

By Eileen Sauer.

Email: [eileen@sauer.as](mailto:eileen@sauer.as)

For any corrections / updates / additions.

# EJB Specification v1.1 Matrix

Hosted on:  
**TheServerSide.com**  
 Your J2EE Community

## Component Contract Matrix

The following matrix shows you:

- Interfaces
- callbacks for those interfaces
- when they occur, under what conditions
- related page numbers in the EJB 1.1 spec.

| Callback   | Entity Beans   |      |   |      | Session Beans   |      |  |      |
|--|--|------|---|------|---|------|--|------|
|  | CMP  | When | BMP   | When | Stateful  | When | Stateless  | When |
| <i>javax.ejb.EJBHome interface extends java.rmi.Remote</i> |  |      |   |      |   |      |  |      |
| getEJBMetaData()   | Get EJBMetaData interface  |      |   |      | get EJBMetaData interface   |      |  |      |
| getHomeHandle()  | Get handle <sup>5</sup> for home object  |      |   |      | Get handle <sup>5</sup> for home object   |      |  |      |
| remove(Handle)   | Remove EJB object identified by handle   |      |   |      | Remove EJB object identified by handle  |      |  |      |
| remove(PK)   | Remove EJB object identified by PK   |      |   |      | N/A – throws RemoteException  |      |  |      |
| <i>Home interface extends javax.ejb.EJBHome</i>            |  |      |   |      |   |      |  |      |
| general  | Create/find/remote entity object p89   |      |   |      | Allowed to have super interfaces p77  |      |  |      |
| create()   | Zero or more create() methods p89 <sup>4</sup><br>Calls newInstance(), setEntityContext(), ejbCreate(), ejbPostCreate()<br>Arguments typically used to initialize state, must return remote interface<br>Entity bean instance life cycle, p102 |      |   |      | Must have at least one ejbCreate() p75<br>Calls newInstance(), setSessionCtx(), ejbCreate()<br>Must return remote interface (p77), unlike ejbCreate() |      | Must have only one ejbCreate()/no args<br>Calls newInstance(), setSessionCtx() ejbCreate()<br>Must return remote interface (p77), unlike ejbCreate() |      |
| findByPrimary Key()  | One method<br>Can find object, return remote interface   |      | One method<br>Do SELECT, check if exists<br>Can find object, return remote interface          |      | N/A   |      |  |      |
| findXXX()  | May have one or more<br>Single/multiple finders p114<br>Return remote interface or Collection  |      | May have one or more<br>Single/multiple finders p114<br>Return remote interface or Collection |      | N/A   |      |  |      |
| remove()   | N/A  |      |   |      | ejbRemove() - error if client invokes while in transaction  |      |  |      |
| <i>javax.ejb.EJBObject extends java.rmi.Remote</i>         |  |      |   |      |   |      |  |      |
| getEJBHome()   | Get home interface<br>Implementation generated by container provider tools at deployment time p98  |      |   |      | Implementation provided by container p43  |      |  |      |
| getHandle()  | Get handle <sup>5</sup> for entity object  |      |   |      | Implementation provided by container p43  |      |  |      |
| getPrimaryKey()  | Valid - get identity of entity object  |      |   |      | N/A   |      | N/A - throws RemoteException   |      |
| isIdentical()  | Use this or equals(), == not guaranteed (because hashCode() not guaranteed) p93<br>If same home and same PK, considered identical p92  |      |   |      | Valid   |      | True if same SB, same home interface   |      |
| remove()   | Valid?? p94  |      |   |      |   |      |  |      |
| <i>Remote interface extends javax.ejb.EJBObject</i>        |  |      |   |      |   |      |  |      |

|   |  |   |  |   |
|---|--|---|--|---|
| general   | Client can: invoke business methods, obtain reference to home interface, pass reference as parameter/return value, obtain primary key, obtain handle, remove entity object p92   |   |  | Allowed to have super interfaces p77  |
| <i>javax.ejb.EnterpriseBean extends java.io.Serializable</i>    |  |   |  |   |
| serialVersionUID  | Inherited from Serializable  |   |  |   |
| <i>javax.ejb.EntityBean extends javax.ejb.EnterpriseBean</i>    |  |   |  |   |
| general   | Container provides security, concurrency, transactions, persistence etc p86<br>Multiple clients can access concurrently<br>All EBs are persistent; lifetime not limited to lifetime of VM process p92<br>Represent independent business objects that have independent entities/lifecycle |   |  | N/A   |
| fields  | Must not code explicit data access<br>Defer to container<br>Use cmp-field elements of DD to ID fields container must load/store p130   |   |  |   |
| methods   | Method calls via home/remote interface may be remote. Call must go through container/parameters passed by value are copied. <i>Make coarse-grained</i> p99   |   |  | N/A   |
| <i>javax.ejb.SessionBean extends javax.ejb.EnterpriseBean</i>   |  |   |  |   |
| general   | N/A  |   |  | Unique ID assigned by container<br>Anonymous to client, hide identity   |
|   |  |   |  | N/A<br>Instance has no state for a <i>specific</i> client p67   |
| methods   | N/A  |   |  | Delegated to bean instance<br>If instance is passivated, it's reactivated   |
| args/return types   | N/A  |   |  | Must be valid types for RMI/IOP   |
| instance fields   | N/A  |   |  | Can be: non-transient, serializable object, null, remote/home interface ref, etc p51  |
| <i>Bean implementation implements EntityBean or SessionBean</i> |  |   |  |   |
| general   | Pooled state - not associated with identity. Done via ejbPassivate or ejbRemove()<br>Ready state - assigned identity, via ejbCreate, ejbPostCreate or ejbActivate() p103<br>Class must be public, not abstract, not final, must have public default constructor, no finalize() p121      |   |  | Must implement business methods and ejbCreate()<br>Class must be public, not abstract, not final, must have public default constructor, no finalize(), may implement remote interface p75 |
| newInstance   | Called by container  |   |  | Unspecified transaction context   |
| setSessionContext   | N/A  |   |  | Unspecified transaction context<br>May be passivated  |
| setEntityContext  | Allows instance to invoke container services/get info about invoker p103<br>Use to allocate resources held by instance for lifetime<br>Executes within unspecified transaction context, ID of EB not available here  |   |  | N/A   |
| unsetEntityContext  | Remove instance in pool, frees resources allocated by setEntityContext()<br>Unspecified transaction context  |   |  | N/A   |
| ejbCreate()   | Match create()<br>Container does INSERT<br>Returns PK<br>Validate client args, has txn ctx   | home.create()<br>After setEntityContext() | Match create()<br>Do INSERT<br>Returns PK<br>Validate client args, has txn ctx | home.create()<br>After setEntityContext()   |
|   |  |   |  | Match create() p76<br>Unspecified transaction context<br>Returns void (unlike create()) p77   |

|                 |  |  |   |  |   |   |   |
|-----------------|--|--|---|--|---|---|---|
| ejbPostCreate() | Match create()<br>Returns void<br>Entity object ID is available  | After setContext()<br>and ejbCreate()  | Match create()<br>Returns void<br>Entity object ID is available | After setContext()<br>and ejbCreate()                                    | N/A   |   |   |
| ejbLoad()       | Container syncs instance with DB before ejbLoad()<br>Txn ctx of bus. method p106   | Depends... p108<br>After ejbActivate, before method call<br>Unreliable when NotSupported 113 | Do SELECT   |  | N/A   |   |   |
| ejbStore()      | Container syncs DB with instance<br>Same txn ctx as previous ejbLoad or ejbCreate  | Before passivate, after last business method<br>Unreliable when NotSupported 113             | Do UPDATE to sync DB with instance                              |  | N/A   |   |   |
| ejbActivate()   | EB can acquire additional resources needed in ready state<br>Unspec'd txn ctx<br>Container assigns ID                                    | Transition from pooled to ready state  | EB can acquire additional resources needed in ready state       | Transition from pooled to ready state                                    | Optionally obtain state<br>Not allowed during a txn <sup>1</sup><br>Unspecified txn context   | When client invokes   | Never called because container creates  |
| ejbPassivate()  | Container removes bean ID<br>Container doesn't serialize instance <sup>1</sup><br>Unspec'd txn ctx                                       | Within a txn p103<br>After ejbStore()<br>Transition from ready to pooled                     | Removes bean ID<br>Unspecified txn ctx                          | Within a txn p103<br>After ejbStore()<br>Transition from ready to pooled | Optionally store state<br>Release resources (e.g. JDBC connections)<br>May serialize instance <sup>1</sup><br>Unspecified txn context | <ul style="list-style-type: none"> <li>➤ Not in txn</li> <li>➤ Timed out or LRU</li> <li>➤ App server runs out of free beans <sup>1</sup></li> </ul>  | Never called because container destroys   |
| ejbRemove()     | Container removes bean ID <sup>1</sup><br>Remove entity object because client called remove()<br>Transaction ctx of remove() method p106 | Remote interface remove() or EB home.remove()<br>Transition from pooled to ready             | Do DELETE   |  | Clean up state<br>Release resources<br>Unspecified txn context  | <p>Upon:</p> <ul style="list-style-type: none"> <li>➤ Bean timeout</li> <li>➤ Server failure</li> </ul> <p>Not upon:</p> <ul style="list-style-type: none"> <li>➤ Container crash</li> <li>➤ System exceptn</li> <li>➤ Client timeout while bean passivated</li> </ul> <p><b>May need clean up:</b> section 6.6.3</p> | <p>Not upon:</p> <ul style="list-style-type: none"> <li>➤ Container crash</li> <li>➤ System exceptn</li> <li>➤ Client timeout while bean passivated</li> </ul> <p><b>May need clean up:</b> section 6.6.3</p> |

|  |  |   |   |   |
|--|--|---|---|---|
| ejbFindByPrimaryKey()                                    | One or more<br>Match findByPK<br>Return remote int or Collection<br>Txn ctx of find method   | home.findByPrimaryKey()   | One or more methods<br>Match findByPK<br>Do SELECT, if exists, return PK<br>Return remote interface or Collection<br>Txn ctx of find method | N/A - anonymous to client               |
| ejbFindXXX()   | Match findXXX()<br>Return remote interface or Collection<br>Txn ctx of find method p133  | Match findXXX()<br>Do SELECT, if exists, return PK<br>Return remote interface or Collection<br>Txn ctx of find method |   | N/A                                     |
| <i>SessionSynchronization interface</i>                  |  |   |   |   |
|  |  |   | <i>Container-managed transactions</i>   |   |
| afterBegin()   |  | N/A   | After new txn begun, same txn context as beforeComp() and business methods  | N/A - must not implement this interface |
| beforeCompletion()                                       |  | N/A   | Before commit; if setRollbackOnly true, not run else you can cache updated data. Same txn context as afterBegin() and bus methods           | N/A - must not implement this interface |
| afterCompletion()  |  | N/A   | After commit/rollback; clean up state if rollback causes inconsistency. Status of txn: committed (true) or rolled back (false)              | N/A - must not implement this interface |
| <i>EJBMetaData interface</i>                             |  |   |   |   |
|  |  |   | Allow dynamic invocation of beans   |   |
| getPrimaryKeyClass                                       |  |   |   | N/A - anonymous to client               |
| <i>javax.ejb.Handle interface</i>                        |  |   |   |   |
| general  | Lifetime/scope implementation specific. At minimum: can store/retrieve across JVMs p95. <i>Not</i> required to store/retrieve across containers.     |   |   |   |
| serialVersionUID   | Inherited from java.io.Serializable  |   |   |   |
| getEJBObject()   |  |   |   | SB exists, get a reference              |
| <i>HomeHandle interface extends java.io.Serializable</i> |  |   |   |   |
| general  | Handle for the home interface  |   |   |   |
| serialVersionUID   | Inherited from java.io.Serializable  |   |   | N/A                                     |
| getEJBHome() <sup>3</sup>                                |  |   |   | N/A                                     |
| getHandle()  |  |   |   | N/A                                     |
| <i>PK class</i>  |  |   |   |   |
| general  | Can be any class that is legal value type in RMI-IIOP, some restrictions 9.2.9<br>May be specific to an EB, but multiple EBs can use the same PK p93 |   |   | N/A                                     |
| implementation   | primkey-field of DD p134<br>Can be unspecified p134  | Must be specified   |   | N/A                                     |
| equals()   | Recommended: implement this  |   |   | N/A                                     |
| hashCode()   | Recommended: implement this  |   |   | N/A                                     |
| <i>javax.rmi.PortableRemoteObject</i>                    |  |   |   |   |
| narrow()   | Must use to convert result of getEJBObject() method invoked on handle to EB's remote or home interface type p94, 96                                  |   | Use for interoperability with EJB containers.<br>Cast likely to fail if container implementation uses RMI/IIOP p47                          |   |

## Transaction Matrix

| Callback   | Entity Beans  |  | Session Beans  |   |
|--|---|--|--|---|
|  | Container-managed transactions  |  | Container-managed transactions   | Bean-managed transactions   |
| <i>General</i>                                     |   |  |  |   |
|  | No nested transactions. Container not required to support JTS, only required to support JTA UserTransaction p155. <i>Application exceptions don't automatically cause rollback</i> p168. Don't retain state between transactions <sup>1</sup> Can provide caching and updates to multiple data sources. Isolation levels p160 |  |  |   |
| <i>EJBContext interface</i>                        |   |  |  |   |
| getCallerPrincipal                                 | May be different in each business method (if moved from pooled to ready) p104<br>Container must have client security context  |  | Return java.security.Principal - identity of invoker<br>Container must have client security context                        |   |
| getEJBHome   |   |  | Return home interface  |   |
| setRollbackOnly                                    | Should only be used if EB methods have transaction context p112   |  | Instance can mark current transaction for rollback. Container must have meaningful transaction context p71                 | N/A   |
| getRollbackOnly                                    | Should only be used if EB methods have transaction context p112   |  | Instance can check if current transaction is marked for rollback<br>Container must have meaningful transaction context p71 | N/A   |
| getUserTransaction                                 |   |  | N/A  | Return UT interface so bean can:<br>➤ Demarcate transactions<br>➤ Obtain transaction status |
| isCallerInRole(string roleName)                    | May be different in each business method (if moved from pooled to ready) p104<br>Container must have client security context  |  | Does caller have a particular role?<br>Container must have client security context   |   |
| <i>EntityContext interface extends EJBContext</i>  |   |  |  |   |
| getEJBObject                                       | Instance can obtain identity of entity object<br>Identity remains associated until passivated/removed p105<br>Disallowed in EB methods where there is no ID p112  |  | Return remote interface  |   |
| getPrimaryKey                                      | Obtain primary key of entity object<br>Might be different each time instance is moved from pooled to ready p104   |  | N/A  |   |
| <i>SessionContext interface extends EJBContext</i> |   |  |  |   |
| getEJBObject                                       |   |  | Return remote interface  |   |
| <i>Transaction Attributes</i>                      |   |  |  |   |
|  | <i>Client has transaction context</i>   |  | <i>Client has NO transaction context</i>   |   |
| NotSupported                                       | Container suspends txn<br>May have unspecified txn context<br>ejbStore(), ejbLoad() unreliable  | Runs<br>May have unspecified txn context<br>ejbStore(), ejbLoad() unreliable | ← Same   |   |
| Required   | Runs<br>Supports set/getRollbackOnly()  | Container creates txn<br>Supports set/getRollbackOnly()                      | ← Same   |   |
| Supports   | Runs (code should be txn-insensitive)<br>May have unspecified txn context   | Runs (code should be txn-insensitive)<br>May have unspecified txn context    | ← Same   |   |

|  |   |   |        |   |
|--|---|---|--------|---|
| RequiresNew                                    | Start one<br>Supports set/getRollbackOnly()   | Suspend txn and start new txn<br>Supports set/getRollbackOnly() | ← Same |   |
| Mandatory                                      | Run<br>Supports set/getRollbackOnly()   | Throw exception<br>Supports set/getRollbackOnly()               | ← Same |   |
| Never  | Throw exception<br>May have unspecified txn context   | Run<br>May have unspecified txn context                         | ← Same |   |
| <i>javax.transaction.UserTransaction (JTA)</i> |   |   |        |   |
| general  | Must not use sql.Connection interface: commit(), setAutoCommit(), rollback(). Must not use UserTransaction p161 |   |        | Enables programmatic transaction demarcation. Don't use sql.Connection  |
| begin()  | N/A   |   |        | Must refresh copy of possibly stale data  |
| commit()                                       | N/A   |   |        | Stateless SBs must commit before method returns p161. Not guaranteed to get same UserTxn across method calls<br>Stateful guaranteed to get same UserTxn until txn terminates <sup>2</sup> |
| getStatus()                                    | N/A   |   |        | Use in place of getRollbackOnly()   |
| rollback()                                     | N/A   |   |        | Use in place of setRollbackOnly()   |

<sup>1</sup> This information comes from a newsgroup/ mailing list – *caveat emptor!*

<sup>2</sup> Danger of long-running transactions if your transaction spans method calls. Not normally done.

<sup>3</sup> Source: EJB API.

<sup>4</sup> This is because with legacy data, entity objects may exist before the container and entity beans deploy. Also, entity objects may be created by inserting rows into the database, not by create(). The rows are still accessible via finder methods. p92

<sup>5</sup> A *handle* is an object that identifies an object on network

To join EJB-INTEREST, send email to [listserv@java.sun.com](mailto:listserv@java.sun.com) and put in the body of the message: subscribe EJB-INTEREST.

#### There are quirks with:

- keywords transient, abstract, final, static and finalize() method
- re-entrancy (no loopbacks) p120
- distributed diamonds (local diamonds are supported by the container)

At best – use caution. At worst – disallowed (maybe it's the other way around)

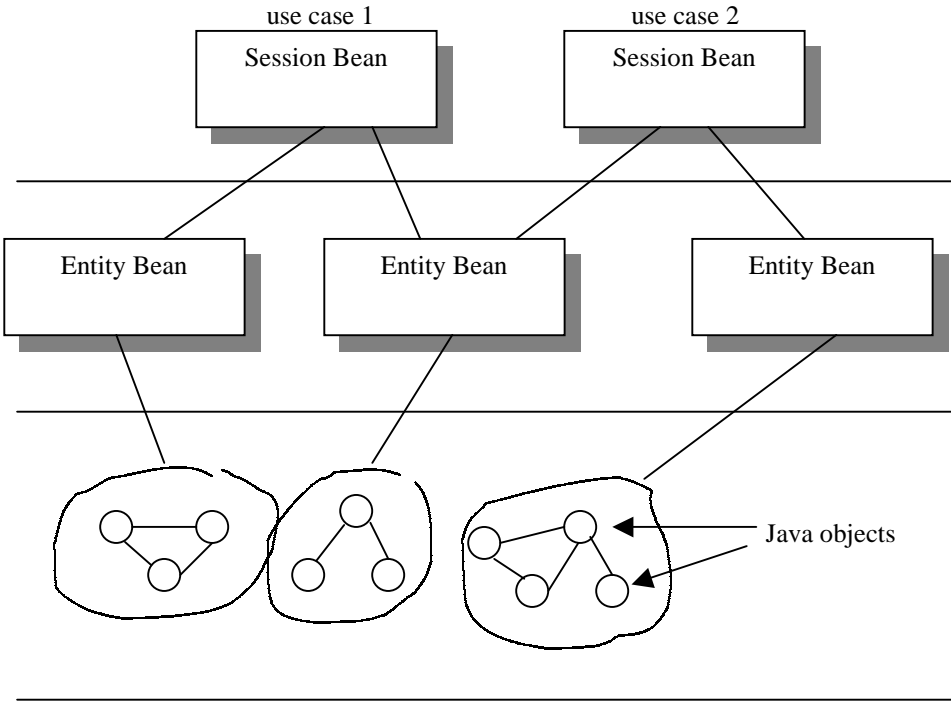
#### Questions:

- FS: What if, in a method, you spawn off 10 threads and join them back up before the end of the method? Legal? p79
- How to handle EJBContext?? EJB-INTEREST Transaction context vs session context (application context). Recommendation: use stateful to maintain your application state, short burst transactions to your underlying persistence. Question about serialization/storing of EJBContext. Same as SessionContext? EntityContext?? Only during stateful beans, activation and passivation don't apply to stateless.
- Stateless session beans good for maintaining open connection to COBOL, legacy system: life of bean instance, not limited to client life cycle, activate/passivate don't affect
- Handling large transactions: start, pause, suspend, abort. JMS, can't multithread. EJB no "oneway" semantics. Need event model?
- Spec doesn't address optimistic concurrency control. Will be part of spec 2.0
- Remote debugging API??

# Recommended Design

## Workflow

## Intelligent Apps



You write – coarse-grained use cases

You write for maximum flexibility:  
➤ determine how coarsely grained  
➤ handle impedance mismatch (e.g. return 3 rows for one EB)

Domain Model: Cocobase-generated  
*Hard part:* Optimal componentization of your Java objects. In general: domain object + all of its dependent objects)  
Exception: isolate intensive processing.  
Lazy instantiate to avoid cascade loads