of DataPower XI52, as shown in Figure 34.

**Figure 34. Configure Web Service Proxy window**
3. Once you have created and named the new Web Service Proxy, click on the Add WSRR Subscription button (on the right) to configure it.
To keep this demonstration short, we have not shown the additional security configuration steps that would be required if, as would be the normal practice, the communication was over HTTPS. See the Resources section for the configuration of security on the DataPower appliance.

**Figure 35. Configure Web Service Proxy window with new WSRR description parameters**

4. The WSRR subscription needs a name and the name and namespace of the WSDL or concept to which it is subscribing, available from the object's details screen in either the WSRR Business Space UI or the WSRR Console view (see Figure 37). If a WSRR Server entry is not already available in list, you need to configure it on the DataPower appliance in this step. The WSRR Server is a reusable object, therefore a generic-style name is recommended. It provides a linkage to the WSRR instance for as many subscriptions as desired. Clicking on the plus sign next to the WSRR Server text box opens the dialog to configure this item, as shown in Figure 37.

**Figure 36. Schematic of DataPower WSRR subscriptions**
Figure 37. Configure WSRR Server pop-up

5. As shown in Figure 37, the WSRR Server needs the:
   - Name
   - URL of the WSRR instance to which you are subscribing
   - Valid user ID and password
   - Version number

The SSL Proxy Profile is only necessary if WebSphere Application Server security is enabled.

**Note**: We have not shown the additional security configuration steps that would be required if, as would be the normal practice, the communication was over HTTPS. See the Resources section for the configuration of security on the DataPower appliance.
6. If desired, the Synchronization Method may be set to:
   - **Manual**: The user must fetch any updates to the subscribed WSDL or concept.
   - **Poll**: DataPower fetches the updates to the subscribed object at a selected interval.
   - **Automatic**: WSRR 7.5 and later and DataPower 5.0.0 and later (or 5.0.5 and later for DataPower XI52 VE) automatically synchronizes with WSRR, receiving updates to the subscribed object as they happen.

Deciding the best synchronization method for your enterprise is determined by your IT Release and Configuration Management processes.

7. If **Poll Synchronization** is selected (Figure 39), you need to enter a refresh interval into the text box below the **Synchronization Method** box. Used only with the Poll method, the **refresh interval** is the interval in seconds between regularly scheduled WSRR queries used to synchronize the local copy of the subscribed resource with the WSRR-resident version. When you select **Manual** (Figure 40) or **Automatic Synchronization** (Figure 41), the refresh interval text box disappears.

**Figure 39. Synchronization method – Poll (with refresh interval)**
8. Using the version information to manage WSDLs in WSRR is recommended as a best practice. To enable this support, select the **Use WSDL Version** radio button and enter the **WSDL version value** of the desired WSDL in WSRR that you want to subscribe to (Figure 41).

**Figure 41. Synchronization method - Automatic**

9. Next, select a **Local Endpoint Handler** (Figure 42). Create one with the desired protocol and port settings, if one is not already available.

**Figure 42. Local Endpoint Handler**
10. You may want to change the URI to a recognizable name, rather than getting it from the WSDL. If so, uncheck the box next to From WSDL under URI and enter the desired name into the text box that will appear. Click the Add button when you are done to save your settings, and then click Next. The WSDL subscription is activated and retrieves the policies attached to the SLA and SLD (Figure 43).

**Figure 43. Local Endpoint handler showing successful WSDL activation**

![Local Endpoint handler showing successful WSDL activation](image)

11. Verify that the Endpoint Handler Summary shows as "up and configured" and that the WSDL Status shows as "Okay". Click on the SLA Policy Details button. You can see the view in Figure 44.

Figure 44 is referred to as the DataPower "Policy Model" view. This view provides the details on the policies consumed by the appliance.
12. Drill down into the EligibilityServiceProxy to show the subscriptions for the SLD, the SLA, and the Policy Documents attached:
   - The Policy Model button shows the preprocessed input data.
   - Drilling down on the eligibility service proxy to the attachments allows you to inspect the attachments referred to as "Policy Subjects".
   - The Reject Message button allows you to view policy assertions and consumer identification details.

The DataPower Rules view

The view shown in Figure 45 provides details on the SLM configuration that is generated by the appliance.
**Figure 45. DataPower Rules view**

<table>
<thead>
<tr>
<th>SLA Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy Node</td>
</tr>
</tbody>
</table>

This section lists the auto-generated DataPower rules for policy attachments.

- The **DataPower Rules** button allows you to view the post-processed, such as the output, data from the policy.
- Drilling down from the WSRR subscription allows you to inspect the attachment nodes, which are the “Policy Subjects”.
- Clicking on the **DataPower configuration artifact** graphics allows you to view the rules and the actions.

**Troubleshooting**

To set the level of logging or to access many other troubleshooting tools, click on the **Troubleshooting** link on the Control Panel of the DataPower, and select from the available options shown (see Figure 46). Logs may be set for alert, critical, debug, emergency, error, information, notice, or warning. If an intensive level of logging is enabled (anything other than notice), it may hinder DataPower’s performance.
Figure 46. Troubleshooting panel

In Figure 47, you can see an example of the DataPower log.
Testing the scenario

We will now test the scenario using the Curl utility along with a DataPower XML Firewall configured as a loopback. This simple approach to testing DataPower services is described more fully in the IBM WebSphere DataPower SOA Appliance Handbook.

Make sure you have access to the "Curl" utility, or a similar tool that can send SOAP and HTTP messages. Curl is a command line utility used for sending requests using HTTPS or HTTP (among other protocols). It is simple, yet powerful, since Curl sends exactly what you typed. Therefore, you can control the exact payload, headers, or other options without seeing unexpected behavior that might be caused by more complex and less transparent utilities.

First, test the EligibilityService proxy by sending a single message using the following Curl command. Note that this command executes the above mentioned Service, which was registered in WSRR using the specific consumer ACSV000 and context ACSSLA. The Curl command is simply:

curl -k -d@ValidateRequest.xml --header "ConsumerId: ACSV000" --header "ContextId: ACSSLA" http://<datapower-host>:6100/EligibilityService
If you refer back to Figure 44, the SLA for the specific consumer is a maximum of 10 messages per minute. Therefore, the first 10 times that you execute the command, you see the return as shown in Figure 48.

**Figure 48. Curl command window showing successful message**

If you continue to send Curl commands in rapid succession, when you reach the 11th message within one minute, the DataPower will reject the 11th message by invoking the rule from the Service Level Agreement. In Figure 49, you see that the DataPower returns a Fault and that the Fault Code is rejected.

**Figure 49. Curl command window showing rejected message**

Now for testing the scenario, there is no real backend server for DataPower to call. However, by using an XML Firewall in loopback configuration, there is no need for a backend server. This is because the XML Firewall service itself generates a response to the client after processing the request message. This is very useful in developing, testing, or debugging a service where the backend server is not available by using the loopback on DataPower to substitute for the backend server.

Figure 50 shows the configuration screen for the XML Firewall. You can see in the lower left that the firewall type is chosen from the drop-down to be "Loopback". In this case, we have assigned an unused port number 2054 to be used by the Web Service proxy.
Figure 50. DataPower Configure XML Firewall window showing loopback

Figure 51 shows the Web Service Proxy with the remote endpoint XML Loopback firewall at Port 2054.
Extending the scenario

This article provided one of many possible examples of using the Service Policy Gateway.

Part 4 (future article) in the series will provide a detailed example of how to create and enforce fully custom policies with a combination of WSRR and DataPower appliances, based on the information in a recently published IBM Redbook, SOA Policy, Service Gateway, and SLA Management. This IBM Redbook provides a comprehensive description of all the topics above, enabling you to create an enterprise-class automated and centralized policy management system, including how:

- Business users can use the SOA Policy Solution to help create the SLAs for their business services to deliver on promises for business performance.
- IT Architects can use the SOA Policy Solution to architect the policy solution patterns that standardize the runtime policy usage at their organizations.
- Developers can select specific policy patterns to implement the non-functional requirements that are associated with their projects.
- Operations groups can provide information about operating needs and create standardized monitoring policy for operational action at runtime.
The book also covers the following topics:

- Centralizing policy administration, enforcing, and monitoring of runtime policies for traffic management
- Enforcing SLAs and service mediation
- Creating customized policies
- An approach on reusing policies

Conclusion

This article has demonstrated one example of how the Service Policy Gateway pattern, combined with WSRR and the DataPower appliance, can leverage declarative policy to enable different SLA and QoS behaviors for different consumers of the same service, including how:

- These declarative policies are authored and governed from WSRR.
- The DataPower Web Service proxy can automatically subscribe to these policies.
- These policies provide business agility while maintaining operational control and stability.

Part 4 (future article) in the series will provide a detailed example of how to create and enforce fully custom policies with a combination of WSRR and DataPower appliances, based on the information in a recently published IBM Redbook, *SOA Policy, Service Gateway, and SLA Management*. 
## Downloads

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code samples</td>
<td>download_files.zip</td>
<td>478KB</td>
</tr>
</tbody>
</table>
Resources

- **Learn**
  - **Part 1: Leveraging WS-MediationPolicy capabilities** explains the new DataPower mediation policy (WS-MediationPolicy V1.6) capabilities and syntax. The article also demonstrates how to create, attach, and deploy them using WSRR and IBM Integration Developer environments.
  - **Part 2: Authoring and enforcing custom policy vocabularies** explains the use of custom policy vocabularies to deploy specific proxy processing patterns not covered by built-in policy domains.
  - For information about DataPower configuration and DataPower processing policy, see the WebSphere DataPower SOA Appliance Handbook or the WebSphere DataPower Information Center.
  - For more general treatment of seller Policy use cases and the use of WSRR and DataPower together to implement the SOA policy Gateway pattern, see the IBM Redbook: SOA Policy, Service Gateway, and SLA Management.
  - For an introduction to the Web Services Policy language, see Web Services Policy 1.5 – Primer.
  - To learn more about the best practices on defining policy assertions, see Web Services Policy 1.5 - Guidelines for Policy Assertion Authors.
  - The Primer and Guidelines documents should not be considered as replacements. For more detailed information about WS-Policy, see:
    - Web Services Policy 1.5 – Framework specification
    - Web Services Policy 1.5 – Attachment specification

- **Discuss**
  - Participate in the WebSphere Service Registry and Repository discussion forum.
  - Participate in the WebSphere DataPower discussion forum.
About the authors

Mario E. De Armas

Mario De Armas is a Senior Technical Staff Member on the WebSphere DataPower Appliance development team. He has worked on many different areas and features of DataPower appliances. He is currently the Software Architect responsible for DataPower's integration with WebSphere Service Registry and Repository, Web services, WS-Policy, and SOA Governance functionality. His past DataPower projects included Technical Lead for XB60, AS1 protocol handler, B2B gateway, appliance management functionality, database integration, and user interface customization features.

Steve Groeger

Steve Groeger is a Senior Systems Engineer in the WebSphere Service Registry and Repository (WSRR) team working as part of the development team in the United Kingdom. He has worked on various aspects of WSRR and has most recently been working on the integration of WSRR with WebSphere DataPower and WS-Policy (WS-Mediation policy). Steve has a BSc (Honors) degree in Mathematics and Computing from Southbank Polytechnic, London, United Kingdom, and has worked at IBM for over 26 years.

Phil Schaadt

Phil Schaadt leads the SOA Governance and Connectivity team at Haddon Hill Group Inc. He focuses on policy-driven SOA implementations at large enterprises in healthcare, retail, and finance. He has been the overall Solution Architect for client implementations using WebSphere Service Registry and Repository (WSRR), WebSphere DataPower, IBM Workload Deployer, WebSphere Process Server, WebSphere Operational Decision Management, and WebSphere Message Broker. He was General Manager of Technical Strategy, Architecture, Standards, and Consulting at Bank of America, as well as Vice-President of Technical Research and Consulting in TRW's commercial Systems Integration Group for the world's largest financial services, telecommunications, and healthcare companies.

Richard Duran

Rich Duran is one of the world’s leading implementers of WebSphere Service Registry and Repository (WSRR), customizing and integrating SOA Governance
including advanced policy integration with WebSphere DataPower for multiple large healthcare, retail, and financial clients. Rich has been a continuous participant in the IBM Hursley (United Kingdom) Lab’s WSRR Early Design, Early Access, and beta programs as well as certified in DataPower appliances.

Deborah Zippel

Deb Zippel specializes in SOA governance and technical process design and documentation as well as business process modeling and state and rules elaboration. She is currently focused on WebSphere Service Registry and Repository (WSRR) using WebSphere DataPower, including developing step-by-step user manuals and customized training for multiple Haddon Hill Group clients in healthcare, retail, and financial services.

© Copyright IBM Corporation 2013
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)