Standard configuration for JMS MDBs

Version 3
27th June 2012

This document proposes how the EJB, JCA and JMS specifications might be enhanced to provide a standard, portable way for applications to configure JMS MDBs.

Contents

1. Changes to the EJB Specification for EJB MDBs

This section proposes some changes to the section of the EJB specification that covers how JMS message-driven beans are configured. The EJB 3.1 specification currently defines four activation properties for JMS message-driven beans. This section proposes adding a further four activation properties.

The container remains free to decide whether to support its built-in JMS provider using a resource adapter or not. However it must allow the application to configure a MDB that uses the built-in JMS provider using the activation properties defined here.

Both the container and any JMS resource adapters are free to support activation properties in addition to those listed here. However applications which use non-standard activation properties may not be portable.

1.1. Activation properties currently specified in EJB 3.1

The EJB 3.1 specification currently defines the following activation properties for JMS message-driven beans.

MDB Activation property	Existing description in EJB 3.1

`acknowledgeMode`	*EJB 3.1 section 5.4.15:* JMS message-driven beans should not attempt to use the JMS API for message acknowledgment. Message acknowledgment is automatically handled by the container. If the message-driven bean uses container-managed transaction demarcation, message acknowledgment is handled automatically as a part of the transaction commit. If bean-managed transaction demarcation is used, the message receipt cannot be part of the bean-managed transaction, and, in this case, the receipt is acknowledged by the container. If bean-managed transaction demarcation is used, the Bean Provider can indicate whether JMS `AUTO_ACKNOWLEDGE` semantics or `DUPS_OK_ACKNOWLEDGE` semantics should apply by using the activationConfig element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the acknowledgment mode is `acknowledgeMode`. If the `acknowledgeMode` property is not specified, JMS `AUTO_ACKNOWLEDGE` semantics are assumed. The value of the `acknowledgeMode` property must be either `Auto-acknowledge` or `Dups-ok-acknowledge` for a JMS message-driven bean.
`messageSelector`	*EJB 3.1 section 5.4.16:* The Bean Provider may declare the JMS message selector to be used in determining which messages a JMS message-driven bean is to receive. If the Bean Provider wishes to restrict the messages that a JMS message-driven bean receives, the Bean Provider can specify the value of the message selector by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the message selector is `messageSelector`.
`destinationType`	*EJB 3.1 section 5.4.17.1:* The Bean Provider may provide advice to the Deployer as to whether a message-driven bean is intended to be associated with a queue or a topic by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the destination type associated with the bean is `destinationType`. The value for this property must be either `javax.jms.Queue` or `javax.jms.Topic` for a JMS message-driven bean.
`subscriptionDurability`	*EJB 3.1 section 5.4.17.1:* If the message-driven bean is intended to be used with a topic, the Bean Provider may further indicate

	whether a durable or non-durable subscription should be used by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify whether a durable or non-durable subscription should be used is `subscriptionDurability`. The value for this property must be either `Durable` or `NonDurable` for a JMS message-driven bean. If a topic subscription is specified and `subscriptionDurability` is not specified, a non-durable subscription is assumed.

1.2. Proposed new activation properties for EJB 3.2

It is now proposed that section 5.4 of the EJB 3.2 specification be extended to define the following additional properties for JMS message-driven beans.

The following table lists the additional properties that are proposed, and a proposed text for the EJB spec.

MDB Activation property	**Proposed description for EJB 3.2**
`destinationLookup`	*Proposed text:* The bean provider or deployer may specify the JMS queue or topic from which a JMS message-driven bean is to receive messages. The lookup name of an administratively-defined `Queue` or `Topic` object may be specified by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the lookup name is `destinationLookup`.
`connectionFactoryLookup`	*Proposed text:* The bean provider or deployer may specify the JMS connection factory that will be used to connect to the JMS provider from which a JMS message-driven bean is to receive messages. The lookup name of an administratively-defined `ConnectionFactory` object may be specified by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the lookup name is `connectionFactoryLookup`.
`clientId`	*Proposed text:* The bean provider or deployer may specify the JMS client identifier that will be used when connecting to the

	JMS provider from which a JMS message-driven bean is to receive messages. The client identifier may be specified by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the client identifier is `clientId`. If this property is not specified then the client identifier will be left unset.
`subscriptionName`	*Proposed text:* If the message-driven bean is intended to be used with a topic, and the bean provider has indicated that a durable subscription should be used by specifying the `subscriptionDurability` property to `Durable`, then the bean provider or deployer may specify the name of the durable subscription. The name of the durable subscription may be specified by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used to specify the name of the durable subscription is `subscriptionName`. If a durable subscription is specified but `subscriptionName` is not specified then the container will set the name of the durable subscription to be a name which is unique to the deployed MDB. If the message-driven bean is deployed into a clustered application server then the `shareSubscriptions` property will be used to determine whether the durable subscription name generated by the container will be the same or different for each instance in the cluster.
`shareSubscriptions`	*Proposed text:* This property is only used if the message-driven bean is deployed into a clustered application server. If message-driven bean is intended to be used with a topic and the bean provider or deployer has specified that a *durable* subscription be used but has not specified a durable subscription name then the bean provider or deployer may specify whether the durable subscription name generated by the container will be the same or different for each instance in the cluster. If message-driven bean is intended to be used with a topic and the bean provider or deployer has specified that a *non-durable*

	subscription be used then the bean provider or deployer may specify whether the same non-durable subscription should be used for each instance in the cluster. This may be specified by using the `activationConfig` element of the `MessageDriven` annotation or by using the `activation-config-property` deployment descriptor element. The property name used is `shareSubscriptions`. This property may have the string values "true" or "false". A value of `true` means that the same durable subscription name or non-durable subscription will be used for each instance in the cluster. A value of `false` means that a different durable subscription name or non-durable subscription will be used for each instance in the cluster. By default a value of `true` is assumed.

2. Changes to the JMS specification to define a JMS resource adapter

This section proposes that a new section be added to the JMS specification to specify a "standard" JMS resource adapter. A JMS provider will be required to provide a resource adapter which confirms to this specification.

This is not intended to prevent the creation of non-standard JMS resource adapters which do not conform to this specification.

The new section will be based on the existing Appendix B of JCA 1.6, which recommends a number of activation properties for a JMS resource adapter, and is intended to replace it.

In addition the new section will define some responsibilities of the container to make various pieces of information available via JNDI.

2.1. Resource Adapter provider's responsibility

The following section defines a number of `ActivationSpec` properties that a standard JMS resource adapter must support.

2.1.1. ActivationSpec properties that the resource adapter must support

The following table defines the ActivationSpec properties that a standard JMS resource adapter must support. Such a resource adapter may support additional properties to those listed.

ActivationSpec property	Description
`destinationLookup`	*This property is new in JMS 2.0* This property may be used to specify the lookup name of an administratively-defined `Queue` or `Topic` object which defines the JMS queue or topic from which the endpoint (message-driven bean) is to receive messages.
`connectionFactoryLookup`	*This property is new in JMS 2.0* This property may be used to specify the lookup name of an administratively-defined `ConnectionFactory` object that will be used to connect to the JMS provider from which the endpoint (message-driven bean) is to receive messages.
`acknowledgeMode`	*This property is already recommended in JCA 1.6.* If bean-managed transaction demarcation is used, this property may be used to indicate whether JMS `AUTO_ACKNOWLEDGE` semantics or `DUPS_OK_ACKNOWLEDGE` semantics should apply. This property may be set to either `Auto-acknowledge` or `Dups-ok-acknowledge`. If this property is not specified, a default of `Auto-acknowledge` will be used.
`messageSelector`	*This property is already recommended in JCA 1.6.* This property may be used to specify a message selector. If this property is not specified then a message selector will not be used.
`destinationType`	*This property is already recommended in JCA 1.6.* This property may be used to specify whether the specified destination is a queue or topic. The valid values are `javax.jms.Queue` or `javax.jms.Topic`.
`subscriptionDurability`	*This property is already recommended in JCA 1.6.* This property only applies to endpoints (message-driven beans) that receive messages published to a topic. It may be used to specify whether the subscription is durable or non-durable. This property may be set to either `Durable` or `NonDurable`. If this property is not specified, a default of `NonDurable` will be used.

`clientId`	*This property is already recommended in JCA 1.6.* This property may be used to specify the client identifier that will be used when connecting to the JMS provider from which the endpoint (message-driven bean) is to receive messages. If this property is not specified then the client identifier will be left unset.
`subscriptionName`	*This property is already recommended in JCA 1.6 though the requirement that the resource adapter generate a default value is new in JMS 2.0.* This property only applies to endpoints (message-driven beans) that receive messages published to a topic. If the `subscriptionDurability` property has been used to specify that the subscription is durable then the `subscriptionName` property may be used to specify the name of the durable subscription. If a durable subscription is specified but `subscriptionName` is not specified then the resource adapter will set the name of the durable subscription to be a name which is unique to the deployed MDB. If the message-driven bean is deployed into a clustered application server then the `shareSubscriptions` property will be used to determine whether the durable subscription name generated by the resource adapter will be the same or different for each instance in the cluster.

`shareSubscriptions`	*This property is new in JMS 2.0* This property only applies to message-driven beans that receive messages published to a topic and are deployed into a clustered application server. If a *durable* subscription is specified but `subscriptionName` is not specified then the `shareSubscription` property may be used to specify whether the durable subscription name generated by the resource adapter should be the same or different for each instance in the cluster. If a non-*durable* subscription is specified then the `shareSubscription` property may be used to specify whether the same non-durable subscription should be used for each instance in the cluster. This property may have the string values `true` or `false`. A value of `true` means that the same durable subscription name or non-durable subscription will be used for each instance in the cluster. A value of `false` means that a different durable subscription name or non-durable subscription will be used for each instance in the cluster. By default a value of `true` is assumed.

In addition to the `ActivationSpec` properties listed above, JCA 1.6 also recommends a property `destination`. However this is not clearly defined. The specification states that "for example [the destination] could be encapsulated as textual data… or as a private object available only at deployment time". The specification also states that the `destination` property value may also be an object that implements the `javax.jms.Destination` interface, without defining how this would be specified by the deployer.

The new property `destinationLookup` makes the `destination` property unnecessary, and so the `destination` property will not be neither recommended nor required in JMS 2.0.

2.2. Container provider's responsibility

This section describes the responsibilities of the *container* when using a resource adapter to integrate with a JMS provider.

2.2.1. Container responsibility to interpret <message-destination-link> for a MDB

EJB 3.1 section 16.9.2 states that the destination from which a message-driven bean receives messages may be specified using a `message-destination-link` deployment descriptor element. If such an element is specified then the container must determine the lookup name of the required destination and pass this information to the resource adapter using the `ActivationSpec` property `destinationLookup`.

2.2.2. Container responsibility to make MDB deployment information available via JNDI

The container is responsible for making certain MDB deployment information available in JNDI so that it can be looked up by the resource adapter when its `endpointActivation` method is called. This is defined in the following table:

JNDI lookup name	Type	Description
`java:/comp/env/uniqueMDBName`	`String`	*This is new in JMS 2.0* A name which is unique to the deployed *MDB.* The resource adapter may use this when constructing a default durable subscription name.
`java:comp/env/instanceName` or `java:global/instanceName` (to be decided)	`String`	*This property is new in JMS 2.0* If the MDB has been deployed into a clustered application server, then this is a name which identifies the application server instance within the cluster. If the application server is not clustered then this must be set to an empty String. The resource adapter may use this when constructing a default durable subscription name.

2.2.3. Container responsibility to make deployment information available via JNDI

The container is responsible for making certain deployment information available in JNDI so that it can be looked up by resource adapter objects such as managed connections when used directly by the application.

Since this information is not specific to JMS it should probably be defined in the JCA spec rather than the JMS spec.

JNDI lookup name	Type	Description
`java:/global/env/inAppClientContainer`	`String`	*This property is new in JMS 2.0* If the application running in a Java EE application client container, then the container must set this property to `true`. If the application running in a Java EE web or EJB container, then the container must set this property to `false`. The resource adapter may use this information to implement different behaviour when running in a Java EE application client container than when running in a Java EE EJB or web container.

3. Issues

- It has been suggested that there is no need to specify the activation properties that can be specified for a MDB separately from the activation properties supported by a standard resource adapter.

  However I still think that the activation properties of a MDB need to be specified separately for two reasons:
  - The container may choose to implement JMS support without using a resource adapter. It would be inappropriate to expect the vendor to consult a resource adapter specification to discover how a MDB may be configured.
  - The EJB specification contains some subtle language about some activation properties being specified by the "bean provider" and used to provide advice to the "deployer". This terminology has been maintained in the changes proposed above. However this language is not appropriate to the resource adapter implementation and so needs to be stated separately.

- The above proposals do not specify any restrictions on the valid characters and maximum length of `uniqueMDBName` and `instanceName` (which are set by the container). In addition they do not specify any restrictions on the valid characters and maximum length of a durable subscription name. It would be desirable to do so.

- Should we say that clientID should not be set when the application is deployed into a clustered container? (some providers break JMS and allow this).