

# EJBs in Shark



---

# Table of Contents

1. Introduction .....	1
2. Configuration and EAR generation .....	2
Configuration for JOnAS .....	2
Configuration for JBOSS .....	3
Configuration for Geronimo .....	4
3. Deployment .....	8
Deployment in JOnAS .....	8
Deployment in JBOSS .....	8
Deployment in Geronimo .....	8
4. Using Shark Swing Admin Application .....	10
JOnAS Settings .....	10
JBOSS Settings .....	10
Geronimo Settings .....	11

---

# Chapter 1. Introduction

An Enterprise Java Bean (EJB) is a server-side software component that can be deployed in a distributed multi tier environment.

An EJB container can hold four major categories of beans:

- Session Beans
  - Stateless Session Beans
  - Stateful Session Beans
- Entity Beans
- Message Driven Beans (MDBs or Message Beans)

EJBs used in shark are session beans (stateless and stateful).

Stateless Session Beans are distributed objects that do not have state associated with them thus allowing concurrent access to the bean. The contents of instance variables are not guaranteed to be preserved across method calls. The lack of overhead to maintain a conversation with the calling program makes them less resource-intensive than stateful beans.

Session beans are used to implement business objects that hold client-specific business logic. The state of an object consists of the values of its instance variables. In a stateful session bean, the instance variables represent the state of a unique client-bean session. Because the client interacts with its bean, this state is often called the conversational state.

Note: The shark beans build process is done and tested for JOnAS Tomcat (4.5.2, 4.7.5), JBOSS (4.x) and Geronimo Tomcat 1.1.1 EJB containers.

---

# Chapter 2. Configuration and EAR generation

The configuration file used for shark beans configuration is *Shark.conf* and is located in *sharkejb.war* (in *WEB-INF/classes*), which is inside EAR file.

For now, shark is defined to work with HSQL database.

Beside this configuration file, property file describing database defined by *DataSourceName* should be set on the proper place for each EJB container.

This chapter explains more detail all settings needed for different EJB containers.

When proper configuration files are set to the desired values, EAR should be generated. In *configure.properties* file (located in *<shark\_install\_dir>* directory) should be set the name of EJB container for which will the EAR be generated for. The name of this parameter is *ejb\_container* and the possible values are *jboss*, *jonas*, *geronimo* and *jboss-ws*. The default value is *jboss*.

```
# EJB container for which the ear file will be built, pick one of:
# jboss, jboss-ws, jonas, geronimo
# NOTE: if you pick jboss-ws, it will build ear for JBoss with possibility
# to expose beans as WebServices
ejb_container=jboss
```

Value *jboss-ws* means that EAR file will be built for JBOSS server with possibility to expose shark's stateless session beans as WebServices.

Also, in this file can be set the database that will be used (the default value is HSQL):

```
# database vendor, pick one of:
# db2, hsql, informix, msql, msql2005, mysql, oracle, postgresql, sybase
db_loader_job=msql
```

and the location of JDBC driver (needed for all databases except HSQL):

```
# directory containing JDBC driver jar/zip files
db_ext_dirs=C:/drivers/MSQL
```

For the database defined in this property file will all configuration files needed for the EJB be generated for.

After all parameters are set, the EAR file is being generated by running *configure.bat* script (for windows) or *configure.sh* script (for linux), located in *<shark\_install\_dir>* directory. After generation, the proper EAR file and other needed files are placed in *<shark\_install\_dir>/EJB* directory.

## Configuration for JOnAS

The configuration file that will be generated for JOnAS EAR is called *Shark.conf.in* and is located in *<shark\_install\_dir>/dist/EJB/conf* directory. During the build of EAR, the parameters specific for JOnAS will be set to the following values:

```
DatabaseManager.defaults.XATransactionManagerLookupName=java:comp/UserTransaction
SharkTxSynchronizationFactory.XATransactionManagerLookupName=java:comp/UserTransaction
DatabaseManager.DB.sharkdb.Connection.DataSourceName=jndi:sharkdb
```

The parameter *DataSourceName* has the following form: "jndi:<jndi name of the database defined in JOnAS>". In this example, jndi name is *sharkdb*.

The property file describing the database is called *sharkdb.properties.in* and is located in `<shark_install_dir>/dist/EJB/conf/jonas` directory. During the EAR generation, this file will be generated for the database defined in `configure.properties` file under the name *sharkdb.properties*. After generation, this file should be placed in `<jonas_root>/conf` directory.

Here is given the example for PostgreSQL database:

```
jdbc.wrapper=org.enhydra.jdbc.standard.StandardXADataSource
jdbc.minconpool=12
jdbc.maxconpool=180
jdbc.connmaxage=30
jdbc.connchecklevel=1
datasource.description=Shark WfEngine DataSource
jdbc.connteststmt=SELECT 1
datasource.name=sharkdb

datasource.classname=org.postgresql.Driver
datasource.url=jdbc:postgresql://localhost/shark
datasource.username=sa
datasource.password=sa
```

Database drivers, if not already added to JOnAS, should be added somewhere in `<jonas_root>/lib` directory (for example, in `<jonas_root>/lib/commons/jonas`).

When JOnAS is started, the database must be deployed (JOnAS server loads data source, related jdbc drivers, and registers the data sources into JNDI). The easiest way to do it is to use JOnAS Admin tool (URL `http://<localhost:connection_port>/jonasAdmin`). By default, the url is `http://localhost:9000/jonasAdmin/`.

By default, username for JOnAS admin is "jonas", and password is also "jonas".

When admin application appears, go to Domain -> Server JOnAS -> Resources -> Database (JDBC).

Then, on Deployment card, choose sharkdb database and deploy it.

The another way for database deployment is to add database to deployment list defined by `jonas.service.dbm.datasources` parameter in `<jonas_root>/conf/jonas.properties` file. This parameter contains a coma-separated list of DataSources that will be deployed at JOnAS start.

For example:

```
jonas.service.dbm.datasources HSQL1,sharkdb
```

## Configuration for JBOSS

The configuration file template that will be generated for JBOSS EAR is called *Shark.conf.in* and is located in `<shark_install_dir>/dist/EJB/conf`. During the build of EAR, the parameters specific for JBOSS are set to the following values:

```
DatabaseManager.defaults.XATransactionManagerLookupName=java:/TransactionManager
SharkTxSynchronizationFactory.XATransactionManagerLookupName=java:/TransactionManager
DatabaseManager.DB.sharkdb.Connection.DataSourceName=jndi:java:sharkdb
```

The parameter `DataSourceName` has the following form: `"jndi:java:<jndi name of the database defined in JBOSS >"`. In this example, jndi name is *sharkdb*.

The JBOSS standard for file name which defines database is: `<jndi_name>-ds.xml`.

The property file describing the database is called *sharkdb-ds.xml.in* and is located in `<shark_install_dir>/dist/EJB/conf/jboss` directory (or in `<shark_install_dir>/dist/EJB/conf/jboss-ws` directory for EAR with beans that can be exposed as web services). During the EAR generation, this file will be generated for the database defined in `configure.properties` file under the name *sharkdb-ds.xml*. After generation, this file should be placed in JBOSS server's deploy directory.

Here is given the example for MySQL database:

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

<datasources>
  <local-tx-datasource>
    <jndi-name>sharkdb</jndi-name>
    <connection-url>jdbc:mysql://localhost/shark</connection-url>
    <driver-class>org.gjt.mm.mysql.Driver</driver-class>
    <user-name>root</user-name>
    <password>sa</password>
  </local-tx-datasource>
</datasources>
```

Database drivers, if not already added to JBoss, should be added in JBOSS server's lib directory.

When JBOSS is started, the database is automatically deployed into JBOSS (JBOSS server loads data source, related jdbc drivers, and registers the data sources into JNDI).

## Configuration for Geronimo

The configuration file that will be generated for Geronimo EAR is called *Shark.conf.in* and is located in `<shark_install_dir>/dist/EJB/conf` directory. During the build of EAR, the parameters specific for Geronimo are set to the following values:

```
EnvironmentType=geronimo
DatabaseManager.DB.sharkdb.Connection.DataSourceName=jndi:jdbc/sharkBase
DatabaseManager.DB.sharkdb.ObjectId.ClassName=org.enhydra.dods.jta.GeronimoJTAObjectIdAllocator
DatabaseManager.defaults.TransactionFactory=org.enhydra.dods.jta.GeronimoDBTransactionFactory
```

The parameter `DataSourceName` has the following form: "jndi:<jndi name of the database defined in Geronimo>". In this example, jndi name is *jdbc/sharkBase*.

Geronimo needs some additional configuration files. In all files in which the database jndi name is needed, it is set to value "jdbc/sharkBase". The configuration file that defines deployment plan (database) is called *sharkdb.xml.in* and is located in `<shark_install_dir>/dist/EJB/conf/geronimo` directory. In this file should be the database name be set to value *sharkdb* (because of various settings in all configuration files). During the EAR generation, this file will be generated for the database defined in `configure.properties` file under the name *sharkdb.xml*.

The example of *sharkdb.xml* file that defines Postgresql database:

```
<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector-1.1">
  <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1">
    <dep:moduleId>
      <dep:groupId>console.dbpool</dep:groupId>
      <dep:artifactId>sharkdb</dep:artifactId>
      <dep:version>1.0</dep:version>
      <dep:type>rar</dep:type>
    </dep:moduleId>
    <dep:dependencies>
      <dep:dependency>
        <dep:groupId>sharkdbdriver</dep:groupId>
        <dep:artifactId>sharkdbdriver</dep:artifactId>
```

```

        <dep:version>1.0</dep:version>
        <dep:type>jar</dep:type>
      </dep:dependency>
    </dep:dependencies>
  </dep:environment>
  <resourceadapter>
    <outbound-resourceadapter>
      <connection-definition>

<connectionfactory-interface>javax.sql.DataSource</connectionfactory-interface>
      <connectiondefinition-instance>
        <name>sharkdb</name>
        <config-property-setting
name="Driver">org.postgresql.Driver</config-property-setting>
        <config-property-setting name="UserName">sa</config-property-setting>
        <config-property-setting name="Password">sa</config-property-setting>
        <config-property-setting
name="ConnectionURL">jdbc:postgresql://localhost/shark</config-property-setting>
        <connectionmanager>
          <local-transaction/>
          <single-pool>
            <max-size>10</max-size>
            <min-size>0</min-size>
            <match-one/>
          </single-pool>
          </connectionmanager>
        </connectiondefinition-instance>
      </connection-definition>
    </outbound-resourceadapter>
  </resourceadapter>
</connector>

```

Also, proper jdbc drivers should be added to geronimo's repository (if not added yet). Jdbc driver should be also set in sharkdb.xml file. To make the story general, the universal name for the database driver is set:

```

<dep:dependencies>
  <dep:dependency>
    <dep:groupId>sharkdbdriver</dep:groupId>
    <dep:artifactId>sharkdbdriver</dep:artifactId>
    <dep:version>1.0</dep:version>
    <dep:type>jar</dep:type>
  </dep:dependency>
</dep:dependencies>

```

So, jdbc driver should be named sharkdbdriver-1.0.jar and should be placed in <geronimo\_root>/repository/sharkdbdriver/sharkdbdriver/1.0 directory. When driver is renamed and placed in repository directory as explained, it should be added to the *Common libs* list that can be viewed by using Geronimo Admin application. By default, the url is <http://localhost:8080/console>. Off course, in order to use the console, Apache Geronimo must be running.

By default, username for Apache Geronimo admin is "system", and password is "manager".

When admin application appears, go to Services -> Common Libs, and check if in the list appears desired database drivers.

As mentioned, there are some differences in the number and content of configuration files between Apache Geronimo and other EJB servers. Here are explained additional Apache Geronimo's configuration files.

File *application.xml*, located in EAR's META-INF directory, contains one additional connector:

```

<?xml version="1.0" encoding="UTF-8"?>
<application>
  . . .
  <module>
    <connector>tranql-connector-1.2.rar</connector>
  </module>

```



```
</application>
```

File *geronimo-application.xml*, located in EAR's META-INF directory, is specific for Apache Geronimo and looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.1">
  <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1">
    <dep:moduleId>
      <dep:groupId>enhydra</dep:groupId>
      <dep:artifactId>Shark</dep:artifactId>
      <dep:version>1.0</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
    <dep:dependencies/>
    <dep:hidden-classes/>
    <dep:non-overridable-classes/>
  </dep:environment>
  <module>
    <connector>tranql-connector-1.2.rar</connector>
    <alt-dd>sharkdb.xml</alt-dd>
  </module>
</application>
```

If *web.xml* file exists in the EAR, in the same location must also exist *geronimo-web.xml* file.

For example, if *web.xml* file looks like:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
. . .
<web-app>
  <display-name>shark-ejb</display-name>
  <description>Shark EJB sample JSP application.</description>
</web-app>
```

The *geronimo-web.xml* file can look like:

```
<web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1"
  xmlns:naming="http://geronimo.apache.org/xml/ns/naming-1.1">
  <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1">
    <dep:moduleId>
      <dep:groupId>enhydra</dep:groupId>
      <dep:artifactId>shark</dep:artifactId>
      <dep:version>1.0</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
    <dep:dependencies/>
    <dep:hidden-classes/>
    <dep:non-overridable-classes/>
  </dep:environment>
  <context-root>/shark</context-root>
</web-app>
```

File *openejb-jar.xml*, located in META-INF directory of jar file (sharkejb.jar) that contains shark beans (inside EAR), looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<openejb-jar xmlns="http://www.openejb.org/xml/ns/openejb-jar-2.1">
  <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1">
    <dep:moduleId>
      <dep:groupId>enhydra</dep:groupId>
      <dep:artifactId>Shark</dep:artifactId>
      <dep:version>1.0</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
    <dep:dependencies/>
    <dep:hidden-classes/>
    <dep:non-overridable-classes/>
  </dep:environment>
```

```
<cmp-connection-factory>
  <resource-link>sharkdb</resource-link>
</cmp-connection-factory>

<enterprise-beans>
  <session>
    <ejb-name>WAPIEJB</ejb-name>
    <jndi-name>org/enhydra/shark/ejb/WAPIEJB</jndi-name>
    <resource-ref>
      <ref-name>jdbc/sharkBase</ref-name>
      <resource-link>sharkdb</resource-link>
    </resource-ref>
  </session>

  . . .
  <session>
    ... defined for every bean ...
  </session>
  . . .
</enterprise-beans>
</openejb-jar>
```

If database in deployment plan (file sharkdb.xml) is set to name jdbc/sharkBase (as explained in this chapter), none of this additional files needn't be changed. Only deployment plan sharkdb.xml, if not set for the proper database in configure.properties file, must be changed manually to the desired values before EAR generation.

File *geronimo\_client.properties* is located in <shark\_install\_dir>/.dist/EJB/conf/geronimo directory and contains the values needed for JNDI context creation. By default, this file contains geronimo's defaults:

```
java.naming.factory.initial=org.openejb.client.RemoteInitialContextFactory
java.naming.provider.url=localhost:4201
java.naming.security.principal=system
java.naming.security.credentials=manager
```

---

## Chapter 3. Deployment

After EAR generation, EAR file is placed in <shark\_install\_dir>/EJB directory along with all necessary data.

After generation, EAR must be placed in EJB container and deployed.

### Deployment in JOnAS

For this container, EAR (sharkejb-jonas.ear) should be placed in <jonas\_root>/apps directory and deployed by using JOnAS admin tool (URL <http://localhost:9000/jonasAdmin> [<http://localhost:9000/jonasAdmin/>]).

By default, username for JOnAS admin is "jonas", and password is also "jonas".

When admin application appears, go to Domain -> Server JOnAS -> Deployment -> Applications (EAR). Then, on Deployment card, choose sharkejb-jonas.ear and deploy it.

The another way for EAR deployment is to place EAR in directory defined by jonas.service.ear.autoloadpath parameter in <jonas\_root>/conf/jonas.properties file. By default, the value of this parameter is autoload, so, EAR should be placed in <jonas\_root>/apps/autoload directory and the EARs placed in this directory will be deployed by the JOnAS Server at launch time.

### Deployment in JBOSS

For this container, EAR (sharkejb-jboss.ear or sharkejb-jboss-ws.ear) should be placed in JBOSS server's deploy directory, and the jboss.zip file should be unpacked to %JBOSS\_HOME% folder.

When placed there, EAR is automatically deployed into JBOSS. If this EAR already exists, EAR will be automatically deployed in JBOSS during the start of the JBOSS.

JBoss deployment also includes SharkWebClient application for administering engine, which can be accessed through:

<http://localhost:8080/sharkWebClient>

if you run JBoss on a local machine and with default settings.

### Deployment in Geronimo

For this container, EAR (sharkejb-geronimo.ear) should be deployed by using geronimo admin tool (URL <http://localhost:8080/console> [<http://localhost:8080/console>] ><http://localhost:8080/console>]).

By default, username for geronimo admin is "system", and password is "manager".

When admin application appears, go to Applications -> Deploy New. There are two things that should be deployed by using this form.

First, the deployment plan should be deployed. In archive input, browse to *tranql-connector-1.2.rar* file and for plan browse to *sharkdb.xml* file (both files are located in <shark\_install\_dir>/EJB directory) and click *Install* button.

Second, the EAR should be deployed. In arrive input, browse to the EAR file (located in <shark\_install\_dir>/EJB directory) and click *Install* button.

If everything was OK, shark database (sharkdb) should appear in the list of database pools (path Services -> Database Pools in admin application ),

and shark EAR (sharkejb-geronimo.ear) should appear in the list of application EARs (path Applications -> Application EARs in admin application ).

---

# Chapter 4. Using Shark Swing Admin Application

Shark swing admin application can work with shark enterprise beans. To achieve this, some parameters must be set in *SharkClient.conf* file, located in <shark\_install\_dir>/conf directory.

This chapter explains this settings for different EJB containers.

After setting these parameters, start and work with shark admin is the same as in the POJO case.

## JOnAS Settings

For JOnAS, in *SharkClient.conf* file (located in <shark\_install\_dir>/conf directory), the following parameters should be set to the following values:

```
#####
# Section related to client applications
#####
# some applications can work with shark deployed as POJO or as EJB
# for such applications, default client type is POJO
ClientType=EJB

# defines user transaction timeout (max time that transaction can last before automatically
# being rolled back). Default is 300 sec
UserTransaction.Timeout=300

# if application works with EJB shark, the name for user transaction lookup should be
# defined. Default is name for the JBoss deployment.
XaUserTransactionLookupName=javax.transaction.UserTransaction

# if application works with EJB shark, the name for initial context factory should be
# specified. Default is name for the JBoss deployment.
java.naming.factory.initial=org.objectweb.carol.jndi.spi.MultiOrbInitialContextFactory

# if application works with EJB shark, the URL prefix for packages should be
# specified. Default is name for the JBoss deployment.
java.naming.factory.url.pkgs=org.objectweb.jonas.naming

# if application works with EJB shark, the URL of service provider should be
# specified. Default is name for the JBoss deployment on the local machine.
java.naming.provider.url=rmi://localhost:1099
```

The JOnAS jar file *client.jar* (located in <jonas\_root>/lib directory) must be copied to <shark\_install\_dir>/lib/contrib directory. Also, after EAR deployment in JOnAS, in <shark\_install\_dir>/lib directory should be added sharkejb.jar with generated stubs and skeletons from JOnAS. This jar is located in <jonas\_root>/work/apps/jonas/sharkejb\_<latest\_date>-<latest\_time> directory.

## JBoss Settings

For JBoss, in *SharkClient.conf* file (located in <shark\_install\_dir>/conf directory), the following parameter should be set to the following values:

```
#####
# Section related to client applications
#####
```

```
# some applications can work with shark deployed as POJO or as EJB
# for such applications, default client type is POJO
ClientType=EJB

# defines user transaction timeout (max time that transaction can last before automatically
# being rolled back). Default is 300 sec
UserTransaction.Timeout=300

# if application works with EJB shark, the name for user transaction lookup should be
# defined. Default is name for the JBoss deployment.
JaUserTransactionLookupName=UserTransaction

# if application works with EJB shark, the name for initial context factory should be
# specified. Default is name for the JBoss deployment.
java.naming.factory.initial=org.jboss.naming.NamingContextFactory

# if application works with EJB shark, the URL prefix for packages should be
# specified. Default is name for the JBoss deployment.
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interface

# if application works with EJB shark, the URL of service provider should be
# specified. Default is name for the JBoss deployment on the local machine.
java.naming.provider.url=jnp://localhost:1099
```

The parameter *ClientType* must be set to the value *EJB*. Other mentioned parameters need not be set, since shark is by default configured that it's client applications work with JBOSS (when *ClientType* is "EJB").

The JBOSS jar file *jbossall-client.jar* (located in JBOSS server's client directory) must be copied to <shark\_install\_dir>/lib/contrib directory.

For shark EAR with beans that can be exposed as web services, the parameter *ClientType* must be set to the value *WS*.

```
ClientType=WS
```

## Geronimo Settings

For geronimo, in *SharkClient.conf* file (located in <shark\_install\_dir>/conf directory), the following parameters should be set to the following values:

```
#####
# Section related to client applications
#####
# some applications can work with shark deployed as POJO or as EJB
# for such applications, default client type is POJO
ClientType=EJB_GERONIMO

# defines user transaction timeout (max time that transaction can last before automatically
# being rolled back). Default is 300 sec
UserTransaction.Timeout=300

# if application works with EJB shark, the name for initial context factory should be
# specified. Default is name for the JBoss deployment.
java.naming.factory.initial=org.openejb.client.RemoteInitialContextFactory

# if application works with EJB shark, the URL prefix for packages should be
# specified. Default is name for the JBoss deployment.
java.naming.factory.url.pkgs=org.apache.geronimo.naming

# if application works with EJB shark, the URL of service provider should be
# specified. Default is name for the JBoss deployment on the local machine.
java.naming.provider.url=localhost:4201

# if application works with EJB shark, only for geronimo these two parameters
# should be defined (default principal is system and credential manager)
```

```
java.naming.security.principal=system  
java.naming.security.credentials=manager
```

Beside geronimo's jar files that are already added in shark, *geronimo-j2ee\_1.4\_spec-1.1.jar* file (located in <geronimo\_root>/repository/org/apache/geronimo/specs/geronimo-j2ee\_1.4\_spec/1.1 directory) must be copied to <shark\_install\_dir>/lib/contrib directory.