Storing transaction and compensation logs in a relational database for high availability and disaster recovery in IBM Business Process Manager

Hong Yan Wang
Chris Richardson
Yu Zhang
Hang Yin

Learn how to leverage the latest WebSphere® Application Server feature by configuring an IBM® Business Process Manager transaction and compensation logs into a relational database for high availability and disaster recovery purposes.

Introduction

Since IBM Business Process Manager (BPM) V8.0.1.2, V8.5.0.1, and above, it’s possible to configure WebSphere transaction logs and compensation logs into a relational database. This configuration can eliminate the requirement of a shared file system for high availability purposes and eliminate the requirement of a consistency group offered by storage subsystem for disaster recovery. In this tutorial, we will demonstrate how to configure IBM BPM to use this feature.

Background

In previous tutorials, Using Oracle Database File System for disaster recovery on IBM Business Process Manager and Faster disaster recovery in IBM Business Process Manager, we demonstrated how to configure WebSphere transaction and compensation logs into a database file system (DBFS) for transaction high availability and disaster recovery. You can now eliminate the prerequisite of DBFS because IBM Business Process Manager (BPM) V8.0.1.2 and V8.5.0.1 now allow transaction logs and compensation logs to be stored in a relational database instead of the default option in operating system files. This new feature allows customers, particularly those with an investment in database high availability (HA) and disaster recovery (DR) technologies, to purely rely on database level replication to achieve HA and DR objectives.

In order to leverage this feature, you must configure the transaction log location and the compensation log location for each server in the cluster before enabling high availability by setting the TransactionLogDirectory and CompensationLogDirectory attributes for each server. Each server in a cluster must reference unique transaction log and compensation log locations.
specifying a distinct tablesuffix property for each, multiple servers do not contend for relational
database management system (RDBMS) resources. Furthermore, each server in the cluster must
reference the same data source to enable peer recovery.

**Tips:**

- A compensation log is required by *IBM BPM Advanced Edition* only. There is no need to
  enable it on *IBM BPM Standard Edition*.
- In a typical three-cluster topology, transaction logs should be configured on both Application
  and Support cluster's members. The compensation log should be configured on Application
  cluster members.
- Transaction and compensation tables will be created automatically during server startup.
- There is no need to configure transaction and compensation logs for messaging engine
  cluster members because the messaging engine does not run any business processes.

**Performance impact**

Before reading the detailed steps, keep in mind that comparing with the previous option of "storing
transaction and compensation logs into file system", the new approach may introduce additional
overhead for some workloads. Therefore, conduct a performance test to evaluate whether
the transaction throughput and response time can meet business requirements when storing
transaction and compensation logs into the database.

**Implementation details**

The procedure consists of the following steps:

1. **Configure non-transactional data sources for transaction and compensation recovery logs.**
2. **Configure transaction and compensation logs into the database.**
3. **Enable cluster high availability.**
4. **Restart the deployment environment and verify.**

The demonstration environment we used to illustrate this procedure is shown in Table 1.

**Table 1. Demo environment**

<table>
<thead>
<tr>
<th>Product</th>
<th>Detail</th>
<th>Machine</th>
</tr>
</thead>
</table>
| BPM     | IBM BPM 8.5.5 Advanced PS 3 cluster:  
  • AppCluster  
  • AppClusterMember1  
  • AppClusterMember2  
  • SupCluster  
  • SupClusterMember1  
  • SupClusterMember2  
  • MECcluster  
  • MECClusterMember1  
  • MECClusterMember2 | 9.111.125.111 - dmgr  
9.111.125.111 - Node1  
9.111.125.112 - Node2 |
| DB      | IBM DB2® 10.5 | 9.115.196.246 |
Configuring non-transactional data sources for transaction and compensation recovery log storage

In order to store transaction and compensation recovery logs in a database, you need to create non-transactional data sources first. Those data sources will be used by the WebSphere Transaction Manager to store transaction information.

1. Create the connection pool JDBC provider. You can skip this step if the provider exists. Create a JDBC provider for your specific RDBMS implementation. Specify an implementation type of "non-XA".
2. Select Resources > JDBC > JDBC providers.
3. From the Scope list, select Cluster=AppCluster and then click New.
4. Specify the following, then click Next:
   a. Database type: <your_database_type> (for example, DB2)
   b. Provider type: <your_database_provider_type> (for example, DB2 Using IBM JCC Provider)
   c. Implementation type: <your_implementation_name> (for example, Connection pool data source)
   d. Name: <your_provider_name> (for example, DB2 Using IBM JCC Driver)
   The sample is shown in Figure 1.
5. In the "Enter database class path information" page, use the default required values. Click Next.
6. In the "Summary" page, click Finish.
7. Repeat the above step to create the corresponding JDBC provider in the support cluster.
8. Create a J2C authentication data alias. You can skip this step if the proper authentication data alias exists. Create a JAAS J2C authentication data alias. This defines the security credential that is used to connect to the database. The credential defined here must have sufficient privileges to create tables in the database. Otherwise, you have to manually create the tables. The corresponding DDL can be found in this Technote.
9. Select Security > Global security. In the Authentication area, select Java Authentication and Authorization Service, click J2C authentication data (Figure 2), and then click New. The sample is shown in Figure 2.
10. Specify the following, then click OK:
   a. **Alias**: `<your_alias>` (for example, BPM_TX_DB_ALIAS)
   b. **User ID**: `<your_user_ID>` (for example, db2inst1)
   c. **Password**: `<your_password>` (for example, test)
11. Create new data sources by using the JDBC provider created in Step 1. Its component managed authentication alias must be set to the JAAS alias created in Step 2.
12. Select Resources > JDBC > Data sources.
13. From the Scope list, select Cluster=AppCluster, then click New.
14. Specify the following, then click Next:
   a. **Data source name**: `<your_dataSource_name>` (for example, tranlog)
   b. **JNDI name**: `<your_JNDI_name>` (for example, jdbc/tranlog). The sample is shown in Figure 3.
15. In the "Select JDBC Provider" page, choose **select an existing JDBC provider** from the list, select the JDBC provider created in Step 2, then click **Next**. The sample is shown in Figure 4.

**Figure 4. Select JDBC provider**

16. In the "Enter database specific properties for the data source" page, enter your database information, then click **Next**:
   a. **Database name:** `<your_database_name>` (for example, CMNDBf)
   b. **Port number:** `<your_port_number>` (for example, 50000)
   c. **Server name:** `<your_server_name>` (for example, 9.115.196.246). The sample is shown in Figure 5.
17. In the "Setup security aliases" page, select the JAAS J2C authentication data alias created in Step 2, then click **Next**. The sample is shown in Figure 6.

**Figure 6. Setup security aliases**

18. In the "Summary" page, review all the information and then click **Finish**. The sample is shown in Figure 7.

**Figure 5. Enter database properties**
19. Repeat the above step to create another JDBC data source in the support cluster scope.
20. Configure the new data sources to be non-transactional by completing the following steps:
   a. Open your newly created data source, **Resources > JDBC > Data sources > your_data_source** (for example, tranlog on AppCluster).
   b. Under **Additional Properties**, click **WebSphere Application Server data source properties**.
   c. Select the **Non-transactional data source** check box. The sample is shown in Figure 8.
   d. Save your changes.

![Figure 7. Summary](image-url)
21. Repeat the above step to update the configuration of the JDBC data source in the support cluster scope.

Configuring transaction and compensation logs into the database

This includes two parts:

- Configure the application cluster and support cluster members' transaction logs into the database.
- Configure the application cluster members' compensation logs into the database.

1. In the WebSphere Application Server console, select one cluster member and then configure its transaction log locations to use the data source created in the previous steps.
2. Click Servers > Server Types > WebSphere application servers > AppClusterMember1.
3. In the Container Settings section, select Container Services > Transaction service. Select the Configuration tab. Then set the Transaction log directory to the following:
   `custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=your_data_source_jndi_name
tablesuffix=your_suffix`

   The sample is shown in Figure 9.

   **Note:** "data_source_jndi_name" is the JNDI name of the non-transactional data source created previously. The `tablesuffix` field is a string that you must set to uniquely label each member of your HA cluster.
Restriction: If you are using an Oracle® database, the length of the `tablesuffix` string must not exceed 15 characters.

Figure 9. Configure transaction log into the database

4. Repeat the above steps for the remaining application cluster members and support cluster members.

5. Configure the application cluster members' compensation log locations into the database *(IBM BPM Advanced Edition only)*.

6. Click **Servers > Server Types > WebSphere application servers > your_server_name**.

7. In the **Container Settings** section, select **Container Services > Compensation service**. Select the **Configuration** tab. Then set the **Recovery log directory** to:

   ```
   custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=your_data_source_jndi_name,
   tablesuffix=your_suffix
   ```

   For example:

   ```
   custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=jdbc/tranlog,tablesuffix=App1
   ```

   The sample is shown in Figure 10.

   **Note:** "data_source_jndi_name" is the JNDI name of the non-transactional data source created previously. The `tablesuffix` field is a string that you must set to uniquely label each member of your HA cluster.

   **Restriction:** If you are using an Oracle® database, the length of the `tablesuffix` string must not exceed 15 characters.
8. Repeat the above steps for the remaining application cluster members.

Table 2 and Table 3 show the final configuration for all the cluster members.

### Table 2. Demo environment's transaction log location settings

<table>
<thead>
<tr>
<th>Cluster name</th>
<th>Server name</th>
<th>Transaction log location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SupCluster</td>
<td>SupClusterMember1</td>
<td>custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=jdbc/tranlog,tablesuffix=Sup1</td>
</tr>
<tr>
<td>SupCluster</td>
<td>SupClusterMember2</td>
<td>custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=jdbc/tranlog,tablesuffix=Sup2</td>
</tr>
<tr>
<td>AppCluster</td>
<td>AppClusterMember1</td>
<td>custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=jdbc/tranlog,tablesuffix=App1</td>
</tr>
</tbody>
</table>

### Table 2. Demo environment's compensation log location settings

<table>
<thead>
<tr>
<th>Cluster name</th>
<th>Server name</th>
<th>Compensation log location</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppCluster</td>
<td>AppClusterMember1</td>
<td>custom://com.ibm.rls.jdbc.SQLRecoveryLog?datasource=jdbc/tranlog,tablesuffix=App1</td>
</tr>
</tbody>
</table>
Storing transaction and compensation logs in a relational database for high availability and disaster recovery in IBM Business Process Manager

<table>
<thead>
<tr>
<th>AppCluster</th>
<th>AppClusterMember2</th>
<th>datasource=jdbc/ tranlog,tablesuffix=App1</th>
<th>datasource=jdbc/ tranlog,tablesuffix=App2</th>
</tr>
</thead>
</table>

**Note:** Since the tables created for the transaction and compensation log locations have different prefixes, you can use the same "tablesuffix" for both transaction and compensation log locations in one specific server.

### Enabling cluster high availability

After setting each server's transaction and compensation log locations to use the database, you can enable the cluster's high availability.

For the Application cluster and Support cluster, repeat the following steps:

1. Click **Servers > Clusters > WebSphere application server clusters > your_cluster_name** (for example, AppCluster).
2. Check the **Enable failover of transaction log recovery** check box. The sample is shown in Figure 11.
3. Click **OK**.

![Figure 11. Enable cluster HA](image)

### Restarting and verifying the deployment environment

In the WebSphere admin console, restart the deployment environment. During server startup, the WebSphere transaction service will try to create the related tables in the specified database. On the database side, you can see the following newly created tables:
Tranlog placement and schema considerations

In order to ensure data integrity, we recommend that you put the transaction and compensation logs into the database where the main workload is running. For instance, the IBM BPM database should be used for BPD type workloads and the Common database should be used for BPEL type workloads.

Regarding the schema for the transaction and compensation data source, it's fine to either reuse the existing schema or use a new one. The only requirement is to make sure the schema has enough privileges.

Limitations

As stated in the WebSphere Application Server technote, the current implementation for storing transaction and compensation logs in a relational database does not ensure support for any database-specific features that the database clients (in this case, WebSphere Application Server Transaction Service) might use to enhance HA capabilities. For example, client-specific features such as DB2 HADR or Oracle RAC DataGuard that allow a reconnection to another database instance, in the event of a failure, might not be supported. The consequence of the limitation is that IBM BPM has to be restarted after the database has been successfully failed over to a standby database. This limitation will be removed in future IBM BPM releases.

Conclusion

This tutorial provided instructions to help you configure IBM BPM transaction and compensation logs into a relational database for high availability and disaster recovery purposes.

Acknowledgments

The authors would like to thank Karri S. Carlson-Newmann, Eric Herness, and Neil Young for reviewing this tutorial and providing valuable suggestions and comments.
Related topics

- Technote: WebSphere Application Server: Storing transaction and compensation logs in a relational database for high availability
- IBM WebSphere Business Process Management V8.5 Information Center
- Faster disaster recovery in IBM Business Process Manager
- Using Oracle Database File System for disaster recovery on IBM Business Process Manager

© Copyright IBM Corporation 2014
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)