EJB Basics – By Example

- Dave Landers
- BEA Systems, Inc.
- dave.landers@4dv.net
- dave.landers@bea.com
Agenda

- EJB Overview
- Parts of an EJB
  - Component Interface
  - Home Interface
  - Implementation
  - Deployment Descriptor
- Writing simple EJBs
- Limitations
- Etc.
What is an EJB?

- Enterprise Java Beans
  - Not JavaBeans
- Architecture for server-side components
- Lots of services provided for you by Container
  - Transactions, security, etc.
  - Most are declarative – no coding
    - You still have to think about and understand these services
- You get a lot for a little work
Where is an EJB

Browser

Swing App

JSP
Web Server

EJB

EJB

Database

Server
EJB Features

- Transactions
- Security
- Database
- Distributed Components
  - RMI, CORBA
- Container Managed Persistence
- Performance and Scalability
  - Pooling, Load Balancing, etc.
- Descriptor-Based Features
  - Deployment of new code to running server
Kinds of EJB

- Session Beans
- Entity Beans
- Message Driven Beans

- Each has its use
- Coding is similar for each
Session Beans

- Usually “business methods”
  - They do something
- Often used to provide coarse grained access
  - Interact with several other EJBs, services, etc.
- Two kinds
  - Stateless
    - Most common
  - Stateful
    - State maintained between method calls
    - State is “conversational” not “durable”
      - Think memory or files, not databases
Entity Beans

- Usually represent “data”
  - Often stored in database

- Durable persistence
  - Survives “crash” of container
  - Container- or Bean-Managed (CMP or BMP)
  - CRUD: Create, Read, Update, Delete

- Unique Primary Key identifies individual Entities

- Relationships to other Entities
Parts of an EJB

- **Home Interface**
  - Factory for creating, finding, deleting EJBs
  - Looked up from JNDI

- **Component Interface**
  - Client makes method call on these interfaces

- **Implementation Class**
  - You write this to implement the EJB
  - Only the container calls it

- **Primary Key class for Entities**

- **Deployment Descriptor(s)**
  - Instructions to the container
Local vs. Remote

- Remote
  - Distributed calls (RMI/CORBA)
  - Pass-by-Value (Serialized)

- Local (EJB 2.0)
  - Must be in same JVM
  - Pass-by-Reference
  - No RemoteException

- EJB can have Either or Both
- Can have the Same or Different Methods
EJB Component Interface

- Remote and/or Local
- extend EJBObject or EJBLocalObject
- Add your methods
  - Entities usually have get/set methods
  - Sessions usually have “operations”
  - If Remote, methods must throw java.rmi.RemoteException
- All arguments and return values for Remote must be Serializable
EJB Home Interface

- extends EJBHome or EJBLocalHome
  - One for each component interface

Requirements
- Create methods (not required, but common for Entity)
  - Entity findByPrimaryKey

Optional
- Several Create methods
  - Except just one for Stateless
- Entity Beans
  - Other finders
  - Home Methods
The Implementation Class

- Implements SessionBean or EntityBean
  - No difference in interface for Stateful / Stateless

- Can Not implement Component or Home Interface
  - Not allowed
  - Method signatures differ slightly
  - Javac can’t help you get it right
    - Vendor “EJB Compiler” tools
    - IDE features
    - Errors when you deploy
Code Break

- Simple Stateless Session Bean
- HelloWorld
- Only one method
  - String sayHello()
- Local and Remote Interfaces
- Only one implementation class
Writing the Implementation Class

Component Interface Methods

- The business methods that do the work
- Same signature
  - Do not throw RemoteException
  - Should pick a better Exception
Implementation...

Container Callbacks – called on state changes

➢ Session Beans
  • setSessionContext
  • ejbCreate
    ✓ Matches create(...) method(s) from Home
    ✓ Only no-args for Stateless
  • ejbActivate, ejbPassivate
    ✓ Used in Stateful only
  • ejbRemove
Simple Stateful Session Bean

Counter

Keeps an int as State

One "business" method

```java
int getNext()
{
    return counter++;
}
```
Special things for Entity Beans

- Primary Key Class
  - Identity for the instance – standard or custom class

- Finders to do lookups (SQL SELECT queries)
  - findByPrimaryKey (required), findXXX (optional)
  - Return Remote/Local or Collection
    - Implementation returns PK or Collection of PK

- Home Methods (EJB 2.0)
  - Not instance specific
    - Like static methods on an object
  - Prior to EJB 2.0, was usually in companion Session Bean
Entity Bean Implementation...

- Container Callbacks – called on state changes
  - `setEntityContext`, `unsetEntityContext`
  - `ejbActivate`, `ejbPassivate`
  - `ejbCreate`
    - Matches `create(...)` method(s) from `Home`
    - BMP Entity – SQL INSERT, return Primary Key
  - `ejbPostCreate` – Entity Beans
    - Called after EJB created in Container – has identity
    - Do things that need EJB reference - establish relationships
  - `ejbRemove`
    - BMP Entity – SQL DELETE
  - `ejbLoad`, `ejbStore`
    - Entity Beans only – BMP SQL SELECT & UPDATE
Implementation...

- Other Entity Home Methods
  - findByPrimaryKey, findXXX, other home methods
  - Similar signature as in Home
    - Method names prefix with “ejb” and next letter upper-cased

- Return Types
  - Session ejbCreate() returns void
  - Entity ejbCreate() returns Primary Key type
    - BMP returns PK instance, CMP returns null
  - Entity ejbFindXXX() returns PK or Collection of PK
Entity Beans
– Container Managed Persistence

- Abstract implementation class
- Each “attribute” has abstract get/set methods
  - Implemented by the container
  - Mapping described in deployment descriptor
- Bind finders to database using EJB-QL
  - Kind of like SQL
  - In deployment descriptor
- Select Methods allow EJB-QL to be used from home or business methods
- Can also set up relationships with other Entity Beans
Code Break

- Simple Entity Bean
- Property
- Persists Key and Value pairs in Database
- Container Managed Persistence
Entity Beans
– Bean Managed Persistence

- You write the database code
- JDBC and SQL in
  - `ejbCreate`, `ejbRemove`
  - `ejbLoad`, `ejbStore`
  - Finders, select methods
- Maintain state of `EJBObject` vs. Database
- Good if:
  - Binding is too complex for CMP / EJB-QL
  - Persistence is not to database
  - You are a control freak or have extra time....
Code Break

- Property
- Bean Managed Persistence
- Extends CMP class and provides just persistence methods
EJB Lifecycle

For All EJB Types:
- Lifecycle of Instances is managed by the Container
- EJB gets callbacks at appropriate times
- More State == More Complicated

Stateless Session
- Does Not Exist

- newInstance
- setSessionContext()
- ejbCreate()
- ejbRemove()
- Method Ready Pool
- method
Stateful Session Lifecycle

- Does Not Exist
  - newInstance
  - setSessionContext()
  - ejbPassivate()
  - ejbCreate()

- Method Ready
  - ejbRemove()

- Passive
  - ejbPassivate()
    - (LRU victim)
  - ejbActivate()
    - (method)

System Exception from any method
Entity Lifecycle

- newInstance
- setEntityContext()
- unsetEntityContext()
- ejbCreate()
- ejbPostCreate()
- ejbActivate()
- ejbPassivate()
- ejbFindX()
- ejbSelectX()
- ejbStore()
- ejbRemove()
- ejbHomeX()
- ejbLoad()

System Exception from any method

Does Not Exist

Pooled

Ready
About Exceptions

- Throw any exception that makes sense

Application Exceptions
- These are non-`RuntimeExceptions`
- In your throws clause
- You must handle things like rollback

System Exceptions
- `RuntimeException`, `Error` (and subclasses)
- Container must
  - Log it
  - Rollback transaction
  - Destroy bean instance
Deployment Descriptor

- In EJB jar file: META-INF/ejb-jar.xml
- Declares each EJB and
  - References to other EJBs (ejb-ref)
  - Environment Properties (env-entry)
  - Database and other resources (resource-ref)
  - Security restrictions
  - Transaction settings

- CMP Definitions
  - Fields and Queries

- Vendor specific Descriptor
  - Server specific configurations, tunings, etc.
Code Break

Deployment Descriptor for other examples
Client view of EJB

- Lookup Home with JNDI
- Create or find bean
- Make method calls
Client EJB calls

- Context jndiCtx = new InitialContext();
  - Might need JNDI properties for server connection

- Object o = jndiCtx.lookup( "beanJndiName" );
- MyLocalHome = (MyLocalHome) jndiCtx.lookup( "localBeanJndiName" );

- MyRemoteHome home = (MyRemoteHome) PortableRemoteObject.narrow( o, MyRemoteHome.class );
  - Narrow needed only if Remote
Use EJBHome to get EJBObject

- MyEJB ejbObj = home.create();
- MyEJB ejbObj = home.create( args );
- MyEntity ejbObj = home.findByPrimaryKey( pk );
- MyEntity ejbObj = home.findTheOne( ... );
- Collection c = home.findTopTen( ... );

➤ If Remote, objects retrieved from collection must be Narrowed
ejbObject.remove()

- When done with Stateful Session “conversation”
- Not necessary for Stateless Session
- Removes Entity from database

Don’t use equals()

- ejbObject.isIdentical( EJBOBJECT other )
- For Entity or Stateful Session

ejbObject.getPrimaryKey()

- Returns Entity PK
Code Break

- Client code for examples
- JSP
Message Driven Beans

- No client interfaces (component or home)
- Implement
  - MessageDrivenBean
  - javax.jms.MessageListener
- One method:
  - onMessage(javax.jms.Message m)
- Tied to JMS Destination when deployed
- EJB 2.1 will let MDB receive non-JMS messages
EJB Limitations

- No Read/write static fields
  - Might not be accessible by all EJBs
  - Container might spread across multiple JVMs

- No Thread synchronization or thread management
  - Might not work like you think
    - Container spread across multiple JVMs
  - Mess up the container pooling, load balancing, etc.

- No File I/O
  - Database is better

- No Server sockets or multicast

- No ClassLoader games or Native Library Loading
EJB Limitations

- Usually OK for EJB to use objects that do these things

- But:
  - Read the rationale in the EJB spec
    - Chapter 24.1.2 – Programming Restrictions
  - Make sure you are aware of the reasons for the restrictions
  - Food for Thought:
    - What is a Singleton?
      - Consider distributed system, multiple ClassLoaders, multiple JVMs, multiple computers....
EJB Limitations

Never give away “this”

- Object reference to implementation
- It “belongs” to the container
- Get interfaces from SessionContext or EntityContext and pass these around instead
  - context.getEJBObject()
  - context.getEJBLocalObject()
Design hints

- Use locals where possible
  - Intra-server calls
  - Hide some things from Remote clients
    - Only deployed code has access

- Use “queries” rather than “bulk” finders
  - If finder would return a lot
  - Finder returns EJBObject, might swamp container pool
  - Query as Home method returning PKs
  - Can use as needed
  - Or set up finder to return “reasonable sized” sets
    - Like cursor
Design hints

- **Entity Value Object**
  - Serializable object representing state of Entity
  - Client can work with this rather than lots of remote calls (individual get/set methods)
  - Entity has getValue / setValue

- Consider where state is kept
  - Client (memory or HttpSession) vs. Stateful Session

- Session Bean methods to access groups of EJBs
  - Rather than all on the client
  - Allows control of logic, transactions, security, etc.
What’s New – EJB 2.1

- Expanded EJB-QL
  - Adds useful things SQL users are used to like ORDER BY, MIN, MAX, SUM, COUNT, etc.

- Timer Service
  - EJBs can get timed callbacks from container

- Web Services
  - Stateless Session Beans as Web Services endpoints
  - JAX-RPC
Summary

- Component Interface
  - Contract for the Component
  - Local and/or Remote

- Home Interface
  - Factory
  - Create, Find, etc.

- Implementation
  - Container Required Stuff
  - Your Code for Component and Home

- Deployment Descriptor
Book Recommendations

- Enterprise JavaBeans
  - Richard Monson-Haefel
- Mastering Enterprise JavaBeans
  - Ed Rowman, et. al.
  - www.theserverside.com
- Practical Java Programming Language Guide
  - Peter Haggar
- Effective Java Programming Language Guide
  - Joshua Bloch
- Mr. Bunny's Big Cup o' Java
  - Carlton Egremont III
Web References

- EJB Spec
  - Go ahead, it’s only 572 pages

- J2EE API Docs
  - http://java.sun.com/j2ee/sdk_1.3/techdocs/api

- The Server Side
  - http://www.theserverside.com
  - News, Patterns, Discussion, Downloads, etc.
The End – Thank You

- Please fill out evaluations

- Example Code
  - On the conference CDROM
  - http://www.avitek.com/landers

  dave.landers@4dv.net
dave.landers@bea.com
Container Callbacks – called on state changes

- `setSessionContext`, `setEntityContext`, `unsetEntityContext`
- `ejbActivate`, `ejbPassivate`
- `ejbCreate`
  - Matches `create(...) from Home`
  - BMP Entity – SQL INSERT, return Primary Key
- `ejbPostCreate` – Entity Beans
  - Called after EJB created in Container – has identity
  - Do things that need EJB reference - establish relationships
- `ejbRemove`
  - BMP Entity – SQL DELETE
- `ejbLoad`, `ejbStore`
  - Entity Beans only – BMP SQL SELECT & UPDATE