Oracle® TimesTen In-Memory Database

Deploy TimesTen JDBC Driver with Web Applications

Release 22.1

F87406-03

May 2024

Deploy TimesTen JDBC Driver with Web Applications

This document explains how to deploy the TimesTen JDBC driver in the WAR file of a web application. The information in this document applies to TimesTen 22 (version 22.1.1.20.0 and higher) and TimesTen 18.1 (version 18.1.4.40.0 and higher).

Background

Web applications packaged and deployed as WAR files may include a JDBC driver JAR library in the `WEB-INF/lib` folder to enable database access. This JDBC driver is typically a Java Type 4 driver.

The TimesTen JDBC driver is a Type 1 driver that relies on native code libraries to access the database. Previously, in order to enable TimesTen access within application servers like Tomcat, the following steps were required:

- Installing TimesTen on the application server.
- Configuring the application server to access the TimesTen JDBC driver JAR file. For Tomcat, this involves copying the TimesTen JDBC driver JAR file to the application server `/lib` directory.
- Configuring connection pool services for TimesTen JDBC driver connections used by web applications.
- In Linux, it is necessary to set `LD_LIBRARY_PATH` or `java.library.path` (or `DYLD_LIBRARY_PATH` on Mac platforms) or `java.library.path` to include the `/lib` directory of the TimesTen installation.

These requirements may be prohibitive, especially when deploying on multiple servers. TimesTen now provides a JDBC driver feature that eliminates these requirements by packaging both the Java and the native components of the driver entirely within an application WAR file.

Process for Deploying TimesTen JDBC Driver WAR

Developers of web applications that access TimesTen 18.1 or higher may now take the following actions to include the client/server version of the TimesTen JDBC driver within the application WAR file:

1. Use the `ttmkLiteClient` script from a TimesTen instance to copy the set of required driver files.
2. Package the driver file set in the WAR file.
3. Add application code or preconfigure the application server to set the `com.timesten.jdbc.client.use.single.shared.library` and `java.library.path` properties before connecting to TimesTen. Alternatively, set the `com.timesten.jdbc.shared.library` property before connecting to TimesTen.

These actions allow a deployed web application to access TimesTen from an application server without any prior configuration.

To learn how to encrypt client/server communication with your web application and create certificates and configure TLS, see Transport Layer Security for TimesTen Client/Server in the *Oracle TimesTen In-Memory Database Security Guide*. TLS requires TimesTen release 18.1.3.2.0 or higher.

Task 1: Create the Driver File Set

To install TimesTen, create a TimesTen client instance, and produce the set of TimesTen JDBC driver files for an application WAR file. Perform the following steps on a single machine that matches the platform running the target application server:

1. Install TimesTen by unzipping the distribution file. The umask setting is recommended by TimesTen so that installation files have appropriate permissions. This example assumes that the TimesTen installer is in the `/sw/ttinstallers` directory.

   ```
   % cd /sw
   % umask 022
   % mkdir ttinstalldir
   % cd ttinstalldir
   % unzip /sw/ttinstallers/timesten2211200.server.linux8664.zip
   ```

   Note that the distribution files on the Mac platforms use `macos64`, for example, `timesten1814420.client.macos64.zip` or `timesten2211230.client.macos64.zip`.

2. From the installation `bin` directory (`/sw/ttinstalldir/tt22.1.1.20.0/bin`), use the `ttInstanceCreate` utility to create a TimesTen client instance. Use the default values when prompted as shown below.

   ```
   % ./ttInstanceCreate -clientonly -name client -location /sw/
   ttinstancedir
   Creating instance in /sw/ttinstancedir/client ...
   The 22.1 Release Notes are located here :
   '/sw/ttinstalldir/tt22.1.1.20.0/README.html'
   ```

```
Instance created successfully.
```

The location you specify (for example, `/sw/ttinstancedir`) must already exist. Ensure that the instance directory is created and has the same name (for example, `client`) that you specified while running the `ttInstanceCreate` utility.

3. Configure the environment for the TimesTen installation by sourcing the `ttenv.csh` or `ttenv.sh` script file located in the `bin` directory (`/sw/ttinstancedir/client/bin`).
4. Execute the `ttmkLiteClient` script, located in the instance `install/support` directory, specifying the desired destination directory for the client driver files. The destination directory is created. Optionally, specify the JDK version for the TimesTen JDBC driver classes. JDK 8 is the default version. See *Oracle TimesTen In-Memory Database Release Notes* for information about JDK versions supported by TimesTen.
   The destination directory should be located in or accessible from the application development environment so that it can be packaged into the WAR file.

   ```
   % /sw/ttinstancedir/client/install/support/ttmkLiteClient -jdk
   8 /app/src/main/resources/liteclient
   ```

   Usage notes:

   ```
   usage: ttmkLiteClient [options] destination-directory
   options are:
   -jdk VERSION (defaults to 8)
   -verbose
   Copies files required to include a TimesTen JDBC client library in
   a WAR file.
   ```

5. When you create and copy certificates, the `clientWallet` directory is created under the TimesTen `conf` directory (`conf/mywallets/clientWallet`). To learn how to encrypt client/server communication with your web application and create certificates and configure TLS, see Transport Layer Security for TimesTen Client/Server in the *Oracle TimesTen In-Memory Database Security Guide*.

   Note:

   If you intend to use TLS for encrypted client/server connections, also include the `clientWallet` directory containing TLS certificates in the driver file set.

The destination directory or lite client root directory (for example, `/app/src/main/resources/liteclient`) contains:

- `conf`: Directory containing the connection configuration files (and optionally includes a subdirectory `clientWallet` for TLS under the `conf` directory (`conf/mywallets/clientWallet`)

- `libttJdbcCS_all.so` (or `libttJdbcCS_all.dylib` on the Mac platform): In TimesTen 22.1 and 18.1, this is the native shared library. In TimesTen 22.1, there are six other shared libraries in `/app/src/main/resources/liteclient/lib`:

Linux platform	Mac platform
`libccme_asym.so`	`libccme_asym.dylib`
`libccme_base.so`	`libccme_base.dylib`
`libccme_base_non_fips.so`	`libccme_base_non_fips.dylib`
`libccme_ecc.so`	`libccme_ecc.dylib`
`libccme_ecc_non_fips.so`	`libccme_ecc_non_fips.dylib`
`libcryptocme.so`	`libcryptocme.dylib`

- `manifest.txt`: Information regarding the source of the file set
- `nls`: Directory containing language and character set data files
- The TimesTen JDBC driver for your selected JDK version: Not all JDK releases are supported. For TimesTen release 18.1 after JDK 10 and for TimesTen release 22.1 in general, only Long-Term Support (LTS) JDK releases are supported. If the JDK version selected is not supported, `ttmkLiteClient` selects the JDBC driver for the most recent but earlier supported JDK release. The filename will be `ttjdbc`*`version`*`.jar`, where *`version`* reflects the target JDK version. In the following examples, `ttjdbc11.jar` is returned for JDK 11 and unsupported JDK 14. Because JDK 14 is unsupported, `ttmkLiteClient` falls back to JDK 11:

```
sh-4.4$ /sw/ttinstancedir/client/install/support/ttmkLiteClient -
jdk 11 /app/src/main/resources/liteclient
sh-4.4$ ls /app/src/main/resources/liteclient/tt*
/app/src/main/resources/liteclient/ttjdbc11.jar

sh-4.4$ /sw/ttinstancedir/client/install/support/ttmkLiteClient -
jdk 14 /app/src/main/resources/liteclient
Unsupported Java version 14, falling back to 11
sh-4.4$ ls /app/src/main/resources/liteclient/tt*
/app/src/main/resources/liteclient/ttjdbc11.jar
```

  - The JAR files available in release 22.1.1: `ttjdbc8.jar`, `ttjdbc11.jar`, `ttjdbc17.jar`, and `ttjdbc21.jar`
  - The JAR files available in release 18.1.4: `ttjdbc8.jar`, `ttjdbc9.jar`, `ttjdbc10.jar`, `ttjdbc11.jar`, `ttjdbc17.jar`, and `ttjdbc21.jar`

Task 2: Package the Driver File Set in the WAR file

The application build process requires modifications during packaging to include the TimesTen JDBC driver file set in the WAR file.

1. Copy the driver files directory (created by the `ttmkLiteClient` script) to a `WEB-INF` subdirectory within the WAR file. The driver files should be placed there for security purposes. Resources within `WEB-INF` are not part of the public document tree.
2. Exclude the TimesTen JDBC driver JAR file from Step 1. Instead, the driver JAR file must be copied to the `WEB-INF/lib` directory. When the application starts, the JDBC driver classes are automatically located on the classpath.

For a project managed by Maven, these steps could be implemented as shown below. This example uses the JDBC Driver for JDK 8 (`ttjdbc8.jar`). In this example, the driver files produced by the `ttmkLiteClient` script are copied from the directory `/app/src/main/resources/liteclient` to `/WEB-INF/liteclient` and `/WEB-INF/lib` in the WAR file.

```
<resources>
  <resource>
    <targetPath>${project.build.directory}/${project.name}/WEB-INF/
liteclient</targetPath>
    <filtering>false</filtering>
    <directory>${basedir}/src/main/resources/liteclient</directory>
    <excludes>
      <exclude>ttjdbc8.jar</exclude>
    </excludes>
  </resource>
  <resource>
    <targetPath>${project.build.directory}/${project.name}/WEB-INF/
lib</targetPath>
    <filtering>false</filtering>
    <directory>${basedir}/src/main/resources/liteclient</directory>
    <includes>
      <include>ttjdbc8.jar</include>
    </includes>
  </resource>
</resources>
```

Task 3: Set JVM Properties

When the web application is deployed to the application server, two JVM properties must be set before the application can successfully connect to a TimesTen database. You can either statically configure the application server environment before the server starts, or dynamically configure the server at runtime before the first connection to TimesTen. Both methods are described here.

1. Set the `com.timesten.jdbc.client.use.single.shared.library` system property to `true`. This directs the TimesTen JDBC driver to load a native shared library named `libttJdbcCS_all.so`. This is the native library copied by the `ttmkLiteClient` script.

To set the property, you can modify the Java command that starts the application server to include the following option:

```
-Dcom.timesten.jdbc.client.use.single.shared.library=true
```

Alternatively, at runtime, the web application could execute this code after it is deployed and before connecting to TimesTen.

```
System.setProperty("com.timesten.jdbc.client.use.single.shared.libra
ry","true");
```

2. Update the `java.library.path` system property to include the server directory where the `libttJdbcCS_all.so` shared library is deployed.
   For example, when a web application is deployed to Tomcat, the WAR files are unpacked into an application directory in the Tomcat `webapps` directory. If the JDBC driver file set was packaged in the `WEB-INF/liteclient` directory, then the full path to the server directory that contains the `libttJdbcCS_all.so` library is:

   ```
   tomcat_dir/webapps/application_dir/WEB-INF/liteclient
   ```

   This directory must be included in the `java.library.path` system property before any TimesTen connection is established.

   To set the property, you can modify the Java command that starts the application server to include the following option:

   ```
   -Djava.library.path=".:tomcat_dir/webapps/application_dir/WEB-INF/
   liteclient"
   ```

   Note:

   On Linux, instead of updating `java.library.path`, you can update `LD_LIBRARY_PATH` (or `DYLD_LIBRARY_PATH` on Mac platform) within the application server's environment. See *Restrictions*.

After both `com.timesten.jdbc.client.use.single.shared.library` and `java.library.path` are set, the web application can proceed to establish connections to a TimesTen database. Both properties should be set only once by the application before the first connection.

Configure TimesTen Connection

You must also configure TimesTen connections. Connection information for a web application can be stored in either of two ways, in a resource definition or in TimesTen configuration files.

In Tomcat, the JDBC driver connection information for a JNDI DataSource is located in the `context.xml` file:

```
<Context>
   <Resource
      type="javax.sql.DataSource"
      name="jdbc/TIMESTEN"
      driverClassName="com.timesten.jdbc.TimesTenDriver"

url="jdbc:timesten:client:TTC_SERVER_DSN=MyDB;TTC_SERVER=myserver;TCP_P
ORT=8113"/>
</Context>
```

In the example above, all required connection information to connect to a TimesTen server DSN called `MyDB` is provided in the resource definition. However, it is also possible to locate some of this connection information in the `sys.odbc.ini` and `sys.ttconnect.ini` configuration files located in the `conf` subdirectory created by the `ttmkLiteClient` script.

The resource definition below contains only the essential information required by Tomcat to establish a TimesTen connection to a client/server DSN called `MyDBCS`.

```
<Context>
   <Resource
      type="javax.sql.DataSource"
      name="jdbc/TIMESTEN"
      driverClassName="com.timesten.jdbc.TimesTenDriver"
      url="jdbc:timesten:client:MyDBCS"/>
</Context>
```

The remaining TimesTen connection parameters are defined in the `sys.odbc.ini` and `sys.ttconnect.ini` configuration files.

Note:

After making changes to the `sys.odbc.ini` file when configuring TimesTen connection using DSN, ensure to refresh the `sys.odbc.ini` file in the `WEB-INF/liteclient` (for the Maven example above, it would be `${basedir}/src/main/resources/liteclient`) directory.

`sys.odbc.ini` file:

```
[ODBC Data Sources]
MyDBCS=TimesTen 22.1 Client Driver
[MyDBCS]
TTC_SERVER_DSN=MyDB
TTC_SERVER=my_server
```

```
ConnectionName=Tomcat
ConnectionCharacterSet=AL32UTF8
WaitForConnect=1
```

`sys.ttconnect.ini` file:

```
[my_server]
Description=TimesTen Server
Network_Address=myserver
TCP_PORT=8113
```

Additional Configuration for TLS

To learn how to encrypt client/server communication with your web application and create certificates and configure TLS, see Transport Layer Security for TimesTen Client/Server in the *Oracle TimesTen In-Memory Database Security Guide*.

> **Note:**
>
> To use TLS, the driver file set must include the `clientWallet` directory (when you created certificates), located in the directory `conf/mywallets/clientWallet`. When you run your application, the location of the `clientWallet` directory must be specified in your TLS configuration file `sys.odbc.ini`.

You must set `WalletDir`, `Encryption`, `CipherSuites`, and `SSLClientAuthentication` in the DSN definition. For example, in the client DSN definition in `sys.odbc.ini`:

```
WalletDir=tomcat_dir/webapps/application_dir/WEB-INF/liteclient/conf/
mywallets/clientWallet
Encryption=requested
CipherSuites=SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
SSLClientAuthentication=1
```

The `WalletDir` attribute points to the location of the `clientWallet` directory on the application server file system when the application is deployed.

Restrictions

This section describes restrictions when deploying the TimesTen JDBC driver with a web application.

- Tomcat: There are known restrictions when using the TimesTen JDBC driver WAR configuration with the Tomcat application server:
  - When a TimesTen web application is deployed to a Tomcat server and at least one connection to TimesTen has been established, it is not possible to

redeploy or restart the application without restarting the application server. This restriction is related to the application class loader. When a Tomcat application is restarted, a new class loader is instantiated. When the new class loader attempts to load the TimesTen shared library, the operation fails because the class loader associated with the previous instance of the application still exists and is still associated with the shared library.

  – For the same reason, different Tomcat applications that connect to TimesTen using the JDBC driver WAR configuration cannot coexist in the same JVM.

  The solution to both problems is to configure the TimesTen JDBC driver as a system-managed library that is shared and available to all applications in the Tomcat JVM.

- Linux: On Linux, instead of updating `java.library.path`, you can update the `LD_LIBRARY_PATH` environment variable where the application server runs.

```
export LD_LIBRARY_PATH=tomcat_dir/webapps/application_dir/WEB-INF/
liteclient:$LD_LIBRARY_PATH
```

  However, it may not be feasible to preconfigure the application server to set `java.library.path`. In this case, it's possible to load the shared library at runtime by explicitly setting the `com.timesten.jdbc.shared.library` system property with the full path to `libttJdbcCS_all.so`. This must be done before connecting to TimesTen.

  The following example procedure sets `com.timesten.jdbc.shared.library` with the full path to the TimesTen shared library before an initial connection to TimesTen.

```
// Configure TimesTen to use the lite client shared library
System.setProperty("com.timesten.jdbc.shared.library",
      "/tomcat_dir/webapps/application_dir/WEB-INF/liteclient/
libttJdbcCS_all.so");
// Connect to TimesTen ...
javax.sql.DataSource ds =
      (javax.sql.DataSource) ctx.lookup ("java:comp/env/jdbc/
TIMESTEN");
conn = ds.getConnection();
```

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 that have purchased support 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 TimesTen In-Memory Database Deploy TimesTen JDBC Driver with Web Applications, Release 22.1
F87406-03