Reladomo Test Resource

October 16, 2006

Table of Contents

1. Creating test cases using Reladomo objects

This document explains the steps required to use Reladomo objects in unit tests.

2. MithraTestResource Introduction

MithraTestResource supports writing test cases that use Reladomo objects. To use this test resource, you'll need to do the following:

- Provide a Reladomo runtime configuration that uses ConnectionManagerForTests (or a subclass) as its connection manager. The runtime configuration must include all Reladomo objects that you'll use for your test(s).
- Provide a data file that has the data you'll use for your tests. The format for the data is specified at the end of this document.

You may use multiple data files or different data for different tests. All the files used are picked up from the classpath. Typically, you'll want to use the same data for the same suite. In other words, you'll setup the MithraTestResource in your setUp() method (as opposed to each individual test). A typical setUp method might look like this:

```
this.mithraTestResource = new MithraTestResource
    ("services/Reladomo/TestPostingEngineReladomoConfiguration.xml");

ConnectionManagerForTests connectionManagerForTestDb =
 ConnectionManagerForTests.getInstance("test_db");

this.mithraTestResource.createDatabaseForStringSourceAttribute(
connectionManager,TEST_ACMAP_CODE, "testIntradayData.txt");

this.mithraTestResource.setUp();
```

You must also call the mithraTestResource.tearDown() in your tearDown() method.

.1. Reladomo runtime configuration for tests

The runtime configuration is used by the MithraTestReosurce to create the neccesary table in the in memory database used for tests. Also, the MithraTestResource initializes Reladomo using that runtime configuration. It is required that a "resourceName" property is defined.

```
 <MithraRuntime>

  <ConnectionManager className="com.gs.fw.common.mithra.test.ConnectionManagerForT
        <Property name="resourceName" value="test_db"/>

  <MithraObjectConfiguration className="com.gs.fw.myapp.Foo" cacheType="partial"/
>

  <MithraObjectConfiguration className="com.gs.fw.myapp.Bar" cacheType="partial"/
>
        ...
    </ConnectionManager>
 </MithraRuntime>
```

More than one <ConnectionManager></ConnectionManager> section can be defined in the runtime configuration in case test cases with more than one database are created. Each section must specify a "resourceName" corresponding to the name of the database to be used in the test cases(s). You must group together under the same section all the MithraObjectConfiguration that are going to reside in the same database.

```
<MithraRuntime>

  <ConnectionManager className="com.gs.fw.common.mithra.test.ConnectionManagerForT
        <Property name="resourceName" value="test_db"/>

  <MithraObjectConfiguration className="com.gs.fw.myapp.Foo" cacheType="partial"/
>

  <MithraObjectConfiguration className="com.gs.fw.myapp.Bar" cacheType="partial"/
>
        ...
    </ConnectionManager>

  <ConnectionManager className="com.gs.fw.common.mithra.test.ConnectionManagerForT
        <Property name="resourceName" value="desk_db"/>

  <MithraObjectConfiguration className="com.gs.fw.myapp.Product" cacheType="partia
>

  <MithraObjectConfiguration className="com.gs.fw.myapp.Account" cacheType="partia
>
        ...
    </ConnectionManager>
 </MithraRuntime>
```

.1. ConnectionManagerForTests

The ConnectionManagerForTests is a convenient class that implement all the ConnectionManager interfaces. This class contains factory methods to create ConnectionManager for the specified reosurce or database name.

```
```

```
/**
 * This method returns an instance of a ConnectionManagerForTests.
 * The properties must contain a property with name "resourceName".
 * The value of this property is the database name used by the
 * connection manager when creating a Connection.
 *
 * @param properties Properties use to create the instance of the
 ConnectionManagerForTests
 * @return An instance of a ConnectionManagerForTests
 */
public static ConnectionManagerForTests getInstance(Properties
 properties)

/**
 * This method returns an instance of a ConnectionManagerForTests.
 *
 * @param resourceName The database name used by the connection
 manager when creating a Connection.
 * @return An instance of a ConnectionManagerForTests
 */
public static ConnectionManagerForTests getInstance(String
 resourceName)


 /**
 * This method returns an instance of a ConnectionManagerForTests.
 *
 * @param dbName The database name used by the connection manager when
 creating a Connection.
 * @return An instance of a ConnectionManagerForTests
 */
public static ConnectionManagerForTests getInstanceForDbName(String
 dbName)
```

The resourceName used to create an instance of a ConnectionManager must be the same resourceName used in the runtime configuration.

3. MithraTestResource Detailed API

MithraTestResource provides convenience methods to perform database (starts the transaction manager, starts the database server, create the database, create tables, drop tables, insert data, and delete data from the tables) and Reladomo lifecycle operations (Initialize Reladomo objects, load Reladomo cache).

public MithraTestResource(String configFilename)

The MithraTestResource constructor expects the filename of the Reladomo configuration XML file. During construction, several things are being performed:

- Starts the database server.
- Initializes the Reladomo objects.
- Creates all the database tables for the Reladomo objects.

The Reladomo configuration file is an xml file that specifies the Reladomo objects and the connection manager.

Three methods are provided to create the database:

```
public void createSingleDatabase(SourcelessConnectionManager
 connectionManager, String testDataFilename)
    throws InitialiseException, ParseException
```

This method creates a database with the name specified in databaseName using the specified connectionManager. Before calling this method in MithraTestResource, a default source must be specified in the connectionManager. The connectionManager will use this default source to get connections. Be aware that only one default source can be specified and calling this method with a databaseName different than the default source may cause unexpected results.

```
public void
 createDatabaseForStringSourceAttribute(ObjectSourceConnectionManager
 connectionManager,
           String sourceAttribute, String testDataFilename)
                    throws InitialiseException, ParseException

public void
 createDatabaseForIntSourceAttribute(IntSourceConnectionManager
 connectionManager,
            int sourceAttribute, String testDataFilename)
                    throws InitialiseException, ParseException
```

These methods create a database with the name specified in databaseName using the specified connectionManager. The connectionManager maps the sourceAttribute to the databaseName.

Where :

- connectionManager - The connection manager that will be used for the tests. This value should be the same as the connection manager specified in the MithraConfiguration xml file.
- sourceAttribute - This value is the acmap code. The connection manager will use this value as the key to map a database name. For testing purposes, if the database is created using a String source attribute, the name of the database and the sourceAttribute will be the same.
- testDataFileName - The name of the test data file.

```
public void setUp() throws Exception
```

This method performs several things for you:

- Gets the Mithra manager.
- Sets the transaction manager provider in the Mithra manager.
- Populates database tables (from the file specified during the database creation).
- Loads Reladomo cache.

This method needs to be called from the setUp() method in the test cases

```
public void tearDown() throws Exception
```

This method cleans the database tables after the test. This method should be called from the tearDown() method in the test cases.

The following is an example of how to use the MithraTestResource. In the setup method of the test case, an instance of a MithraTestResource is created. The instance is used to create a database and populate the tables in the setup method and to clean the data in the teardown method.

```
MihtraTestResource  mihtraTestResource = null;
protected void setUp() throws Exception
{
   String xmlFile = "MithraConfigurationTest.xml"
   mithraTestResource = new MithraTestResource(xmlFile);

   ConnectionManagerForTests connMgrForTestDb =
        ConnectionManagerForTests.getInstance("test_db");

   ConnectionManagerForTests connMgrForDeskDb =
        ConnectionManagerForTests.getInstance("desk_db");

   mithraTestResource.createSingleDatabase(
        connMgForTestDb, "mithraTestData.txt");

   mithraTestResource.createDatabaseForStringSourceAttribute(
        connMgForDeskDb, "deskA", "mithraTestDataForDesk.txt");

   mithraTestResource.createDatabaseForStringSourceAttribute(
        connMgForDeskDb, "deskB", "mithraTestDataForDesk.txt");

   mithraTestResource.setUp();
}

protected void tearDown() throws Exception
{
     mithraTestResource.tearDown();
}
```

Please, refer to MithraTestAbstract.java for a complete example of how to use the MithraTestResource.

5. Test data file format

```
class <classname>
<attribute name 1>, <attribute name 2>... <attribute name N>
<value 1, 1>, <value 1, 2> ... <value 1, N>
<value 2, 1>, <value 2, 2> ... <value 2, N>
...
...
<value M, 1>, <value M, 2> ... <value M, N>
```

Where:

- classname = Fully qualified MithraObject classname.
- attribute name = Name of the attribute as specified in the MithraObject configuration xml file.
- value = The value for the specified attribute.

There are few things worth mentioning about the values in the test data file:

- Comments - Java style comments are allowed. The parsing mechanism used to read the test data files will recognize and ignore single-line (//) and multi-line () Java style comments. Comments can be placed anywhere in the file.
- String attributes - Java like Strings are allowed. The parser will recognize escape sequences inside a string value.
- Date attributes - The value is specified within double quotes. The format used for date attributes is "yyyy-MM-dd".
- Timestamp attributes - The value is specified within double quotes. The format used for timestamp attributes is "yyyy-MM-dd HH:mm:ss.SSS".

  Where:

  - yyyy - year
  - MM - month
  - dd - day in the month
  - HH - hour
  - mm - minutes
  - ss - seconds
  - SSS - milliseconds
- Numeric attributes (byte, short, integer, long, float, double)
- Boolean attributes - The values allowed for this type of attributes are: true or false. Please note that these values are not specified within single nor double quotes.
- Null values - In every row of data there must be a value for every single attribute specified. In the case that the attribute accepts null values, the value null (no quotes) can be specified.

Each line represents a row that will be inserted into the database. White space is generally ignored, except when it acts as line separators (you may not split the data for a single row across multiple lines).

This is an example of how the test data can be specified for an Account and ExchangeRate objects:

```
class <classname>
class com.gs.fw.common.mithra.test.domain.Account
accountNumber, code, trialId, pnlGroupId
"7410161001", "74101610", "001A", "999A"
"7410161101", "74101611", "001A", "999A"
"7410161201", "74101612", "001A", "999A"
```

```
"7410161301", "74101613", "001A", "999A"
"null trial", "testnull", null, "NULLTEST"
"with trial", "testnull", "001A", "NULLTEST"

class com.gs.fw.common.mithra.test.domain.ExchangeRate
currency, source, date, exchangeRate
"EUR", 10, "2004-09-30 18:30:00.0", 1.25
"USD", 10, "2004-09-30 18:30:00.0", 1

"GBP", 10, "2004-09-30 18:30:00.0", 1.81
"JPY", 10, "2004-09-30 18:30:00.0", .009
```

For an example of a Reladomo test data file, please refer to mithraTestDataStringSourceA.txt