Oracle JCA Adapter for Coherence

Compatibilty

When using a version of the Coherence Adapter, you are not able to interact with a remote Coherence cluster other than the same version or a newer version. For example, when using the Coherence Adapter in 12.1.3, you would not be able to interact with a remote Coherence cluster other than 12.1.3 or a newer version. This is due to a Coherence product limitation on the use of the Coherence extend client with older versions of the extend proxy.

Specifically, regarding compatibility between a Coherence server and an extend client, only forward compatibility is maintained from extend clients to cluster proxy servers. That is, an extend client can connect to cluster servers that have either the same or higher version numbers, but not the lower version numbers.

Oracle Coherence Adapter Features

You can use the Oracle Coherence Adapter to perform the following activities associated with Oracle Coherence:

- Add a Cache Entry: Create a new entry in the Coherence Cache.
- Remove Cache entries: Identify an item to be removed from the Cache, and the system removes the entry from the Cache. You can also remove multiple entries from the cache by providing a filter or search criteria which match the multiple records in cache.
- Get Cache entry value: After specifying an entry to obtain the associated value, the system returns the value of that entry to you.
- Query Cache: After you identify the Cache, and specify search criteria, the system returns the entries that match the search criteria.

Basic Use Cases

There are two basic use cases for the Coherence Adapter.

- A Coherence Adapter connecting to a local cluster. This configuration supports transactional caches.
- A Coherence Adapter connecting to a standalone Coherence cluster or a WebLogic Server. This configuration does not support transactional caches.

Configuring the Coherence Adapter Connection to a Remote Cluster

You can use the Coherence Adapter to access remote caches.

The name of the remote cache is captured in a configuration file; in the following example, the file is called extend-config.xml file. The location of this configuration file needs to be specified as the value for property CacheConfigLocation. In addition to containing the cache name, the configuration file also requires information to connect to a remote Coherence cluster.
The access to remote caches is enabled using <remote-cache-scheme>.
Note that the ManagedConnectionFactory example here also points to a location for the PojoJarFile.

Example - ManagedConnectionFactory for a Remote Cluster

```
<connection-instance>
<jndi-name>eis/Coherence/Remote</jndi-name>
  <connection-properties>
   <properties>
     <property>
       <name>CacheConfigLocation</name>
        <value>/scratch/amahajan/Temp/coherence/dhqa/extend-
config.xml</value>
     </property>
   <property>
     <name>ClassLoaderMode</name>
     <value>CUSTOM</value>
   </property>
   <property>
     <name>PojoJarFile</name>
     <value>/scratch/amahajan/Temp/coherence/dhqa/book.jar</value>
 </property>
 <property>
   <name>WLSExtendProxy</name>
   <value>false</value>
</property>
</properties>
</connection-properties>
</connection-instance>
```

Example - Sample extend-config to Define a Remote Cache
In this example, the extend configuration enables the adapter to connect to an extend proxy running at address 10.240.82.123 and listening on port 14777.

```
<?xml version="1.0"?>

<!DOCTYPE cache-config SYSTEM "cache-config.dtd">
```

```
<cache-config>
   <caching-scheme-mapping>
      <cache-mapping>
         <cache-name>samples-cache</cache-name>
         <scheme-name>extend-dist</scheme-name>
      </cache-mapping>
      <cache-mapping>
         <cache-name>samples-cache-binxml</cache-name>
         <scheme-name>extend-dist</scheme-name>
      </cache-mapping>
    </caching-scheme-mapping>
    <caching-schemes>
   <remote-cache-scheme>
      <scheme-name>extend-dist</scheme-name>
      <service-name>ExtendTcpCacheService</service-name>
      <initiator-config>
        <tcp-initiator>
          <remote-addresses>
            <socket-address>
              <address>10.240.82.123</address>
              <port>14777</port>
            </socket-address>
          </remote-addresses>
          <connect-timeout>10s</connect-timeout>
        </tcp-initiator>
        <outgoing-message-handler>
          <request-timeout>5s</request-timeout>
        </outgoing-message-handler>
      </initiator-config>
    </remote-cache-scheme>
   </caching-schemes>
</cache-config>
```

Coherence Adapter Connection to Local Cluster

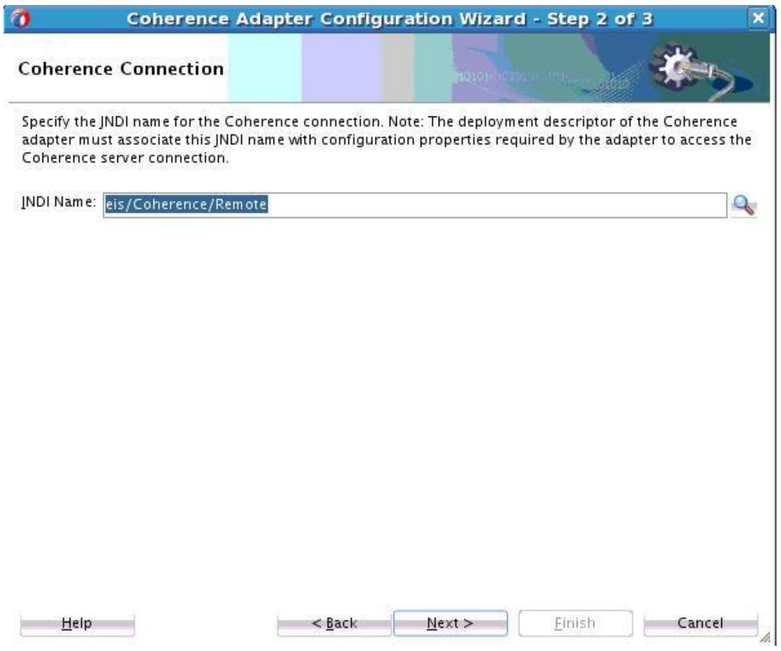
The Coherence Adapter provides a default connection factory to connect to an out-of-box Coherence cache and also a cache called adapter-local.
Coherence adapter employs transactions for access to local caches (local coherence cluster). So the supported scheme for local caches is <transactional-scheme>

Configuring the Coherence Adapter

Learn how to configure the Coherence Adapter using the Coherence Adapter Configuration Wizard.

1. In the **Application Navigator** of JDeveloper, select **New Application**. The **Create Generic Application - Name your application** page is displayed.
2. Drag and drop the Coherence Adapter from the Components window of Oracle JDeveloper. The **Coherence Adapter Welcome Screen** appears. Click Next.
3. On the Coherence Connection screen, specify a JNDI name for the Coherence Connection. Select **Next**.

   Figure 14-1 The Coherence Adapter Configuration Wizard Service Name Screen

   Coherence Adapter Configuration Wizard - Step 2 of 3

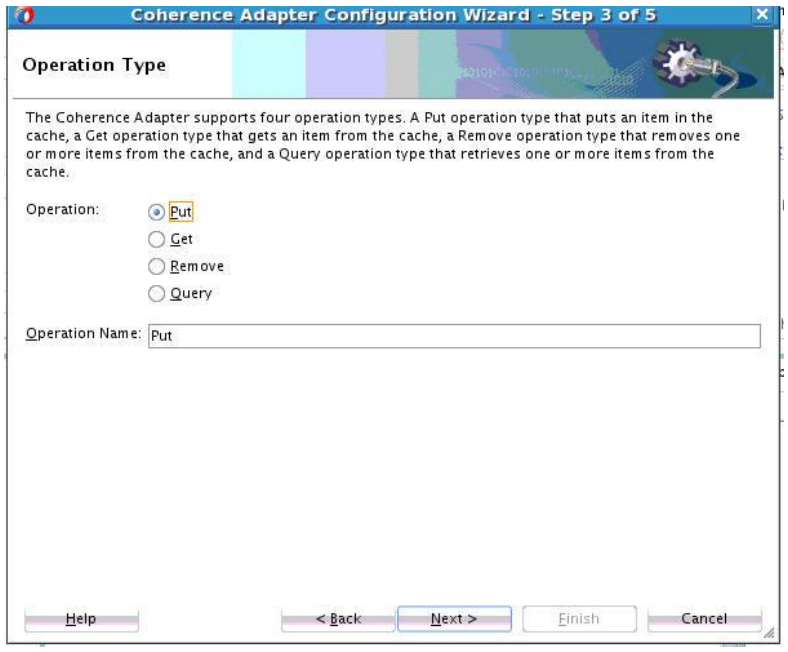
   Coherence Connection

   Specify the JNDI name for the Coherence connection. Note: The deployment descriptor of the Coherence adapter must associate this JNDI name with configuration properties required by the adapter to access the Coherence server connection.

   JNDI Name: eis/Coherence/Remote

4. On the **Operation Type** Page, specify a valid operation against the Coherence cache that you want to perform. When you import an existing WSDL, the operation name is pre-populated. Operations are **Put (Put item in cache)**, **Get (Get item from cache)**, **Remove (Remove item from cache**) or **Query (Query item from cache)**. In the screen example below, **Put** is selected.

   Figure 14-2 Coherence Adapter Configuration Wizard Operation Type Page with Put Selected



Coherence Adapter Configuration Wizard - Step 3 of 5

Operation Type

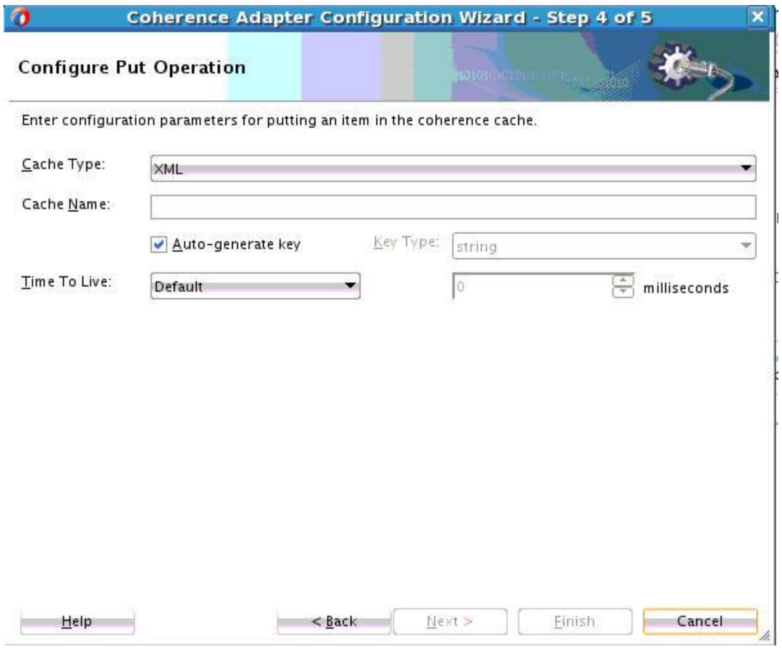
The Coherence Adapter supports four operation types. A Put operation type that puts an item in the cache, a Get operation type that gets an item from the cache, a Remove operation type that removes one or more items from the cache, and a Query operation type that retrieves one or more items from the cache.

Operation: [✓ Put] Get Remove Query

Operation Name: Put

5. If you selected Put, the **Put Page** appears. This page captures the configuration parameters for the Put operation. On this screen, you indicate the following fields:
   - **Cache Type** – This is a drop-down combo-box with values – XML, POJO. Use the value that corresponds to the type of item you want to put into the cache.
   - **Cache Name** – Enter a cache name in this text field. This is the name that uniquely identifies the Coherence cache.
   - **Key** – Either enter a key in the text field, or check the auto-generate checkbox to have the key generated. If checked, the key is automatically generated by the Coherence runtime. The key auto-generate process sets the Key Type to String.
   - **Key Type** – Key Type and Key are disabled, if you choose to enter a filter It is enabled if you choose to enter a Key Type and Key. Here Key Type combo-box with list of Java simple types: string, integer, long, float, double.
   - **Auto-generate Key** – Check this box if you want the Coherence Adapter Configuration Wizard to generate a key for you.
   - **Time to Live** – Choose Default, Always, or Custom. If you choose Custom, you can specify a value in milliseconds. This value indicates how long an entry should remain in the Coherence cache. The default is that the message never expires. The Time To Live property is applicable more for Remote Caches than for Local caches. For a Local cache, the entry always remains in the cache until you remove it or the SOA server terminates.

Figure 14-3 The Coherence Adapter Configuration Wizard Configure Put Operation Page



Enter configuration parameters for putting an item in the coherence cache.

Cache Type: XML

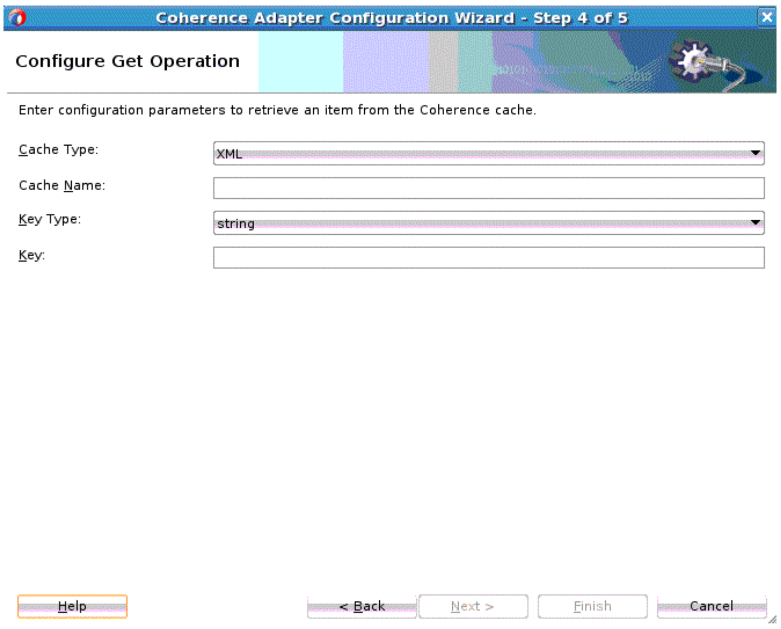
Cache Name:

[✓] Auto-generate key   Key Type: string

Time To Live: Default   0 milliseconds

6. If you had specified the **Get item from cache** operation on the Coherence Adapter Operation Type page, the Configure Get Operation screen appears:

Figure 14-4 The Coherence Adapter Configure Get Operation Page



Enter configuration parameters to retrieve an item from the Coherence cache.

Cache Type: XML

Cache Name:

Key Type: string

Key:

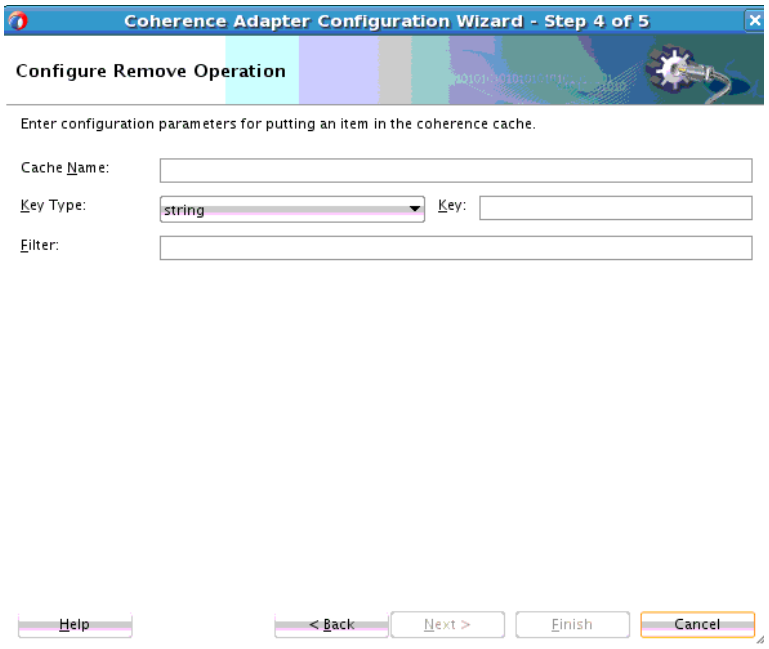
Fill in the following fields to retrieve items from cache:

- **Cache Type** – This is a drop-down combo-box with values – XML, POJO. Choose the value that corresponds to the type of entry you want to retrieve.
- **Cache Name** – Enter a cache name from where the item is to be retrieved.

- **Key Type** – This field is always enabled for a Get operation.
- **Key** – Enter the key of the cache entry in this text field.

7. If you had chosen the **Remove from cache** option on the Operation Type screen, the Remove Item from Cache screen would appear:

Figure 14-5 The Coherence Adapter Configuration Configure Remove Operation Screen


Coherence Adapter Configuration Wizard - Step 4 of 5

Configure Remove Operation

Enter configuration parameters for putting an item in the coherence cache.

Cache Name:

Key Type: string

Key:

Filter:

Enter the configuration parameters to remove an item from Cache, and click Next:

- **Cache Name** –The name of the cache item.
- **Key Type** – This is enabled when performing a Remove operation.
- **Key** – The key for the cache item.
- **Filter** – A string filter for the cache name.You can specify a key or a filter but you cannot specify both.

Note:

When using a filter for a Remove operation, the Coherence Adapter does not report the count of entries affected by the remove operation, regardless of whether the remove operation is successful.

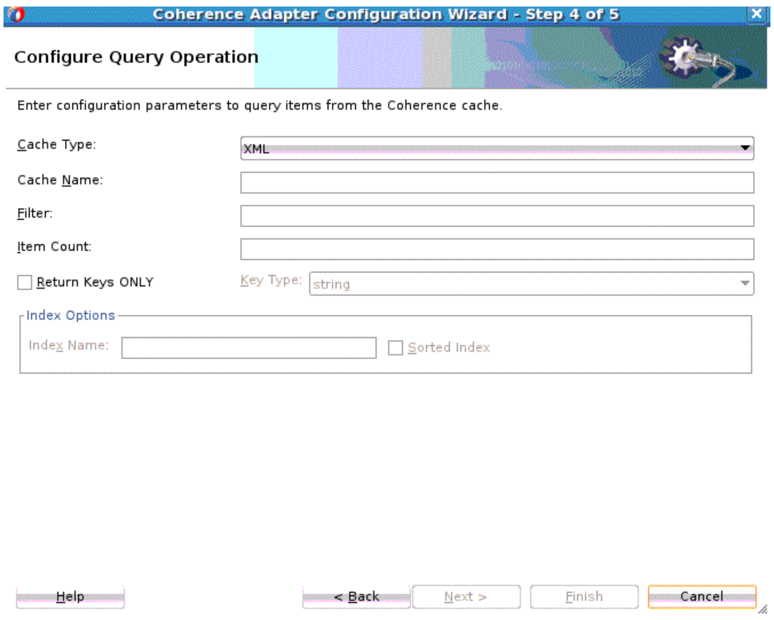
When using a key to remove a specific entry, the Coherence Adapter does report the count, which is always 1 if a Coherence Remove operation is successful.

Querying Items in the Coherence Cache

Learn how to query items in the Coherence cache.

1. To do, select **Query** from the Operations Screen. The Coherence Adapter Configure Query Operation screen appears.

   Figure 14-6 The Coherence Adapter Configure Query Operation Screen


Coherence Adapter Configuration Wizard - Step 4 of 5

Configure Query Operation

Enter configuration parameters to query items from the Coherence cache.

Cache Type: XML

Cache Name:

Filter:

Item Count:

Return Keys ONLY   Key Type: string

Index Options

Index Name: Sorted Index

2. On the Configure Query Operation screen fill in the following:
   - **Cache Type** - Select XML or POJO from the dropdown list.
   - **Cache Name** -The name of the cache.
   - **Filter** - You can enter a Coherence Query Language filter expression manually. If you do not specify a Filter, the Coherence Adapter Configuration Wizard warns that all items are to be returned.
   - **Item Count** - An integer to specify the limit to the item count returned from the query.
   - **Index Name** - (Optional) A index name to be created in the cache. Select the Sorted checkbox to create a sorted index.
   - **Return Keys ONLY** - Check the box and supply a return key type. If this box is checked, only the keys will be returned whose values match the entries that are returned from the query.
     Index and Ordered field key types are only valid for the POJO Cache Type. For the XML Cache Type, you can only use key() token in the filter expression.

     This means that when you specify a filter expression you can only use the special token key(), which means the cache object key. For example, you could use either of these keys:

     ```
     key() = 1234
     key() = 5678
     ```

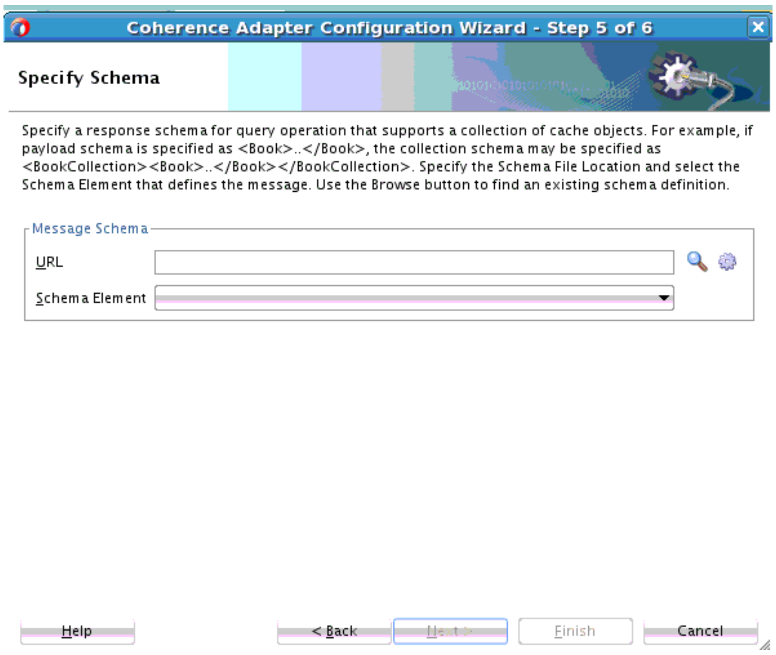
If you are using the POJO cache type, you can create a filter expression that uses both tokens from the POJO object as well as the special built-in key() token.

Defining Messages for Put, Get and Query Operations if XML is Chosen

Learn how to define messages for Put, Get and Query Operations if you have chosen XML as the cache type.

If you have chosen XML as the cache type on any of the Put, Get or Query operations, the Specify Schema Page appears. On this page, you select a schema for the Coherence cache object. See Figure 14-7.

Figure 14-7 Coherence Adapter Configuration Wizard Messages (Specify Schema) Screen



Specify a response schema for query operation that supports a collection of cache objects. For example, if payload schema is specified as <Book>..</Book>, the collection schema may be specified as <BookCollection><Book>..</Book></BookCollection>. Specify the Schema File Location and select the Schema Element that defines the message. Use the Browse button to find an existing schema definition.

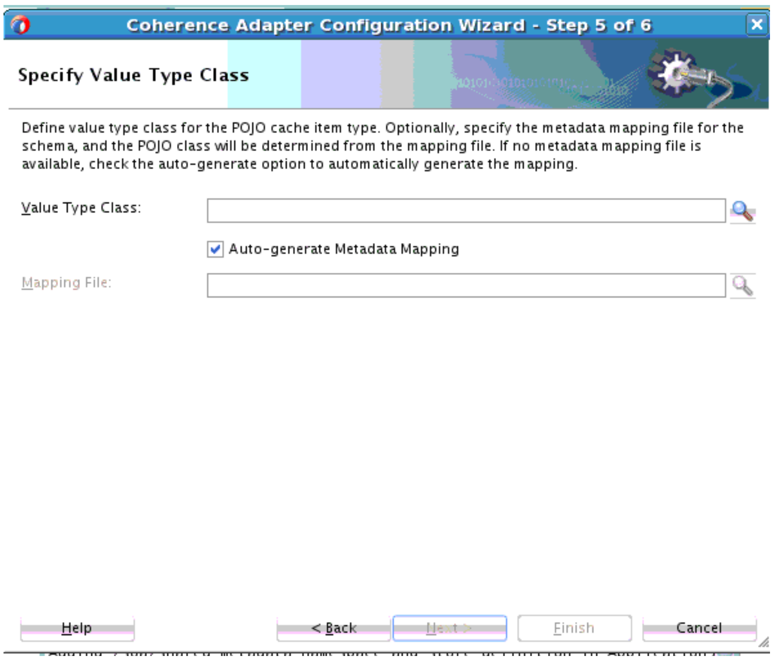
1. Specify the Schema File Location in the **URL** field and select the Schema element that defines the elements in the incoming files. Use the Browse button to find an existing schema definition.
2. Some considerations that are applicable to your choices at this point:
   - For a Put operation, a request message is generated that corresponds to schema generated for the value type specified, or to a schema you supplied on the Schema Page. A response message is with the ReturnIdentifier of the cache entry created in the Coherence server.
   - For Get operation, the response message points to the schema for the value type, or the schema you supplied on the Schema Page. An empty request message is created.

- For Query operation, if the filter expression has bind variables, a request message containing elements for these bind variables will be generated. The response message will be the schema generated for the value type, or the schema you supplied on the Schema Page.

Defining Messages for Put, Get and Query Operations if Pojo is Chosen

If you choose POJO as the cache type for a Get, Put, or Query operation, the **Specify Value Type Class Screen** appears.

Figure 14-8 Coherence Adapter Configuration Wizard Value Type Class Screen



Coherence Adapter Configuration Wizard - Step 5 of 6

Specify Value Type Class

Define value type class for the POJO cache item type. Optionally, specify the metadata mapping file for the schema, and the POJO class will be determined from the mapping file. If no metadata mapping file is available, check the auto-generate option to automatically generate the mapping.

Value Type Class:

[✓] Auto-generate Metadata Mapping

Mapping File:

To define messages when you have specified POJO as the cache type:

1. Enter the value type for the POJO cache type for the PUT operation in the **Value Type** box. You can use the browser to find a value type.
2. Optionally, enter the metadata mapping file for the schema in the **Metadata Mapping** field.

   The Metadata mapping file helps in POJO to XML and XML to POJO conversion. The mapping file you selected is copied to the JDeveloper project. You can use the browser to find the metadata mapping file. The mapping file is stored in the JDeveloper project. (When you provide XML, the Adapter converts the XML to an object before adding it to the cache. The reverse happens when a you query an object in the cache. The Adapter converts the object stored in cache to XML and passes it along as output to you.)

If a mapping file is specified, the Coherence Adapter automatically generates the schema.

If you do not specify a metadata file, the Coherence Adapter Configuration wizard automatically generates the mapping file by introspecting the value type class.

Note:

: For a Query Operation type, if you select Return Keys, the Messages page is not displayed for either XML or POJO Cache Types.

Coherence Adapter Files and Artifacts

This section provides examples of Coherence Adapter design-time artifacts.

JCA File

The JCA file stores JCA property values for each of the supported operations.

Example - JCA File Created

```
<adapter-config name="cohPut1" adapter="Coherence Adapter"
wsdlLocation="../WSDLs/cohPut1.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
      <connection-factory location="eis/coherence"/>
       <endpoint-interaction portType="Put_ptt" operation="Put">
    <interaction-spec className=
"oracle.tip.adapter.coherence.CoherenceInteractionSpec">
   <property name="CacheName" value="TestCache"/>
      <property name="Key" value="ABC12345678"/>
      <property name="KeyType" value="String"/>
      <property name="ValueType" value="com.coherence.vt.Book"/>
      <property name="TimeToLive" value="60"/>
      <property name="MappingsMetadataFile" value="book-oxm-mappings.xml"/>
    </interaction-spec>
  </endpoint-interaction>
</adapter-config>
```

WSDL for Put Operation

The following example shows the WSDL for the Coherence Adapter Put Operation

Example - WSDL for Coherence Adapter Put Operation

```
<wsdl:definitions name="cohPut1" targetNamespace=
"http://xmlns.oracle.com/pcbpel/adapter/coherence/Application1/Project1/coh
Put1"
     xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:tns=http://xmlns.oracle.com/pcbpel/adapter/coherence/Application1/
                 Project1/cohPut1
     xmlns:imp1=" http://xmlns.oracle.com/pcbpel/adapter/coherence"
     xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/">
    <plt:partnerLinkType name="Put_plt">
        <plt:role name="Put_role">
            <plt:portType name="tns:Put_ptt"/>
        </plt:role>
    </plt:partnerLinkType>
 <wsdl:types>
        <schema targetNamespace=
‟http://xmlns.oracle.com/pcbpel/adapter/coherence/Application1/Project1/coh
Put1"  xmlns="http://www.w3.org/2001/XMLSchema">
    <import namespace=" http://xmlns.oracle.com/pcbpel/adapter/coherence "
schemaLocation="xsd/book_cache.xsd"/>
        </schema>
        <schema <schema targetNamespace= ‟http://xmlns.oracle.com/pcbpel
/adapter/coherence/Application1/Project1/cohPut1"
xmlns="http://www.w3.org/2001/XMLSchema">
           <element name="returnId" type="xsd:string"/>
         </schema>
    </wsdl:types>
    <wsdl:message name="Request_msg">
         <wsdl:part name="body" element="imp1:book"/>
    </wsdl:message>
    <wsdl:message name="Response_msg">
        <wsdl:part name="body" element="tns: returnId "/>
    </wsdl:message>
      <wsdl:portType name="Put_ptt">
        <wsdl:operation name="Put">
            <wsdl:input message="tns:Request_msg"/>
            <wsdl:output message="tns:Response_msg"/>
        </wsdl:operation>
    </wsdl:portType>
</wsdl:definitions>
```

WSDL for Remove with Filter Expression Having Bind Variables

shows the WSDL for the Coherence Remove Operation with a Filter Expression

Example - WSDL for Coherence Remove Operation with a Filter Expression

```
<wsdl:definitions name="cohRem1"
    targetNamespace="http://xmlns.oracle.com/pcbpel/adapter
        /coherence/Application1/Project1/cohRem1"
     xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:tns=http://xmlns.oracle.com/pcbpel/adapter/coherence
/Application1/Project1/cohRem1
     xmlns:plt="http://schemas.xmlsoap.org/ws/
                2003/05/partner-link/">
    <plt:partnerLinkType name="Remove_plt">
        <plt:role name="Remove_role">
            <plt:portType name="tns:Remove_ptt"/>
        </plt:role>
    </plt:partnerLinkType>
    <wsdl:types>
          <schema targetNamespace=
             “http://xmlns.oracle.com/pcbpel/adapter
            /coherence/Application1/Project1/cohRem1"
xmlns="http://www.w3.org/2001/XMLSchema">
<element name="RemoveRequest">
  <complexType>
     <element name="bind1" type="string"/>
     <element name="bind2" type="string"/>
               </complexType>
              </element>
         </schema>
<schema targetNamespace= “http://xmlns.oracle.com/
         pcbpel/adapter/coherence
       /Application1/Project1/cohRem1"
         xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="ReturnCount" type="integer"/>
        </schema>
    </wsdl:types>
    <wsdl:message name="Request_msg">
        <wsdl:part name="body" element="tns:RemoveRequest "/>
    </wsdl:message>
    <wsdl:message name="Response_msg">
        <wsdl:part name="body" element="tns: ReturnCount "/>
    </wsdl:message>
    <wsdl:portType name="Remove_ptt">
        <wsdl:operation name="Remove">
            <wsdl:input message="tns:Request_msg"/>
            <wsdl:output message="tns:Response_msg"/>
        </wsdl:operation>
    </wsdl:portType>
</wsdl:definitions>
```

WSDL for Get Operation

The following example shows a generated WSDL for the Coherence Get Operation.

Example - WSDL for Get Operation

```
<wsdl:definitions name="cohRem1" targetNamespace=
         "http://xmlns.oracle.com/
     pcbpel/adapter/coherence/Application1/Project1/cohGet1"
     xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
     xmlns:tns=http://xmlns.oracle.com/pcbpel/
       adapter/coherence/Application1/Project1/cohGet1
     xmlns:imp1=" http://xmlns.oracle.com/pcbpel/
adapter/coherence"
     xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/">
    <plt:partnerLinkType name="Get_plt">
        <plt:role name="Get_role">
            <plt:portType name="tns:Get_ptt"/>
        </plt:role>
    </plt:partnerLinkType>
    <wsdl:types>
        <schema targetNamespace=
“http://xmlns.oracle.com/pcbpel/adapter
           /coherence/Application1/Project1/cohGet1"
           xmlns="http://www.w3.org/2001/XMLSchema">
          <element name="empty"><complexType/></element>
               </schema>
        <schema targetNamespace=
             “http://xmlns.oracle.com/pcbpel/adapter/
              coherence/Application1/Project1/cohGet1"
xmlns="http://www.w3.org/2001/XMLSchema">
<import namespace=
      "http://xmlns.oracle.com/pcbpel/adapter/coherence
           "schemaLocation="xsd/book_cache.xsd"/>
            </schema>
    </wsdl:types>
    <wsdl:message name="Request_msg">
        <wsdl:part name="body" element="tns:empty "/>
    </wsdl:message>
    <wsdl:message name="Response_msg">
        <wsdl:part name="body" element="imp1:book "/>
```

```
   </wsdl:message>
    <wsdl:portType name="Get_ptt">
        <wsdl:operation name="Get">
            <wsdl:input message="tns:Request_msg"/>
            <wsdl:output message="tns:Response_msg"/>
        </wsdl:operation>
    </wsdl:portType>
</wsdl:definitions>
```

WSDL for Query with Filter Expression having Bind Variables

The following example shows the WSDL for the Query Operation, with a Filter Expression including Bind variables.

Example - WSDL for Query with Filter Expression Having Bind Variables

```
<wsdl:definitions name="cohQuery1"
targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/coherence
/Application1/Project1/cohQuery1"
     xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
     xmlns:tns=http://xmlns.oracle.com/pcbpel/adapter
/coherence/Application1/Project1/cohQuery1
     xmlns:imp1=" http://xmlns.oracle.com/pcbpel/
                  adapter/coherence"
     xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/">
    <plt:partnerLinkType name="Query_plt">
       <plt:role name="Query_role">
        <plt:portType name="tns:Query_ptt"/>
        </plt:role>
    </plt:partnerLinkType>
    <wsdl:types>
<schema targetNamespace= ‰http://xmlns.oracle.com/
              pcbpel/adapter/coherence
             /Application1/Project1/cohQuery1"
xmlns="http://www.w3.org/2001/XMLSchema">
<element name="QueryRequest">
  <complexType>
     <element name="bind1" type="string"/>
     <element name="bind2" type="string"/>
               </complexType>
              </element>
         </schema>
<schema targetNamespace= ‰http://xmlns.oracle.com/
        pcbpel/adapter/coherence
       /Application1/Project1/cohQuery1"
xmlns="http://www.w3.org/2001/XMLSchema">
<import namespace=" http://xmlns.oracle.com/
            pcbpel/adapter/coherence"
              schemaLocation="xsd/book_cache.xsd"/>
            </schema>
    </wsdl:types>
        <wsdl:message name="Request_msg">
        <wsdl:part name="body" element="tns:QueryRequest "/>
        </wsdl:message>
        <wsdl:message name="Response_msg">
        <wsdl:part name="body" element="imp1:book "/>
        </wsdl:message>
    <wsdl:portType name="Query_ptt">
        <wsdl:operation name="Query">
            <wsdl:input message="tns:Request_msg"/>
            <wsdl:output message="tns:Response_msg"/>
        </wsdl:operation>
    </wsdl:portType>
</wsdl:definitions>
```

Tips for Using the Coherence Adapter

The following tips would come handy while using the Coherence Adapter.

If you see the following error:

```
Execute of operation 'Put' failed due to: Service "TransactionalCache" has
been started by a different configurable cache factory.; nested exception
is:  java.lang.IllegalStateException: Service "TransactionalCache" has been
started by a different configurable cache factory.
```

This message means that the Service TransactionalCache is defined in multiple places. Obtaining this error is not limited only to the Coherence Adapter Put operation. You can get this exception for all operations supported by Coherence Adapter.

This issue occurs when you have the configuration files as below.

The first configuration file, a-config.xml contains the following:

```
<cache-config>
 <caching-scheme-mapping>
  <cache-mapping>
   <cache-name>adapter-local</cache-name>
   <scheme-name>transactional</scheme-name>
  </cache-mapping>
 </caching-scheme-mapping>
 <caching-schemes>
  <transactional-scheme>
   <scheme-name>transactional</scheme-name>
   <service-name>TransactionalCache</service-name>
   <autostart>true</autostart>
  </transactional-scheme>
 </caching-schemes>
</cache-config>
```

The second configuration file, b-config.xml, contains the following:

```
<cache-config>
 <caching-scheme-mapping>
  <cache-mapping>
   <cache-name>movie-local</cache-name>
   <scheme-name>transactional</scheme-name>
  </cache-mapping>
 </caching-scheme-mapping>

 <caching-schemes>
  <transactional-scheme>
   <scheme-name>transactional</scheme-name>
   <service-name>TransactionalCache</service-name>
   <autostart>true</autostart>
  </transactional-scheme>
 </caching-schemes>
</cache-config>
```

You use a-config.xml for one jndi and b-config.xml in another jndi and two composites use these two different jndis. When you deploy the first configuration, it completes acceptably, but the second deployment fails because the service TransactionalCache has already been started by the first one.
The solution is to have only one config.xml file and use the same config.xml across different jndis. For example, you can use the following config.xml file across different jndis.

```
<cache-config>
 <caching-scheme-mapping>
  <cache-mapping>
   <cache-name>adapter-local</cache-name>
   <scheme-name>transactional</scheme-name>
  </cache-mapping>
  <cache-mapping>
   <cache-name>movie-local</cache-name>
   <scheme-name>transactional</scheme-name>
  </cache-mapping>
 </caching-scheme-mapping>
 <caching-schemes>
  <transactional-scheme>
   <scheme-name>transactional</scheme-name>
   <service-name>TransactionalCache</service-name>
   <autostart>true</autostart>
  </transactional-scheme>
 </caching-schemes>
</cache-config>
```