

Alternate Descriptors

Example alternate-descriptors can be browsed at <https://github.com/apache/tomee/tree/master/examples/alternate-descriptors>

See the [Alternate Descriptors](#) page for the full details of how this feature works.

For our example we'll use the standard "moviefun" code which contains a **Movie** entity and **Movies** session bean. To add a twist for testing and demonstrate alternate descriptors, we will create an interceptor that will be used only in our test cases.

To add this to our application, we simply need a **test.ejb-jar.xml** in the same location that the regular **ejb-jar.xml** would be expected.

That gives us the following files in our project:

- src/main/resources/META-INF/ejb-jar.xml
- src/main/resources/META-INF/persistence.xml
- src/main/resources/META-INF/test.ejb-jar.xml

The test.ejb-jar.xml

The normal **ejb-jar.xml** simply contains `<ejb-jar/>`, however the **test.ejb-jar.xml** we add an extra interceptor:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ejb-jar xmlns="http://java.sun.com/xml/ns/javaee">
  <assembly-descriptor>
    <interceptor-binding>
      <ejb-name>Movies</ejb-name>
      <interceptor-class>org.superbiz.altd.MoviesTest$Interceptor</interceptor-class>
    </interceptor-binding>
  </assembly-descriptor>
</ejb-jar>
```

The TestCase

To enable our **test.ejb-jar.xml** in the test case, we simply set the **openejb.altd.prefix** property when creating the embedded **EJBContainer**

```
public class MoviesTest extends TestCase {
```

```
@EJB
private Movies movies;
```

```
@Resource
private UserTransaction userTransaction;
```

```
@PersistenceContext
private EntityManager entityManager;
```

```
public void setUp() throws Exception {
    Properties p = new Properties();
    p.put("movieDatabase", "new://Resource?type=DataSource");
    p.put("movieDatabase.JdbcDriver", "org.hsqldb.jdbcDriver");
    p.put("movieDatabase.JdbcUrl", "jdbc:hsqldb:mem:moviedb");
```

```
p.put("openejb.altdb.prefix", "test");
```

```
    EJBContainer.createEJBContainer(p).getContext().bind("inject", this);
}
```

```
public void test() throws Exception {
```

```
    userTransaction.begin();
```

```
    try {
        entityManager.persist(new Movie("Quentin Tarantino", "Reservoir Dogs", 1992));
        entityManager.persist(new Movie("Joel Coen", "Fargo", 1996));
        entityManager.persist(new Movie("Joel Coen", "The Big Lebowski", 1998));
```

```
List<Movie> list = movies.getMovies();
assertEquals("List.size()", 3, list.size());
```

```
    for (Movie movie : list) {
        movies.deleteMovie(movie);
    }
```

```
assertEquals("Movies.getMovies()", 0, movies.getMovies().size());
```

```
    } finally {
        try {
            userTransaction.commit();
            fail("Transaction should have been rolled back");
        } catch (RollbackException e) {
            // Good, we don't want to clean up the db
        }
    }
}
```

```
public static class Interceptor {
```

```
@Resource
private SessionContext sessionContext;
```

```
@AroundInvoke
public Object invoke(InvocationContext context) throws Exception {
```

```
    sessionContext.setRollbackOnly();
```

```
        return context.proceed();
    }
}
```

As noted in [the documentation](#), several prefixes can be used at once.

Running

```
-----
T E S T S
-----
```

```
Running org.superbiz.altd.MoviesTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/alternate-descriptors
INFO - openejb.base = /Users/dblevins/examples/alternate-descriptors
INFO - Using 'javax.ejb.embeddable.EJBContainer=true'
```

```

INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Configuring Service(id=movieDatabase, type=Resource, provider-id=Default JDBC
Database)
INFO - Found EjbModule in classpath: /Users/dblevins/examples/alternate-
descriptors/target/classes
INFO - Beginning load: /Users/dblevins/examples/alternate-descriptors/target/classes
INFO - AltDD ejb-jar.xml -> file:/Users/dblevins/examples/alternate-
descriptors/target/classes/META-INF/test.ejb-jar.xml
INFO - Configuring enterprise application: /Users/dblevins/examples/alternate-
descriptors
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably
using an older Runtime.
INFO - Configuring Service(id=Default Stateful Container, type=Container, provider-
id=Default Stateful Container)
INFO - Auto-creating a container for bean Movies: Container(type=STATEFUL, id=Default
Stateful Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.altdd.MoviesTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Configuring PersistenceUnit(name=movie-unit)
INFO - Auto-creating a Resource with id 'movieDatabaseNonJta' of type 'DataSource for
'movie-unit'.
INFO - Configuring Service(id=movieDatabaseNonJta, type=Resource, provider-
id=movieDatabase)
INFO - Adjusting PersistenceUnit movie-unit <non-jta-data-source> to Resource ID
'movieDatabaseNonJta' from 'movieDatabaseUnmanaged'
INFO - Enterprise application "/Users/dblevins/examples/alternate-descriptors" loaded.
INFO - Assembling app: /Users/dblevins/examples/alternate-descriptors
INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 411ms
INFO - Jndi(name="java:global/alternate-descriptors/Movies!org.superbiz.altdd.Movies")
INFO - Jndi(name="java:global/alternate-descriptors/Movies")
INFO -
Jndi(name="java:global/EjbModule1893321675/org.superbiz.altdd.MoviesTest!org.superbiz.
altdd.MoviesTest")
INFO - Jndi(name="java:global/EjbModule1893321675/org.superbiz.altdd.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateful
Container)
INFO - Created Ejb(deployment-id=org.superbiz.altdd.MoviesTest, ejb-
name=org.superbiz.altdd.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateful
Container)
INFO - Started Ejb(deployment-id=org.superbiz.altdd.MoviesTest, ejb-
name=org.superbiz.altdd.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/alternate-descriptors)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.569 sec

```

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

Warning on Tooling

If you split your descriptors into separate directories, this support will not work. Specifically, this will not work:

- src/main/resources/META-INF/ejb-jar.xml
- src/main/resources/META-INF/persistence.xml
- src/**test**/resources/META-INF/test.ejb-jar.xml

This support is **not** aware of any Maven, Gradle, Ant, IntelliJ, NetBeans, Eclipse or other settings.