LDAP as a Naming Service in Enterprise Application Integration

Integrating JMS Adapter of IBM Sterling B2B Integrator and IBM WebSphere MQ with OpenLDAP as a Naming Service

Kanagarajan Ramakrishnan Maheshwari

January 20, 2014

In enterprise integrations, applications across different systems are integrated by various means and techniques. In such large-scale integration, there is a need to identify the key integration components or elements and maintain them in a secure centralized location that provides easy and dynamic access. Here, Lightweight Directory Access Protocol (LDAP) is used as a naming service to discover most of the common integration elements. LDAP satisfies the needs and challenges that are faced in a typical integration scenario. The article explains how the LDAP-based naming service is used for application integration. The scenario explains how the JMS queue adapter of IBM® Sterling B2B Integrator® is integrated with another downstream system, IBM® WebSphere® MQ. The JMS adapter application looks up the OpenLDAP registry for key integration elements, such as queue connection details. Using these integration parameters, the JMS adapter connects to the appropriate queue in WebSphere MQ and delivers the message.

Introduction

Large enterprise integration architectures across multiple complex systems are governed by many key best practices in the industry. For centralized management and security reasons, it is recommended to configure and fetch the integration parameters from a naming service. LDAP is preferred as a naming service primarily because it is a lightweight directory. LDAP can be used in different modes. It can be used to provide authentication and naming services. The LDAP Naming Model uses the Directory Information Tree (DIT) structure to hold the name/value information.

The LDAP-based naming service provides the advantage of compatibility across multiple platforms and supports multiple vendors. It acts as a central repository for data, allowing data to be shared by users, applications, and different naming services within the corporate intranets and across extranets.

The JMS queue adapter of Sterling B2B Integrator can be used to exchange messages with a remote JMS server as part of a business process-based application. The JMS queue adapter
can be configured to work in send mode, synchronous mode, or asynchronous receive mode. The JMS adapter takes input parameters, such as Remote Queue Name and Remote Queue Connection Factory. These parameters can be directly configured in the setup, but they can also be accessed or looked up from any standard naming service, such as JNDI or LDAP. In enterprise integration practice, the parameters are ideally maintained in a centralized repository, such as an LDAP server. The article explains the way to integrate with the JMS adapter; to use LDAP as the naming service; to get the key parameters such as queue connection details from an LDAP server and connect to the message queues for message exchange.

Scenario overview

The article describes an enterprise application integration scenario in which LDAP is used as a naming service to integrate with other applications or systems. In this scenario, a fictitious retail company, Marine & Co., uses Sterling B2B Integrator to enable all of its Business-to-Business (B2B) transactions. Marine & Co. uses the OpenLDAP server for authentication. The OpenLDAP is also used as the centralized naming service for all of its enterprise integration deployments that are configured so far. There is a new requirement to transfer the processed document (for example, an invoice document) to the audit administration unit. The document from the workflow, at the end of its process orchestration, needs to be delivered to a WebSphere MQ based queuing system. This new addition is introduced to continue further processing by its administrative unit for annual audit.

In this scenario, the context of illustration is provided on the Windows® operating system. For other platforms, refer to the corresponding platform documentation.

B2B integration engine

Sterling B2B Integrator V5.2.4 is used to exchange documents between the trading partners.

LDAP server

OpenLDAP V2.0.27 is the LDAP server for the naming service.

Message queuing infrastructure

The destination queues are set up using WebSphere MQ V7.1. Sterling B2B Integrator V5.2.4 Business Process uses the JMS queue adapter to establish the connectivity with the queuing infrastructure.

In the example, the integration elements or configuration details from one system are referenced in the configuration of another system. Hence, for more clarity, all the needed parameters are given a variable name of P1, P2, and so on, for ease of reference. Refer to Table 1 for details.

Figure 1 shows an enterprise application integration scenario with LDAP as a naming or directory service for discovering the queuing infrastructure details.
LDAP connection details

**Prerequisite:** It is assumed that the OpenLDAP server is installed and configured appropriately.

From the OpenLDAP server, note the following important parameters. These parameters are used in connection configurations for locating the endpoint.

1. LDAP server suffix (parameter P1): dc=mycompany,dc=com
2. LDAP server binding URL (parameter P2): ldap://10.11.24.75:389

**Figure 2** shows the OpenLDAP server details that were originally configured during installation.
3. Directory Manager DN (parameter P3): cn=Manager,dc=mycompany,dc=com
4. Password (parameter P4): password

Figure 3 shows the directory configuration details of the OpenLDAP.

Figure 3. Directory configuration details of the OpenLDAP

Next, go to the OpenLDAP installation folder and choose the configuration file (for example, slapd.conf). Then, uncomment the inclusions for the OpenLDAP schema, Java™ schema, and core schema, as follows:
# View slapd.conf for details on configuration options.
#
ucdata-path "C:/Program Files/OpenLDAP/ucdata"
include "C:/Program Files/OpenLDAP/schema/core.schema"
include "C:/Program Files/OpenLDAP/schema/cosine.schema"
include "C:/Program Files/OpenLDAP/schema/inetorgperson.schema"
include "C:/Program Files/OpenLDAP/schema/nis.schema"
include "C:/Program Files/OpenLDAP/schema/corba.schema"
include "C:/Program Files/OpenLDAP/schema/java.schema"
include "C:/Program Files/OpenLDAP/schema/krb5-kdc.schema"
include "C:/Program Files/OpenLDAP/schema/openldap.schema"

Check whether the OpenLDAP server is started. If it has not started, initiate the startup process.

## Configuring WebSphere MQ

### Prerequisite: It is assumed that WebSphere MQ is installed and available for queuing operations.

Configure the necessary WebSphere MQ objects, as follows:

1. Create a queue manager (parameter P5: QMLDAP).
2. Create a local queue (parameter P6: Q1).
3. Note the listening port (parameter P7: 1416).
4. Note the default server connection channel (parameter P8: SYSTEM.DEF.SVR CONN).
5. Note the IP address of the WebSphere MQ server (parameter P9: 10.11.24.75).

Figure 4 shows that the Queue Manager is active.
Figure 4. Queue Manager is active

6. Ensure that the Queue Manager and the Listener are active and in running mode.

Figure 5 shows that the Listener is running.
Configuring the JMSAdmin utility

The JMSAdmin utility is made available as part of the WebSphere MQ installation. Perform the following steps to register and bind the services to the LDAP with the help of the JMSAdmin utility so that the queuing parameters are available in the LDAP naming service as a lookup object and can be located to establish the queue connection.

1. Navigate to `<Install directory of WebSphere MQ>/java/bin`.
2. Create a config file (for example, LdapConfigFile.conf) with the following entries:

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
PROVIDER_URL=ldap://10.11.24.75:389/dc=mycompany,dc=com
PROVIDER_USERDN=cn=Manager,dc=mycompany,dc=com
SECURITY_AUTHENTICATION=simple
PROVIDER_PASSWORD=password
```

Where:
- PROVIDER_URL contains the value of the parameter P2 followed by P1.
- PROVIDER_USERDN contains the value of the parameter P3.
- PROVIDER_PASSWORD contains the value of the parameter P4.

Alternatively, you can open the JMSAdmin.config file and alter the content of this file to match the entries that are shown.
3. From the command prompt, run the JMSAdmin.bat file with the following attributes or parameter:

```
JMSAdmin.bat -v -cfg <configfile Name say, LdapConfigFile.conf >
```

ex: JMSAdmin.bat -v -cfg LdapConfigFile.conf

The prompt changes to **InitCtx>**.

4. Use the following commands to bind the queue connection factory and the queue to their logical LDAP lookup names:

```
def qcf(qm) qmgr(QMLDAP) tran(client) chan(SYSTEM.DEF.SVR CONN) host(10.11.24.75)+
      port(1416)
define q(q) qmgr(QMLDAP) queue(Q1)
```

Where:

- `qm` is the logical name that is given to the queue connection factory. However, the object is bound to the context as `cn=qm` and is referred as `cn=qm` (parameter **P10**).
- `q` is the logical name that is given to the queue. However, the object is bound to the context as `cn=q` and is referred as `cn=q` (parameter **P11**).
- `QMLDAP` is the value of the parameter **P5**.
- `Q1` is the value of the parameter **P6**.
- `SYSTEM.DEF.SVR CONN` is the value of the parameter **P8**.
- `10.11.24.75` is the value of the parameter **P9**.
- `1416` is the value of the parameter **P7**.

5. Finally, enter the following command to check whether the objects are bound to the context:

```
> display ctx
```

Refer to the screen shown in **Figure 6**, where all the objects bound to its context are listed.
Now that the necessary objects are bound and are available in the LDAP directory service to look up and discover these objects, they can be looked up from any application service and can be used for further connection and integration with them.

**Configuring the JMS queue adapter in Sterling B2B Integrator**

Configure the JMS queue adapter in Sterling B2B Integrator V5.2.4 with the appropriate LDAP connection information and remote queuing logical names that are bound to the LDAP.

1. Navigate to the B2B Integrator Services Configuration section.
2. Choose JMS queue adapter as the service type, and configure it as shown in the following example:

   ```
   Initial Context Factory: com.sun.jndi.ldap.LdapCtxFactory
   URL: ldap://10.11.24.75:389/dc=mycompany,dc=com
   Remote Queue Name: cn=q
   Remote Queue Connection Factory: cn=qm
   Remote user name: cn=Manager,dc=mycompany,dc=com
   Remote Password: password
   Do turn on debug messages? yes
   Queue Type: choose Queue Send
   ```
Where:

- URL is the same as the PROVIDER_URL, with the value of parameter P2 followed by P1.
- Remote Queue Name contains the value of the parameter P11.
- Remote Queue Connection Factory contains the value of the parameter P10.
- Remote user name contains the value of the parameter P3.
- Remote Password contains the value of the parameter P4.

Figure 7 shows the JMS adapter configuration summary.

**Figure 7. JMS adapter configuration summary**

![JMS adapter configuration summary](image)

3. Save the service settings.
4. Refresh the service, for the backend changes to get reflected.
5. After the service is configured, check the advanced status.
6. When the service is in enabled status, it is available for use with the Business Process.
7. Test the connectivity by sending files or data to the JMS adapter by running the configured workflow.
8. Connect to the WebSphere MQ with the WebSphere MQ Explorer client, and browse the queue to verify that the messages reached the appropriate queue.

**Validation**

When the service configuration is enabled, the JMS queue adapter can be used as a participant in the Business Process Definition.

**Example business process**

**Listing 1. Sample business process to test the JMS queue adapter**

```xml
<process name="JMSQueueSender">
  <sequence>
    <operation name="JMS Queue Adapter">
      <participant name="JMSQAdapter"/>
      <output message="JMSQAdapter_InMsg">
        <assign to="." from="*"/></assign>
      </output>
    </operation>
    <input message="inmsg">
    </input>
  </operation>
  </sequence>
</process>
```

The original invoice processing workflow can now be modified to call the JMSQueueSender Business Process (BP). The BP delivers the processed invoice document to the configured queue in WebSphere MQ for further processing by the administrative unit for audit.

**Table 1** provides a mapping of all the parameters and referenced values across all the systems.

**Table 1. Mapping of parameters and referenced values across all systems**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>LDAP Server Suffix</td>
<td>dc=mycompany,dc=com</td>
</tr>
<tr>
<td>P2</td>
<td>LDAP Server Binding</td>
<td>ldap://10.11.24.75:389</td>
</tr>
<tr>
<td>P3</td>
<td>Directory Manager DN</td>
<td>cn=Manager,dc=mycompany,dc=com</td>
</tr>
<tr>
<td>P4</td>
<td>LDAP Password</td>
<td>Password</td>
</tr>
<tr>
<td>P5</td>
<td>Queue Manager Name</td>
<td>QMLLDAP</td>
</tr>
<tr>
<td>P6</td>
<td>Local Queue Name</td>
<td>Q1</td>
</tr>
<tr>
<td>P7</td>
<td>Listening Port</td>
<td>1416</td>
</tr>
<tr>
<td>----</td>
<td>----------------</td>
<td>------</td>
</tr>
<tr>
<td>P8</td>
<td>Server Connection channel</td>
<td>SYSTEM.DEF.SVR.CONN</td>
</tr>
<tr>
<td>P9</td>
<td>IP address of the WebSphere MQ Server</td>
<td>10.11.24.75</td>
</tr>
<tr>
<td>P10</td>
<td>Logical JNDI name of queue connection factory</td>
<td>cn=qm</td>
</tr>
<tr>
<td>P11</td>
<td>Logical JNDI name of queue</td>
<td>cn=q</td>
</tr>
</tbody>
</table>

## Conclusion

This article describes the enterprise application integration with LDAP as a naming service. It illustrates a scenario, integrating the OpenLDAP server as a naming service to discover the remote queuing parameters for the transaction of B2B documents from Sterling B2B Integrator's JMS queue adapter to WebSphere MQ's messaging system. The scope of this article covers the JMS queue adapter. To configure the messaging infrastructure to use a topic instead of a queue, use the corresponding topic name and topic connection factory details instead of the queue name and queue connection details in this illustration.

© Copyright IBM Corporation 2014

**Trademarks**
(www.ibm.com/developerworks/ibm/trademarks/)