

Bean Provider Responsibility	Container Provider Responsibility		Client View	
<b>Application Exceptions</b>  <b>Throw ApplicationException</b>  Requirements: -Exception must be a subtype of Exception and not be a subtype of RuntimeException or RemoteException -Exception can be CreateException, RemoveException, FinderException and subclasses thereof -Possible to invoke setRollbackOnly() before throwing the application exception -throw ApplicationException before modifying DB -No EJB(Local)Object removal!!	<b>Caller tx</b> (Required, Mandatory, Supports)	<b>Throw ApplicationException</b>	<b>Receive ApplicationException</b>	if (ctx.getRollbackOnly()) //impossible to commit tx else //do more work and commit tx
	<b>BMT</b> (SB only)			if (ut.getStatus()== <small>Status.STATUS_MARKED_ROLLBACK</small> ) //impossible to commit tx else //do more work and commit tx
	<b>No tx</b> (NotSupported, Never, Supports)			Client tx (if existing) is not marked for rollback
	<b>Cont. tx</b> (Required, RequiresNew)			if (getRollbackOnly()) //rollback tx else //commit tx  <b>Throw ApplicationException</b>
<b>Other Exceptions</b>  if ( <b>RuntimeException</b> ) propagate it else wrap checked exception or any other error in <b>EJBException</b> and throw it	<b>Caller tx</b> (Required, Mandatory, Supports)	<b>-Log</b> <b>-Mark tx for RB</b> (only if tx exists) <b>-Discard instance</b> and release managed resources  <div style="text-align: center; font-size: 1.5em; font-weight: bold; color: #00008B;">MDB</div>	<b>Receive RemoteException or TR(L)E EJBException</b>	<b>Throw TransactionRolledback-(Local)Exception</b>
	<b>BMT</b> (SB, MDB)			Throw: <b>-RemoteException to remote clients</b> and  <b>-EJBException to local clients</b>
	<b>No tx</b> (NotSupported, Never, Supports)			Client tx may or may not have been marked for rollback
	<b>Cont. tx</b> (Required, RequiresNew)			

The table applies to entity and session (stateful and stateless) beans. Only the blue part applies to message-driven beans.

The exception flow goes from the left to the right. An exception is usually thrown by the underlying system or by the bean provider, and goes through the container to the client (only for entity and session beans). Exception handling is performed according to the transaction context in which the business method that throws the exception is run.

# Strategies for dealing with RemoteException or EJBException on the client's side.

