

Transaction Rollback

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

Help us document this example! Click the blue pencil icon in the upper right to edit this page.

CustomRuntimeException

```
package org.superbiz.txrollback;

import javax.ejb.ApplicationException;

@ApplicationException
public class CustomRuntimeException extends RuntimeException {

    public CustomRuntimeException() {
    }

    public CustomRuntimeException(String s) {
        super(s);
    }

    public CustomRuntimeException(String s, Throwable throwable) {
        super(s, throwable);
    }

    public CustomRuntimeException(Throwable throwable) {
        super(throwable);
    }
}
```

Movie

```
package org.superbiz.txrollback;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity(name = "Movie")
public class Movie {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;
```

```
private String director;  
private String title;  
private int year;  
  
public Movie() {  
}  
  
public Movie(String director, String title, int year) {  
    this.director = director;  
    this.title = title;  
    this.year = year;  
}  
  
public String getDirector() {  
    return director;  
}  
  
public void setDirector(String director) {  
    this.director = director;  
}  
  
public String getTitle() {  
    return title;  
}  
  
public void setTitle(String title) {  
    this.title = title;  
}  
  
public int getYear() {  
    return year;  
}  
  
public void setYear(int year) {  
    this.year = year;  
}  
}
```

Movies

```

package org.superbiz.txrollback;

import javax.annotation.Resource;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.PersistenceContextType;
import javax.persistence.Query;
import java.util.List;

//START SNIPPET: code
@Stateless
public class Movies {

    @PersistenceContext(unitName = "movie-unit", type = PersistenceContextType
.TRANSACTION)
    private EntityManager entityManager;

    @Resource
    private SessionContext sessionContext;

    public void addMovie(Movie movie) throws Exception {
        entityManager.persist(movie);
    }

    public void deleteMovie(Movie movie) throws Exception {
        entityManager.remove(movie);
    }

    public List<Movie> getMovies() throws Exception {
        Query query = entityManager.createQuery("SELECT m from Movie as m");
        return query.getResultList();
    }

    public void callSetRollbackOnly() {
        sessionContext.setRollbackOnly();
    }

    public void throwUncheckedException() {
        throw new RuntimeException("Throwing unchecked exceptions will rollback a
transaction");
    }

    public void throwApplicationException() {
        throw new CustomRuntimeException("This is marked @ApplicationException, so no
TX rollback");
    }
}

```

persistence.xml

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">

  <persistence-unit name="movie-unit">
    <jta-data-source>movieDatabase</jta-data-source>
    <non-jta-data-source>movieDatabaseUnmanaged</non-jta-data-source>
    <class>org.superbiz.testinjection.MoviesTest.Movie</class>

    <properties>
      <property name="openjpa.jdbc.SynchronizeMappings" value=
"buildSchema(ForeignKeys=true)"/>
    </properties>
  </persistence-unit>
</persistence>
```

MoviesTest

```
package org.superbiz.txrollback;

import junit.framework.TestCase;

import javax.annotation.Resource;
import javax.ejb.EJB;
import javax.ejb.embeddable.EJBContainer;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.transaction.RollbackException;
import javax.transaction.UserTransaction;
import java.util.List;
import java.util.Properties;

//START SNIPPET: code
public class MoviesTest extends TestCase {

    @EJB
    private Movies movies;

    @Resource
    private UserTransaction userTransaction;

    @PersistenceContext
    private EntityManager entityManager;

    private EJBContainer ejbContainer;

    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" + System
.currentTimeMillis());

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

    @Override
    protected void tearDown() throws Exception {
        ejbContainer.close();
    }

    /**
     * Standard successful transaction scenario. The data created inside
     * the transaction is visible after the transaction completes.
     * <p/>
     * Note that UserTransaction is only usable by Bean-Managed Transaction
     * beans, which can be specified with @TransactionManagement(BEAN)
     */
    public void testCommit() 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());
        } finally {
            userTransaction.commit();
        }

        // Transaction was committed
        List<Movie> list = movies.getMovies();
        assertEquals("List.size()", 3, list.size());
    }

    /**
     * Standard transaction rollback scenario. The data created inside
     * the transaction is not visible after the transaction completes.
     */
    public void testUserTransactionRollback() 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());
    } finally {
        userTransaction.rollback();
    }

    // Transaction was rolled back
    List<Movie> list = movies.getMovies();
    assertEquals("List.size()", 0, list.size());
}

/**
 * Transaction is marked for rollback inside the bean via
 * calling the javax.ejb.SessionContext.setRollbackOnly() method
 * <p/>
 * This is the cleanest way to make a transaction rollback.
 */
public void testMarkedRollback() 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());

        movies.callSetRollbackOnly();
    } finally {
        try {
            userTransaction.commit();
            fail("A RollbackException should have been thrown");
        } catch (RollbackException e) {
            // Pass
        }
    }

    // Transaction was rolled back
    List<Movie> list = movies.getMovies();
    assertEquals("List.size()", 0, list.size());
}

```

```

/**
 * Throwing an unchecked exception from a bean will cause
 * the container to call setRollbackOnly() and discard the
 * bean instance from further use without calling any @PreDestroy
 * methods on the bean instance.
 */
public void testExceptionBasedRollback() 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());

        try {
            movies.throwUncheckedException();
        } catch (RuntimeException e) {
            // Good, this will cause the tx to rollback
        }
    } finally {
        try {
            userTransaction.commit();
            fail("A RollbackException should have been thrown");
        } catch (RollbackException e) {
            // Pass
        }
    }

    // Transaction was rolled back
    List<Movie> list = movies.getMovies();
    assertEquals("List.size()", 0, list.size());
}

/**
 * It is still possible to throw unchecked (runtime) exceptions
 * without dooming the transaction by marking the exception
 * with the @ApplicationException annotation or in the ejb-jar.xml
 * deployment descriptor via the <application-exception> tag
 */
public void testCommit2() 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());

try {
    movies.throwApplicationException();
} catch (RuntimeException e) {
    // This will *not* cause the tx to rollback
    // because it is marked as an @ApplicationException
}
} finally {
    userTransaction.commit();
}

// Transaction was committed
List<Movie> list = movies.getMovies();
assertEquals("List.size()", 3, list.size());
}
}

```

Running

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

```

Running org.superbiz.txrollback.MoviesTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/transaction-rollback
INFO - openejb.base = /Users/dblevins/examples/transaction-rollback
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/transaction-
rollback/target/classes
INFO - Beginning load: /Users/dblevins/examples/transaction-rollback/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/transaction-
rollback
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably
using an older Runtime.
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)

```

```

INFO - Auto-creating a container for bean Movies: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.txrollback.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/transaction-rollback" loaded.
INFO - Assembling app: /Users/dblevins/examples/transaction-rollback
INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 412ms
INFO - Jndi(name="java:global/transaction-
rollback/Movies!org.superbiz.txrollback.Movies")
INFO - Jndi(name="java:global/transaction-rollback/Movies")
INFO -
Jndi(name="java:global/EjbModule1718375554/org.superbiz.txrollback.MoviesTest!org.supe
rbiz.txrollback.MoviesTest")
INFO - Jndi(name="java:global/EjbModule1718375554/org.superbiz.txrollback.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/transaction-rollback)
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetaDataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetaDataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetaDataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
INFO - Undeploying app: /Users/dblevins/examples/transaction-rollback
INFO - Closing DataSource: movieDatabase
INFO - Closing DataSource: movieDatabaseNonJta
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/transaction-rollback
INFO - openejb.base = /Users/dblevins/examples/transaction-rollback
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/transaction-
rollback/target/classes
INFO - Beginning load: /Users/dblevins/examples/transaction-rollback/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/transaction-
rollback
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean Movies: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.txrollback.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/transaction-rollback" loaded.
INFO - Assembling app: /Users/dblevins/examples/transaction-rollback
INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 5ms
INFO - Jndi(name="java:global/transaction-
rollback/Movies!org.superbiz.txrollback.Movies")
INFO - Jndi(name="java:global/transaction-rollback/Movies")
INFO -
Jndi(name="java:global/EjbModule935567559/org.superbiz.txrollback.MoviesTest!org.super
biz.txrollback.MoviesTest")
INFO - Jndi(name="java:global/EjbModule935567559/org.superbiz.txrollback.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/transaction-rollback)
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.

```

```

WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
INFO - Undeploying app: /Users/dblevins/examples/transaction-rollback
INFO - Closing DataSource: movieDatabase
INFO - Closing DataSource: movieDatabaseNonJta
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/transaction-rollback
INFO - openejb.base = /Users/dblevins/examples/transaction-rollback
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/transaction-
rollback/target/classes
INFO - Beginning load: /Users/dblevins/examples/transaction-rollback/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/transaction-
rollback
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean Movies: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.txrollback.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/transaction-rollback" loaded.
INFO - Assembling app: /Users/dblevins/examples/transaction-rollback
INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 5ms
INFO - Jndi(name="java:global/transaction-
rollback/Movies!org.superbiz.txrollback.Movies")
INFO - Jndi(name="java:global/transaction-rollback/Movies")
INFO -
Jndi(name="java:global/EjbModule1961109485/org.superbiz.txrollback.MoviesTest!org.supe
rbiz.txrollback.MoviesTest")
INFO - Jndi(name="java:global/EjbModule1961109485/org.superbiz.txrollback.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-

```

```

name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/transaction-rollback)
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
INFO - Undeploying app: /Users/dblevins/examples/transaction-rollback
INFO - Closing DataSource: movieDatabase
INFO - Closing DataSource: movieDatabaseNonJta
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/transaction-rollback
INFO - openejb.base = /Users/dblevins/examples/transaction-rollback
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/transaction-
rollback/target/classes
INFO - Beginning load: /Users/dblevins/examples/transaction-rollback/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/transaction-
rollback
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean Movies: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.txrollback.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/transaction-rollback" loaded.
INFO - Assembling app: /Users/dblevins/examples/transaction-rollback

```

```

INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 5ms
INFO - Jndi(name="java:global/transaction-
rollback/Movies!org.superbiz.txrollback.Movies")
INFO - Jndi(name="java:global/transaction-rollback/Movies")
INFO -
Jndi(name="java:global/EjbModule419651577/org.superbiz.txrollback.MoviesTest!org.super
biz.txrollback.MoviesTest")
INFO - Jndi(name="java:global/EjbModule419651577/org.superbiz.txrollback.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/transaction-rollback)
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetaDataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetaDataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetaDataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
INFO - Undeploying app: /Users/dblevins/examples/transaction-rollback
INFO - Closing DataSource: movieDatabase
INFO - Closing DataSource: movieDatabaseNonJta
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/transaction-rollback
INFO - openejb.base = /Users/dblevins/examples/transaction-rollback
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/transaction-
rollback/target/classes
INFO - Beginning load: /Users/dblevins/examples/transaction-rollback/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/transaction-
rollback
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean Movies: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-

```

```
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.txrollback.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/transaction-rollback" loaded.
INFO - Assembling app: /Users/dblevins/examples/transaction-rollback
INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 4ms
INFO - Jndi(name="java:global/transaction-
rollback/Movies!org.superbiz.txrollback.Movies")
INFO - Jndi(name="java:global/transaction-rollback/Movies")
INFO -
Jndi(name="java:global/EjbModule15169271/org.superbiz.txrollback.MoviesTest!org.superb
iz.txrollback.MoviesTest")
INFO - Jndi(name="java:global/EjbModule15169271/org.superbiz.txrollback.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=org.superbiz.txrollback.MoviesTest, ejb-
name=org.superbiz.txrollback.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/transaction-rollback)
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
WARN - The class "org.superbiz.testinjection.MoviesTest.Movie" listed in the
openjpa.MetadataFactory configuration property could not be loaded by
sun.misc.Launcher$AppClassLoader@39172e08; ignoring.
INFO - Undeploying app: /Users/dblevins/examples/transaction-rollback
INFO - Closing DataSource: movieDatabase
INFO - Closing DataSource: movieDatabaseNonJta
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.586 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
```