

# Embedded JBoss

Rob Di Marco  
416 Software, LLC  
[rob.dimarco@416software.com](mailto:rob.dimarco@416software.com)

<http://www.innovationontherun.com>

# Overview

- What is the Embedded JBoss Container
- How Does It Work
- Examples of How We Use It

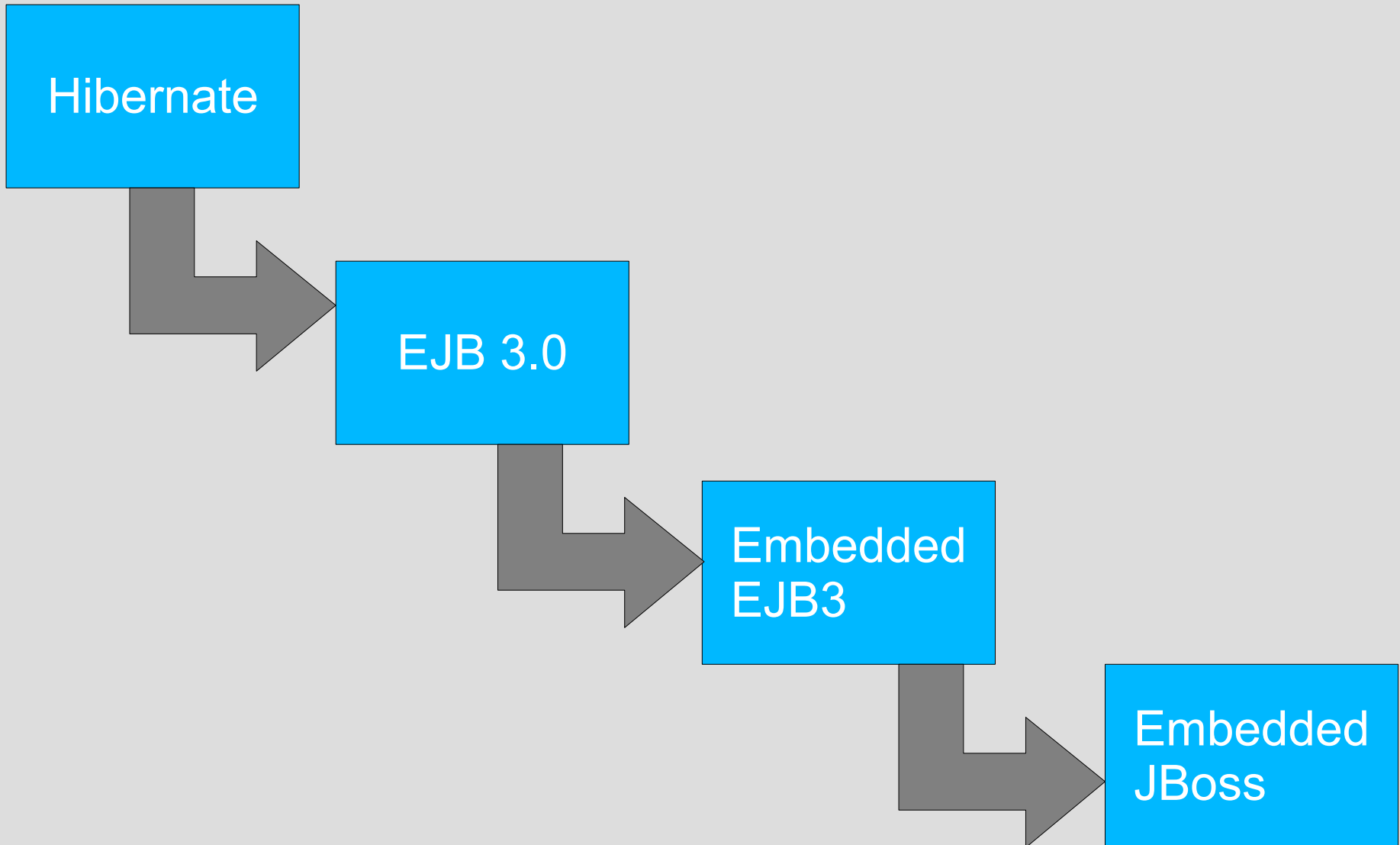
# What is Embedded JBoss

- Allows for embedding of JEE functionality
  - No independent application server required
  - Allows for easy EJB 3.0 and Messaging support
  - Quick startup time
- Composed of JARs and configuration files
  - jboss-embedded-all.jar
  - hibernate-all.jar
  - third-party-all.jar
- Currently in beta
  - Beta released in April 2007
  - No announced plan on final 1.0 release

# Supported JEE/JBoss Features

- JEE
  - JNDI (remoteable)
  - EJB 3.0 (remoteable)
  - JCA
  - JMX MBeans (-service.xml files)
- JBoss
  - \*-ds.xml files (connection pooling)
  - JBoss Security
  - JBoss Messaging
  - MicroContainer beans (-beans.xml files)
  - JBoss Transaction Service

# Where Did This Come From



# How Does It Work

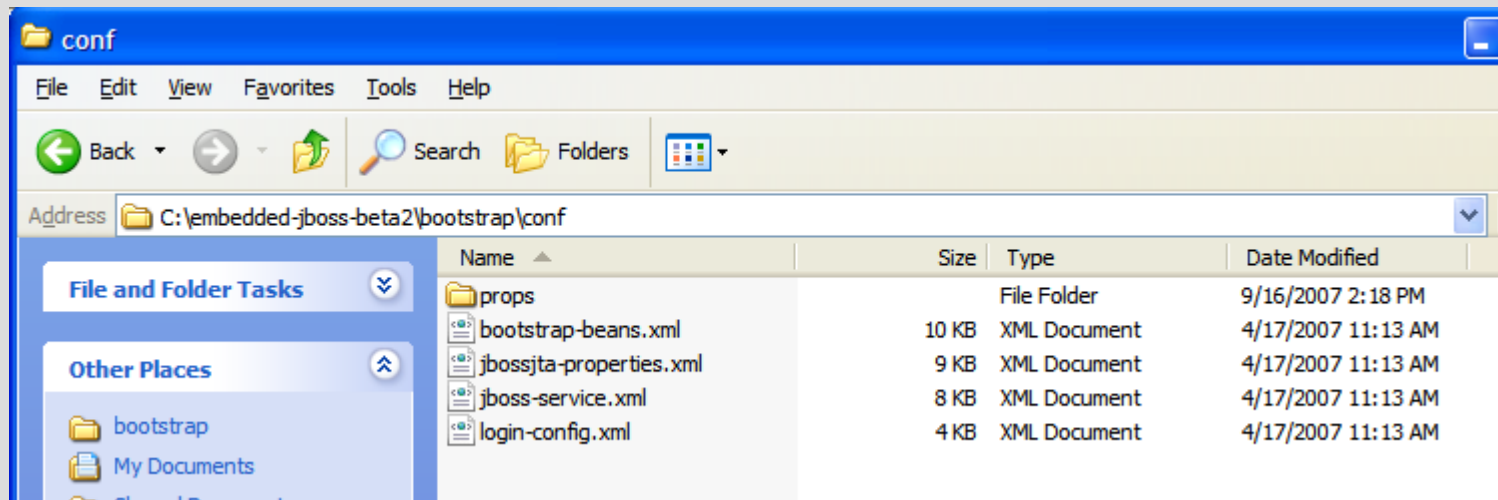
- Bootstrap.java is the programmatic interface
  - startup/shutdown
  - deploy/undeploy
  - Uses bootstrap-beans.xml for basic configuration
- Relies on JBoss 5.0 kernel
  - Pluggable class loading
  - Virtual File System
- Uses standard JBoss configuration classes
  - \*-service.xml, \*-ds.xml, \*-beans.xml

# Diving into Bootstrap.java

- Responsible for all interactions between application and embedded JBoss
- Use `bootstrap()` to start and `shutdown()` to stop
- Deploying an application can be done multiple ways
  - Scanning the classpath for a named resource
  - Specifying the URL of a specific resource to deploy
  - Creating a Virtual Archive that represents your deployment

# About the Examples

- Included in the Embedded JBoss download
- Configuration files in *bootstrap* directory
  - Very similar to the conventional JBoss configurations with additional bootstrap.xml



# The 5 Minute EJB 3.0 Tutorial

- Beans are configured using Java 5 annotations
  - @Stateless
  - @Entity
  - @MessageDriven
- Dependency injection of required resources
  - PersistenceUnit
  - Queues
  - Annotations specify the name, application server delivers the instance

# A Simple POJO Application

- Key Concepts
  - Three ways to deploy an application
  - Can connect to a session bean deployed from the embedded server locally or remotely
  - EJB 3.0 support
- Database configuration information
  - META-INF/persistence.xml
  - \*-ds.xml
- Example run using Ant and JDK 1.5
  - JDK 1.6 will not work due to an open Java bug

# Sample POJO: The Client Code

- Obtain a local session bean interface
- Obtain a remote session bean interface
- Execute methods on the beans

```
public static void run() throws Exception
{
    InitialContext ctx = new InitialContext();
    CustomerDAOLocal local = (CustomerDAOLocal)ctx.lookup("CustomerDAOBean/local");
    CustomerDAORemote remote = (CustomerDAORemote)ctx.lookup("CustomerDAOBean/remote");

    System.out.println("-----");
    int id = local.createCustomer("Gavin");
    Customer cust = local.findCustomer(id);
    System.out.println("    Successfully created and found Gavin from @Local interface");

    id = remote.createCustomer("Emmanuel");
    cust = remote.findCustomer(id);
    System.out.println("    Successfully created and found Emmanuel from @Remote interface");
    System.out.println("-----");
}
```

# Starting and Deploying the Application

- Bootstrap JBoss
- Deploy the application
- Execute the test
- Shutdown JBoss

```
public static void main(String[] args)
{
    Bootstrap.getInstance().bootstrap();

    String resource = "META-INF/persist.xml";
    Bootstrap.getInstance().deployResource(resource);

    run();

    Bootstrap.getInstance().shutdown();
}

Bootstrap.getInstance().deploy(jar);

run();

Bootstrap.getInstance().shutdown();
}
```

```
Tried to add non-URLClassLoader. Ignored
JBosSTS Transaction Service (JTA version) - JBoss Inc.
Setting up property manager MBean and JMX layer
TransactionStatusManagerItem - host: 192.168.1.4 port: 4681
Starting recovery manager
--- Start RecoveryActivators
[com.arjuna.ats.internal.arjuna.recovery.ready] RecoveryManagerImpl is ready on port 4,682
Recovery manager started
Binding TransactionManager JNDI Reference
Bound ConnectionManager 'jboss.jca:service=DataSourceBinding,name=DefaultDS' to JNDI name 'java:DefaultDS'
ServerPeer[0] creating server peer with ID 0
JBoss Messaging 1.2.0.sp1 server [0] started
Queue[queue/PrivateDLQ] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Queue[queue/PrivateExpiryQueue] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Queue[queue/ExpiryQueue] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Queue[queue/QueueWithOwnDLQAndExpiryQueue] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Topic[topic/TopicWithOwnRedeliveryDelay] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Queue[queue/QueueWithOwnRedeliveryDelay] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Topic[topic/TopicWithOwnDLQAndExpiryQueue] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Connector bisocket://192.168.1.4:4457 has leasing enabled, lease period 10000 milliseconds
[/ConnectionFactory, /XAConnectionFactory, java:/ConnectionFactory, java:/XAConnectionFactory] started
Queue[queue/DLQ] started, fullSize=200000, pageSize=2000, downCacheSize=2000
Bound ConnectionManager 'jboss.jca:service=DataSourceBinding,name=JmsXA' to JNDI name 'java:JmsXA'
installing bean: persistence.units:jar=tutorial.jar,unitName=custdb with dependencies:
jboss.jca:name=DefaultDS,service=DataSourceBinding
Starting persistence unit persistence.units:jar=tutorial.jar,unitName=custdb
installing bean: jboss.j2ee:jar=tutorial.jar,name=CustomerDAOBean,service=EJB3 with dependencies:
persistence.units:jar=tutorial.jar,unitName=custdb
STARTED EJB: org.jboss.embedded.tutorial.javase.beans.CustomerDAOBean ejbName: CustomerDAOBean
-----
Successfully created and found Gavin from @Local interface
Successfully created and found Emmanuel from @Remote interface
-----
STOPPED EJB: org.jboss.embedded.tutorial.javase.beans.CustomerDAOBean ejbName: CustomerDAOBean
Stopping persistence unit persistence.units:jar=tutorial.jar,unitName=custdb
Queue[queue/PrivateExpiryQueue] stopped
Queue[queue/PrivateDLQ] stopped
Queue[queue/QueueWithOwnDLQAndExpiryQueue] stopped
Queue[queue/ExpiryQueue] stopped
Topic[topic/TopicWithOwnRedeliveryDelay] stopped
Queue[queue/QueueWithOwnRedeliveryDelay] stopped
Topic[topic/TopicWithOwnDLQAndExpiryQueue] stopped
Queue[queue/DLQ] stopped
[/ConnectionFactory, /XAConnectionFactory, java:/ConnectionFactory, java:/XAConnectionFactory] undeployed
Unbound ConnectionManager 'jboss.jca:service=DataSourceBinding,name=JmsXA' from JNDI name 'java:JmsXA'
org.jboss.jms.server.messagecounter.MessageCounterManager@1b27882 isn't started
JMS ServerPeer[0] stopped
Unbound ConnectionManager 'jboss.jca:service=DataSourceBinding,name=DefaultDS' from JNDI name 'java:DefaultDS'
Database standalone closed clean
Stopping recovery manager
uninstalling bean: jboss.j2ee:jar=tutorial.jar,name=CustomerDAOBean,service=EJB3
uninstalling bean: persistence.units:jar=tutorial.jar,unitName=custdb
```

# Using Embedded JBoss with JUnit

- Can roll your own as described
  - Uses the same code as embedding in a custom application
  - Start JBoss is a one time setup method
  - Shutdown JBoss in a final cleanup method
- Embedded JBoss includes BaseUnitTest
  - Junit 3.x utility class
  - Helps with the setup and tear down methods

# Example Message Driven Bean

- Message Driven Bean defined using EJB 3.0 annotations

```
@MessageDriven(activationConfig =
{
    @ActivationConfigProperty(propertyName="destinationType", propertyValue="javax.jms.Queue"),
    @ActivationConfigProperty(propertyName="destination", propertyValue="queue/example")
})
@Depends("jboss.mq.destination:name=example,service=Queue")
public class ExampleMDB implements MessageListener
{
    public static boolean executed = false;

    public void onMessage(Message recvMsg)
    {
        System.out.println("-----");
        System.out.println("Received message");
        executed = true;
        System.out.println("-----");
    }
}
```

# Testing ExampleMDB

- Bootstrap JBoss
- Deploy the Message Driven Bean
- Send message
- Validate response

```
public void testMDB() throws Exception
{
    Bootstrap.getInstance().bootstrap();
    Bootstrap.getInstance().deployResourceBase(ExampleMDB.class);
    ExampleMDB.executed = false;

    InitialContext ctx = new InitialContext();
    ConnectionFactory factory = (ConnectionFactory) ctx.lookup("ConnectionFactory");
    Destination destination = (Destination) ctx.lookup("queue/example");
    assertNotNull(destination);
    Connection conn = factory.createConnection();
    Session session = conn.createSession(false, Session.AUTO_ACKNOWLEDGE);
    TextMessage message = session.createTextMessage("hello");
    MessageProducer producer = session.createProducer(destination);
    producer.send(message);
    session.close();
    conn.close();
    // wait a second just to make sure message was delivered.
    Thread.sleep(1000);
    assertTrue(ExampleMDB.executed);
    Bootstrap.getInstance().shutdown();
}
```

# Best Practices for Unit Testing

- Start Embedded JBoss as few times as possible
  - Ideal is once per test suite
  - Can use the `BaseTestCase` or `EmbeddedTestCase` to help simplify
- Use in-memory databases
  - Minimizes external dependencies
  - Allows for test data creation
  - Most commonly used is Hypersonic DB
- Make a habit of testing your EJBs!

# Embedding JBoss in Tomcat

- Put JBoss JAR and configuration files in Tomcat directories (not only in the WAR!)
  - common/classes
  - common/lib
  - server/lib
- Bootstrapping done with a server listener
  - Requires modification to server.xml
  - Handles bootstrap and shutdown

```
<Server port="8005" shutdown="SHUTDOWN">  
  
  <!-- Comment these entries out to disable JMX MBeans support used for the  
    administration web application -->  
  <Listener className="org.apache.catalina.core.AprLifecycleListener" />  
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" />  
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />  
  <Listener className="org.apache.catalina.storeconfig.StoreConfigLifecycleListener"/>  
  <!-- Needs to be after other listeners! -->  
  <Listener className="org.jboss.embedded.tomcat.EmbeddedJBossBootstrapListener" />
```

# Deploying Resources From WARs

- Scanning all classpath resources
  - Simple addition of a listener to the **context.xml**
  - Potential performance ramifications

```
<Context>  
  <Listener className="org.jboss.embedded.tomcat.WebinfScanner" />  
</Context>
```

- Configure for a individual WAR
  - Modification to web.xml
  - Specify resources in context parameters

```
<web-app>  
  <context-param>  
    <param-name>jboss.embedded.resources</param-name>  
    <param-value>/WEB-INF/lib/titan-ejbs.jar, /WEB-INF/classes/oracle-ds.xml</param-value>  
  </context-param>  
  
  <listener>  
    <listener-class>org.jboss.embedded.ServletContextResourceScanner</listener-class>  
  </listener>  
  
</web-app>
```

# Accessing the EJBs from the WAR

- Same code as for non-embedded EJB references

```
<%@ page import="javax.naming.InitialContext" %>
<%@ page import="org.jboss.embedded.tutorial.war.beans.Customer" %>
<%@ page import="org.jboss.embedded.tutorial.war.beans.CustomerDAOLocal" %>
<%@ page import="org.jboss.embedded.tutorial.war.beans.CustomerDAORemote" %>

<html>
<body>
<%
    InitialContext ctx = new InitialContext();
    CustomerDAOLocal local = (CustomerDAOLocal) ctx.lookup("CustomerDAOBean/local");
    CustomerDAORemote remote = (CustomerDAORemote) ctx.lookup("CustomerDAOBean/remote");

    int id = local.createCustomer("Gavin");
    Customer cust = local.findCustomer(id);
%>
<p>
    Successfully created and found Gavin from @Local interface: <%=cust.getName() %>
</p>
<%
    id = remote.createCustomer("Emmanuel");
    cust = remote.findCustomer(id);
%>
<p>
    Successfully created and found Emmanuel from @Remote interface
</p>
DONE!
</body>
</html>
```

# Wrap Up

- Embedded JBoss can be a powerful way to use JEE functionality in non-traditional places
  - EJB Unit Tests!
  - Standalone applications
  - Tomcat only deployment environments
- Embedded JBoss is beta software.
  - There may be bugs
    - Does not work with Java 1.6
    - Cannot run in Windows XP from directories with a space in their path
  - Error messages may be sparse
  - Small user community

# Useful Resources

- Downloads

- [https://sourceforge.net/project/showfiles.php?group\\_id=22866&packa](https://sourceforge.net/project/showfiles.php?group_id=22866&packa)

- User Guide

- <http://wiki.jboss.org/wiki/Wiki.jsp?page=EmbeddedJBoss>

- Forum

- <http://www.jboss.com/index.html?module=bb&op=viewforum&f=265>

- Bill Burke Description

- [http://blogs.jboss.com/blog/bburke/2007/04/17/Embedded\\_JBoss\\_JE](http://blogs.jboss.com/blog/bburke/2007/04/17/Embedded_JBoss_JE)

- Source Code

- <http://fisheye.jboss.org/browse/JBossAS/trunk/embedded>