Oracle® TopLink

Release Notes

Release 12c (12.1.2)

E40213-01

June 2013

This chapter describes issues associated with Oracle TopLink. It includes the following topics:

- Section 1, "TopLink Object-Relational Issues"
- Section 2, "Oracle Database Extensions with TopLink"
- Section 3, "Allowing Zero Value Primary Keys"
- Section 4, "Managed Servers on Sybase with JCA Oracle Database Service"
- Section 5, "Logging Configuration with EclipseLink Using Container Managed JPA"
- Section 6, "Documentation Accessibility"

1 TopLink Object-Relational Issues

This section contains information on the following issues:

- Section 1.1, "Cannot set EclipseLink log level in WLS System MBean Browser"
- Section 1.2, "UnitOfWork.release() not Supported with External Transaction Control"
- Section 1.3, "Returning Policy for UPDATE with Optimistic Locking"
- Section 1.4, "JDBC Drivers returning Timestamps as Strings"
- Section 1.5, "Unit of Work does not add Deleted Objects to Change Set"

1.1 Cannot set EclipseLink log level in WLS System MBean Browser

Use Oracle Enterprise Manager to set the EclipseLink log level; do not use the WLS System MBean Browser to complete this action.

1.2 UnitOfWork.release() not Supported with External Transaction Control

A unit of work synchronized with a Java Transaction API (JTA) will throw an exception if it is released. If the current transaction requires its changes to not be persisted, the JTA transaction must be rolled back.

When in a container-demarcated transaction, call `setRollbackOnly()` on the EJB/session context:

```
@Stateless
public class MySessionBean
{    @Resource
    SessionContext sc;

    public void someMethod()
    {
        ...
```

```
        sc.setRollbackOnly();
    }
}
```

When in a bean-demarcated transaction then you call `rollback()` on the UserTransaction obtained from the EJB/session context:

```
@Stateless
@TransactionManagement(TransactionManagementType.BEAN)
public class MySessionBean implements SomeInterface
{
    @Resource
    SessionContext sc;

    public void someMethod()
    {
        sc.getUserTransaction().begin();
        ...
        sc.getUserTransaction().rollback();
    }
}
```

1.3 Returning Policy for UPDATE with Optimistic Locking

The returning policy, which allows values modified during `INSERT` and `UPDATE` to be returned and populated in cached objects, does not work in conjunction with numeric version optimistic locking for `UPDATE`. The value returned for all `UPDATE` operations is **1** and does not provide meaningful locking protection.

Do not use a returning policy for `UPDATE` in conjunction with numeric optimistic locking.

The use of returning policy for `INSERT` when using optimistic locking works correctly.

1.4 JDBC Drivers returning Timestamps as Strings

TopLink assumes that date and time information returned from the server will use `Timestamp`. If the JDBC driver returns a `String` for the current date, TopLink will throw an exception. This is the case when using a DB2 JDBC driver.

To work around this issue, consider using a driver that returns Timestamp (such as COM.ibm.db2.jdbc.app.DB2Driver) or change the policy to use local time instead of server time.

Another option is to use a query re-director on the ValueReadQuery used by the platform:

```
ValueReadQuery vrq = new ValueReadQuery(
    "SELECT to_char(sysdate, 'YYYY-MM-DD HH:MM:SS.SSSSS') FROM DUAL"
);
vrq.setRedirector(new TSQueryRedirector());
...
class TSQueryRedirector implements QueryRedirector
{
    public Object invokeQuery(DatabaseQuery query, Record arguments, Session session)
    {
        String value = (String)session.executeQuery(query);
        return ConversionManager.getDefaultManager().convertObject(
            value, java.sql.Timestamp.class
        );
    }
}
```

1.5 Unit of Work does not add Deleted Objects to Change Set

When accessing the change set of a Unit of Work to determine what has changed, objects that are pending deletion (such as `uow.deleteObject( )` and `uow.deleteAllObjects( )`) will not be returned from the result set.

The objects pending deletion are only available through the Unit of Work `getDeletedObjects` call.

2 Oracle Database Extensions with TopLink

This section contains information on the following:

- Section 2.1, "Template JAR for Spatial and XDB Support in Oracle WebLogic Server"

2.1 Template JAR for Spatial and XDB Support in Oracle WebLogic Server

To fully support Oracle Spatial and Oracle XDB mapping capabilities (in both standalone Oracle WebLogic Server and the Oracle JDeveloper integrated WebLogic Server), you must use the `toplink-spatial-template.jar` and `toplink-xdb-template.jar` to extend the WebLogic Server domain to support Oracle Spatial and XDB, respectively.

To extend your WebLogic Server domain:

1. Download the `toplink-spatial-template.jar` (to support Oracle Spatial) and `toplink-xdb-template.jar` (to support Oracle XDB) files from:
   - `http://download.oracle.com/otn/java/toplink/111110/toplink-spatial-template.jar`
   - `http://download.oracle.com/otn/java/toplink/111110/toplink-xdb-template.jar`
2. Use Table 1, " To Support Oracle Spatial" or Table 2, " To Support Oracle XDB" to determine which files to copy.

Table 1 To Support Oracle Spatial

Copy this file	From...[1]	To...[2]
sdoapi.jar	*<ORACLE_DATABASE_HOME>*/md/jlib	*<WEBLOGIC_HOME>*/server/lib

[1] These are the default locations. Your actual location may vary depending on your specific environment, installed options, and version.

[2] When using Oracle JDeveloper integrated WebLogic Server, the *<WEBLOGIC_HOME>* is located within the *<JDEVELOPER_HOME>* directory.

Table 2 To Support Oracle XDB

Copy this file	From...[1]	To...[2]
xdb.jar	*<ORACLE_DATABASE_HOME>*/rdbms/jlib	*<WEBLOGIC_HOME>*/server/lib
xml.jar	*<ORACLE_DATABASE_HOME>*/lib	*<WEBLOGIC_HOME>*/server/lib
xmlparserv2.jar	*<ORACLE_DATABASE_HOME>*/lib	*<WEBLOGIC_HOME>*/server/lib
orai18n-mapping.jar[3]	*<ORACLE_DATABASE_HOME>*/jlib	*<WEBLOGIC_HOME>*/server/lib

[1] These are the default locations. Your actual location may vary depending on your specific environment, installed options, and version.

[2] When using Oracle JDeveloper integrated WebLogic Server, the *<WEBLOGIC_HOME>* is located within the *<JDEVELOPER_HOME>* directory.

[3] Use **orai18n-mapping.jar** for Oracle Database 11.2 and higher.

Note: Although the actual JAR file may be named differently in your **From** directory, the file must be named as shown, when copied to the **To** directory.

3. Launch the Config Wizard (`<WEBLOGIC_HOME>/common/bin/config.sh` (or `.bat`).
4. Select **Extend an existing WebLogic domain**.
5. Browse and select your WebLogic Server domain.

   When using JDeveloper with integrated WebLogic Server, the typical WebLogic Server domain location may be similar to:

   - In Windows environments: `%APPDATA%\JDeveloper\systemXX.XX.XX.XX\DefaultDomain`

     where `XX.XX.XX.XX` is the unique number of the product build.

     For Windows platforms, you must enable the **Show hidden files and folders** folder option.
   - In non-Windows environments, the default location is under the current user's default home directory: `<$Home>/DefaultDomain`

   Refer to the Oracle JDeveloper documentation for details.
6. Select **Extend my domain using an existing extension template**.
7. Browse and select the required template JAR (**toplink-spatial-template.jar** for Oracle Spatial, **toplink-xdb-template.jar** for Oracle XDB).
8. Complete the remaining pages of the wizard.

3 Allowing Zero Value Primary Keys

By default, EclipseLink interprets *zero* as *null* for primitive types that cannot be null (such as `int` and `long`) causing zero to be an invalid value for primary keys. You can modify this setting by using the `allow-zero-id` property in the `persistence.xml` file. Valid values are:

- **true** – EclipseLink interprets zero values as *zero*. This permits primary keys to use a value of zero.
- **false** (default) – EclipseLink interprets zero as *null*.

Refer the *EclipseLink User's Guide* at `http://wiki.eclipse.org/EclipseLink/UserGuide` for more information.

4 Managed Servers on Sybase with JCA Oracle Database Service

When using a JCA service with the Oracle Database adapter in a cluster to perform database operations on a Sybase database, the managed nodes in the cluster process the messages and may attempt to perform duplicate operations.

Because supported versions of Sybase do not support Oracle TopLink record locking, Sybase allows the duplicate operation attempts.

5 Logging Configuration with EclipseLink Using Container Managed JPA

By default, EclipseLink users in container managed JPA will use the Oracle WebLogic Server logging options to report all log messages generated by EclipseLink. Refer to "Configuring WebLogic Logging Services" in *Oracle® Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

To use the EclipseLink native logging configuration, add the following property to your `persistence.xml` file:

```
<property name="eclipselink.logging.logger" value="DefaultLogger"/>
```

6 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle Fusion Middleware, Release 12c (12.1.2) TopLink Release Notes
E40213-01