

Configuring Oracle TimesTen 11.2.2 for J2EE Application Servers and Object-Relational Mapping Frameworks

An Oracle White Paper
January 2012

Configuring Oracle TimesTen 11.2.2 for J2EE Application Servers and Object-Relational Mapping Frameworks

Introduction	3
Conventions	3
Prerequisites	4
Troubleshooting	4
TimesTen for JBoss AS 5.1.....	6
Configuring the TimesTen JDBC driver	6
Configuring CMP type mappings for TimesTen	6
Configuring TimesTen data sources	8
Using JBoss JPA with TimesTen.....	12
TimesTen for Oracle WebLogic Server 10.3	13
Configuring the TimesTen JDBC driver	13
Configuring TimesTen data sources	13
Using Weblogic Server JPA with TimesTen.....	21
TimesTen for Oracle GlassFish Server 3.1	22
Configuring the TimesTen JDBC driver	22
Configuring TimesTen connections	22
Configuring a TimesTen JDBC connection pool	23
Configuring a TimesTen JDBC resource	25
Using GlassFish Server JPA with TimesTen	26
TimesTen for WebSphere Application Server 7.....	27
Configuring TimesTen connections	27
Configuring a TimesTen JDBC provider	28
Configuring TimesTen data sources	30
Using WebSphere Application Server JPA with TimesTen	32
TimesTen for EclipseLink 2	33
Configuring the TimesTen JDBC driver for EclipseLink	33
TimesTen for Hibernate 3.5	35
Configuring the TimesTen JDBC driver for Hibernate.....	35
Hibernate configuration properties for TimesTen	35
TimesTen for OpenJPA 2.....	39
Configuring the TimesTen JDBC driver for OpenJPA	39

Configuring Oracle TimesTen 11.2.2 for J2EE Application Servers and Object-Relational Mapping Frameworks

INTRODUCTION

This whitepaper is for application developers who use and administer Oracle TimesTen JDBC and for system administrators who configure and manage the TimesTen database. It provides information about configuring J2EE application servers and object-relational mapping frameworks for use with Oracle TimesTen 11.2.2.

Additional information and code examples demonstrating the use of TimesTen with Java technologies is available in the Oracle TimesTen Quick Start Guide. The TimesTen Quick Start Guide is an optional component of a TimesTen installation.

CONVENTIONS

This document uses the following conventions:

tt_install_dir: The path to the directory where Oracle TimesTen is installed.

gs_install_dir: The path to the directory where Oracle GlassFish Server is installed.

gs_domain: The name of the directory that contains a GlassFish Server domain configuration.

jboss_install_dir: The path to the directory where JBoss Application Server is installed.

server_name: The directory name for a configured JBoss application server.

wl_install_dir: The path to the directory where Oracle Weblogic Server is installed.

wl_domain_dir: The path to the directory containing a Weblogic Server domain configuration.

PREREQUISITES

The sample configurations provided in this document assume that you have installed both TimesTen and the target J2EE application server or object-relational mapping framework on the same machine. In addition, the following TimesTen items are required:

- A TimesTen direct driver DSN called `SAMPLEDB_1122` defined in the `.odbc.ini` ODBC configuration file or in the ODBC Data Source Administrator dialog box on Windows platforms.
- A TimesTen client/server driver DSN called `SAMPLEDBCS_1122` defined in the `.odbc.ini` ODBC configuration file or in the ODBC Data Source Administrator dialog box on Windows platforms. This `SAMPLEDBCS_1122` DSN should be configured to connect to the database defined by the `SAMPLEDB_1122` direct driver DSN.
- An internal TimesTen user account with the user name `scott` and password `'tiger'` defined in the database associated with the `SAMPLEDB_1122` DSN.

Test and configure the two DSNs and the `scott/tiger` user account with the TimesTen `ttIsql` utility before following the procedures and using the configurations included in this document.

TROUBLESHOOTING

- The TimesTen JDBC driver loads native shared libraries at runtime. This requires the Java Virtual Machine (JVM) to use the same architecture as the TimesTen JDBC driver. For example, a 32-bit JVM cannot connect to a TimesTen database using a 64-bit version of the TimesTen JDBC driver and vice versa. Attempts to mix architectures in this way will result in a runtime error when the Java application attempts to connect to the TimesTen database.
- The TimesTen JDBC driver includes a tracing facility that can help diagnose difficult problems. To enable the tracing facility you must enable a Java system property called `timesten.tracefile` for the JVM accessing the TimesTen JDBC driver. You can enable tracing from the Java command line using the syntax `-Dtimesten.tracefile=filename` where *filename* is the complete path to a local output file.
- When attempting to establish a connection to a TimesTen database from within the JVM, a TimesTen JDBC driver exception with a message like “Cannot attach database shared-memory segment” can occur. This problem is likely to occur on 32-bit platforms when the system is unable to allocate or map enough contiguous shared memory to hold the TimesTen

database. It can also occur when there is a conflict with the value of the `PLSQL_MEMORY_ADDRESS` TimesTen connection attribute. For detailed instructions on how to resolve this and other similar connection problems see the *Oracle TimesTen In-Memory Database Troubleshooting Procedures Guide* and the *Oracle TimesTen In-Memory Database Installation Guide*. We recommend these troubleshooting steps:

- a. Verify that it is possible to connect to the TimesTen database outside of the JVM. You can use the `ttIsql` utility for this purpose.
- b. Try increasing the maximum amount of shared memory available to the system. See the *Oracle TimesTen In-Memory Database Installation Guide* for platform specific details.
- c. Try decreasing the memory size of the TimesTen database by recreating the database with lower values for the `Permsize` and `TempSize` connection attributes.
- d. Try adjusting the value of the `PLSQL_MEMORY_ADDRESS` connection attribute. If PL/SQL support is not required for the TimesTen database then disable it by recreating the TimesTen database with the `PLSQL` connection attribute set to 0.
- e. Terminate any unnecessary processes on the system to make more memory available.
- f. If possible, try a client/server TimesTen JDBC connection instead of a direct connection.
- g. If possible, try a 64-bit version of the TimesTen JDBC driver in conjunction with a 64-bit JVM.

TIMESTEN FOR JBOSS AS 5.1

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with JBoss 5.1 Application Server. This document assumes that Oracle TimesTen and JBoss Application Server have been installed on the same machine.

Configuring the TimesTen JDBC driver

Before you can use JBoss with TimesTen databases, the JBoss server environment must have access to the TimesTen JDBC driver jar file and the TimesTen shared libraries.

1. Copy the TimesTen JDBC driver jar file from `tt_install_dir/lib` to `jboss_install_dir/server/server_name/lib` where `server_name` is the directory of a configured JBoss server. If the JBoss server is using the Java 5 runtime then copy the `ttjdbc5.jar` TimesTen driver jar file. If the JBoss server is using the Java 6 runtime then copy the `ttjdbc6.jar` TimesTen driver jar file.
2. Set the `LD_LIBRARY_PATH` environment variable (or the equivalent variable for your OS) to include the `tt_install_dir/lib` directory where TimesTen shared libraries are located. On Windows, set the `PATH` environment variable to the `tt_install_dir/bin` directory where the TimesTen DLLs are located. This variable must be set for the environment where JBoss runs.
3. Restart the JBoss Server.

Configuring CMP type mappings for TimesTen

For TimesTen to work with various features of Container Managed Persistence (CMP) in JBoss, a database vendor specific JDBC to SQL type mapping definition may be required. An example type mapping definition for TimesTen 11.2.2 (`timesten-type-mapping.xml`) is located under the `quickstart` directory in your TimesTen installation.

Example of the type mapping descriptor for TimesTen:

```
<type-mapping>
  <name>TimesTen</name>
  <row-locking-template>
    SELECT ?1 FROM ?2 WHERE ?3 FOR UPDATE
  </row-locking-template>
  <pk-constraint-template>
    PRIMARY KEY (?2)
  </pk-constraint-template>
```

```

<fk-constraint-template>
  ALTER TABLE ?1 ADD CONSTRAINT ?2 FOREIGN KEY (?3) REFERENCES ?4
(?5)
</fk-constraint-template>
<add-column-template>
  ALTER TABLE ?1 ADD COLUMN ?2 ?3
</add-column-template>
<drop-column-template>
  ALTER TABLE ?1 DROP COLUMN ?2
</drop-column-template>
<alias-header-prefix>t</alias-header-prefix>
<alias-header-suffix></alias-header-suffix>
<alias-max-length>30</alias-max-length>
<subquery-supported>>true</subquery-supported>
<true-mapping>(1=1)</true-mapping>
<false-mapping>(1=0)</false-mapping>

<function-mapping>
  <function-name>concat</function-name>
  <function-sql>CONCAT (?1, ?2)</function-sql>
</function-mapping>
<function-mapping>
  <function-name>substring</function-name>
  <function-sql>SUBSTRING (?1, ?2, ?3)</function-sql>
</function-mapping>
<function-mapping>
  <function-name>lcase</function-name>
  <function-sql>LOWER (?1)</function-sql>
</function-mapping>
<function-mapping>
  <function-name>length</function-name>
  <function-sql>LENGTH (?1)</function-sql>
</function-mapping>
<function-mapping>
  <function-name>locate</function-name>
  <function-sql>INSTR (?2, ?1, ?3)</function-sql>
</function-mapping>
<function-mapping>
  <function-name>ucase</function-name>
  <function-sql>UPPER (?1)</function-sql>
</function-mapping>
<function-mapping>
  <function-name>count</function-name>
  <function-sql>COUNT (?1)</function-sql>
</function-mapping>

<mapping>
  <java-type>java.lang.Boolean</java-type>
  <jdbc-type>TINYINT</jdbc-type>
  <sql-type>TT_TINYINT</sql-type>
</mapping>
<mapping>
  <java-type>java.lang.Byte</java-type>
  <jdbc-type>TINYINT</jdbc-type>
  <sql-type>TT_TINYINT</sql-type>
</mapping>
<mapping>
  <java-type>java.lang.Short</java-type>
  <jdbc-type>SMALLINT</jdbc-type>
  <sql-type>TT_SMALLINT</sql-type>
</mapping>
<mapping>
  <java-type>java.lang.Integer</java-type>
  <jdbc-type>INTEGER</jdbc-type>
  <sql-type>INTEGER</sql-type>
</mapping>
<mapping>
  <java-type>java.lang.Long</java-type>
  <jdbc-type>BIGINT</jdbc-type>
  <sql-type>TT_BIGINT</sql-type>
</mapping>
<mapping>
  <java-type>java.lang.Float</java-type>

```

```

        <jdbc-type>REAL</jdbc-type>
        <sql-type>BINARY_FLOAT</sql-type>
    </mapping>
    <mapping>
        <java-type>java.math.BigDecimal</java-type>
        <jdbc-type>DECIMAL</jdbc-type>
        <sql-type>DECIMAL (38,15)</sql-type>
    </mapping>
    <mapping>
        <java-type>java.lang.Double</java-type>
        <jdbc-type>DOUBLE</jdbc-type>
        <sql-type>BINARY_DOUBLE</sql-type>
    </mapping>
    <mapping>
        <java-type>java.lang.Character</java-type>
        <jdbc-type>CHAR</jdbc-type>
        <sql-type>CHARACTER</sql-type>
    </mapping>
    <mapping>
        <java-type>java.lang.String</java-type>
        <jdbc-type>VARCHAR</jdbc-type>
        <sql-type>VARCHAR (256)</sql-type>
    </mapping>
    <mapping>
        <java-type>java.sql.Date</java-type>
        <jdbc-type>DATE</jdbc-type>
        <sql-type>DATE</sql-type>
    </mapping>
    <mapping>
        <java-type>java.sql.Time</java-type>
        <jdbc-type>TIME</jdbc-type>
        <sql-type>TT_TIME</sql-type>
    </mapping>
    <mapping>
        <java-type>java.sql.Timestamp</java-type>
        <jdbc-type>TIMESTAMP</jdbc-type>
        <sql-type>TIMESTAMP</sql-type>
    </mapping>
    <mapping>
        <java-type>java.lang.Object</java-type>
        <!-- directly serializable objects -->
        <jdbc-type>BLOB</jdbc-type>
        <sql-type>BLOB</sql-type>
    </mapping>
</type-mapping>

```

You should insert this TimesTen specific type mapping definition into the `<type-mappings>` section of the `jboss_install_dir/server/server_name/conf/standardjbossmp-jdbc.xml` configuration file where `server_name` is the directory of a configured JBoss server. This type mapping can then be referenced by JBoss data source deployment descriptors.

Configuring TimesTen data sources

The TimesTen JDBC driver supports four different types of connections to TimesTen databases.

- **Direct access with local transactions only** - This configuration provides the fastest database performance. The TimesTen database must reside on the same machine as the JBoss server.

- **Direct access with XA distributed transaction support** - This configuration uses JTA to support distributed transactions. The TimesTen database must reside on the same machine as the JBoss server.
- **Client/server access with local transactions only** - The TimesTen database can reside on a remote machine that is running the TimesTen server.
- **Client/server access with XA distributed transaction support** - This configuration uses JTA to support distributed transactions. The TimesTen database can reside on a remote machine that is running the TimesTen server.

Each type of TimesTen JDBC driver connection requires a unique JBoss data source configuration. A JBoss data source deployment descriptor file is named using a `*-ds.xml` pattern and is placed in the `jboss_install_dir/server/server_name/deploy` directory of the JBoss installation. A sample JBoss data source deployment descriptor file (`timesten-ds.xml`) that includes all four types of TimesTen JDBC driver configurations is located under the `quickstart` directory in your TimesTen installation.

Example of the deployment descriptor file, it includes 4 distinct types of connections for TimesTen:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ===== -->
<!-- -->
<!-- TimesTen datasource configurations for the TptbmAS benchmark. -->
<!-- -->
<!-- ===== -->

<!-- TimesTen local transaction datasources -->

<datasources>

  <local-tx-datasource>
    <jndi-name>TbtbmDirectDS</jndi-name>
    <connection-url>jdbc:timesten:direct:SAMPLEDB_1122</connection-url>
    <driver-class>com.timesten.jdbc.TimesTenDriver</driver-class>

    <user-name>SCOTT</user-name>
    <password>tiger</password>
    <transaction-isolation>
      TRANSACTION_READ_COMMITTED
    </transaction-isolation>

    <!--pooling parameters-->
    <min-pool-size>5</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <blocking-timeout-millis>5000</blocking-timeout-millis>
    <idle-timeout-minutes>15</idle-timeout-minutes>

    <prepared-statement-cache-size>32</prepared-statement-cache-size>

    <!-- sql to call when connection is created or validated
    <new-connection-sql>SELECT * FROM DUAL</new-connection-sql>
```

```

<check-valid-connection-sql>
    SELECT * FROM DUAL
</check-valid-connection-sql>
-->

<!-- corresponding type-mapping in the standardjbosscmp-jdbc.xml -->
<metadata>
    <type-mapping>TimesTen</type-mapping>
</metadata>
</local-tx-datasource>

<local-tx-datasource>
    <jndi-name>TbtbmClientDS</jndi-name>
    <connection-url>jdbc:timesten:client:SAMPLEDBCS_1122</connection-
url>
    <driver-class>com.timesten.jdbc.TimesTenDriver</driver-class>

    <user-name>SCOTT</user-name>
    <password>tiger</password>
    <transaction-isolation>
        TRANSACTION_READ_COMMITTED
    </transaction-isolation>

    <!--pooling parameters-->
    <min-pool-size>5</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <blocking-timeout-millis>5000</blocking-timeout-millis>
    <idle-timeout-minutes>15</idle-timeout-minutes>

    <prepared-statement-cache-size>32</prepared-statement-cache-size>

    <!-- sql to call when connection is created or validated
    <new-connection-sql>SELECT * FROM DUAL</new-connection-sql>
    <check-valid-connection-sql>
        SELECT * FROM DUAL
    </check-valid-connection-sql>
    -->

    <!-- corresponding type-mapping in the standardjbosscmp-jdbc.xml -->
    <metadata>
        <type-mapping>TimesTen</type-mapping>
    </metadata>
</local-tx-datasource>

<!-- TimesTen XA transaction datasources -->
<xa-datasource>
    <jndi-name>TbtbmDirectXADS</jndi-name>
    <xa-datasource-class>
        com.timesten.jdbc.xa.TimesTenXADataSource
    </xa-datasource-class>
    <xa-datasource-property name="Url">
        jdbc:timesten:direct:SAMPLEDB_1122
    </xa-datasource-property>

    <user-name>SCOTT</user-name>
    <password>tiger</password>
    <transaction-isolation>
        TRANSACTION_READ_COMMITTED
    </transaction-isolation>

    <!--pooling parameters-->
    <min-pool-size>5</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <blocking-timeout-millis>5000</blocking-timeout-millis>
    <idle-timeout-minutes>15</idle-timeout-minutes>

    <prepared-statement-cache-size>32</prepared-statement-cache-size>

```

```

    <!-- This is required by TimesTen XA data sources. If it is not
    included then XA transactions can fail with various transaction
    management errors including javax.transaction.xa.XAException:
    errorCode=XAER_PROTO -->
    <track-connection-by-tx/>

    <!-- sql to call when connection is created or validated
    <new-connection-sql>SELECT * FROM DUAL</new-connection-sql>

    <check-valid-connection-sql>
        SELECT * FROM DUAL
    </check-valid-connection-sql>
    -->

    <!-- corresponding type-mapping in the standardjbosscomp-jdbc.xml -->
    <metadata>
        <type-mapping>TimesTen</type-mapping>
    </metadata>
</xa-datasource>

<xa-datasource>
    <jndi-name>TbtbmClientXADS</jndi-name>
    <xa-datasource-class>
        com.timesten.jdbc.xa.TimesTenXADataSource
    </xa-datasource-class>
    <xa-datasource-property name="Url">
        jdbc:timesten:client:SAMPLEDBCS_1122
    </xa-datasource-property>

    <user-name>SCOTT</user-name>
    <password>tiger</password>
    <transaction-isolation>
        TRANSACTION_READ_COMMITTED
    </transaction-isolation>

    <!--pooling parameters-->
    <min-pool-size>5</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <blocking-timeout-millis>5000</blocking-timeout-millis>
    <idle-timeout-minutes>15</idle-timeout-minutes>

    <prepared-statement-cache-size>32</prepared-statement-cache-size>

    <!-- This is required by TimesTen XA data sources. If it is not
    included then XA transactions can fail with various transaction
    management errors including javax.transaction.xa.XAException:
    errorCode=XAER_PROTO -->
    <track-connection-by-tx/>

    <!-- sql to call when connection is created or validated
    <new-connection-sql>SELECT * FROM DUAL</new-connection-sql>
    <check-valid-connection-sql>
        SELECT * FROM DUAL
    </check-valid-connection-sql>
    -->

    <!-- corresponding type-mapping in the standardjbosscomp-jdbc.xml -->
    <metadata>
        <type-mapping>TimesTen</type-mapping>
    </metadata>
</xa-datasource>

</datasources>

```

The key difference between the configurations is the TimesTen driver class name and the TimesTen URL. TimesTen connections that support local transactions only

use the `com.timesten.jdbc.TimesTenDriver` class. Connections that require distributed transaction support through JTA use the `com.timesten.jdbc.xa.TimesTenXADataSource` class.

Using JBoss JPA with TimesTen

JBoss server utilizes the Hibernate object-relational mapping framework to support the Java Persistence API (JPA). TimesTen supports Hibernate applications by providing a custom SQL dialect class called `TimesTenDialect1122`. This dialect class should be used by JBoss applications that connect to TimesTen using the Hibernate framework. For more information on using the TimesTen SQL dialect see the TimesTen for Hibernate section later in this document and the TimesTen Quick Start Guide.

TIMESTEN FOR ORACLE WEBLOGIC SERVER 10.3

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with Oracle WebLogic Server 10.3.4.0. This document assumes that Oracle TimesTen and Oracle WebLogic Server have been installed on the same machine.

Configuring the TimesTen JDBC driver

Before you can use WebLogic Server to access TimesTen databases, the server environment must have access to the TimesTen JDBC driver jar file and the native TimesTen shared libraries.

1. Edit the `wl_install_dir/common/bin/commEnv.sh` script on UNIX or the `wl_install_dir/common/bin/commEnv.cmd` script on Windows. Append the path to the TimesTen JDBC driver jar file to the `WEBLOGIC_CLASSPATH` variable defined in the file. The Java 6 runtime version of the TimesTen driver jar file at `tt_install_dir/lib/ttjdbc6.jar` should be used.
2. Modify the `LD_LIBRARY_PATH` (or equivalent variable for you OS) to locate the TimesTen shared libraries. Append the `tt_install_dir/lib` path to the `LD_LIBRARY_PATH` (or equivalent) variable in the `wl_install_dir/common/bin/commEnv.sh` file. On Windows platforms, append the `PATH` environment variable with the `tt_install_dir/bin` directory where TimesTen DLLs are located.
3. Restart the WebLogic Server.

Configuring TimesTen data sources

The TimesTen JDBC driver supports four types of connections to TimesTen databases.

- **Direct access with non-XA local transactions only** - This configuration provides the fastest database performance. The TimesTen database must reside on the same machine as WebLogic Server.
- **Direct access with TimesTen XA distributed transaction support** - The driver uses JTA to supports distributed transactions. The TimesTen database must reside on the same machine as WebLogic Server.
- **Client/server access with non-XA local transactions only** - The TimesTen database can reside on a remote machine that is running the TimesTen server.

- **Client/server access with TimesTen XA distributed transaction support** - The driver uses JTA to supports distributed transactions. The TimesTen database can reside on a remote machine that is running the TimesTen server.

Each type of TimesTen JDBC driver connection requires a unique Weblogic Server data source configuration. The examples below reference a TimesTen direct access DSN named **SAMPLEDB_1122** and a client/server access DSN named **SAMPLEDBCS_1122**. These TimesTen DSNs must be configured in the `.odbc.ini` or `sys.odbc.ini` file or the Windows Data Source Administrator before the examples can work.

A data source is usually configured using the WebLogic Server Administration Console. Below is an example procedure for configuring a TimesTen data source within the console.

1. Connect to the WebLogic Server Administration Console and click **Data Sources** under the Services section
2. Click the **New** button and then click **Generic Data Source**.
3. At the **JDBC Data Source Properties** screen type a data source name and a JNDI name for the data source. In the **Database Type** drop down list box select **Other**. The screen should look similar to figure 3.1.

Figure 3.1 JDBC Data source properties screen

The screenshot shows the 'Create a New JDBC Data Source' wizard in the WebLogic Server Administration Console. The page title is 'Home > Summary of JDBC Data Sources'. The breadcrumb trail is 'Home > Summary of JDBC Data Sources'. The main heading is 'Create a New JDBC Data Source'. Below the heading are buttons for 'Back', 'Next', 'Finish', and 'Cancel'. The section is titled 'JDBC Data Source Properties' and contains the following text: 'The following properties will be used to identify your new JDBC data source. * Indicates required fields'. The first question is 'What would you like to name your new JDBC data source?'. The '* Name:' field contains 'TptbmDS'. The second question is 'What JNDI name would you like to assign to your new JDBC Data Source?'. The '* JNDI Name:' field contains 'jdbc/TptbmDS'. The third question is 'What database type would you like to select?'. The 'Database Type:' dropdown menu is set to 'Other'. At the bottom of the form are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

Click **Next**.

4. On the following screen the **Database Driver** drop down list value should be set to **Other**. Click **Next**.
5. At the **Transaction Options** screen, make sure that the **Supports Global Transactions** check box is checked. If an XA connection is not required by the application then select one of the three transaction protocol options. The screen should look similar to figure 3.2.

Figure 3.2 Transaction Options screen

Home Log Out Preferences Record Help Welcome, weblogic Connected to: base_domain

Home > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

Transaction Options

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

Supports Global Transactions

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the *Logging Last Resource* (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

Logging Last Resource

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.

Emulate Two-Phase Commit

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.

One-Phase Commit

Back Next Finish Cancel

Click **Next**. (Note: If you require a TimesTen XA connection for distributed JTA transactions, the transaction protocol options on this screen are ignored. Instead, WebLogic uses the TimesTen implementation of XAResource to control transactions.)

6. At the **Connection Properties** screen enter a database name, host name and port. These three attributes are required by WebLogic Server but are not used by the TimesTen JDBC driver. Specify the TimesTen database user's name in the **Database User Name** field and the TimesTen database user's password in the **Password** and **Confirm Password** fields. The screen should look similar to figure 3.3.

Figure 3.3 Connection Properties screen

Home Log Out Preferences Record Help Welcome, weblogic Connected to: base_domain
Home > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

Connection Properties
Define Connection Properties.

What is the name of the database you would like to connect to?

Database Name: TPTBM

What is the name or IP address of the database server?

Host Name: localhost

What is the port on the database server used to connect to the database?

Port: 53384

What database account user name do you want to use to create database connections?

Database User Name: scott

What is the database account password to use to create database connections?

Password: ●●●●

Confirm Password: ●●●●

Back Next Finish Cancel

Click **Next**.

- At the **Test Database Connection** screen enter the name of the TimesTen JDBC driver. If a TimesTen XA connection is required then specify the `com.timesten.jdbc.xa.TimesTenXADataSource` class. Otherwise specify the `com.timesten.jdbc.TimesTenDriver` class. Enter a TimesTen JDBC URL for the TimesTen DSN that the data source should connect to. A TimesTen URL takes the form: `jdbc:timesten:[direct|client]:DSN`. In the Properties field enter `user=tt_username` where `tt_username` is the name of the TimesTen database user. Given a direct DSN named **SAMPLEDB_1122** and a TimesTen database user called **scott** the screen should look similar to figure 3.4.

Figure 3.4 Test Database Connection screen

Home Log Out Preferences Record Help Welcome, weblogic Connected to: base_domain

Home > Summary of JDBC Data Sources

Create a New JDBC Data Source

Test Configuration Back Next Finish Cancel

Test Database Connection

Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool?
(Note that this driver class must be in the classpath of any server to which it is deployed.)

Driver Class Name: oracle.jdbc.OracleDriver

What is the URL of the database to connect to? The format of the URL varies by JDBC driver.

URL: jdbc:oracle:thin:@localhost:1521:xe

What database account user name do you want to use to create database connections?

Database User Name: scott

What is the database account password to use to create database connections?
(Note: for secure password management, enter the password in the Password field instead of the Properties field below)

Password: [masked]

Confirm Password: [masked]

What are the properties to pass to the JDBC driver when creating database connections?

Properties:
user=scott

If the configuration is correct then clicking on the **Test Configuration** button should result in a success message. Click **Next**.

- At the **Select Targets** screen, select the WebLogic servers to be associated with the data source. Click **Finish**.
- The message **All changes have been activated** indicates that the configuration is complete.

When a data source is configured in the WebLogic console an XML file with the name pattern of `*-jdbc.xml` is created in the `wl_domain_dir/config/jdbc` directory. Sample configuration files for the four different types of TimesTen connections are available under the `quickstart` directory in your TimesTen installation. To use these configuration files copy them to the `wl_domain_dir/config/jdbc` directory. The `wl_domain_dir/config/config.xml` file must also be edited to include a reference to the data source configuration file. For example if you use the `TptbmDirectDS-jdbc.xml` file, the `wl_domain_dir/config/config.xml` should include an entry like this in the `<domain>` section:

```
<jdbc-system-resource>
  <name>TptbmDirectDS</name>
  <target>AdminServer</target>
  <descriptor-file-name>jdbc/TptbmDirectDS-
jdbc.xml</descriptor-file-name>
</jdbc-system-resource>
```

Direct access data source

TimesTen Class: `com.timesten.jdbc.TimesTenDriver`

Example URL: `jdbc:timesten:direct:SAMPLEDB_1122`

Example Data Source Configuration File: `TptbmDirectDS-jdbc.xml`

Example data source configuration file

```
<?xml version='1.0' encoding='UTF-8'?>
<jdbc-data-source xmlns="http://xmlns.oracle.com/weblogic/jdbc-data-
source" xmlns:sec="http://xmlns.oracle.com/weblogic/security"
xmlns:wls="http://xmlns.oracle.com/weblogic/security/wls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/jdbc-data-source
http://xmlns.oracle.com/weblogic/jdbc-data-source/1.0/jdbc-data-
source.xsd">
  <name>TptbmDirectDS</name>
  <jdbc-driver-params>
    <url>jdbc:timesten:direct:SAMPLEDB_1122</url>
    <driver-name>com.timesten.jdbc.TimesTenDriver</driver-name>
    <properties>
      <property>
        <name>user</name>
        <value>scott</value>
      </property>
    </properties>
    <password-
encrypted>{AES}q6q4Fxr0eyxqCspjk8aC1LjwChm0r97GqNexeTpdXRk=</password-
encrypted>
  </jdbc-driver-params>
  <jdbc-connection-pool-params>
    <test-table-name></test-table-name>
  </jdbc-connection-pool-params>
  <jdbc-data-source-params>
    <jndi-name>jdbc/TptbmDS</jndi-name>
    <global-transactions-protocol>OnePhaseCommit</global-transactions-
protocol>
```

```
</jdbc-data-source-params>
</jdbc-data-source>
```

Direct access data source with XA distributed transaction support

TimesTen Class: com.timesten.jdbc.xa.TimesTenXADataSource

Example URL: jdbc:timesten:direct:SAMPLEDB_1122

Example Data Source Configuration File: TptbmDirectXADS-jdbc.xml

Example data source configuration file

```
<?xml version='1.0' encoding='UTF-8'?>
<jdbc-data-source xmlns="http://xmlns.oracle.com/weblogic/jdbc-data-
source" xmlns:sec="http://xmlns.oracle.com/weblogic/security"
xmlns:wls="http://xmlns.oracle.com/weblogic/security/wls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/jdbc-data-source
http://xmlns.oracle.com/weblogic/jdbc-data-source/1.0/jdbc-data-
source.xsd">
  <name>TptbmDirectXADS</name>
  <jdbc-driver-params>
    <url>jdbc:timesten:direct:SAMPLEDB_1122</url>
    <driver-name>com.timesten.jdbc.xa.TimesTenXADataSource</driver-name>
    <properties>
      <property>
        <name>user</name>
        <value>scott</value>
      </property>
    </properties>
    <password-
encrypted>{AES}vL3PUTsfZ6d9RbAc3mXRqtBeRdj1A77qMnehvGJnkZc=</password-
encrypted>
  </jdbc-driver-params>
  <jdbc-connection-pool-params>
    <test-table-name></test-table-name>
  </jdbc-connection-pool-params>
  <jdbc-data-source-params>
    <jndi-name>jdbc/TptbmDS</jndi-name>
    <global-transactions-protocol>OnePhaseCommit</global-transactions-
protocol>
  </jdbc-data-source-params>
</jdbc-data-source>
```

Client/Server access data source

TimesTen Class: com.timesten.jdbc.TimesTenDriver

Example URL: jdbc:timesten:client:SAMPLEDBCS_1122

Example Data Source Configuration File: TptbmClientDS-jdbc.xml

Example data source configuration file

```
<?xml version='1.0' encoding='UTF-8'?>
<jdbc-data-source xmlns="http://xmlns.oracle.com/weblogic/jdbc-data-
source" xmlns:sec="http://xmlns.oracle.com/weblogic/security"
xmlns:wls="http://xmlns.oracle.com/weblogic/security/wls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/jdbc-data-source
http://xmlns.oracle.com/weblogic/jdbc-data-source/1.0/jdbc-data-
source.xsd">
  <name>TptbmClientDS</name>
  <jdbc-driver-params>
    <url>jdbc:timesten:client:SAMPLEDBCS_1122</url>
    <driver-name>com.timesten.jdbc.TimesTenDriver</driver-name>
    <properties>
      <property>
        <name>user</name>
        <value>scott</value>
      </property>
    </properties>
    <password-
encrypted>{AES}+0Mm4McqIl5+U1TmjtIk9M6lt3n/6EPJGodOylcYRQg=</password-
encrypted>
  </jdbc-driver-params>
  <jdbc-connection-pool-params>
    <test-table-name></test-table-name>
  </jdbc-connection-pool-params>
  <jdbc-data-source-params>
    <jndi-name>jdbc/TptbmDS</jndi-name>
    <global-transactions-protocol>OnePhaseCommit</global-transactions-
protocol>
  </jdbc-data-source-params>
</jdbc-data-source>
```

Client/Server access data source with XA distributed transaction support

TimesTen Class: com.timesten.jdbc.xa.TimesTenXADataSource

Example URL: jdbc:timesten:client:SAMPLEDBCS_1122

Example Data Source Configuration File: TptbmClientXADS-jdbc.xml

Example data source configuration file

```
<?xml version='1.0' encoding='UTF-8'?>
<jdbc-data-source xmlns="http://xmlns.oracle.com/weblogic/jdbc-data-
source" xmlns:sec="http://xmlns.oracle.com/weblogic/security"
xmlns:wls="http://xmlns.oracle.com/weblogic/security/wls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/jdbc-data-source
http://xmlns.oracle.com/weblogic/jdbc-data-source/1.0/jdbc-data-
source.xsd">
  <name>TptbmClientXADS</name>
  <jdbc-driver-params>
    <url>jdbc:timesten:client:SAMPLEDBCS_1122</url>
```

```

    <driver-name>com.timesten.jdbc.xa.TimesTenXADataSource</driver-name>
    <properties>
      <property>
        <name>user</name>
        <value>scott</value>
      </property>
    </properties>
    <password-
encrypted>{AES}v5eyhu0ZOM05L5cAwyiuatOa/ZYLoYV4czB+/cbIUvo=</password-
encrypted>
    </jdbc-driver-params>
    <jdbc-connection-pool-params>
    <test-table-name></test-table-name>
    </jdbc-connection-pool-params>
    <jdbc-data-source-params>
    <jndi-name>jdbc/TptbmDS</jndi-name>
    <global-transactions-protocol>OnePhaseCommit</global-transactions-
protocol>
    </jdbc-data-source-params>
  </jdbc-data-source>

```

Using Weblogic Server JPA with TimesTen

Weblogic Server supports the Java Persistence API (JPA) through the EclipseLink object-relational mapping framework. TimesTen supports EclipseLink applications. See the TimesTen for EclipseLink section later in this document and the TimesTen Quick Start Guide for additional information.

TIMESTEN FOR ORACLE GLASSFISH SERVER 3.1

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with Oracle GlassFish Server 3.1. This document assumes that Oracle TimesTen and Oracle GlassFish Server have been installed on the same host machine.

Configuring the TimesTen JDBC driver

Before you can use Oracle GlassFish Server to access TimesTen databases, the server environment must have access to the TimesTen JDBC driver jar file and the TimesTen shared libraries.

1. Copy the TimesTen JDBC driver jar file called `ttjdbc6.jar` from `tt_install_dir/lib` to the `gs_install_dir/glassfish/domains/gs_domain/lib` directory.
2. Set the `LD_LIBRARY_PATH` environment variable (or the equivalent variable for your OS) to include the `tt_install_dir/lib` directory. On Windows platforms, set the `PATH` environment variable to the `tt_install_dir/bin` directory where TimesTen DLLs are located. This variable must be set for the environment where GlassFish Server runs.
3. Restart the GlassFish Server.

Configuring TimesTen connections

The TimesTen JDBC driver supports four different types of database connections.

- **Direct access with local transactions only** – This configuration provides the fastest database performance. The TimesTen database must reside on the same machine as GlassFish Server.
- **Direct access with XA distributed transaction support** - The driver uses JTA to supports distributed transactions. The TimesTen database must reside on the same machine as GlassFish Server.
- **Client/server access with local transactions only** - The TimesTen database can reside on a remote machine that is running the TimesTen server.
- **Client/server access with XA distributed transaction support** - The driver uses JTA to supports distributed transactions. The TimesTen database can reside on a remote machine that is running the TimesTen server.

Properties of TimesTen connections

When configuring TimesTen connections in GlassFish Server use the following settings.

Direct access with local transactions only:

TimesTen Class: `com.timesten.jdbc.ObservableConnectionDS`

Example URL Property: `jdbc:timesten:direct:SAMPLEDB_1122`

Direct access with XA distributed transaction support:

TimesTen Class: `com.timesten.jdbc.xa.TimesTenXADataSource`

Example URL Property: `jdbc:timesten:direct:SAMPLEDB_1122`

Client/server access with local transactions only:

TimesTen Class: `com.timesten.jdbc.ObservableConnectionDS`

Example URL Property: `jdbc:timesten:client:SAMPLEDBCS_1122`

Client/server access with XA distributed transaction support:

TimesTen Class: `com.timesten.jdbc.xa.TimesTenXADataSource`

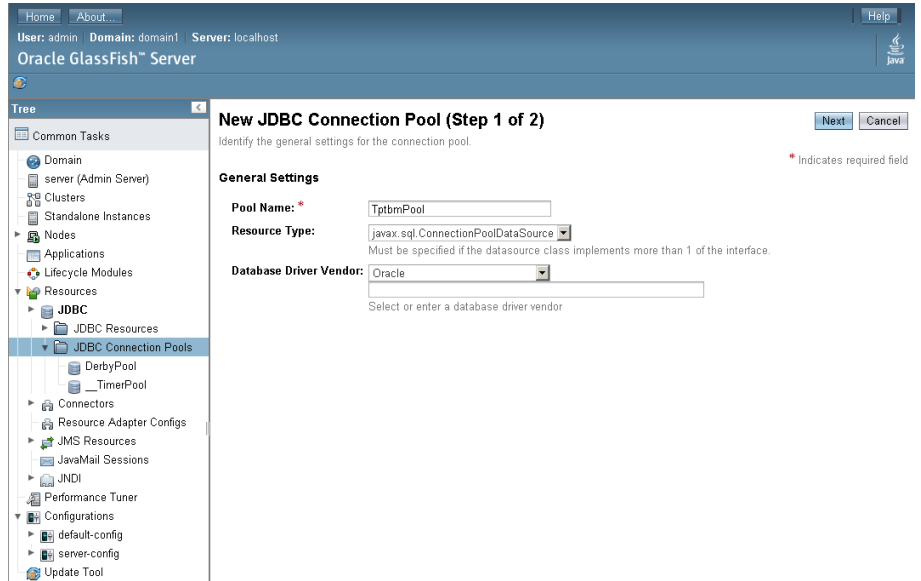
Example URL Property: `jdbc:timesten:client:SAMPLEDBCS_1122`

Configuring a TimesTen JDBC connection pool

To configure a TimesTen JDBC connection pool in GlassFish Server follow these steps.

1. Connect to the GlassFish Server administration console.
2. Select **Resources->JDBC->Connection Pools** from the left hand pane.
3. Click **New**.
4. At the **New JDBC Connection Pool (Step 1 of 2)** screen enter a name for the new pool in the **Name** field. In the **Resource Type** drop down list box select `javax.sql.ConnectionPoolDataSource` if the application requires only local transaction support. If the application requires XA distributed transaction support then select `javax.sql.XADataSource` instead. In the **Database Driver Vendor** field select 'Oracle' from the drop down list box. The screen should look similar to figure 4.1.

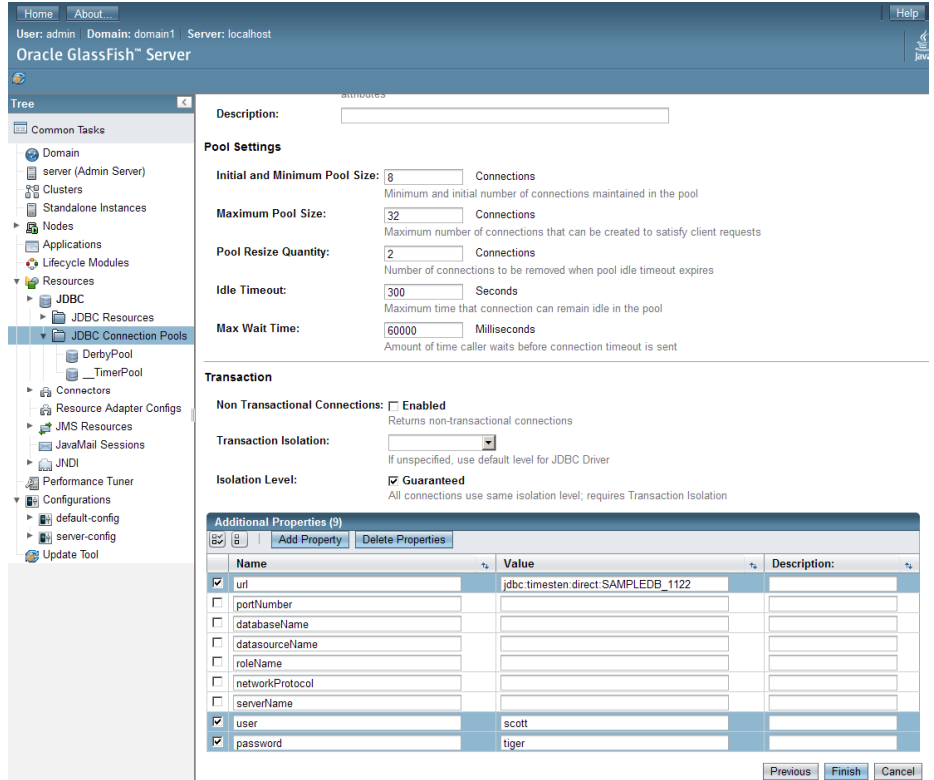
Figure 4.1 New Connection Pool Screen



Click **Next**.

- At the **New Connection Pool (Step 2 of 2)** screen enter `com.timesten.jdbc.ObservableConnectionDS` in the **Datasource Classname** field for local transactions. Enter `com.timesten.jdbc.xa.TimesTenXADataSource` if XA distributed transactions are required.
- Scroll down to the **Additional Properties** section at the bottom of the screen. Click **Add Property**. A new row will appear. Click the check box in the new row. Type 'url' in the **Name** column. In the **Value** column enter the URL for the TimesTen database associated with the connection pool. A TimesTen URL takes the form:
`jdbc:timesten:[direct|client]:DSN`. Specify the name of the TimesTen database user for the connection in the **Value** field of the 'user' property. Specify the TimesTen user's password in the **Value** field for the 'password' property. The screen should look similar to figure 4.2 for a connection to a direct DSN named `SAMPLEDB_1122` using the TimesTen user account `scott/tiger`.

Figure 4.2 Properties screen



Click **Finish**. The TimesTen connection pool is now configured.

Configuring a TimesTen JDBC resource

In order for applications to access the TimesTen database a JDBC resource must also be created and associated with the TimesTen connection pool. Follow these steps.

1. Connect to the GlassFish Server administration console.
2. Select **Resources->JDBC->JDBC Resources** from the left hand pane.
3. Click **New**.
4. At the **New JDBC Resource** screen enter a JNDI name for the resource. In the **Pool Name** field select the name of the TimesTen connection pool

created previously. The screen should look similar to figure 4.3 given a TimesTen connection pool called TptbmPool.

Figure 4.3 JDBC Resources screen



Click **OK**. The TimesTen database configuration for GlassFish Server is now complete.

Using GlassFish Server JPA with TimesTen

GlassFish Server uses the EclipseLink object-relational mapping framework to support the Java Persistence API (JPA). TimesTen supports EclipseLink applications. See the TimesTen for EclipseLink section later in this document and the TimesTen Quick Start Guide for additional information.

TIMESTEN FOR WEBSHERE APPLICATION SERVER 7

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with IBM WebSphere Application Server 7.0.0.11. This document assumes that Oracle TimesTen and WebSphere Application Server have been installed on the same machine.

Configuring TimesTen connections

The TimesTen JDBC driver supports four different types of database connections.

- **Direct access with local transactions only** - This configuration provides the fastest database performance. The TimesTen database must reside on the same machine as WebSphere Application Server.
- **Direct access with XA distributed transaction support** - The driver uses JTA to support distributed transactions. The TimesTen database must reside on the same machine as WebSphere Application Server.
- **Client/server access with local transactions only** - The TimesTen database can reside on a remote machine that is running the TimesTen server.
- **Client/server access with XA distributed transaction support** - - The driver uses JTA to support distributed transactions. The TimesTen database can reside on a remote machine that is running the TimesTen server.

Properties of TimesTen connections

When configuring TimesTen connections in WebSphere use the following settings.

Direct access with local transactions only:

TimesTen Class: `com.timesten.jdbc.ObservableConnectionDS`

Example URL Property: `jdbc:timesten:direct:SAMPLEDB_1122`

Direct access with XA distributed transaction support:

TimesTen Class: `com.timesten.jdbc.xa.TimesTenXADataSource`

Example URL Property: `jdbc:timesten:direct:SAMPLEDB_1122`

Client/server access with local transactions only:

TimesTen Class: `com.timesten.jdbc.ObservableConnectionDS`

Example URL Property: `jdbc:timesten:client:SAMPLEDBCS_1122`

Client/server access with XA distributed transaction support:

TimesTen Class: `com.timesten.jdbc.xa.TimesTenXADataSource`

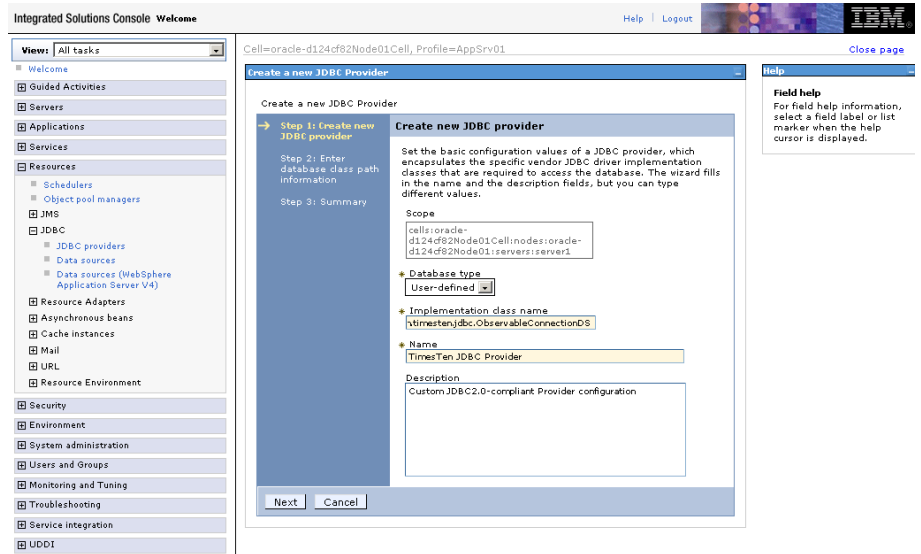
Example URL Property: `jdbc:timesten:client:SAMPLEDBCS_1122`

Configuring a TimesTen JDBC provider

Before you can use WebSphere to access TimesTen databases the server environment must have access to the TimesTen JDBC driver jar file and the native shared libraries. In WebSphere this can be done by configuring a JDBC provider in the WebSphere administration console. Follow these steps.

1. Connect to the WebSphere Administrative Console.
2. Select **Resources->JDBC->JDBC Providers**.
3. Select a scope from the drop down list box.
4. Click **New**.
5. At the **Create new JDBC provider** screen select 'User-defined' for **Database type**. If the application requires distributed XA transaction support then enter `com.timesten.jdbc.xa.TimesTenXADataSource` in the **Implementation class name** field. Otherwise enter `com.timesten.jdbc.ObservableConnectionDS` in this field. Enter a name for the configuration in the **Name** field. The screen should look similar to figure 5.1.

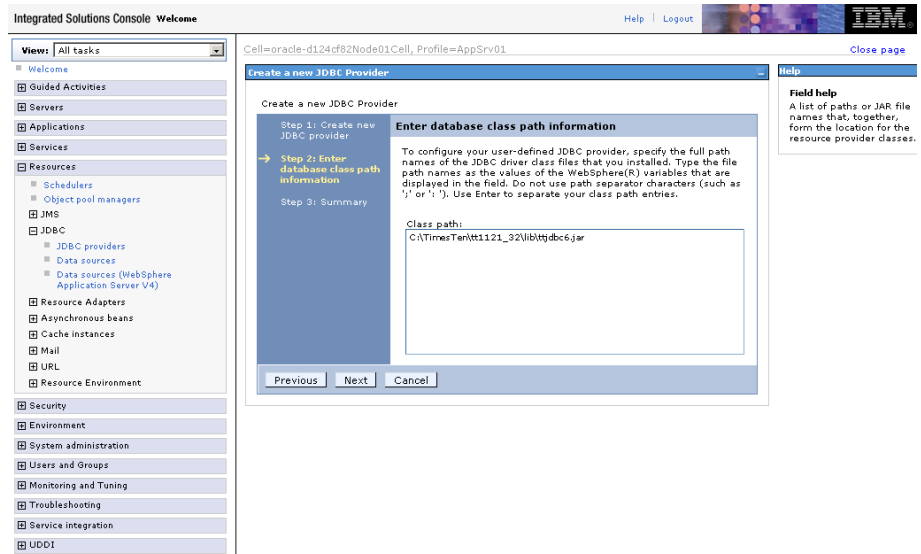
Figure 5.1 Create new JDBC provider screen



Click **Next**.

- At the **Enter database class path information** screen enter the path to the TimesTen JDBC driver located at `tt_install_dir/lib/ttjdbc6.jar`. The screen should look similar to figure 5.2.

Figure 5.2 Enter database class path information screen

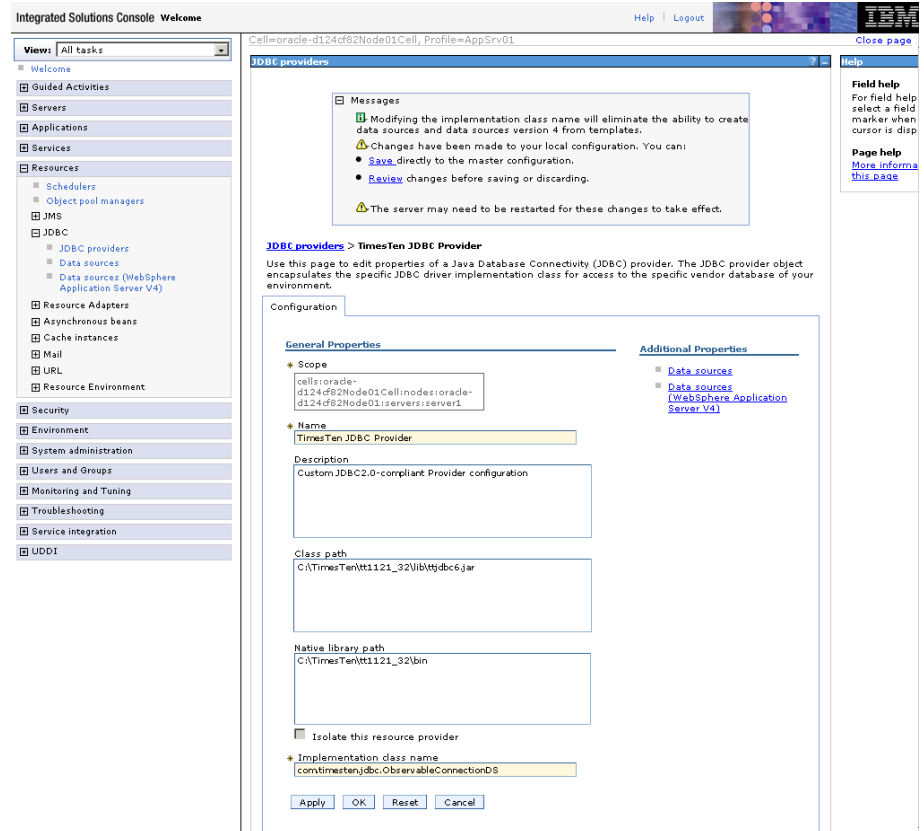


Click **Next**.

- On the summary screen click **Finish**.

8. At the **JDBC providers** screen click the TimesTen provider name that was just created.
9. At the **General properties** screen enter the path to the TimesTen shared libraries in the **Native library path** field. On UNIX platforms the path is `tt_install_dir/lib`. On Windows platforms the path is `tt_install_dir/bin`. The screen should look similar to figure 5.3.

Figure 5.3 General properties screen



Click **Apply**.

10. Click **Save** to save the new configuration.

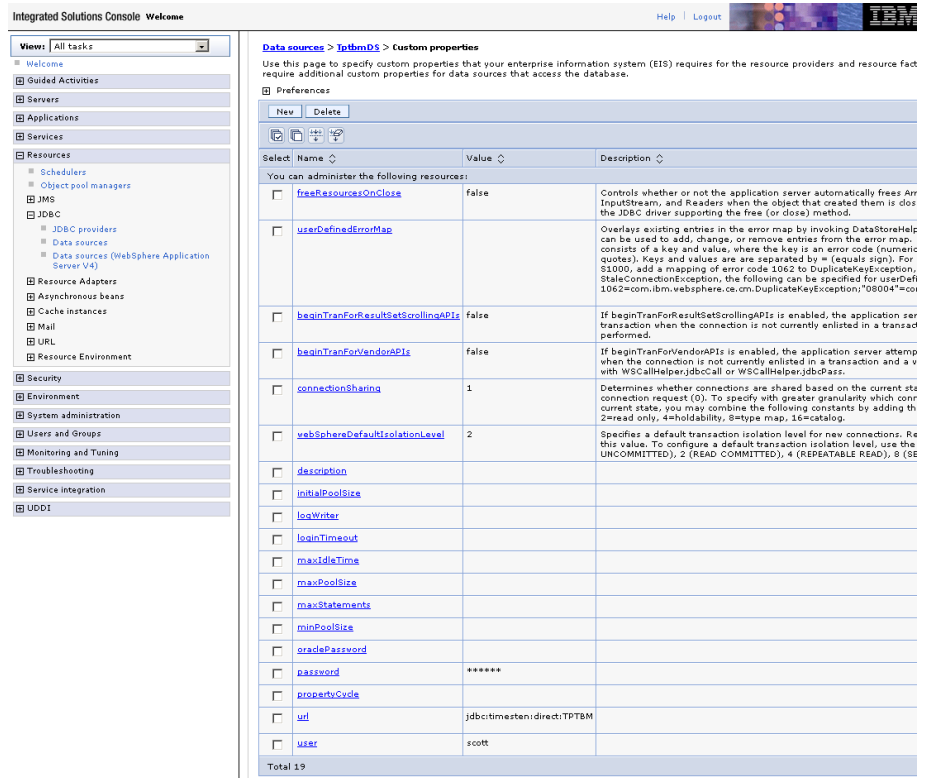
Configuring TimesTen data sources

WebSphere data sources are associated with specific JDBC providers defined in the previous section. Follow these steps to configure TimesTen data sources from the WebSphere administration console.

1. Connect to the WebSphere administration console.
2. Select **Resources->JDBC->Data sources**.
3. Select a scope from the drop down list box.

4. Click the **New** button.
5. At the **Enter basic data source information** screen enter a name and a JNDI name for the data source. Click **Next**.
6. At the **Select JDBC provider** screen select the TimesTen JDBC Provider created in the previous section. Click **Next**.
7. At the **Enter database specific properties for the data source** screen use
use
`com.ibm.websphere.rsadapter.GenericDataStoreHelper` for the **Data store helper class name** field. Make sure that the **Use this database in container managed persistence (CMP)** check box is checked. Click **Next**.
8. On the **Setup security aliases** screen click **Next**.
9. On the following summary screen click **Finish**.
10. At the **Data sources** screen click on the name of the TimesTen data source that was just created.
11. Click the **Custom properties** link.
12. Click the **url** property.
13. In the **value** field enter the TimesTen JDBC URL for the DSN that this data source should connect to. A TimesTen URL takes the form:
`jdbc:timesten:<direct|client>:<DSN>`. Click **OK**.
14. Click the **user** property and enter the TimesTen database user name for the connection in the **value** field. Click **OK**.
15. Click the **password** property and enter the TimesTen database password for the connection in the **value** field. Click **OK**.
16. Click the **webSphereDefaultIsolationLevel** property and enter '2' in the **value** field. Click **OK**.
17. For a direct DSN the screen should look similar to figure 5.4.

Figure 5.4 Data Sources Custom properties screen



18. Click the **Save** link near the top of the screen to save the new configuration.

19. Restart the Websphere server before attempting to make connections to the new TimesTen data source.

Using WebSphere Application Server JPA with TimesTen

WebSphere Server supports the Java Persistence API (JPA) using the OpenJPA object-relational mapping framework. TimesTen supports OpenJPA applications. See the TimesTen for OpenJPA section later in this document and the TimesTen Quick Start Guide for additional information.

TIMESTEN FOR ECLIPSELINK 2

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with EclipseLink 2.1.1 JPA. EclipseLink is a further development of the Oracle TopLink object-relational mapping framework. This document assumes that you have installed Oracle TimesTen and EclipseLink on the same machine.

Configuring the TimesTen JDBC driver for EclipseLink

When accessing a TimesTen database from an EclipseLink application, you must set the CLASSPATH environment variable to include the TimesTen JDBC driver jar file. When using the Java 5 runtime add `tt_install_dir/lib/ttjdbc5.jar` to the CLASSPATH. When using the Java 6 runtime add the `tt_install_dir/lib/ttjdbc6.jar` file to the CLASSPATH.

Set the LD_LIBRARY_PATH (or equivalent for your OS) environment variable to include the path to the TimesTen shared libraries located at `tt_install_dir/lib`. On Windows platforms the PATH environment variable should be set to the `tt_install_dir/bin` directory where TimesTen DLLs are located.

The EclipseLink distribution includes a platform class called `org.eclipse.persistence.platform.database.TimesTen7Platform`. This is the recommended platform class for TimesTen EclipseLink applications. An example `persistence-eclipselink2.xml` configuration file for EclipseLink JPA and TimesTen is located under the `quickstart` directory in your TimesTen installation.

Example persistence.xml configuration file for EclipseLink

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"
  version="1.0">
  <persistence-unit name="TptbmEclipseLink" transaction-
type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>com.timesten.tptbmas.Tptbm</class>
    <properties>
      <property name="eclipselink.logging.level" value="INFO"/>
      <property name="eclipselink.platform.class.name"
value="org.eclipse.persistence.platform.database.TimesTen7Platform"/>
      <property name="javax.persistence.jdbc.driver"
value="com.timesten.jdbc.TimesTenDriver"/>
      <property name="javax.persistence.jdbc.url"
value="jdbc:timesten:direct:SAMPLEDB_1122"/>
      <property name="javax.persistence.jdbc.password" value="tiger"/>
    </properties>
  </persistence-unit>
</persistence>
```

```
        <property name="javax.persistence.jdbc.user" value="SCOTT"/>
    </properties>
</persistence-unit>
</persistence>
```

TIMESTEN FOR HIBERNATE 3.5

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with Hibernate 3.5. This document assumes that Oracle TimesTen and Hibernate have been installed on the same machine.

Configuring the TimesTen JDBC driver for Hibernate

When accessing a TimesTen database from a Hibernate application, you must set the CLASSPATH environment variable to include the TimesTen JDBC driver jar file. When using the Java 5 runtime add `tt_install_dir/lib/ttjdbc5.jar` to the CLASSPATH. When using the Java 6 runtime add the `tt_install_dir/lib/ttjdbc6.jar` file to the CLASSPATH.

Set the LD_LIBRARY_PATH (or equivalent for your OS) environment variable to include the path to the TimesTen shared libraries located at `tt_install_dir/lib`. On Windows platforms the PATH environment variable should be set to the `tt_install_dir/bin` directory where TimesTen DLLs are located.

A Hibernate SQL dialect class called `org.hibernate.dialect.TimesTenDialect1122` has been developed for use with TimesTen. This dialect is optimized for the latest SQL features of TimesTen and it is the recommended dialect for TimesTen Hibernate applications. This dialect is not included in current versions of the Hibernate distribution. The dialect is available as part of the TimesTen Quick Start example programs that are included as an optional component of a TimesTen installation. The java source code for the dialect is located in the `tt_install_dir/quickstart/sample_code/orm/config/hibernate3` directory of the TimesTen installation. This directory includes an Ant build script and instructions for compiling and using the dialect in TimesTen Hibernate applications.

Hibernate configuration properties for TimesTen

Hibernate uses properties defined in a `hibernate.properties` file or a `hibernate.cfg.xml` configuration file to determine which JDBC driver and which SQL dialect class to use for a session.

When using TimesTen with Hibernate, set the following property values:

```
hibernate.dialect=org.hibernate.dialect.TimesTenDialect1122
hibernate.connection.driver_class=com.timesten.jdbc.TimesTenDriver
hibernate.connection.url=jdbc:timesten:<direct|client>:<DSN>
```

The following property values are recommended when using TimesTen with Hibernate:

```
hibernate.connection.isolation=2
hibernate.jdbc.use_get_generated_keys=false
hibernate.jdbc.use_scrollable_resultset=false
hibernate.jdbc.batch_size=256
```

An example `hibernate.cfg.xml` configuration file is located under the `quickstart` directory in your TimesTen installation.

Example of `hibernate.cfg.xml` configuration file

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

    <!-- a SessionFactory instance -->
    <session-factory name="Tptbm">

        <!-- properties -->
        <property name="hibernate.connection.url">
            jdbc:timesten:direct:SAMPLEDB_1122
        </property>
        <property name="hibernate.connection.username">SCOTT</property>
        <property name="hibernate.connection.password">tiger</property>
        <property name="hibernate.connection.driver_class">
            com.timesten.jdbc.TimesTenDriver
        </property>
        <property name="hibernate.dialect">
            org.hibernate.dialect.TimesTenDialect1122
        </property>
        <!-- Connection.TRANSACTION_READ_COMMITTED = 2 -->
        <property name="hibernate.connection.isolation">2</property>
        <property name="hibernate.jdbc.fetch_size">32</property>
        <property name="hibernate.jdbc.batch_size">256</property>
        <property name="hibernate.jdbc.batch_versioned_data">
            True
        </property>
        <property name="hibernate.jdbc.use_streams_for_binary">
            False
        </property>
        <property name="hibernate.jdbc.use_get_generated_keys">
            False
        </property>
        <property name="hibernate.jdbc.use_scrollable_resultset">
            False
        </property>
        <property name="hibernate.cache.use_query_cache">
            False
        </property>
        <property name="hibernate.cache.use_second_level_cache">
            False
        </property>
    </session-factory>
</hibernate-configuration>
```

```

        <property name="hibernate.show_sql">false</property>
        <property name="hibernate.connection.pool_size">4</property>

        <!-- mapping file -->
        <mapping resource="META-INF/Tptbm.hbm.xml"/>

    </session-factory>
</hibernate-configuration>

```

When using Hibernate with the JPA API the same TimesTen properties can be specified in the `persistence.xml` configuration file. An example JPA configuration file (`persistence-hibernate3.xml`) for a TimesTen Hibernate application is located under the `quickstart` directory in your TimesTen installation.

Example of the `persistence.xml` configuration file

```

<?xml version="1.0" encoding="UTF-8"?>

<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"
             version="1.0">

    <persistence-unit name="TptbmHibernate" transaction-
type="RESOURCE_LOCAL">

        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <class>com.timesten.tptbmas.Tptbm</class>

        <properties>
            <property name="hibernate.connection.url"
                value="jdbc:timesten:direct:SAMPLEDB_1122"/>

            <property name="hibernate.connection.username" value="SCOTT"/>
            <property name="hibernate.connection.password" value="tiger"/>

            <property name="hibernate.connection.driver_class"
                value="com.timesten.jdbc.TimesTenDriver"/>

            <property name="hibernate.dialect"
                value="org.hibernate.dialect.TimesTenDialect1122"/>

            <!-- Connection.TRANSACTION_READ_COMMITTED = 2 -->
            <property name="hibernate.connection.isolation" value="2"/>

            <property name="hibernate.jdbc.fetch_size" value="32"/>
            <property name="hibernate.jdbc.batch_size" value="256"/>
            <property name="hibernate.jdbc.batch_versioned_data"
value="true"/>

            <property name="hibernate.jdbc.use_streams_for_binary"
                value="false"/>
            <property name="hibernate.jdbc.use_get_generated_keys"
                value="false"/>
            <property name="hibernate.jdbc.use_scrollable_resultset"
                value="false"/>

            <property name="hibernate.cache.use_query_cache" value="false"/>
            <property name="hibernate.cache.use_second_level_cache"

```

```
        value="false"/>
        <property name="hibernate.show_sql" value="false"/>
        <property name="hibernate.connection.pool_size" value="4"/>
    </properties>
</persistence-unit>
</persistence>
```

TIMESTEN FOR OPENJPA 2

This section provides descriptions and examples of configuring the TimesTen JDBC driver for use with OpenJPA 2.0.1. This document assumes that Oracle TimesTen and OpenJPA have been installed on the same machine.

Configuring the TimesTen JDBC driver for OpenJPA

When accessing a TimesTen database from an OpenJPA application, you must set the CLASSPATH environment variable to include the TimesTen JDBC driver jar file. When using the Java 5 runtime add `tt_install_dir/lib/ttjdbc5.jar` to the CLASSPATH. When using the Java 6 runtime add the `tt_install_dir/lib/ttjdbc6.jar` file to the CLASSPATH.

Set the LD_LIBRARY_PATH (or equivalent for your OS) environment variable to include the path to the TimesTen shared libraries located at `tt_install_dir/lib`. On Windows platforms the PATH environment variable should be set to the `tt_install_dir/bin` directory where TimesTen DLLs are located.

OpenJPA can be configured with a property called `openjpa.jdbc.DBDictionary`, which defines a SQL dialect for a database provider. For TimesTen connections the recommended value for this property is 'oracle'. An example `persistence-openjpa2.xml` configuration file for OpenJPA and TimesTen is located under the `quickstart` directory in your TimesTen installation.

Example persistence.xml configuration file for OpenJPA

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"
             version="1.0">
  <persistence-unit name="TptbmOpenJPA" transaction-
type="RESOURCE_LOCAL">
    <provider>org.apache.openjpa.persistence.PersistenceProviderImpl</provid
er>
    <class>com.timesten.tptbmas.Tptbm</class>
    <class>com.timesten.tptbmas.TptbmPKey</class>

    <properties>
      <property name="openjpa.ConnectionUserName" value="SCOTT"/>
      <property name="openjpa.ConnectionPassword" value="tiger"/>
      <property name="openjpa.ConnectionURL"
value="jdbc:timesten:direct:SAMPLEDB_1122"/>
      <property name="openjpa.ConnectionDriverName"
value="com.timesten.jdbc.TimesTenDriver"/>

      <property name="openjpa.jdbc.DBDictionary" value="oracle"/>
      <property name="openjpa.ConnectionRetainMode" value="always"/>
      <property name="openjpa.Multithreaded" value="true"/>
    </properties>
  </persistence-unit>
</persistence>
```

```
    </properties>  
  </persistence-unit>  
</persistence>
```




Configuring Oracle TimesTen 11.2.2 for Application Servers and Object-Relational Mapping Frameworks
January 2012

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2012, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.