EJB 2.0 Matrix

Eileen Sauer

Email: eileen@volantec.biz.com

The following matrix shows:

♦ Interfaces
♦ Methods for those interfaces
♦ When they occur, under what conditions
♦ Related sections in the EJB 2.0 Final Release (8/14/01)

2.0 additions:

♦ Message driven beans and integration with JMS
♦ CMP revised for entity beans, added container managed relationships
♦ Defined local interfaces and local home interfaces for session/entity beans
♦ QL – declarative query language
♦ Added select methods for internal use of entity beans with CMP
♦ Added support for business logic methods on home interface
♦ Added declarative run-as security identity functionality for enterprise beans
♦ Defined CORBA interoperability protocol

Clients

	Remote Clients	Local Clients
general	Client of an SB or EB may be remote client or local client. Typically one or the other view will be provided. 4.2.1 ES: PFD1 had dependent objects; these local interfaces in PFD2 essentially replace dependent objects. More consistency, e.g. now create a graph of beans with same interfaces, transactional/security/persistence behavior, etc. Issues: issues with semantic differences, best practices?	
	Access SBs or EBs through the remote interface and remote home interface 5.2 Location independent, Java RMI interfaces 5.2 Args/results passed by value, protects against inadvertent modification, classes decoupled from persistence layer 5.2, 5.4, 10.3.10	Collocated in the same JVM with the SB or EB 5.3 Not location independent, access SBs or EBs through local interface and local home interface 5.3 Args/results passed by reference, can manipulate large amount of data 5.3, 5.4
client types	Can be Java application, applet, servlet. Can be mapped to non-Java clients 5.2	May be another enterprise bean (SB, EB or MDB) 5.3
determining view	Loose coupling between client and bean, potentially expensive 5.4 Requires javax.rmi.PortableRemoteObject.narrow() instead of Java casts 5.4	Supports fine-grained component access 5.4
includes	Home/component interfaces, object identity, metadata interface, handle	Home/component interfaces, object identity
home interface	extends javax.ejb.EJBHome home.createXXX() returns remote interface, unlike ejbCreate() 6.3.1	extends javax.ejb.EJBLocalHome home.createXXX() returns local interface, unlike ejbCreate() 6.4.1
home lookup	Context ctx = new InitialContext(); cartHome cartHome = (CartHome)javax.rmi.PortableRemoteObject.narrow(ctx.lookup("java:comp/env/ejb/cart"), CartHome.class); 6.2.1	Context ctx = new InitialContext(); CartHome cartHome = (CartHome) ctx.lookup("java:comp/env/ejb/cart"); 6.2.1
component interface	extends javax.ejb.EJBObject	extends javax.ejb.EJBLocalObject Reference must not be passed out of the JVM in which it was created 6.7.2
exceptions	javax.ejb.NoSuchObjectException java.rmi.RemoteException	javax.ejb.NoSuchObjectLocalException javax.ejb.EJBException

Component Contract

Method	Entity Beans		Session Beans	
	CMP	BMP	Stateful	Stateless
general	6 roles: Enterprise Bean Provider, Application Assembler, Deployer, EJB Server Provider, EJB Container Provider, System Administrator 3.1.1 – 6 The specification does not address optimistic concurrency control, and doesn't prescribe how the container is to handle transactions. This is designed to give the container as much freedom as possible. Also, optimistic concurrency control is sometimes done on the DB layer. Developers are expected to catch SQLExceptions resulting from optimistic transaction rollbacks, and manually code the retry.			
javax.ejb.EJBHome interface extends java.rmi.Remote				
general	Generated by deployment tools, implements EJB remote home interface 7.11.2, 9.9.2, 10.7.5			
getEJBMetaData()	Get EJBMetaData interface for EB 9.5		Get EJBMetaData interface for SB 6.3	
getHomeHandle()	Get handle for remote home interface 9.5, 9.12		Get handle for remote home interface 6.3	
remove(Handle)	Container calls ejbRemove(). Entity representation should be removed from DB. Bean instance returned to pool. Subsequent client calls to bean raises NoSuchObjectException or NoSuchObjectLocalException. 9.5.3, 9.6.3	Container calls ejbRemove(). Entity representation should be removed from DB. Bean instance returned to pool. Subsequent client calls to bean raises NoSuchObjectException or NoSuchObjectLocalException. 9.5.3, 9.6.3	Client stub invalidated. (Bean may remain in the pool. ejbRemove() isn't necessarily called.) Subsequent client calls to bean raises NoSuchObjectException or NoSuchObjectLocalException. Attempt to remove while in transaction raises RemoveException 6.3.2, 6.4.2, 7.8.1	Container calls ejbRemove(). Subsequent client calls to bean raises NoSuchObjectException or NoSuchObjectLocalException. 6.3.2, 6.4.2, 7.6
remove(PK)	Same	Same	Raises javax.ejb.RemoveException. 6.3.2, 6.4.2	Raises javax.ejb.RemoveException. 6.3.2, 6.4.2
javax.ejb.EJBLocalHome interface				
general	Generated by deployment tools, implements EJB local home interface 7.11.4, 9.9.2, 10.7.7			
Home interface extends javax.ejb.EJBHome or javax.ejb.EJBLocalHome				
general	**Home interface is allowed to have super interfaces,** subject to RMI/IIOP rules 7.10.6, 7.10.8 Example of inheritance 7.2, Figure 17			
	Have runtime object identity maintained by container, based on PK 10.3.5 Subsequent access after removed causes java.rmi.NoSuchObjectException 9.5.3 Arbitrary method names must not start with create/find/remove 9.5.4, 10.6.10,12 Entities may be created/deleted via direct database manipulation 9.7		Has unique identity assigned by container at create time 6.9.1 Conversational state: instance's field values plus transitive closure 7.4	All session objects with same home have same object identity 6.9.2 May contain state across client-invoked calls, e.g. open DB connection 6.8
	Create/finder/home 10.6.10, 10.6.12	Create/remove/find/home 12.1.2		
createXXX()	Zero or more create() methods 9.5.1, 9.6.1 Arguments typically used to initialize state 9.6.1 Return type is the entity bean's remote interface (unlike ejbCreate()) 9.5.1, 11.2.9 Must throw java.rmi.RemoteException and javax.ejb.CreateException 9.5.1 Container calls obtains an available bean from the pool, calls ejbCreate(), ejbPostCreate(). (If the container must first instantiate a new instance to grow the pool, newInstance() and setEntityContext() are called before ejbCreate() is called.) 10.5.1, 12.1.3		One or more create() methods 6.3.1 Container calls newInstance(), setSessionContext(), ejbCreate() 7.5.5 Return component interface 6.3.1, 6.4.1	Only one create()/no args 7.8 Container calls newInstance(), setSessionContext(), ejbCreate() 7.8.1 Return component interface 7.8 Container can perform instance creation at any time 7.8.1

Method	Entity Beans		Session Beans	
	CMP	BMP	Stateful	Stateless
findByPrimary Key()	Home interface must have 9.5.2 Locate object by PK, return remote interface 9.5.2 Must throw FinderException 9.5.2 Argument must be Object 10.8.3 Container adds CM fields, generate PK	Home interface must have 11.2.9 Must throw FinderException 9.5.2 Do SELECT, check if exists Can find object, return remote interface	Home interface must not define finder methods 5.5	
findXXX()	May have one or more 9.5.2 Every finder method except ejbFBPK() must be associated with query element in DD 10.7.3 Single/multiple finders 10.5.6.1-2 Return remote interface or Collection	May have one or more Single/multiple finders 11.1.8.1-2 Return remote interface, Collection or Enumeration 11.1.8.2	Home interface must not define finder methods 6.6	
javax.ejb.EJBObject interface extends java.rmi.Remote				
general	Generated by container tools, implements EJB's remote interface 7.5.4, 7.11.3, 10.4.1 Delegates invocation of business method to the matching business method implemented in EJB class 7.5.4			
	Remote interface restrictions 10.3.10.1			
getEJBHome()	Get entity object's remote home interface 9.12		Get session object's remote home interface 6.5	
getHandle()	Get handle for entity object 9.11		Get session object's handle 6.5	
getPrimaryKey()	Get identity of entity object (same value as from local interface) 9.8		Session objects hide identity - throws java.rmi.RemoteException 6.6, 6.9.3	
isIdentical(EJBObj)	If same home and same PK, considered identical 9.8 Result of equals() unspecified hashCode() not guaranteed 9.8		Test if session object is identical with another session object 6.5, 6.9.1	Always returns true when comparing session objects of same SB 6.9.2
remove()	Remove the entity object 8.4		Remove session object 6.3.2, 6.5 Subsequent invocations on removed SB raise NoSuchObjectException 6.7.1	Container may not necessarily remove an instance 6.8
javax.ejb.EJBLocalObject interface				
general	Generated by container tools, implements EJB's local interface 7.5.4, 7.11.5, 10.4.1, 10.7.8 Delegates invocation of business method to the matching business method implemented in EJB class 7.5.4			
getEJBLocalHome()	Obtain EB's local home interface		Obtain SB's local home interface Allowed to have super interfaces 7.10.7	
getPrimaryKey()	Obtain PK of EB's local object (same value as from remote interface) 9.8		Session objects hide their identity – throw javax.ejb.EJBException 6.6, 6.9.3	
isIdentical(EJBLocal Object)	Test if a given EB local object is identical to the invoked EB local object		Test if session object is identical with another session object 6.9.1	Always returns true when comparing session objects of same SB 6.9.2
remove()	Remove the EB local object		Remove the SB local object Subsequent invocations on removed SB raise NoSuchObjectLocalException 6.7.2	
Component interface extends javax.ejb.EJBObject or javax.ejb.EJBLocalObject				
general	Defines the operations that allow the client to access the EJB object's identity 4.2.1 Invoke business methods defined in component interface 6.7.2			
	Allowed to have super interfaces, subject to RMI-IIOP rules 10.6.2		Allowed to have super interfaces, subject to RMI/IIOP rules 7.10.5	
javax.ejb.EnterpriseBean extends java.io.Serializable				
serialVersionUID	Inherited from Serializable			
javax.ejb.EntityBean or javax.ejb.SessionBean extends javax.ejb.EnterpriseBean				

_____

Method	Entity Beans		Session Beans	
	CMP	**BMP**	**Stateful**	**Stateless**
general	Container provides security, concurrency, transactions, persistence, etc 8.1 Entity objects are persistent objects, not limited to lifetime of VM 9.7 Multiple clients can access entities concurrently. Transactions isolate clients 9.7 Ideally, client should be able to use object reference across server crash 9.9.9 Object reference should become invalid only when EB is removed/reconfigured 9.9.9 RuntimeException thrown from EB method results in transition to "does not exist" 10.5.1		Session objects have ID that generally doesn't survive container crash/restart 6.1 newInstance, setSessionContext, ejbCreate, ejbRemove, ejbPassivate, ejbActivate and afterCompletion called with unspecified transaction context 7.5.7 RuntimeException thrown from any SB method results in transition to "does not exist" 7.6.2, 7.8.3 Stateful only- Clients not allowed to make concurrent calls. Container *may* (optionally) throw RemoteException (remote) or EJBException (local). 7.5.6	
*Bean implementation implements javax.ejb.EntityBean or javax.ejb.SessionBean – **independent of local/remote client view***				
general	Attempted invocations on entity objects that don't exist result in java.rmi.NoSuchObjectException 9.7.1 May use instance variables to maintain non-persistent state, e.g. JMS connection 10.3.9 Must have public constructor with no arguments 10.6.2 Standard exceptions: Create/DuplicateKey/Finder/ObjectNotFound/Remove 10.5.8		SBs must manage cached database data with respect to transactions and caching persistent state of EBs with CMP 7.1 As a simple rule, declaring session bean's fields as transient is discouraged 7.4.1 Class must be public, not abstract, not final, must have public default constructor, no finalize(), may implement component interface 7.10.2	
	Must define bean as abstract class 10.4.1 CMP and CMR fields must *not* be defined in entity bean class 9.4.1 Specify CMP and CMR fields in DD		Clients cannot make concurrent calls. Container may throw RemoteException or EJBException 7.5.6 If attempted access to resource manager or enterprise bean is not allowed, behavior is undefined 6.6.1 Invocations after RuntimeException result in java.rmi.NoSuchException 7.6.2	No concurrent restriction on stateless beans 7.5.6 If attempted access to resource manager or enterprise bean is not allowed, behavior is undefined 7.8.2 Client view after RuntimeException: session object continues exist, container can delegate to other instances 7.8.3
fields	Use cmp-field elements of DD to ID fields container must load/store p130 Rely on tools to generate methods which persist data 9.5.3.1 Once the primary key is set in ejbCreateXXX(), other methods may not change it. 10.3.5	Once the primary key is set in ejbCreateXXX(), other methods may not change it.	Non-transient fields can be serializable object, null, remote/remote home interface ref, local/local home interface ref, SessionContext, etc. Must assume content of transient fields may be lost between passivation and activation 7.4.1	
business methods	Zero or more business methods 10.6.8 Must not start with "ejb", must be public, not final/static 10.6.8		Delegated to bean instance If instance is passivated, it's reactivated 7.11.3 Must not start with "ejb", must be public, not final/static 7.10.4 Throws clause may define arbitrary application exceptions 7.10.4	
deployment	Bean provider must specify relationships using relationship element 10.6.14 Must provide unique names to designate EBs using ejb-name and abstract-schema-name DD elements 10.6.14 Must define query for each finder/select except findByPK() using ejb-ql 10.6.14 May need to encode EJB QL characters ">", "<" since DD is XML 10.6.14		Deployment tool is responsible for generating: ♦ Session EJBHome/EJBLocalHome ♦ Session EJBObject/EJBLocalObject ♦ Handle classes for home/remote interfaces ♦ EJBMetaData class 7.11.1 – 7.11.7	
newInstance	Called by container		Called by container	
set<Session|Entity> Context	Pass reference to EntityContext to EB instance 10.5.2, 12.1.4.1 Unspecified transaction context, ID of EB not available here 10.5.2, 12.1.4.1 Use to allocate resources held by instance for its lifetime 10.5.2, 12.1.4.1		If stored in session bean's conversational state, container must be able to save and restore across passivation 7.4.1 Associates a session bean instance with its context maintained by container 7.5.1	
	EB must not attempt to access persistent state 10.5.2	Cannot be specific to entity object identity, may be reused 12.1.4.1		

Method	Entity Beans		Session Beans	
	CMP	BMP	Stateful	Stateless
unsetEntityContext	Container can remove instance in pool 10.5.2, 12.1.4.1 Unspecified transaction context 10.5.2, 12.1.4.1 Container invokes before terminating life of instance 10.5.2, 12.1.4.1 Use to free resources held by instance 10.5.2, 12.1.4.1		N/A	
	EB must not attempt to access persistent state 10.5.2	ID of entity object not available 12.1.4.1		
ejbCreateXXX()	Match home.createXXX() 10.5.2 Should validate args, initialize all CMP fields, and return *null*. 10.5.2 Should *not* attempt to set CMR relationship fields yet. 10.5.2 Container does DB INSERT after method returns. 10.5.2 Has transaction context 10.5.2 Throws CreateException 10.6.4 Legal to reuse PK of removed EB 10.5.2	Match home.createXXX() 12.1.4.1 Returns PK 12.1.4.1 Validate client args/has txn ctx 12.1.4.1 INSERT a record in DB 12.1.4.1 May throw CreateException 12.1.8.1 Must have unique PK 12.1.8.2 Throw DuplicateKeyException if tries to create EB with duplicate key 12.1.8.2 Legal to reuse PK of removed EB 12.2.3	Match signature of create() 7.5.5 Must have at least one ejbCreate() 7.5.5	Match signature of create() 7.8 Only one ejbCreate()/no args 7.8
			Must return void (unlike create()) 7.10.3 Method arguments must be legal for RMI/IIOP 7.10.3 EJB 1.1 or 2.0 bean should throw javax.ejb.EJBException or another RuntimeException to indicate non-application exceptions to container 7.10.3 Must throw javax.ejb.CreateException 7.10.8	
ejbPostCreateXXX()	Match home.createXXX() 10.5.2 Returns void 10.5.2 Entity object ID is available 10.5.2 After setContext()/ejbCreate() 10.5.2 Bean provider may reference other local EBs 10.5.2 Same txn ctx as ejbCreate 10.5.2	Match home.createXXX() 12.1.4.1 Returns void 12.1.4.1 Entity object ID is available 12.1.4.1 Same txn ctx as ejbCreate 12.1.4.1 May throw application exceptions or CreateException 12.1.8.1	N/A	
ejbActivate()	EB can acquire additional resources needed in ready state 10.5.2, 12.1.4.1 Unspecified transaction context 10.5.2, 12.1.4.1 ID associated until passivated/removed 10.5.2, 12.1.4.1		Open resources (socket, DB) 7.4 Signals the instance it has just been reactivated 7.5.1	Never called because container creates
	EB must not access persistent state 10.5.2	Should not use ejbActivate() to read state from DB, use ejbLoad() 12.1.4.1		
ejbPassivate()	Disassociates instance from ID 10.5.2, 12.1.4.1 Release resources 10.5.2, 12.1.4.1 Unspecified transaction context 10.5.2, 12.1.4.1		Container may passivate only when instance is **not** in a transaction 7.3 Close open resources (DB connections) 7.4 Container serializes after passivation 7.4.1 Rules for destroying instance: 7.4.1, 7.6 If client attempts to invoke, gets java.rmi.NoSuchObjectException or javax.ejb.NoSuchObjectLocalException7.6	Passivation not needed 6.8 Creates instances if needed 6.8 Container may destroy instances that are not needed 6.8
	Must not access persistent state 10.5.2	Should not use ejbPassivate() to write state to DB, use ejbStore() 12.1.4.1		
ejbLoad()	Container syncs instance with DB 10.5.2 Recompute dependencies on persistent state, e.g. uncompress text 10.5.2 Careful passing state of local EB 10.3.9 Transaction ctx of bus. method 10.5.2	Do SELECT to sync instance with state in DB 12.1.4.1 Transaction ctx of bus method 12.1.4.1 Unreliable if unspec'd txn ctx 12.1.6.1	N/A	

Method	Entity Beans		Session Beans	
	CMP	BMP	Stateful	Stateless
ejbStore()	Container syncs DB with instance 10.5.2 Update instance if necessary, e.g. compress text before storage 10.5.2 Same txn ctx as previous ejbLoad or ejbCreate 10.5.2	Do UPDATE to sync DB with state of instance 12.1.4.1 Unreliable if unspec'd txn ctx 12.1.6.1	N/A	
ejbRemove()	Remove entity object because client called remove() or cascade-delete exists 10.5.2 Transaction ctx of remove() method 10.5.2	Match client remove() 12.1.4.1 Transaction ctx of remove() 12.1.4.1 DELETE record from DB 12.1.4.1 Release any resource it would normally release in ejbPassivate 12.1.4.1	Signals that the instance is in process of being removed by container 7.5.1 Typically releases same resources as in ejbPassivate 7.5.1 Invoked when client calls remove on home or remote interface 7.6 Subsequent invocations by client raise java.rmi.NoSuchObjectException 7.6 Can't call when object is in transaction; raises java.rmi.RemoveException 7.6 Container can call after bean expires 7.6 *Not* invoked upon: ➢ Container crash, system exception ➢Client timeout while bean passivated **May need clean up** 7.6.3	*Not* invoked upon: ➢ Container crash ➢ System exception ➢Client timeout while bean passivated **May need clean up**: section 7.6.3 Container invokes when it no longer needs instance 7.8.1
ejbFindByPrimaryKey()	Not specified in query DD 10.7.3 Match findByPK Match zero or one entities Txn ctx of find method	Match findByPK 12.1.7.1 Do SELECT, if exists, return PK Match zero or one entities 12.1.7.1 Txn ctx of find method	N/A - anonymous to client	
ejbFindXXX()	Bean provider doesn't write 10.5.2 Return PK or Collection 10.5.2 Exposed to client 9.6.7 EJBQL SELECT optional 9.6.7 Return EBs of same type 9.6.7	Match findXXX() 12.1.4.1 Txn ctx of find method 12.1.4.1 Do SELECT, if exists, return PK or Collection 12.1.4.1 Return EBs of same type 12.1.4.1 May throw FinderException 12.1.8.3	N/A – session objects hide their identity 6.6	
ejbHomeXXX()	Zero or more EB home methods, matches client-invoked XXX() home method 10.5.2, 10.6.2, 12.1.4.1 Transaction context of XXX() home method 10.5.2, 12.1.4.1 Bean provider does implementation 10.5.2		N/A	
ejbSelectXXX()	Zero or more methods, may be called by EB home method while in pooled state 10.5.2 Special query method not exposed to client in home interface 10.5.2 Must be abstract 10.5.2 Txn ctx of invoking business method Associated with EJB QL element 10.6.7 Return single/multiple values 10.6.7	N/A	N/A	
javax.ejb.EntityContext or javax.ejb.SessionContext interface extends javax.ejb.EJBContext interface				

Method	Entity Beans		Session Beans	
	CMP	BMP	Stateful	Stateless
	EntityContext		SessionContext	
general	Obtain reference to EJB object associated with this instance 10.5.4		If attempted method invocation not allowed, container throws java.lang.IllegalStateException 7.6.1, 7.8.2	
getEJBLocalObj	Obtain reference to EJB local object associated with this instance 10.5.4		Returns session bean's local interface 7.5.2	
getEJBObject	Obtain reference to EJB object associated with this instance 10.5.4 Bean can use Java cast, doesn't need narrow() 10.7.15		Returns session bean's remote interface 7.5.2	
getPrimaryKey	Obtain primary key of EJB object currently associated with this instance Might be different each time instance is moved from pooled to ready 10.5.1		N/A	
javax.ejb.EJBMetaData interface				
general	Not EB specific; container may use a single class for all deployed EBs 10.7.11		Allow application assembly tools to discover information about beans, allows loose client/server binding 6.3	
getEJBHome	Obtain the remote home interface of the enterprise bean			
getHomeInterface Class	Obtain the class object for the enterprise bean's remote home interface			
getPrimaryKeyClass	Obtain the class object for the enterprise bean's primary key class		Session objects hide their identity – throws java.lang.RuntimeException 6.6	
getRemoteInterface Class	Obtain the class object for the enterprise bean's remote interface			
isSession	Test if enterprise bean's type is "session"			
isStatelessSession	Test if enterprise bean's type is "stateless session"			
javax.ejb.Handle interface extends java.io.Serializable				
general	Handle identifies the entity object on the network 9.11 Lifetime/scope implementation specific. Can store/retrieve across JVMs. 9.11 Handle support across container upgrades/replacements not required 9.11 Handle support across different containers not required 9.11 Not EB specific; container may use a single class for all deployed EBs 10.7.9		Handle can be held beyond the life of a client process by serializing to persistent store. When deserialized later, session object it returns will work as long as the session object still exists on server (timeout or crash may destroy it) 6.6	
getEJBObject()	Client code must use javax.rmi.PortableRemoteObject.narrow() to convert result to remote interface type 6.6, 9.11			
HomeHandle interface extends java.io.Serializable				
general	Handle for the remote home interface. Useful if client needs to use home interface in the future but doesn't know the JNDI name 9.12 Not EB specific; container may use a single class for all deployed EBs 10.7.10		Handle for the remote home interface.	
getEJBHome()	Must use javax.rmi.PortableRemoteObject.narrow(…) to convert result to the home interface type 9.12		Return session bean's remote home interface 7.5.2	
PK class				
general	Can be any class that is legal value type in RMI-IIOP 9.8 May be specific to an EB, but multiple EBs can use the same PK 9.8 If client obtains two references from same home, can determine if they refer to the same entity by comparing PKs using equals() method 9.8 Must implement equals() and hashCode() 10.6.13		N/A	

Method	Entity Beans		Session Beans	
	CMP	BMP	Stateful	Stateless
implementation	Can map to single/mult fields 10.8 `primkey-field` of DD 10.8.1 Mult: PK class must be public with public constructor/no args 10.8.2 Names must be subset of CMR 10.8.2 Can be unspecified 10.8.3 May not be visible as cmp-field 10.3.5 Must not reset PK after set 10.3.5		N/A	
javax.ejb.SessionSynchronization interface				
general	N/A		Session bean can optionally implement. Provides SB with txn synchronization notification to, e.g. manage DB data cached within transactions 7.5.3 afterBegin/beforeCompletion/busn methods between executed with the same txn ctx 7.5.7 **Stateless session bean must not implement this interface 6.8** Application Assembler can specify only Required, RequiresNew, Mandatory 16.4.1	
			Container-managed transactions	Bean-managed transactions
afterBegin()	N/A		Signals that a new txn has begun Invoked before first business method within transaction 7.5.3	N/A - must not implement this interface 7.5.3
beforeCompletion()	N/A		Before commit; write out cached DB updates 7.5.3 Container may not call if transaction marked for rollback 7.6	N/A - must not implement this interface 7.5.3
afterCompletion()	N/A		Signals current txn completed. Status of txn: committed (true) or rolled back (false). Clean up conversational state if rollback causes inconsistency 7.4.2	N/A - must not implement this interface 7.5.3

	Message Driven Beans	
general	Instead of using RMI, clients invoke MDBs by sending messages to Destination (Queue or Topic) for which MDB is the MessageListener 15.3 Since RMI is not used, MDBs do not require home and remote interfaces, are entirely stateless, and can be pooled. 15 MDBs asynchronous message consumers, invoked by container on arrival of JMS message 15 Client's JNDI name space can include JMS Destinations of MDBs installed in multiple EJB containers on multiple machines on a network 15.3 Container provides security, concurrency, transactions and other services for the MDB 15.4 If non-durable topic subscription, container is responsible for making sure MDB subscription is active 15.4 MDBs should not use JMS API for message acknowledgement. Handled automatically by container 15.4.8	
deployment	Bean provider can use DD to indicate if MDB is used with a topic or queue, and if topic, whether topic subscriptions are durable or not 15.4 Bean provider may provide advice to deployer as to whether an MDB is intended to be associated with a queue or topic by using message-driven-destination 15.4.9 If MDB is intended to be used with a topic, use subscription-durability to specify whether durable or non-durable. Non-durable assumed 15.4.9 Only DD txn attributes Required and NotSupported are legal. Required always starts a brand new transaction, as the client's transaction context is not shared. 15.4.7	
javax.ejb.MessageDrivenBean interface extends javax.ejb.EnterpriseBean		
general	Container calls newInstance, setMessageDrivenContext(), ejbCreate() 15.4.4 Container allows instances of MDB to execute concurrently, no guarantees made as to the exact order in which messages are delivered 15.4.6 newInstance, setMessageDrivenContext, ejbCreate, ejbRemove called with unspecified transaction context 15.4.7 Class must be public, not final, not abstract, must have public constructor with no args, may have super interfaces, allowed to implement other methods 15.7.2	
ejbCreate()	Must have one ejbCreate(), no arguments 15.4.4 May allocate resources 15.4.11 Throws must not define any application exceptions 15.7.3	
ejbRemove()	Container invokes before it ends the life of an MDB Bean provider cannot assume container will always invoke. Will not be called on crash of container or system exception thrown (and therefore may not release resources allocated in ejbCreate()). Clean up may be needed 15.4.11	
setMessageDriven Context(ctx)	Set associated message-driven context	
onMessage	onMessage() is called when a message has arrived for the bean to service. Contains business logic to process message 15.4.2 If an MDB throws a RuntimeException, the container will garbage collect it and instantiate a new 15.4.10 In bean-managed txn, programmer must commit transaction before onMessage() returns 17.3.3.	
javax.ejb.MessageDrivenContext interface extends EJBContext		
	Container-managed transactions	*Bean-managed transactions*
general	Either Required or NotSupport must be used. Other attributes meaningless 14.4.7 Parameters of createQueueSession(transacted, ackMode) and createTopicSession(transacted, ackMode) are ignored. Specify true, 0 16.3.5 Bean provider should not use acknowledge(). Message acknowledgement in unspecified transaction context handled by container 16.3.5	Message receipt is not part of transaction. If message receipt needs to be part of transaction, CMT with Required must be used 14.4.7
setRollbackOnly	Mark current transaction so that only outcome is a rollback 14.4.3	N/A
getRollbackOnly	Test if current transaction was marked for rollback 14.4.3	N/A
getUserTransactn	N/A	Return UserTransaction interface used to demarcate txns and obtain status 14.4.3
getCallerPrincipal	Inherited from EJBContext interface. Must not call this method 14.4.3	
isCallerInRole	Inherited from EJBContext interface. Must not call this method 14.4.3	
getEJBHome	Inherited from EJBContext interface. Must not call this method 14.4.3	

Container Managed Persistence

	Entity Beans
general	Bean provider develops EBs and relationships. Also designs an abstract persistence schema for each EB. Schema defines container-managed fields/relationships 10.1 Bean provider specifies schema in DD. Deployer uses container provider tools to map fields/relationships to database/persistent store 10.1 Distinction between client view, EB and abstract persistence schema. Allows EBs to be redeployed across different containers/persistent data stores 10.2
relationship	Element defines CMR between CMP EBs 10.3
local only	Relationships defined in terms of local interfaces. Use for related beans. Internal invocations are call by reference ???, usually no checking of method permissions 10.3
EB provider programming contracts	EB class must be abstract. CMP/CMR fields must not be defined in EB class (virtual, accessed only through get/set methods) 10.3.1 CMP/CMR fields specified in DD using cmp-field/cmr-field elements. Field names must begin with lowercase letter 10.3.1 Bean provider must define accessor methods for CMP/CMR fields a get/set methods. Methods must be public, abstract and match CMP/CMR field in DD, where name of cmp-field or cmr-field has been uppercased and prefixed by "get"/"set" 10.3.1 Accessor methods for 1-* or *-* relationships utilize java.util.Collection or java.util.Set (List and Map included in a later version of this specification) 10.3.1 EB local interface type or collection can be cmr-field, not cmp-field 10.3.1 Accessor methods, local interface types and collection classes must not be exposed in remote interface of EB 10.3.1 Once PK for EB has been set, bean provider must not attempt to change via set accessor methods on PK cmp-fields. Do not expose set accessor for PK cmp-fields in component interface of EB. Java types assigned to cmp-fields restricted to Java primitive types and Java serializable types 10.3.1
EB provider view of relationships	1-1, 1-* or *-*, may be bi-directional or uni-directional 10.3.2 Unidirectional implemented with cmr-field on EB sender, and none on receiver. Enables bean provider to restrict visibility of a relationship 10.3.2 EB with no local interface can only have unidirectional relationships from itself to other EBs 10.3.2
Dependent value classes	Public, concrete class, serializable, may be legacy class that bean provider wishes to use internally 10.3.3, 10.6.3 Can be value of cmp-field, cannot be value of cmr-field. Cannot have a member that is an EB or local EJB object 10.3.3 Get accessor method for cmp-field for a dependent value class returns a copy. Assignment causes value to be copied to target cmp-field. 10.3.3
Remove protocols	Via: a. remove() method on EB's component/home interface or b. cascade-delete specification in DD 10.3.4
remove()	When remove() is invoked, ejbRemove() is called. After ejbRemove(), container must remove entity object from all relationships and remove representation 10.3.4.1 Get accessor of a removed element in 1-1 relationship returns null; in 1-* or *-* returns a Collection without element 10.3.4.1 Attempted invocation of method on removed element raises NoSuchObjectLocalException. Assignment attempt to cmr-field raises IllegalArgumentException 10.3.4.1 Container must then cascade removal to all other related EBs where cascade-delete option was specified 10.3.4.1
cascade-delete	Used to specify lifetime of one or more entity objects is dependent upon lifetime of another entity object 10.3.4.2 Can only be specified where other end of relationship has multiplicity 1. Cannot be specified for *-* relationships 10.3.4.2
Updating relationships	For assignment of cmr-field 1-1 relationship to another of same type, object is effectively moved 10.3.6 For 1-* or *-* relationships, methods of Collection have usual semantics, except add() and AddAll(), which move objects that already exist, else just add 10.3.6 Warning with Iterators: don't modify container managed collection while iteration is in progress, raises IllegalStateException 10.3.6 Semantics of set accessor applied to collection-valued cmr-field is also determined by referential integrity semantics 10.3.6.2
EJB QL	Define query methods (finders/select) for EBs with CMP in a portable way. Compilable to target language of database/persistent store 11.1, 11.2.1 Uses abstract persistence schemas of EBs/their relationships for data model. Defines operators/expressions based on data model 11.1 EJB QL query is a string with SELECT/FROM/WHERE. *EJB QL ::= select_clause from_clause [where_clause]* 11.2 EBs designated by abstract schema names 11.2.2 Value of query result is abstract schema type 11.2.4 ID variable (11.2.5.2) followed by navigation operator and cmp-field/cmr-field is path expression. Navigation of CMR path expression may be complex **11.2.5.6**

Transactions

Method	Entity Beans	Session Beans	
	Container-managed transactions	**Container-managed transactions**	**Bean-managed transactions [3]**
General			
	No nested transactions. Container not required to support JTS, only required to support JTA UserTransaction p155. *Application exceptions don't automatically cause rollback p168. Don't retain state between transactions[1]* Can provide caching and updates to multiple data sources. Isolation levels p160		
javax.ejb.EJBContext interface			
getCallerPrincipal	May be different in each business method (if moved from pooled to ready) 9.6.1 Returns java.security.Principal that identifies invoker 10.5.4 Container must have client security context	Return java.security.Principal that identifies invoker 7.5.2 Container must have client security context 7.8.2	
getEJBHome	Return EJB's remote home interface 7.5.2, 10.5.4		
getEJBLocalHome	Return EJB's local home interface 7.5.2, 10.5.4		
getRollbackOnly	Returns true iff current txn marked for rollback. 10.5.4 Should only be used if EB methods have transaction context 10.5.5	True iff txn marked for rollback. 7.5.2 Throws java.lang.IllegalStateException if instance not associated with txn 7.6.1	N/A 16.3.3.1
getUserTransaction	N/A - EBs must not call this method 10.5.4	N/A	Return UserTX interface so bean can: ➢ Start/Stop transactions ➢ Obtain transaction status 7.5.2
isCallerInRole(String roleName)	May be different in each business method (if moved from pooled to ready) 9.6.1 Test if caller has a particular role 10.5.4 Container must have client security context 7.6.1	Test if caller has a particular role 7.5.2	
setRollbackOnly	Mark current txn for rollback 10.5.4 Should only be used if EB methods have transaction context 10.5.5	Mark current txn for rollback 7.5.2 Must having active txn 7.6.1	N/A 16.3.3.1
Transaction Attributes (Session Beans & Entity Beans only, see MDB section above for their attributes)			
	Client has transaction context	**Client has NO transaction context**	
NotSupported	Container suspends existing txn 17.6.2. Bad for EB: When ejbStore(), ejbLoad() are called is unspecified 12.1.6.1	Runs w/o starting a txn 17.6.2.1 Bad for EB: When ejbStore(), ejbLoad() are called is unspecified 12.1.6.1	
Required	Runs within Client's txn 17.6.2.2. Supports set/getRollbackOnly()	Container creates txn 17.6.2.2. Supports set/getRollbackOnly()	
Supports	Runs within Client's txn. Code should be txn-insensitive. Bad for EB because when ejbStore(), ejbLoad() happen is unspecified 12.1.6.1	Runs w/o starting a txn. Code should be txn-insensitive. Bad for EB because when ejbStore(), ejbLoad() happen is unspecified 12.1.6.1	
RequiresNew	Suspend client txn, start new txn, run, commit txn, resume client txn. 17.6.2.4 Supports set/getRollbackOnly()	Start a txn 17.6.2.4 Supports set/getRollbackOnly()	
Mandatory	Run within Client txn 17.6.2.5 Supports set/getRollbackOnly()	Throws TransactionRequired[Local]Exception 17.6.2.5	
Never	Throws RemoteException or EJBException 17.6.2.6 Bad for EB: When ejbStore(), ejbLoad() happen is unspecified 12.1.6.1	Runs w/o starting a TX 17.6.2.6 Bad for EB: When ejbStore(), ejbLoad() happen is unspecified 12.1.6.1	
javax.ejb.transaction.UserTransaction (JTA)			
begin()	N/A		17.1.1, 17.2.4, 17.6.1
commit()	N/A		17.1.1, 17.2.4, 17.6.1
getStatus()	N/A		17.3.3
rollback()	N/A		17.3.3