Part 1: Integrating WebSphere Service Registry and Repository with WebSphere MQ

This section describes integrating WebSphere Service Registry and Repository with WebSphere MQ.
Value proposition

In many enterprises, WebSphere MQ applications play a key role in enterprise message processing and act as service providers. These applications can be considered as services within a service-oriented architecture (SOA) solution, as they are loosely coupled and interoperate irrespective of the platforms on which they run.

There are a number of advantages in registering WebSphere MQ applications as services in WSRR. They are:

- Cataloging WebSphere MQ applications as software assets.
- Applying SOA Governance and perform change impact analysis.
- Identifying assets for reuse.
- Management and auditing.
- Monitoring of WebSphere MQ Services, for example, usage statistics and health monitoring.

WebSphere MQ Applications in SOA

WebSphere MQ applications can be described as services for use in SOA using the WebSphere MQ Service Definition specification and the WebSphere MQ IRI specification. These specifications are available as SupportPac MA93 and can be downloaded from the SupportPac site at the following address:


The WebSphere MQ Service Definition specification defines the standard for describing WebSphere MQ applications as services using WSDL and the WebSphere MQ IRI specification defines the standard for addressing WebSphere MQ message destinations as URIs in a service definition.

Cataloging WebSphere MQ Applications in WSDL for Reuse in SOA, REDP-4350 discusses these specifications and provides an example scenario that shows how existing WebSphere MQ applications can be described as services using a MQ Service Definition.

Using MQ Service Definition to describe WebSphere MQ applications allows these applications to be available as Web services within SOA and expands their availability to different service consumers, thus promoting reuse and allowing them to be managed as standard Web services.
WebSphere MQ Service Definition Wizard

WebSphere MQ V7 provides a Service Definition Wizard in WebSphere MQ Explorer, which enables creation of WSDL for an WebSphere MQ application. This wizard allows easy creation of WMQ WSDL using WMQ Service Binding and generates the WSDL document based on WebSphere MQ Service Definition and WebSphere MQ IRI specifications. The following steps show how to create a MQ WSDL document using the Service Definition Wizard.

Before generating a WSDL document for a WebSphere MQ application, a Service Definition Repository needs to be created, as shown in the following steps. This repository will be used to store the WebSphere MQ Service definitions.

Perform these steps:

1. Right-click Service Definition Repositories and click Add Repository, as shown in Figure 1.

![Figure 1 Create Service Definition Repository](image-url)
2. Enter the name of the new Service Definition Repository, as shown in Figure 2, and click **Finish**.

![Add New Service Definition Repository](image1)

*Figure 2 Enter the Repository Name*

3. Right-click **Account Services** and select **New Service Definition**, as shown in Figure 3.

![IBM WebSphere MQ Explorer](image2)

*Figure 3 Create a new Service Definition*

4. Enter the name of Service Definition and select a Message Exchange Pattern, as shown in Figure 4 on page 5.
A Message Exchange Pattern describes the message pattern required by the WMQ application for which the WSDL document is being generated. WebSphere MQ Service Definition provides support for Request-Response and One-Way message exchange patterns. The Message Exchange Pattern should be selected as One-Way or Request-Response, depending on the application.

5. Click **Next** to select a Binding Type for the service, depending on the target application, as shown in Figure 5.
WMQ Service Binding provides support for native MQ and WMQ SOAP binding. The Binding Type should be selected as MQ or SOAP/MQ, depending on whether the application expects a WebSphere MQ message or a SOAP message.

6. Click **Next** and create an operation, as shown in Figure 6.

![Figure 6 Create an operation](image)

*Figure 6 Create an operation*
7. Select the property **Input destination** and provide the input destination queue name, input destination queue manager, and connection properties for the request message, as shown in Figure 7.

![Figure 7: Provide Input destination properties](image)
8. Select the property **Input message schema** and enter the schema for the request message payload, as shown in Figure 8.

![Figure 8: Input message schema](image)

The message should be modeled as an XML schema.
9. Select the property **Output destination** and provide the input destination queue name, input destination queue manager, and connection properties for the request message, as shown in Figure 9.

![Figure 9 Provide output destination properties](image_url)
10. Select the property **Output message schema** and enter the schema for the response message payload, as shown in Figure 10.

![Figure 10  Output message schema](image)

11. Click **Finish** to create the service definition.

This process generates the AccountCreationV1_0_MQPort service definition, as shown in Example 1.

**Example 1  AccountCreationV1_0_MQPort service definition**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:importinms="http://www.jkhl.com/Account"
xmlns:tns="http://tempuri.org/AccountCreationV1_0_MQPort"
xmlns:WebSphere MQservice="http://www.ibm.com/xmlns/prod/WebSphereMQ/bindings/1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="AccountCreationV1_0_MQPort"
targetNamespace="http://tempuri.org/AccountCreationV1_0_MQPort">
  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="http://www.jkhl.com/Account"
schemaLocation="AccountCreationSchema.xsd"/>
    </xsd:schema>
  </types>
  <message name="createAccount_Input">
    <part name="createAccount_Input_Part" type="importinms:CustomerBO"/>
  </message>
  <message name="createAccount_Output">
```
<part name="createAccount_Output_Part" type="importinms:TrueOrFalseBO"/>
</message>
<portType name="AccountCreationV1_0_MQPort_PortType">
<operation name="createAccount">
  <input message="tns:createAccount_Input"/>
  <output message="tns:createAccount_Output"/>
</operation>
</portType>
<binding name="AccountCreationV1_0_MQPort_WebSphere MQ_Binding" type="tns:AccountCreationV1_0_MQPort_PortType">
<WebSphere MQservice:binding/>
<operation name="createAccount">
  <input>
    <WebSphere MQservice:body/>
  </input>
  <output>
    <WebSphere MQservice:body/>
  </output>
</operation>
</binding>
<service name="AccountCreationV1_0_MQPort">
  <port binding="tns:AccountCreationV1_0_MQPort_WebSphere MQ_Binding" name="AccountCreationV1_0_MQPort_WebSphere MQ_Port">
    <WebSphere MQservice:address location="WebSphere MQ:/msg/queue/REQUEST.CREATEACCOUNT.QUEUE@CORP.ACCOUNTS.QM"/>
    <WebSphere MQservice:replyTo>WebSphere MQ:/msg/queue/RESPONSE.CREATEACCOUNT.QUEUE@CORP.ACCOUNTS.QM</WebSphere MQservice:replyTo>
  </port>
</service>
</definitions>
**WebSphere MQ Services in WSRR**

WSRR provides support for the WebSphere MQ Service Definition WSDL document through the WebSphere MQ integration feature. The WebSphere MQ integration feature is used to load WebSphere MQ WSDL documents into WSRR. WSRR V6.3 provides this feature in the Governance Enablement Profile, which is the default profile. This feature can be enabled in any profile by following the instructions in WSRR Information Center, which can be found at the following address:


When a WebSphere MQ WSDL document is loaded into WSRR, it automatically builds:

- The WSDL logical port object on which the MQ queue and connection were specified in the MQ WSDL document, along with relationships between queue managers, queues, and connections
- Representations of the queue managers
- Representations of the queues and connections

**Loading a WebSphere MQ service into WSRR**

The following steps show how to export the MQ Service definition using WebSphere MQ Explorer.
Perform these steps:

1. From the Service definition repository, right-click the WebSphere MQ Service Definition that needs to be exported and click Export, as shown in Figure 11.

![Figure 11 Export the MQ Service Definition](image)
2. Provide the destination directory for the export and click **OK**, as shown in Figure 12.

![Export](image)

*Figure 12  Export to destination folder*

This generates the AccountCreationV1_0_MQPort.wsdl document in the directory provided as the destination folder during export.

3. Load the WSDL document and dependant XML schema into WSRR, as shown in Figure 13 on page 15.
WebSphere MQ Objects in WSRR

With the WebSphere MQ integration feature enabled, when loading the WSDL document, WSRR automatically parses the MQ WSDLs and builds the MQ entities from the service definition.

The WebSphere MQ entities that are created in WSRR when loading the WebSphere MQ Service Definition WSDL can be viewed from Development and SOA Governance Perspectives in the Governance Enablement Profile. The following steps show how the WebSphere MQ entities are created.
Perform these steps:

1. Select View → Service Model → MQ Services → Queue Managers, as shown in Figure 14.

![Figure 14 View created MQ entities](image1)

2. The Queue Manager entities created by WSRR appear, as shown in Figure 15.

![Figure 15 Queue Manager entities](image2)
3. Click the WebSphere MQ Queue Manager CORP.ACCOUNTS.QM object to see its properties and dependent MQ Queues, as shown in Figure 16.

![Queue Manager entity properties](image)

Figure 16  Queue Manager entity properties

One key advantage of registering MQ Services with WSRR is to enable SOA governance and perform change impact analysis. Impact analysis shows which objects have a dependency on an entity. Determining which resources are going to be affected by changes to an entity will help manage service availability.

The following steps show how to perform impact analysis on the Queue Manager entity CORP.ACCOUNTS.QM.

Perform these steps:

1. Click **Impact Analysis** on the Queue Manager properties window, as shown in Figure 16.
2. In the Built-in relationships pane, select **WSDL service to WSDL port**, and in the Custom relationships pane, select **sm63_queueManager**, **sm63_requestQueue**, **sm63_responseQueue**, and **sm63_connectQueueManager**, as shown in Figure 17.

![Figure 17 Queue Manager impact analysis](image)

3. Click **Go** to view a graphical view of Impact Analysis for the Queue Manager CORP.ACCOUNTS.QM entity and which services have a dependency on this Queue Manager, as shown in Figure 18 on page 19.
It is also possible to see the complete WebSphere MQ Service and determine which operations, resources, and message exchange patterns are being used.
Perform the following steps:

1. Select **View → Service Model → Web Services → Services**, as shown in Figure 19.

![Figure 19  View services in WSRR](image-url)
2. Click the graph icon for the service AccountCreationV1_0_MQPort, as shown in Figure 20.

Figure 20   Show a service in graph view
3. The graphical view of the service AccountCreationV1_0_MQPort, showing the objects on which it is dependent, is shown in Figure 21.

![Graphical view of the service](image)

**Figure 21  Graphical view of a service**

**Creating MQ objects in WSRR manually**

The WebSphere MQ Integration feature in WSRR also provides functionality to manually create WebSphere MQ endpoints and WebSphere MQ objects using the WSRR console. This can be done from the Development and SOA Governance Perspectives in the Governance Enablement Profile. The following WebSphere MQ objects can be created manually in WSRR:

- Manual MQ Endpoint
- Manual Client MQ Endpoint
- Manual Client Channel Table MQ Endpoint
- MQ Connection
- MQ Queue
- MQ Queue Manager
- MQ Service Endpoint
WebSphere MQ Objects can be created manually from the WSRR console by clicking **Actions**, selecting **Create → MQ**, and then clicking the WebSphere MQ object that needs to be created, as shown in Figure 22.

![Image of WebSphere MQ objects creation](image.png)

*Figure 22   Creating MQ objects manually in WSRR*

An example scenario provided in “Example scenario 1” on page 35 shows how to create a MQ Service Endpoint and a Manual MQ Endpoint.

Integrating WSRR with WebSphere MQ environments allows organizations to catalog and classify their MQ applications, perform impact analysis, and enable better understanding and increase the visibility of these applications, which promotes reuse and life cycle management.

**Part 2: Integrating WebSphere Registry and Repository with WebSphere Message Broker**

This section describes integrating WebSphere Service Registry and Repository with WebSphere MQ.
Value proposition

WebSphere Message Broker provides advanced transformation and integration functionality and is used as an enterprise service bus for connectivity of enterprise applications over a wide range of protocols and message formats. WebSphere Message Broker plays a key role in SOA implementations and acts as a transformation and connectivity engine. Integrating WSRR with WebSphere Message Broker allows Message Broker to access services registered with WSRR at run time, thus allowing for dynamic connectivity between service consumers and service providers. WebSphere Message Broker can also dynamically retrieve resources from WSRR at run time to be used in message flow processing.

WSRR nodes in WMB

WebSphere Message Broker provides two built-in primitive nodes that can be used to create message flows to dynamically retrieve service endpoints or other resources from WSRR. The two built-in nodes are:

- EndpointLookup node
- RegistryLookup node

The EndpointLookup node is used to retrieve a service endpoint for a WSDL service from WSRR. The node inserts the retrieved service endpoint in the local environment tree for use by subsequent SOAP or HTTPRequest nodes to call the Web service.

The RegistryLookup node is used to retrieve any artifacts that are stored in WSRR, for example, WSDL, XML schema, XSLT, policy documents, and so on. This is a general purpose node that can be used to query and retrieve any document or meta data from WSRR. The node inserts the retrieved artifacts in the local environment tree for use in subsequent processing of the message flow.

These nodes are provided with Message Broker Toolkit V6.1 and can be found in the WebServices folder of the message flow node palette.

WSRR nodes in a message flow

This section addresses the EndpointLookup and RegistryLookup nodes.
**EndpointLookup node**

The EndpointLookup node uses the PortType property to query WSRR to retrieve a service endpoint, as described in the WSDL specified by the port. The node can retrieve one endpoint or multiple matching endpoints, depending on the configuration.

The EndpointLookup node can be configured with multiple properties to define the search criteria for a service in WSRR. The EndpointLookup node configuration properties are shown in Figure 23.

![Figure 23 EndpointLookup node properties](image)

- **Property Name**: PortName
- **Property Type**: String
- **Property Value**: AccountCreationV1_0

Optional Port user defined properties for the query:

- **Property Name**: portType
- **Property Value**: AccountCreationV1_0

Optional Port Classification URI to query based on classification:

- **Classification URI**: http://www.ibm.com/jnlp/jnlp/serviceRegistry/lifecycle/v5.3/lifecycleDefinitionOnline

Retrieve one matching endpoint or all matching.
The PortType information is mandatory and at least one of the PortType Name, PortType name space, or PortType version must be defined to uniquely identify a WSDL service PortType defined in WSRR. The mapping between EndpointLookup node properties and properties in WSRR for PortType are shown in Figure 24.

The User Properties in the EndpointLookup node is optional and can be used to retrieve a service endpoint that is based on user defined properties on the port. The User Properties can be used as filter criteria for the endpoint lookup and can be a static string, an ESQL expression, or a XPATH expression. The mapping between the User Properties in the EndpointLookup node and the published service port in WSRR is shown in Figure 25.
The Classification property in the EndpointLookup node is optional and can be used to retrieve a service endpoint based on the classification on the port in WSRR. The mapping between the Classification property in the EndpointLookup node and the published service port in WSRR is shown in Figure 26.
The Match Policy property can be set to One or All. If the Match Policy property is set to One, the node retrieves the first matching endpoint and places the retrieved endpoint in LocalEnvironment.Destination.HTTP.RequestURL and in LocalEnvironment.Destination.SOAP.Request.Transport.HTTP.WebServiceURL, as shown in Example 2.

Example 2  Match Policy property

```
(0x01000000:Name):Destination     = (  
  (0x01000000:Name):soap = (  
    (0x01000000:Name):Request = (  
      (0x01000000:Name):Transport = (  
        (0x01000000:Name):HTTP = (  
          (0x03000000:NameValue):WebServiceURL =  
            'http://9.42.170.194:9082/AccountCreationV1_0/services/AccountCreationServiceV1_0_ProductionPort' (CHARACTER)  
          )  
        )  
      )  
    )  
  )  
)  

(0x01000000:Name):HTTP = (  
  (0x03000000:NameValue):RequestURL =  
    'http://9.42.170.194:9082/AccountCreationV1_0/services/AccountCreationServiceV1_0_ProductionPort' (CHARACTER)  
)  
)

(0x01000000:Name):ServiceRegistry = (  
  (0x01000000:Name):ITService = (  
    (0x01000000:Name):Endpoint = (  
      (0x03000000:NameValue):Address        =  
        'http://9.42.170.194:9082/AccountCreationV1_0/services/AccountCreationServiceV1_0_ProductionPort' (CHARACTER)  
      )  
    )  
  )  
)  

(0x01000000:Name):PortType       = (  
  (0x03000000:NameValue):name      = 'AccountCreationV1_0' (CHARACTER)  
)  

(0x03000000:NameValue):namespace =  
  'http://www.jkh1.com/AccountCreationV1/interface' (CHARACTER)  
(0x03000000:NameValue):version   = '1.0' (CHARACTER)  
)  

(0x03000000:NameValue):Classification =  
  'http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Online' (CHARACTER)  
)  
)  
)  
```
A subsequent HTTPRequest node or a SOAPRequest node can use this information to select the service dynamically at run time and route the request to the service provider.

If the Match Policy property is set to All, all endpoints matching the lookup query are retrieved from WSRR and are added to the local environment tree. In this case, since multiple endpoints could be retrieved from WSRR, a compute node should be used to select the endpoint address and set it in the LocalEnvironment.Destination URL to be used by a HTTPRequest node or a SOAPRequest node before a SOAPRequest or a HTTPRequest node can use this information.

A typical usage of an EndpointLookup node in a message flow is shown in Figure 27.

The compute node after the EndpointLookup node is used to select the desired endpoint to forward the service request, if multiple endpoints are retrieved from WSRR.

Note: The EndpointLookup node can only retrieve SOAP endpoints and not MQ endpoints. A RegistryLookup node will be required to retrieve an MQ endpoint from WSRR.

RegistryLookup node
The RegistryLookup node uses the name of the artifact to query and retrieve from WSRR. The node places the retrieved artifact in the LocalEnvironment.ServiceRegistry folder to be used in a subsequent compute node for message flow processing, allowing for dynamic processing of the message at run time.
The RegistryLookup node can be configured with multiple properties to define the search for an artifact in WSRR. The RegistryLookup node configuration properties are shown in Figure 28.

![Figure 28 RegistryLookup node properties](image)

The configuration properties for the RegistryLookup node are the same as the ones for the EndpointLookup node. In addition to these properties, the RegistryLookup node has an additional property that defines the depth of the query and the artifacts to be retrieved from WSRR. This property is called Depth Policy and the values for this property are shown in Figure 29.

![Figure 29 RegistryLookup node Depth Policy property](image)
These properties are described in detail in the WebSphere Message Broker Information Center, which is available at the following address:


The RegistryLookup node can be used to query the MQ service definition that is loaded in WSRR. A simple message flow that uses the RegistryLookup node to retrieve the AccountCreationV1_0_MQPort Service Definition shown in Example 1 on page 10 is shown in Figure 30.

![Image of RegistryLookup node in a message flow]

The message flow exposes the MQ Service Definition AccountCreationV1_0_MQPort as a Web service using the HTTP Input node. The RegistryLookup node is used to retrieve the MQ WSDL from WSRR and is configured using the properties shown in Figure 28 on page 30.

The ParseMQWSDL compute node parses the MQ WSDL retrieved by the RegistryLookup node for the service location:

```xml
<wmqservice:address
location="wmq:/msg/queue/REQUEST.CREATEACCOUNT.QUEUE@CORP.ACCOUNTS.QM"/>
```

The queue name from the above service location is parsed and is set in OutputLocalEnvironment.Destination.MQ.DestinationData.queueName and used by the MQOutput node MQCreateAccountRequest to send the request.

The ParseMQWSDL compute node also parses the replyTo location from the MQ WSDL:

```xml
<wmqservice:replyTo>wmq:/msg/queue/RESPONSE.CREATEACCOUNT.QUEUE@CORP.ACCOUNTS.QM</wmqservice:replyTo>
```

Note: WebSphere Message Broker V6.1.0.4 is required to use the Depth Policy property.
The queue name from the replyTo location is parsed and is set in OutputLocalEnvironment.MQ.GET.QueueName so that an MQGet node can pick up the response message for the service request.

The MQGet node forwards the response message to the HTTPReply node so that the response is returned to the client.

The ESQL for the compute node ParseMQWSDL is shown in Example 3.

Example 3  ESQL for the ParseMQWSDL computer node

CREATE COMPUTE MODULE AccountCreate_Compute
   CREATE FUNCTION Main() RETURNS BOOLEAN
   BEGIN

      DECLARE NS1 NAMESPACE 'http://schemas.xmlsoap.org/wsdl/';
      DECLARE NS2 NAMESPACE 'http://www.ibm.com/xmlns/prod/wmq/bindings/1.0';
      DECLARE MQENDPOINT, MQREPLYPROPS CHAR ;
      DECLARE REQUESTQNAME, REQUESTQMNAME, REPLYQNAME, REPLYQMNAME CHAR;
      DECLARE MQIRIPREFIX INTEGER 16;

      SET OutputRoot = InputRoot;

      CREATE FIELD Environment.XMLNSC.SRResult ;
      DECLARE cursor REFERENCE TO Environment.XMLNSC.SRResult;
      CREATE FIRSTCHILD OF cursor Domain('XMLNSC') PARSE
       (InputLocalEnvironment.ServiceRegistry.Entity.content,
        InputProperties.Encoding, InputProperties.CodedCharSetId) ;

      SET MQENDPOINT = Environment.XMLNSC.SRResult.XMLNSC.NS1:definitions.NS1:service.NS1:port .NS2:address.location ;

      SET REQUESTQNAME = SUBSTRING ( SUBSTRING(MQENDPOINT BEFORE '@')
       FROM MQIRIPREFIX ) ;
      SET REQUESTQMNAME = SUBSTRING(MQENDPOINT AFTER '@') ;

      SET MQREPLYPROPS = Environment.XMLNSC.SRResult.XMLNSC.NS1:definitions.NS1:service.NS1:port .NS2:replyTo ;

      SET REPLYQNAME = SUBSTRING ( SUBSTRING(MQREPLYPROPS BEFORE '@')
       FROM MQIRIPREFIX ) ;
      SET REPLYQMNAME = SUBSTRING(MQREPLYPROPS AFTER '@') ;
SET OutputLocalEnvironment.Destination.MQ.DestinationData.queueName = REQUESTQNAME;

SET OutputRoot.MQMD.ReplyToQ = REPLYQNAME;
SET OutputRoot.MQMD.ReplyToQMgr = REPLYQMNAME;

SET OutputLocalEnvironment.MQ.GET.QueueName = REPLYQNAME;

END;

END MODULE;

For the message flow to use the request message and receive the response message successfully, the broker Queue Manager and Queue Manager in the MQ WSDL CORP.ACCOUNTS.QM entity must be defined in a cluster with the request and response queues defined as cluster queues. The request queue REQUEST.CREATEACCOUNT.QUEUE must be defined as a cluster local queue on the Queue Manager CORP.ACCOUNTS.QM entity, and the response queue RESPONSE.CREATEACCOUNT.QUEUE entity must be defined as a cluster local queue on the broker Queue Manager.

**Dynamic search criteria**

The search queries EndpointLookup and RegistryLookup to WSRR can be made dynamic by providing the properties required by the node in LocalEnvironment. This will enable a dynamic search for the endpoint or artifact in WSRR based on the input service request and context of the message flow processing. The required properties can be defined in OutputLocalEnvironment.ServiceRegistryLookupProperties. This can be done by setting these properties within a compute node before the EndpointLookup or RegistryLookup node. A detailed description of these properties can be found in the WebSphere Message Broker Information Center, which is available at the following address:

Configuring WebSphere Message Broker connectivity to WSRR

Before deploying a message flow that uses an EndpointLookup or RegistryLookup node, WebSphere Message Broker needs to be configured to access WSRR. The following command is used to configure WebSphere Message Broker to be able to access WSRR:

```
mqsichangeproperties <BrokerName> -c ServiceRegistries -o DefaultWSRR -n endpointAddress -v http://<WSRR Server Hostname or IP Address>:9080/WSRR6_3/services/WSRRCoreSDOPort
```

**Note:** WebSphere Message Broker can be configured to connect to only one WSRR server at a time using the DefaultWSRR configurable service object. Its not possible to configure Message Broker to connect to multiple WSRR servers.

WebSphere Message Broker can be configured to connect to a security enabled WSRR server. The configuration can be performed by following the instructions in WebSphere Message Broker Information Center, which is available at the following address:


Message Broker will need to be restarted after configuring the connectivity to WSRR.

Caching and cache notification

WebSphere Message Broker provides a caching functionality to store the information retrieved from WSRR. Caching the WSRR lookup results improves the performance of the message processing in Message Broker so that a lookup does not need to occur for every message processed by the message flow. Caching is enabled by default in the Message Broker and the result of the first occurrence of each lookup query to WSRR is stored in the cache. The EndpointLookup and RegistryLookup nodes use the information stored in the broker cache for subsequent message processing. The cached results of the query are discarded after the time specified in the timeout configurable parameter expires. The timeout parameter can be changed by using the following command:

```
mqsichangeproperties <BROKER NAME> -c ServiceRegistries -o DefaultWSRR -n timeout -v <Value in milliseconds>
```

WebSphere Message Broker can also be enabled to use the WSRR notification functionality. WSRR has a publish feature that provides notifications for changes
to artifacts. WSRR uses JMS for this notification functionality and publishes a message to topic jms/SuccessTopic on connection factory jms/SRConnectionFactory after a successful change to the artifacts. Message Broker can be configured to subscribe to this topic so that any changes occurring within WSRR are sent to the Message Broker so that the cache will be updated. The cache notification functionality within Message Broker can be enabled by using the following commands:

```
mqsichangeproperties <BROKER NAME> -c ServiceRegistries -o DefaultWSRR -n enableCacheNotification -v true
mqsichangeproperties <BROKER NAME> -c ServiceRegistries -o DefaultWSRR -n locationJNDIBinding -v iiop://localhost:<bootstrap_address>
```

Message Broker will need to be restarted after enabling the cache and cache notification functionality.

**Example scenarios**

This section provides the following two example scenarios:

1. Create an MQ Service Endpoint and manual MQ endpoint in WSRR to expose the MQ application as a service.
2. Dynamically route a service request to an available service endpoint using the EndpointLookup node.

**Example scenario 1**

This scenario shows a basic example of how an MQ application can be exposed as a service by creating a manual MQ endpoint in WSRR. The example also shows the creation of an MQ Service Endpoint, so that the manual MQ endpoint can be governed.

The manual MQ endpoint is retrieved from WSRR using a RegistryLookup node in a message flow, which exposes the MQ application as a Web service using an HTTP Input node.

The scenario uses a fictitious company, JKHL Enterprises, which is using WSRR as their service repository. The company has an MQ application that processes CreateAccount requests by receiving the request on MQ queue REQUEST.CREATEACCOUNT.QUEUE and sending a response to the MQ queue RESPONSE.CREATEACCOUNT.QUEUE on Queue Manager CORP.ACCOUNTS.QM. By exposing the MQ application as a service, the company can receive CreateAccount requests not only from MQ applications but also from other Web applications that can use the service.
Perform these steps:

1. Click **MQ Service Endpoints**, as shown in Figure 31.

![Figure 31  MQ Service Endpoint](image)

2. Click **New** to create a new MQ Service Endpoint, as shown in Figure 32.

![Figure 32  Create MQ Service Endpoint](image)
3. Specify the properties for the new MQ Service Endpoint, as shown in Figure 33, and click **Add MQ Endpoint** to add a manual MQ Endpoint.
4. Select the Entity type **Manual MQ Endpoint** and click **Create**, as shown in Figure 34.

*Figure 34  Create Manual MQ Endpoint*
5. Specify the properties for the Manual MQ Endpoint and click **OK**, as shown in Figure 35.

![Figure 35 Create Manual MQ Endpoint](image-url)
6. Click **Finish** to complete the creation of the CreateAccount MQ Service Endpoint, as shown in Figure 36.
7. This action creates the CreateAccount MQ Service Endpoint with a relationship to MQ Endpoint CreateAccountMQ and places the endpoint under governance with an initial governance state of Offline, as shown in Figure 37.
8. To change the governance state to Online, the MQ Service Endpoint needs to be classified as one of the environments. In this example, the CreateAccount MQ Service Endpoint is classified as a Development environment, as shown in Figure 38.

Figure 38   Classify MQ Service Endpoint environment
9. To change the governance state to Online, click **Approve For Use**, as shown in Figure 39.
10. A simple message flow to retrieve the manual MQ Endpoint from WSRR can be constructed, as shown in Figure 40.

![Figure 40 Message flow to retrieve MQ Endpoint](image)

11. The RegistryLookup node properties in the message flow that are used to retrieve the manual MQ Endpoint based on classification and the governance state of Online are shown in Figure 41. By retrieving the endpoint based on classification, the current active endpoint can be retrieved, allowing for dynamic routing of the service request.

![Figure 41 RegistryLookup node](image)

12. The compute node Parse MQ Endpoint parses the request and reply queue and queue manager names from the retrieved MQ Endpoint to be used by the message flow to route the request to the request queue and to read the response from the reply queue. The ESQL for this compute node is shown in Example 4 on page 45.
Example 4  ESQL for Parse MQ Endpoint compute node

CREATE COMPUTE MODULE RegistryLookupFlow_Parse_MQ_Endpoint
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN

    DECLARE NS1 NAMESPACE 'http://schemas.xmlsoap.org/wsd1/';
    DECLARE NS2 NAMESPACE
    'http://www.ibm.com/xmlns/prod/wmq/bindings/1.0';
    DECLARE REQUESTQNAME, REQUESTQMNAME, REPLYQNAME, REPLYQMNAME CHAR;

    SET OutputRoot = InputRoot;

    SET REQUESTQMNAME =
    InputLocalEnvironment.ServiceRegistry.Entity.userDefinedProperties[1].value ;
    SET REQUESTQNAME =
    InputLocalEnvironment.ServiceRegistry.Entity.userDefinedProperties[2].value ;

    SET REPLYQMNAME =
    InputLocalEnvironment.ServiceRegistry.Entity.userDefinedProperties[5].value ;
    SET REPLYQNAME =
    InputLocalEnvironment.ServiceRegistry.Entity.userDefinedProperties[7].value ;

    SET OutputLocalEnvironment.Destination.MQ.DestinationData.queueName = REQUESTQNAME ;

    SET OutputRoot.MQMD.ReplyToQ = REPLYQNAME ;
    SET OutputRoot.MQMD.ReplyToQMgr = REPLYQMNAME ;

    SET OutputLocalEnvironment.MQ.GET.QueueName = REPLYQNAME;

    END;

END MODULE;

The sample message flow can be found in a project interchange file that is available as a download with the additional material supplied with this paper.
Example scenario 2

This scenario shows a basic example of how an EndpointLookup node can be used within a message flow to dynamically route the service request to an available service endpoint that is governed in WSRR.

The scenario uses a fictitious company, JKHL Enterprises, which uses WSRR as their service repository. To protect applications from changes to the Account Creation service, JKHL utilizes an EndpointLookup node in a message flow to look up the active endpoint of the Account Creation service in WSRR.

The company has an Account Creation service running in their production environment that requires a minor upgrade. The version that is currently running in production is Version 1_0. Once Version 1_1 of the Account Creation service is deployed and available, the endpoint for Version 1_0 will be classified as offline in WSRR. All service consumer requests for the Account Creation service will be directed to the endpoint for Version 1_1, enabling Version 1_0 to be deprecated with no impact on service consumers and with no interruption to the service.

In this endpoint lookup scenario, WSRR is used to retrieve a single matching endpoint for the Account Creation service, which is classified as online.

Perform these steps:

1. Load the Version 1_0 and Version 1_1 service WSDLs and their dependant artifacts into WSRR, as shown in Figure 42.

Figure 42  Load service WSDLs
2. Initiate the endpoint life cycle for Version 1.0 of the Account Creation service from the Governance tab of the service WSDL document. Select \texttt{InitiateEndpointLifecycle} and click \texttt{Govern}, as shown in Figure 43.

![Figure 43 Initiate endpoint life cycle](image)

3. In the Change Governance State pane, select \texttt{Approve For Use} from the Available state transitions drop-down menu and click \texttt{Transition} to change the classification state of the service to Online, as shown in Figure 44.

![Figure 44 Change service transition state to Approve For Use](image)
This will change the classification of the service WSDL to Online, as shown in Figure 45.

4. Click **Edit Classifications**, as shown in Figure 45, to discover the classification URI of the service WSDL, as shown in Figure 46.
The classification URI will be used in the EndpointLookup node to search for the available endpoint.

5. A simple message flow to retrieve the active endpoint from WSRR can be constructed, as shown in Figure 47.

![Simple message flow](image)

**Figure 47** Simple message flow

6. The EndpointLookup node properties in the message flow are shown in Figure 48.

![EndpointLookup node properties](image)

**Figure 48** EndpointLookup node properties

In order to retrieve the active endpoint from WSRR, the lookup is performed using a name space, and the classification will be a Match Policy of One.
Processing a service request through the message flow and the TraceNode after processing it through the EndpointLookup node will produce the retrieved endpoint for Version 1.0 of the Account Creation service, as shown in Example 5.

**Example 5  Account Creation service endpoint**

```java
(0x01000000:Name):Destination = (
    (0x01000000:Name):SOAP = (        
        (0x01000000:Name):Request = ( 
            (0x01000000:Name):Transport = ( 
                (0x01000000:Name):HTTP = ( 
                    (0x03000000:NameValue):WebServiceURL = 'http://9.42.170.194:9082/AccountCreationV1_0/services/AccountCreationServiceV1_0_ProductionPort' (CHARACTER) ) 
                ) 
            ) 
        ) 
    ) 

(0x01000000:Name):HTTP = ( 
    (0x03000000:NameValue):RequestURL = 'http://9.42.170.194:9082/AccountCreationV1_0/services/AccountCreationServiceV1_0_ProductionPort' (CHARACTER) ) 

(0x01000000:Name):ServiceRegistry = ( 
    (0x01000000:Name):ITSService = ( 
        (0x01000000:Name):Endpoint = ( 
            (0x03000000:NameValue):Address = 'http://9.42.170.194:9082/AccountCreationV1_0/services/AccountCreationServiceV1_0_ProductionPort' (CHARACTER) ) 
        ) 
    ) 

(0x01000000:Name):PortType = ( 
    (0x03000000:NameValue):name = 'AccountCreationV1_0' (CHARACTER) 
    (0x03000000:NameValue):namespace = 'http://www.jkhl.com/AccountCreationV1/interface' (CHARACTER) 
    (0x03000000:NameValue):version = '1.0' (CHARACTER) 
) 

(0x03000000:NameValue):Classification = 'http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Online' (CHARACTER) 
```


When Version 1.1 of the Account Creation service is available, initiate the endpoint life cycle for Version 1.1 of the Account Creation service, change the governance state to Approve For Use, and classify the service as online, using the same procedure described for Version 1.0 of the service.

Because Version 1.1 of the Account Creation service is online, revoke Version 1.0 of the service. You can do this task by going to the Governance tab of the service WSDL, selecting **Revoke From Use** in the Change Governance State drop-down menu, and clicking **Transition**, as shown in Figure 49.

![Figure 49 Revoke Version 1.0 of the service](image)
Processing a service request through the message flow and the TraceNode after processing it through the EndpointLookup node will produce the retrieved endpoint for Version 1_1 of the Account Creation service, as shown in Example 6.

**Example 6  Account Creation service endpoint**

```
(0x01000000:Name):Destination   = (  
  (0x01000000:Name):SOAP = (  
    (0x01000000:Name):Request = (  
      (0x01000000:Name):Transport = (  
        (0x01000000:Name):HTTP = (  
          (0x03000000:NameValue):WebServiceURL = 'http://9.42.170.194:9082/AccountCreationV1_1/services/AccountCreationServiceV1_1_ProductionPort' (CHARACTER) )  
        )  
      )  
    )  
  )  
)(0x01000000:Name):HTTP = (  
  (0x03000000:NameValue):RequestURL = 'http://9.42.170.194:9082/AccountCreationV1_1/services/AccountCreationServiceV1_1_ProductionPort' (CHARACTER) )
)(0x01000000:Name):ServiceRegistry = (  
  (0x01000000:Name):ITSService = (  
    (0x01000000:Name):Endpoint = (  
      (0x03000000:NameValue):Address = 'http://9.42.170.194:9082/AccountCreationV1_1/services/AccountCreationServiceV1_1_ProductionPort' (CHARACTER)  
    )  
  )  
)(0x01000000:Name):PortType = (  
  (0x03000000:NameValue):name = 'AccountCreationV1_1' (CHARACTER)  
)(0x03000000:NameValue):namespace = 'http://www.jkhl.com/AccountCreationV1/interface' (CHARACTER)  
  (0x03000000:NameValue):version = '1.1' (CHARACTER)  
)(0x03000000:NameValue):Classification = 'http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Online' (CHARACTER)  
)  
)  
)  
```
WSRR cache notification must be enabled for WebSphere Message Broker to receive notification of governance state changes on the service in order to retrieve the correct available endpoint.

**Note:** A defect exists in the WebSphere Message Broker cache notification that causes the governance state changes to *not* be reflected in the Message Broker cache. This results in Message Broker retrieving the incorrect endpoint for the service. APAR IC61999 fixes this problem. Contact IBM support to receive an iFix for this APAR.

The sample message flow can be found in a project interchange file that is available as a download with the additional material supplied with this paper.

### The team who wrote this IBM Redpapers publication

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

**Prasad Imandi** is a Senior Software Engineer working in Level 2 support for the IBM AIM organization in Research Triangle Park, North Carolina. He has 14 years of experience working with multiple WebSphere products at IBM. He is currently focused on working with IBM WebSphere Message Broker, IBM WebSphere Business Events, and WSRR products. He also leads product service teams. Prasad works with customers worldwide and focuses on product integration and troubleshooting issues. He holds a Masters Degree in Computer Science and Engineering from Jadavpur University, India. Prasad has previously been an author and contributed to the IBM Redbooks® publication *Getting Started with WebSphere Enterprise Service Bus V6*, SG24-7212.

Thanks to the following people for their contributions to the paper:

- Martin Keen
- Nicole Hargrove
- Ian Heritage
- Wendy Neave
- Laura Olson
- Bhargav Perepa
- Andrew White
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 give 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.

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.

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.

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 illustrates 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. 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 IBM's application programming interfaces.

© Copyright International Business Machines Corporation 2009. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.