



Service Component Architecture JMS Binding Specification Version 1.1

Committee Draft **022 revision 4**

19th June 16th February, 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec-cd02-rev4.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec-cd02-rev4.doc>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec-cd02-rev4.pdf>
(Authoritative)

Previous Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec-cd02.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec-cd02.doc>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec-cd02.pdf>
(Authoritative)

Latest Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec.doc>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-jms-1.1-spec.pdf> (Authoritative)

Latest Approved Version:

Technical Committee:

OASIS Service Component Architecture / Bindings (SCA-Bindings) TC

Chair(s):

Simon Holdsworth, IBM

Editor(s):

Simon Holdsworth, IBM
Khanderao Kand, Oracle
Anish Karmarkar, Oracle
Sanjay Patil, SAP
Piotr Przybylski, IBM

Related work:

This specification replaces or **supersedes**:

- Service Component Architecture JMS Binding Specification Version 1.00, March 21 2007

This specification is related to:

- Service Component Architecture Assembly Model Specification Version 1.1
- Service Component Architecture Policy Framework Specification Version 1.1

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/opencsa/sca/200903>

Deleted: 0

Deleted: 1

Deleted: 21st January

Deleted: 1

Field Code Changed

Field Code Changed

Deleted: 1

Field Code Changed

Deleted: 1

Field Code Changed

Deleted: jmsbinding

Deleted: cd01

Field Code Changed

Deleted: jmsbinding

Deleted: cd01

Field Code Changed

Deleted: jmsbinding

Deleted: cd01

Deleted: supercedes

Deleted: 200712

Deleted: 1

Deleted: 21st January

Deleted: 2008.

sca-binding-jms-1.1-spec-cd02-rev4

Copyright © OASIS® 2006, 2009, All Rights Reserved.

19th June 16th February, 2009

Page 1 of 37

Abstract:

This document defines the concept and behavior of a messaging binding, and a concrete JMS-based binding that provides that behavior.

The binding specified in this document applies to an SCA composite's services and references. The binding is especially well suited for use by services and references of composites that are directly deployed, as opposed to composites that are used as implementations of higher-level components. Services and references of deployed composites become system-level services and references, which are intended to be used by non-SCA clients.

The messaging binding describes a common pattern of behavior that may be followed by messaging-related bindings, including the JMS binding. In particular it describes the manner in which operations are selected based on message content, and the manner in which messages are mapped into the runtime representation. These are specified in a language-neutral manner.

The JMS binding provides JMS-specific details of the connection to the required JMS resources. It supports the use of Queue and Topic type destinations.

Status:

This document was last revised or approved by the OASIS Service Component Architecture / Bindings (SCA-Bindings) TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/sca-bindings/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/sca-bindings/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/sca-bindings/>.

Deleted: 1
Deleted: 21st January
Deleted: 2008.

Notices

Copyright © OASIS® 2006, ~~2009~~, All Rights Reserved.

Deleted: 2008

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", [insert specific trademarked names and abbreviations here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Deleted: 1

Deleted: 21st January

Deleted: 2008.

Table of Contents

1	Introduction	5
1.1	Terminology	5
1.2	Normative References	5
1.3	Non-Normative References	6
1.4	Naming Conventions	6
2	Messaging Bindings	7
3	JMS Binding Schema	8
4	Operation Selectors and Wire Formats	13
4.1	Default Operation Selection	13
4.2	Default Wire Format	13
4.2.1	Example of default wire format	14
5	Policy	16
6	Message Exchange Patterns	17
6.1	One-way message exchange (no Callbacks)	17
6.2	Request/response message exchange (no Callbacks)	17
6.3	JMS User Properties	18
6.4	Callbacks	18
6.4.1	Invocation of operations on a bidirectional interface	18
6.4.2	Invocation of operations on a callback interface	18
6.4.3	Use of JMSReplyTo for callbacks for non-SCA JMS applications	19
7	Examples	20
7.1	Minimal Binding Example	20
7.2	URI Binding Example	20
7.3	Binding with Existing Resources Example	20
7.4	Resource Creation Example	21
7.5	Request/Response Example	21
7.6	Use of Predefined Definitions Example	22
7.7	Subscription with Selector Example	22
7.8	Policy Set Example	22
8	Conformance	24
8.1	SCA JMS Binding XML Document	24
8.2	SCA Runtime	24
A.	JMS XML Binding Schema: sca-binding-jms.xsd	25
B.	Conformance Items	28
C.	Acknowledgements	33
D.	Revision History	36

Field Code Changed	...	[1]
Field Code Changed	...	[2]
Field Code Changed	...	[3]
Field Code Changed	...	[4]
Field Code Changed	...	[5]
Deleted: 2 . Messaging	...	[6]
Formatted	...	[7]
Field Code Changed	...	[8]
Field Code Changed	...	[9]
Deleted: 3 . JMS Bind	...	[10]
Field Code Changed	...	[11]
Field Code Changed	...	[12]
Deleted: 4 . Operation	...	[13]
Field Code Changed	...	[14]
Formatted	...	[15]
Field Code Changed	...	[16]
Deleted: 4.1 Default O	...	[17]
Field Code Changed	...	[18]
Field Code Changed	...	[19]
Deleted: 2...Wire Form	...	[20]
Field Code Changed	...	[21]
Formatted	...	[22]
Field Code Changed	...	[23]
Deleted: 5 . Policy	[24]
Field Code Changed	...	[24]
Formatted	...	[25]
Field Code Changed	...	[26]
Deleted: 6 . Message	...	[27]
Field Code Changed	...	[28]
Formatted	...	[29]
Field Code Changed	...	[30]
Deleted: 6.1 One-way	...	[31]
Field Code Changed	...	[32]
Field Code Changed	...	[33]
Deleted: 6.2	...	[34]
Field Code Changed	...	[35]
Field Code Changed	...	[36]
Deleted: 6.3 JMS User	...	[37]
Field Code Changed	...	[38]
Field Code Changed	...	[39]
Deleted: 4	...	[40]
Field Code Changed	...	[40]
Formatted	...	[41]
Field Code Changed	...	[42]
Deleted: 6.4.1 Invocati	...	[43]
Field Code Changed	...	[44]
Field Code Changed	...	[45]
Deleted: 6.4.2 Invocati	...	[46]
Field Code Changed	...	[47]
Field Code Changed	...	[48]
Deleted: 6.4.3 Use of	...	[49]
Field Code Changed	...	[50]
Formatted	...	[51]
Field Code Changed	...	[52]
Field Code Changed	...	[53]
Field Code Changed	...	[54]
Field Code Changed	...	[55]
Field Code Changed	...	[56]

1 Introduction

This document defines the concept and behavior of a messaging binding, and a concrete JMS-based [JMS] binding that provides that behavior. The binding specified in this document applies to an SCA composite's services and references. The binding is especially well suited for use by services and references of composites that are directly deployed, as opposed to composites that are used as implementations of higher-level components. Services and references of deployed composites become system-level services and references, which are intended to be used by non-SCA clients.

The messaging binding describes a common pattern of behavior that may be followed by messaging-related bindings, including the JMS binding. In particular it describes the manner in which operations are selected based on message content, and the manner in which messages are mapped into the runtime representation. These are specified in a language-neutral manner.

The JMS binding provides JMS-specific details of the connection to the required JMS resources. It supports the use of Queue and Topic type destinations.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This specification uses predefined namespace prefixes throughout; they are given in the following list. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1-1 Prefixes and Namespaces used in this specification

Prefix	Namespace	Notes
xs	"http://www.w3.org/2001/XMLSchema"	Defined by XML Schema 1.0 specification
sca	"http://docs.oasis-open.org/ns/opencsa/sca/200903"	Defined by the SCA specifications

Deleted: 200712"

1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [JMS] JMS Specification <http://java.sun.com/products/jms/>
- [WSDL] E. Christensen et al, *Web Service Description Language (WSDL) 1.1*, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, W3C Note, March 15 2001.
R. Chinnici et al, *Web Service Description Language (WSDL) Version 2.0 Part 1: Core Language*, <http://www.w3.org/TR/2007/REC-wsdl20-20070626/>, W3C Recommendation, June 26 2007.
- [JCA15] Java Connector Architecture Specification Version 1.5 <http://java.sun.com/j2ee/connector/>

Deleted: 1

Deleted: 21st January

Deleted: 2008.

- 32 [IETFJMS] IETF URI Scheme for Java™ Message Service 1.0
33 <http://www.ietf.org/internet-drafts/draft-merrick-jms-uri-05.txt>¹
34 [SCA-Assembly] <http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.html>
35 [SCA-Policy] <http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-spec.pdf>

36 1.3 Non-Normative References

37 TBD TBD

38 1.4 Naming Conventions

39 This specification follows some naming conventions for artifacts defined by the specification. In addition
40 to the conventions defined by section 1.3 of the SCA Assembly Specification [SCA-Assembly], this
41 specification adds three additional conventions:

- 42 • Where the names of elements and attributes consist partially or wholly of acronyms, the letters of the
43 acronyms use the same case. When the acronym appears at the start of the name of an element or
44 an attribute, or after a period, it is in lower case. If it appears elsewhere in the name of an element or
45 an attribute, it is in upper case. For example, an attribute might be named "uri" or "jndiURL".
- 46 • Where the names of types consist partially or wholly of acronyms, the letters of the acronyms are in
47 all upper case. For example, an XML Schema type might be named "JCABinding" or "MessageID".
- 48 • Values, including local parts of QName values, follow the rules for names of elements and attributes
49 as stated above, with the exception that the letters of acronyms are in all upper case. For example, a
50 value might be "JMSTDefault" or "namespaceURI".

¹ Note that this URI scheme is currently in draft. The reference for this specification will be updated when the IETF standard is finalized

Deleted: 1

Deleted: 21st January

Deleted: 2008.

51 2 Messaging Bindings

52 Messaging bindings form a category of SCA bindings that represent the interaction of SCA composites
53 with messaging providers. It is felt that documenting, and following this pattern is beneficial for
54 implementers of messaging bindings, although it is not strictly necessary.

55 This pattern is embodied in the JMS binding, described later.

56 Messaging bindings utilize operation selector and wire format elements to provide the mapping from the
57 native messaging format to an invocation on the target component. A default operation selection and
58 data binding behavior is identified, along with any associated properties.

59 In addition, each operation may have specific properties defined, that may influence the way native
60 messages are processed depending on the operation being invoked.

Comment [SAJH1]: I don't think this section really says anything that is not already said elsewhere and should be deleted

Deleted: 1

Deleted: 21st January

Deleted: 2008.

3 JMS Binding Schema

The JMS binding element is defined by the following schema.

```
63 <binding.jms correlationScheme="QName"?
64     initialContextFactory="xs:anyURI"?
65     jndiURL="xs:anyURI"?
66     requestConnection="QName"?
67     responseConnection="QName"?
68     operationProperties="QName"?
69     name="NCName"?
70     requires="list of QName"?
71     policySets="list of QName"?
72     uri="xs:anyURI"?
73     ... >
74 <destination jndiName="xs:anyURI" type="queue or topic"?
75     create="always or never or ifNotExist"?>
76 <property name="NMTOKEN" type="NMTOKEN"?>*
77 </destination?>
78 <connectionFactory jndiName="xs:anyURI"
79     create="always or never or ifNotExist"?>
80 <property name="NMTOKEN" type="NMTOKEN"?>*
81 </connectionFactory?>
82 <activationSpec jndiName="xs:anyURI"
83     create="always or never or ifNotExist"?>
84 <property name="NMTOKEN" type="NMTOKEN"?>*
85 </activationSpec?>
86
87 <response>
88 <destination jndiName="xs:anyURI" type="queue or topic"?
89     create="always or never or ifNotExist"?>
90 <property name="NMTOKEN" type="NMTOKEN"?>*
91 </destination?>
92 <connectionFactory jndiName="xs:anyURI"
93     create="always or never or ifNotExist"?>
94 <property name="NMTOKEN" type="NMTOKEN"?>*
95 </connectionFactory?>
96 <activationSpec jndiName="xs:anyURI"
97     create="always or never or ifNotExist"?>
98 <property name="NMTOKEN" type="NMTOKEN"?>*
99 </activationSpec?>
100 <wireFormat/>?
101 </response?>
102
103 <resourceAdapter name="NMTOKEN"?>
104 <property name="NMTOKEN" type="NMTOKEN"?>*
105 </resourceAdapter?>
106
107 <headers type="string"?
108     deliveryMode="persistent or nonpersistent"?
109     timeToLive="long"?
110     priority="0 .. 9"?>
111 <property name="NMTOKEN" type="NMTOKEN"?>*
112 </headers?>
113
114 <messageSelection selector="string"?>
115 <property name="NMTOKEN" type="NMTOKEN"?>*
116 </headers?>
117
118 <operationProperties name="string" nativeOperation="string"?>
119 <property name="NMTOKEN" type="NMTOKEN"?>*
```

Deleted: ifNotExist

Deleted: ifNotExist

Deleted: ifNotExist

Deleted: ifNotExist

Deleted: ifNotExist

Deleted: ifNotExist

Deleted: JMSType

Deleted:
JMSDeliveryMode="PERSISTENT or NON_PERSISTENT"?
JMSTimeToLive="long"?
JMSPriority="0 .. 9"

Deleted: subscriptionHeaders JMSSelector

Deleted: 1

Deleted: 21st January

Deleted: 2008.

120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

```
<headers type="string"?
  deliveryMode="persistent or nonpersistent"?
  timeToLive="long"?
  priority="0 .. 9"?>
  <property name="NMTOKEN" type="NMTOKEN"?>*
</headers?>
</operationProperties>*

<wireFormat/>?
<operationSelector/>?
</binding.jms>
```

The binding can be used in one of two ways, either identifying existing JMS resources using JNDI names, or providing the required information to enable the JMS resources to be created.

The **binding.jms** element has the following attributes:

- **/binding.jms** – This is the JMS binding element. The element is extensible so that JMS binding implementers can add additional JMS provider-specific attributes and elements although such extensions are not guaranteed to be portable across runtimes.
- **/binding.jms/@uri** – as defined in the SCA Assembly Specification [SCA-Assembly]. This attribute identifies the destination, connection factory or activation spec, and other properties to be used to send/receive the JMS message. There is an implicit @create="never" for the resources referred to in the @uri attribute.
 - The value of the @uri attribute MUST have the format defined by the IETF URI Scheme for Java™ Message Service 1.0 [IETFJMS] [BJM30001].
 - The following illustrates the structure of the URI and the set of property names that have specific semantics - all other property names are treated as user property names:
 - **jms:indi:<jms-dest?>**
indiURL=<indi-url> &
indiInitialContextFactory=<indi-initial-context-factory> &
indiConnectionFactoryName=<Connection-Factory-Name> &
deliveryMode=<Delivery-Mode> &
timeToLive=<Time-To-Live> &
priority=<Priority> &
<param-name>=<param-value> & ...
- When the @uri attribute is specified, the SCA runtime MUST raise an error if the referenced resources do not already exist [BJM30002].
- When the @uri attribute is specified, the destination element MUST NOT be present [BJM30034].
- An SCA runtime MUST use the values specified in the @uri attribute in preference to corresponding attributes and elements in the binding [BJM30035].
- **/binding.jms/@name** - as defined in the SCA Assembly Specification [SCA-Assembly].
- **/binding.jms/@requires** - as defined in the SCA Assembly Specification [SCA-Assembly].
- **/binding.jms/@policySets** - as defined in the SCA Assembly Specification [SCA-Assembly].
- **/binding.jms/@correlationScheme** – identifies the correlation scheme used when sending reply or callback messages, default value is "sca:messageID".
 - If the value of the @correlationScheme attribute is "sca:messageID" the SCA runtime MUST set the correlation ID of replies to the message ID of the corresponding request [BJM30003].
 - If the value of the @correlationScheme attribute is "sca:correlationID" the SCA runtime MUST set the correlation ID of replies to the correlation ID of the corresponding request [BJM30004].
 - If the value of the @correlationScheme attribute is "sca:none" the SCA runtime MUST NOT set the correlation ID [BJM30005].
 - SCA runtimes MAY allow other values of the @correlationScheme attribute to indicate other correlation schemes [BJM30006].
- **/binding.jms/@initialContextFactory** – the name of the JNDI initial context factory.

Deleted: JMSType

Deleted: JMSDeliveryMode="PERSISTENT or NON_PERSISTENT"?¶
JMSTimeToLive="long"?¶
JMSPriority="0 ..

Deleted: generic

Deleted: type.

Deleted: type

Deleted: (from binding)

Deleted: URI that

Deleted: .
The value of the @uri attribute MUST have the following format, defined by the IETF URI Scheme

Deleted: Java™ Message Service 1.0

Field Code Changed

Deleted: .

Deleted: connectionFactory Name

Deleted: destinationType={queue/topic} .

Deleted: selector=<Selector> & .
<User-Property>=<User-Property-Value

Deleted: When the @uri attribute is specified, the SCA runtime MUST raise an error if the referenced resources do not already exist.¶

Deleted: specification

Deleted: in Section 9, "Binding"

Deleted: specification

Deleted: in Section 9, "Binding"

Deleted: . Possible values for the @correlationScheme attribute are "sca:MessageID" (the default) where the SCA runtime MUST set the correlation ID of replies to the message ID of the corresponding request; "sca:CorrelationID" where the SCA runtime MUST set the correlation ID of replies (... [63])

Deleted: 1

Deleted: 21st January

Deleted: 2008.

- 173 • **/binding.jms/@jndiURL** – the URL for the JNDI provider.
- 174 • **/binding.jms/@requestConnection** – identifies a **binding.jms** element that is present in a definition
 175 document, whose **destination**, **connectionFactory**, **activationSpec** and **resourceAdapter** children
 176 are used to define the values for this binding.
 177 **If the @requestConnection attribute is specified, the binding.jms element MUST NOT contain a**
 178 **destination, connectionFactory, activationSpec or resourceAdapter element [BJM30007].**
- 179 • **/binding.jms/@responseConnection** – identifies a **binding.jms** element that is present in a
 180 definition document, whose **response** child element is used to define the values for this binding.
 181 **If the @responseConnection attribute is specified, the binding.jms element MUST NOT contain a**
 182 **response element [BJM30008].**
- 183 • **/binding.jms/@operationProperties** – identifies a **binding.jms** element that is present in a definition
 184 document, whose **operationProperties** children are used to define the values for this binding.
 185 **If the @operationProperties attribute is specified, the binding.jms element MUST NOT contain an**
 186 **operationProperties element [BJM30009].**
- 187 • **/binding.jms/destination** – identifies the destination that is to be used to process requests by this
 188 binding.
- 189 • **/binding.jms/destination/@type** - the type of the request destination. Valid values are “**queue**” and
 190 “**topic**”. The default value is “**queue**”.
 191 **Whatever the value of the destination/@type attribute, the runtime MUST ensure a single response**
 192 **is delivered for request/response operations [BJM30010].**
- 193 • **binding.jms/destination/@jndiName** – the JNDI name of the JMS Destination that the binding uses
 194 to send or receive messages. The behaviour of this attribute is determined by the value of the
 195 **@create** attribute as follows:
- 196 – **If the @create attribute value for a destination, connectionFactory or activationSpec element is**
 197 **"always" then the @jndiName attribute is optional; if the resource cannot be created at the**
 198 **specified location then the SCA runtime MUST raise an error [BJM30011].**
 199 **If the @jndiName attribute is omitted this specification places no restriction on the JNDI location**
 200 **of the created resource.**
- 201 – **If the @create attribute value for a destination, connectionFactory or activationSpec element is**
 202 **"ifNotExist" then the @jndiName attribute MUST specify the location of the possibly existing**
 203 **resource [BJM30012].**
 204 **If the destination, connectionFactory or activationSpec does not exist at the location identified by**
 205 **the @jndiName attribute, but cannot be created there then the SCA runtime MUST raise an error**
 206 **[BJM30013].**
 207 **If the destination, connectionFactory or activationSpec's @jndiName attribute refers to an**
 208 **existing resource that is not a JMS Destination of the appropriate type, a JMS connection factory**
 209 **or a JMS activation spec respectively then the SCA runtime MUST raise an error [BJM30014].**
- 210 – **If the @create attribute value for a destination, connectionFactory or activationSpec element is**
 211 **"never" then the @jndiName attribute MUST specify the location of the existing resource**
 212 **[BJM30015].**
 213 **If the destination, connection factory or activation spec is not present at the location identified by**
 214 **the @jndiName attribute, or the location refers to a resource of an incorrect type then the SCA**
 215 **runtime MUST raise an error [BJM30016].**
- 216 • **/binding.jms/destination/@create** – indicates whether the destination should be created when the
 217 containing composite is deployed. Valid values are “**always**”, “**never**” and “**ifNotExist**”. The default
 218 value is “**ifNotExist**”.
- 219 • **/binding.jms/destination/property** – defines properties to be used to create the destination, if
 220 required.
- 221 • **/binding.jms/connectionFactory** – identifies the connection factory that the binding uses to process
 222 request messages. The attributes of this element follow [the rules](#), defined for the **destination**
 223 element.
 224 **A binding.jms element MUST NOT include both a connectionFactory element and an**

Deleted: In this case this **binding.jms** element MUST NOT also contain the corresponding elements

Deleted: In this case this **binding.jms** element MUST NOT contain a **response** element

Deleted: In this case this **binding.jms** element MUST NOT contain an **operationProperties** element

Deleted: In either case the runtime MUST ensure a single response is delivered for request/response operations

Deleted: If the **@create** attribute value is “**always**” then the **@jndiName** attribute is optional; if the destination cannot be created at the specified location then the SCA runtime MUST raise an error.

Formatted: Attribute, English (U.S.)

Deleted: <#>If the **@create** attribute value is “**ifNotExist**” then the **@jndiName** attribute MUST specify the location of the possibly existing destination; if the destination does not exist at this location, but cannot be created there then the SCA runtime MUST raise an error. If the **@jndiName** refers to an existing resource other than a JMS Destination of the specified type then the SCA runtime MUST raise an error. ¶ <#>If the **@create** attribute value is “**never**” then the **@jndiName** attribute MUST specify the location of the existing destination; if the destination is not present at the location, or the location refers to a resource other than a JMS Destination of the specified type then the SCA runtime MUST raise an error.¶

Formatted: Attribute, English (U.S.)

Formatted: Attribute, English (U.S.)

Deleted: **ifNotExist**.

Deleted: **ifNotExist**..

Deleted: those

Deleted: 1

Deleted: 21st January

Deleted: 2008.

- 225 **activationSpec** element [BJM30017].
- 226 When the **connectionFactory** element is present, then the destination MUST be defined either by
- 227 the **destination** element or the **@uri** attribute [BJM30018].
- 228 • **/binding.jms/activationSpec** – identifies the activation spec that the binding uses to connect to a
 - 229 JMS destination to process request messages. The attributes of this element follow **the rules** defined
 - 230 for the **destination** element.
 - 231 If the **activationSpec** element is present and the destination is also specified via a **destination**
 - 232 element or the **@uri** attribute then it MUST refer to the same JMS destination as the **activationSpec**
 - 233 [BJM30019].
 - 234 The **activationSpec** element MUST NOT be present when the binding is being used for an SCA
 - 235 reference [BJM30020].
 - 236 • **/binding.jms/response** – defines the resources used for handling response messages (receiving
 - 237 responses for a reference, and sending responses from a service).
 - 238 • **/binding.jms/response/destination** – identifies the destination that is to be used to process
 - 239 responses by this binding. Attributes **follow the rules defined** for the parent's **destination** element.
 - 240 For a service, this destination is used to send responses to messages that have a null value for the
 - 241 **JMSReplyTo** destination. For a reference, this destination is used to receive reply messages
 - 242 • **/binding.jms/response/connectionFactory** – identifies the connection factory that the binding uses
 - 243 to process response messages. The attributes of this element follow those defined for the
 - 244 **destination** element.
 - 245 A **response** element MUST NOT include both a **connectionFactory** element and an **activationSpec**
 - 246 element [BJM30021].
 - 247 • **/binding.jms/response/activationSpec** – identifies the activation spec that the binding uses to
 - 248 connect to a JMS destination to process response messages. The attributes of this element follow
 - 249 those defined for the **destination** element.
 - 250 If a **response/destination** and **response/activationSpec** element are both specified they MUST
 - 251 refer to the same JMS destination [BJM30022].
 - 252 The **response/activationSpec** element MUST NOT be present when the binding is being used for an
 - 253 SCA service [BJM30023].
 - 254 • **/binding.jms/response/wireFormat** – identifies the wire format used by responses sent or received
 - 255 by this binding. This value overrides the **wireFormat** specified at the binding level.
 - 256 • **/binding.jms/headers** – this element specifies values for standard JMS headers.
 - 257 The SCA runtime MUST set JMS headers in messages that it creates to the values specified by the
 - 258 **headers** element unless overridden for the operation being invoked. [BJM30024].
 - 259 These values apply to requests from a reference and responses from a service.
 - 260 • **/binding.jms/headers/@type, @deliveryMode, @timeToLive, @priority** – specifies the value to
 - 261 use for the JMS header property **JMSType, JMSDeliveryMode, JMSTimeToLive or JMSPriority**
 - 262 **respectively**.
 - 263 If the **@uri** attribute includes values for the type, delivery mode, time to live or priority properties then
 - 264 the **@uri** values are used and the **headers** and **operationProperties/headers @type,**
 - 265 **@deliveryMode, @timeToLive or @priority** attributes are ignored [BJM30025].
 - 266 Valid values for **@deliveryMode** are **"persistent"** and **"nonpersistent"**; valid values for **@priority,**
 - 267 are **"0"** to **"9"**.
 - 268 • **/binding.jms/headers/property** – specifies the value for the given JMS user property.
 - 269 For each **header/properties** element the SCA runtime MUST set the named JMS user property to
 - 270 the given value in messages it creates unless overridden for the operation being invoked
 - 271 [BJM30026].
 - 272 • **/binding.jms/messageSelection** - this element allows JMS **message selection** options to be set.
 - 273 These values apply to a service **receiving messages from the request** destination or for a reference
 - 274 **receiving messages from the** callback or reply-to destination.
 - 275 • **/binding.jms/messageSelection/@selector** - specifies the value to use for the JMS selector. **If the**
 - 276 **@uri** attribute includes a value for the message selector then the **@uri** value is used and the
 - 277 **messageSelection/@selector** attribute is ignored [BJM30027].

Deleted: A **binding.jms** element MUST NOT include both this element and an **activationSpec** element. When this element is present, the **destination** element MUST also be present

Deleted: those

Deleted: If a **destination** element is also specified it MUST refer to the same JMS destination as the **activationSpec**. This element MUST NOT be present when the binding is being used for an SCA reference

Deleted: are as

Deleted: A **response** element MUST NOT include both this element and an **activationSpec** element

Deleted: If a response **destination** element is also specified it MUST refer to the same JMS destination as the **activationSpec**. This element MUST NOT be present when the binding is being used for an SCA service

Deleted: that the SCA runtime MUST set to the given $\sqrt{\dots}$ [64]

Deleted: **JMSType, @JMSDeliveryMode,** ... [65]

Deleted: . The value of the **@uri** attribute MUST NOT ... [66]

Deleted: **JMSDeliveryMode** are **"PERSISTENT"** and ... [67]

Deleted: **JMSPriority**

Deleted: that the SCA runtime MUST set

Deleted: specified

Deleted: when creating messages..

Deleted: **subscriptionHeader s**

Deleted: subscription

Deleted: subscribing to the

Deleted: subscribing to the

Deleted: s

Deleted: **subscriptionHeader s/@JMSSelector**

Deleted: The value of the **@uri** attribute MUST NOT ... [68]

Deleted: 1

Deleted: 21st January

Deleted: 2008.

278 • */binding.jms/resourceAdapter* – specifies name, type and properties of the Resource Adapter Java
279 bean.

280 The *resourceAdapter* element MUST be present when JMS resources are to be created for a JMS
281 provider that implements the JCA 1.5 specification [JCA15], and is ignored otherwise [BJM30031].
282 SCA runtimes MAY place restrictions on the properties of the resource adapter Java bean that can be
283 set using the *resourceAdapter* element [BJM30028].

284 ¶ For JMS providers that do not implement the JCA 1.5 specification JCA15, information necessary for
285 resource creation can be added in provider-specific elements or attributes allowed by the extensibility
286 of the *binding.jms* element.

287 • */binding.jms/operationProperties* – specifies various properties that are specific to the processing
288 of a particular operation.

289 • */binding.jms/operationProperties/@name* – The name of the operation in the interface.

290 • */binding.jms/operationProperties/@selectedOperation* – The value generated by the
291 *operationSelector* that corresponds to the operation in the service or reference interface identified
292 by the *operationProperties/@name* attribute. If this attribute is omitted then the value defaults to
293 the value of the *operationProperties/@name* attribute.

294 The value of the *operationProperties/@selectedOperation* attribute MUST be unique across the
295 containing *binding.jms* element [BJM30029].

296 • */binding.jms/operationProperties/property* – specifies properties specific to this operation. These
297 properties are intended to be used to parameterize the *wireFormat* identified for the binding for a
298 particular operation.

299 The SCA runtime SHOULD make the *operationProperties* element corresponding to the
300 *selectedOperation* available to the *wireFormat* implementation [BJM30030].

301 • */binding.jms/operationProperties/headers* – this element specifies values for standard JMS
302 headers. These values apply to requests from a reference and responses from a service.

303 The SCA runtime MUST set JMS headers in messages it creates when the operation identified by the
304 *operationProperties/@name* attribute is invoked to the values specified by the corresponding
305 *operationProperties/headers* element [BJM30032].

306 • */binding.jms/operationProperties/headers/@type, @deliveryMode, @timeToLive, @priority* –
307 specifies the value to use for the JMS header property *JMSType, JMSDeliveryMode, JMSTimeToLive*
308 or *JMSPriority, respectively*.

309 • */binding.jms/operationProperties/headers/property* – specifies the value for the given JMS user
310 property.

311 For each *operationProperties/headers/property* element the SCA runtime MUST set the named
312 JMS user property to the given value in messages it creates when the operation identified by the
313 *operationProperties/@name* attribute is invoked [BJM30033].

314 • */binding.jms/wireFormat* – identifies the wire format used by requests and responses sent or
315 received by this binding.

316 • */binding.jms/operationSelector* – identifies the operation selector used when receiving requests for
317 a service. If specified for a reference this provides the default operation selector for callbacks if not
318 specified via a callback service element.

319 • */binding.jms/@{any}* - this is an extensibility mechanism to allow extensibility via attributes.

320 • */binding.jms/any* – this is an extensibility mechanism to allow extensibility via elements.

321 The *binding.jms* element MUST conform to the XML schema defined in *sca-binding-jms.xsd*
322 [BJM30036].

323 Deployers/assemblers can configure *nonpersistent* for *@deliveryMode* in order to provide higher
324 performance with a decreased quality of service. A *binding.jms* element configured in this way cannot
325 satisfy either of the "atLeastOnce" and "exactlyOnce" policy intents. The SCA Runtime MUST raise an
326 error for this invalid combination at deployment time.

Deleted: This element MUST be present when the JMS resources are to be created for a JMS provider that implements the JCA 1.5 specification

Deleted: JCA15

Deleted: , and is ignored otherwise. SCA runtimes MAY place restrictions on the properties of the RA Java bean that can be set.

Deleted: The value of this attribute MUST be unique across the containing *binding.jms* element.

Deleted: The SCA runtime SHOULD make the *operationProperties* element corresponding to the *selectedOperation* available to the *wireFormat* implementation

Deleted: that the SCA runtime MUST set to the given values for the given operation.

Deleted: *JMSType, @JMSDeliveryMode, @JMSTimeToLive, @JMSPriority*

Deleted: . The SCA runtime MUST use values specified for particular operations in preference to those defined for all operations in the *binding.jms/headers* element or via the binding's *@uri* attribute.

Deleted: that

Deleted: SCA runtime MUST set for the specified

Deleted: when creating messages

Deleted: *NON_PERSISTENT*

Deleted: *JMSDeliveryMode*

Comment [SAJH2]: Pending move of this text for resolution to issue BINDINGS-48

Deleted: 1

Deleted: 21st January

Deleted: 2008.

327

4 Operation Selectors and Wire Formats

328 In general messaging providers deal with message formats and destinations. There is not usually a built-
329 in concept of "operation" that corresponds to that defined in a WSDL portType [WSDL]. Messages have
330 a wire format which corresponds in some way to the schema of an input or output message of an
331 operation in the interface of a service or reference, however additional information is required in order for
332 an SCA runtime to know how to identify the operation and understand the wire format of messages.

333 The process of identifying the operation to be invoked is *operation selection*; the information that
334 describes the contents of messages is a *wire format*. The **binding** element as described in the SCA
335 Assembly **Specification**, [SCA-Assembly] provides the means to identify specific operation selection via
336 the **operationSelector** element and the wire format of messages received and to be sent using the
337 **wireFormat** element. When the JMS binding receives a message, the **operationSelector** is used to
338 generate a selected operation name from the message content. The selected operation name is then
339 mapped to an operation in the service's interface via a matching **operationProperties** element in the
340 JMS binding. If there is no matching element, the operation name is assumed to be the same as the
341 selected operation name.

342 No standard means is provided for linking the **wireFormat** or **operationSelector** elements with the
343 runtime components that implement their **behavior**.

344 The following sections describe the default **operationSelector** and **wireFormat** for a JMS binding.

345 The SCA runtime MUST support the default JMS wire format and operation selector behavior, and MAY
346 provide additional means to override it [BJM40001].

4.1 Default Operation Selection

348 The following defines the default operation selection algorithm when receiving a request at a service,
349 or a callback at a reference. When using the default operation selection algorithm, the selected operation
350 name is determined as follows:

- 351 • If there is only one operation on the service's interface, then that operation is the selected operation
352 name.
- 353 • Otherwise, if the JMS user property "**scaOperationName**" is present, then the value of that user
354 property is used as the selected operation name.
- 355 • Otherwise, if the message is a JMS text or bytes message containing XML, then the selected
356 operation name is the local name of the root element of the XML payload.
- 357 • Otherwise, the selected operation name is "**onMessage**".

358 When a **binding.jms** element specifies the **operationSelector.jmsDefault** element, the SCA runtime
359 MUST use the default operation selection algorithm to determine the selected operation [BJM40008].

360 If no **operationSelector** element is specified then SCA runtimes MUST use
361 **operationSelector.jmsDefault** as the default [BJM40002].

4.2 Default Wire Format

363 The default wire format maps between a **JMSMessage** and the object(s) expected by the component
364 implementation. We encourage component implementers to avoid exposure of JMS APIs to component
365 implementations, however in the case of an existing implementation that expects a **JMSMessage**, this
366 provides for simple reuse of that as an SCA component.

367 When using the default wire format, the message body is mapped to the parameters or return value of the
368 target operation as follows:

- 369 • If there is a single parameter that is a **JMSMessage**, then the **JMSMessage** is passed as is.
- 370 • Otherwise, if the **JMSMessage** is not a JMS text message or bytes message containing XML, it is
371 invalid.

Deleted: specification

Deleted: behaviour

Deleted: This section describes

Deleted: The SCA runtime MUST support this default behavior, and MAY provide additional means to override it.

Deleted: When

Deleted: assumed as

Deleted: .

Deleted: its

Deleted: .

Deleted: taken from

Deleted: .

Deleted: assumed to be

Deleted: The selected operation name is then mapped to an operation in the service's interface via a matching **operationProperties** element in the JMS binding. If there is no matching element, the operation name is assumed to be the same as the selected operation name.¶ The use of this operation selector can be explicitly specified in a **binding.jms** using the **operationSelector.jmsdefault** element; if no **operationSelector** element is specified then SCA runtimes MUST use this as the default.¶

Deleted: The

Deleted: must be

Deleted: ; an SCA runtime MUST be able to receive both forms. When sending messages either form may be used; an SCA runtime MAY provide additional configuration to allow one or other to be selected

Deleted: 1

Deleted: 21st January

Deleted: 2008.

372 • Otherwise if, there is a single parameter, or for the return value, the JMS text or bytes XML payload is
373 the XML serialization of that parameter according to the WSDL schema for the message.

Deleted: If

374 • Otherwise the, multiple parameters, are encoded in XML using the document wrapped style, according
375 to the WSDL schema for the message.

Deleted: If there are

Deleted: , then they

376 When a *binding.jms* element specifies the *wireFormat.jmsDefault* element, the SCA runtime MUST use
377 the default wire format [BJM40009].

378 When using the default wire format to send request messages, if there is a single parameter and the
379 interface includes more than one operation, the SCA runtime MUST set the JMS user property
380 "*scaOperationName*" to the name of the operation being invoked [BJM40003].

381 When using the default wire format an SCA runtime MUST be able to receive both JMS text and bytes
382 messages [BJM40005].

383 When using the default wire format an SCA runtime MUST send either a JMS text or a JMS bytes
384 message [BJM40006].

385 When using the default wire format an SCA runtime MAY provide additional configuration to allow
386 selection between JMS text or bytes messages to be sent [BJM40007].

387 If no *wireFormat* element is specified in a JMS binding then SCA runtimes MUST use
388 *wireFormat.jmsDefault* as the default [BJM40004].

389 **4.2.1 Example of default wire format**

390 For the following interface definition:

```
391 <wsdl:definitions name="Coordinates"  
392 targetNamespace="http://tempuri.org/coordinates"  
393 xmlns:tns="http://tempuri.org/coordinates"  
394 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
395 xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
396 <wsdl:types>  
397 <xsd:schema targetNamespace="http://tempuri.org/coordinates">  
398 <xsd:element name="setCoordinates">  
399 <xsd:complexType>  
400 <xsd:sequence>  
401 <xsd:element name="x" type="xsd:int"/>  
402 <xsd:element name="y" type="xsd:int"/>  
403 </xsd:sequence>  
404 </xsd:complexType>  
405 </xsd:element>  
406 </xsd:schema>  
407 </wsdl:types>  
408  
409 <wsdl:message name="setCoordinatesRequestMsg">  
410 <wsdl:part element="tns:setCoordinates" name="setCoordinatesParameters"/>  
411 </wsdl:message>  
412  
413 <wsdl:portType name="Coordinates">  
414 <wsdl:operation name="setCoordinates">  
415 <wsdl:input message="tns:setCoordinatesRequestMsg"  
416 name="setCoordinatesRequest"/>  
417 </wsdl:operation>  
418 </wsdl:portType>  
419 </wsdl:definitions>
```

Deleted: <#>When sending request messages, if there is a single parameter and the interface includes more than one operation, the SCA runtime MUST set the JMS user property "*scaOperationName*" to the name of the operation being invoked.¶ The use of this wire format can be explicitly specified in a *binding.jms* using the *wireFormat.jmsdefault* element; if no *wireFormat* element is specified then SCA runtimes MUST use this as the default. ¶

Deleted: example, for

420 When the *setCoordinates* operation is invoked via a reference with a JMS binding that uses the default
421 wire format, the message sent from the JMS binding is a JMS text or bytes message with the following
422 content:

Deleted: ¶

```
423 <setCoordinates xmlns="http://tempuri.org/coordinates">  
424 <x>10</x>  
425 <y>5</y>
```

Deleted: 1

Deleted: 21st January

Deleted: 2008.

</setCoordinates>

Deleted: 1
Deleted: 21st January
Deleted: 2008.

427 **5 Policy**

428 The JMS binding provides attributes that control the sending of messages, requests from references and
429 replies from services. These values can be set directly on the binding element for a particular service or
430 reference, or they can be set using policy intents. An example of setting these via intents is shown later.

431 JMS binding implementations MAY support the following standard intents, as defined by the JMS
432 binding's **bindingType**:

```
<bindingType type="binding.jms"  
  alwaysProvides="JMS",  
  mayProvide="atLeastOnce atMostOnce ordered"/>
```

436 The atLeastOnce, atMostOnce and ordered intent are defined in the SCA Policy Specification document
437 in section 8, "Reliability Policy".

Comment [SAJH3]: Pending update to this text for resolution to issue BINDINGS-48

Deleted: `jms`

Deleted: `conversational`

Deleted: The conversational intent is defined in the SCA Assembly Specification document in section 8.3, "Conversational Interfaces".

Deleted: 1

Deleted: 21st January

Deleted: 2008.

438 6 Message Exchange Patterns

439 This section describes the message exchange patterns that are possible when using the JMS binding,
440 including one-way, request/response and callbacks. JMS has a looser concept of message exchange
441 patterns than WSDL, so this section explains how JMS messages that are sent and received by the SCA
442 runtime relate to the WSDL input/output messages. Each operation in a WSDL interface is either one-
443 way or request/response. Callback interfaces may include both one-way and request/response
444 operations.

Deleted: ,
Deleted: and conversations

445 6.1 One-way message exchange (no Callbacks)

446 A one-way message exchange is one where a request message is sent that does not require or expect a
447 corresponding response message. These are represented in WSDL as an operation with an **input**
448 element and no **output** elements and no **fault** elements.

449 For an SCA reference with a JMS binding, when a request message is sent as part of a one-way MEP,
450 the SCA runtime SHOULD NOT set the **JMSReplyTo** destination header in the JMS message that it
451 creates, regardless of whether the JMS binding has a **response** element with a **destination** defined
452 [BJM60001].

453 For an SCA service with a JMS binding, when a request message is received as part of a one-way MEP,
454 the SCA runtime MUST ignore the **JMSReplyTo** destination header in the JMS message, and not raise
455 an error [BJM60002].

456 The use of one-way exchanges when using a bidirectional interface is described in section 6.4.

Deleted: When a request message is sent by a reference with a JMS binding for a one-way MEP, the SCA runtime SHOULD NOT set the **JMSReplyTo** destination header in the JMS message that it creates, regardless of whether the JMS binding has a **response** element with a **destination** defined. ¶
When a request message is received by a service with a JMS binding for a one-way MEP, the SCA runtime MUST ignore the **JMSReplyTo** destination header in the JMS message, and MUST NOT raise an error. ¶

Deleted: 7.4

457 6.2 Request/response message exchange (no Callbacks)

458 A request/response message exchange is one where a request message is sent and a response
459 message is expected, possibly identified by its correlation identifier. These are represented in WSDL as
460 an operation with an **input** element and an **output** and/or a **fault** element.

461 For an SCA reference with a JMS binding, when a request message is sent as part of a request/response
462 MEP, the SCA runtime MUST set a non-null value for the **JMSReplyTo** header in the JMS message it
463 creates for the request [BJM60003].

464 For an SCA reference with a JMS binding, when a request message is sent as part of a request/response
465 MEP, and the JMS binding has a **response** element with a **destination** defined, then the SCA runtime
466 MUST use that destination for the **JMSReplyTo** header in the JMS message it creates for the request
467 [BJM60004].

468 For an SCA reference with a JMS binding, when a request message is sent as part of a request/response
469 MEP, and the JMS binding does not have a **response** element with a **destination** defined, the SCA
470 runtime MUST provide an appropriate destination on which to receive response messages and use that
471 destination for the **JMSReplyTo** header in the JMS message it creates for the request [BJM60005].

472 For an SCA reference with a JMS binding, the SCA runtime MAY choose to receive response messages
473 on the basis of their correlation ID as defined by the binding's **@correlationScheme** attribute, or use a
474 unique destination for each response [BJM60006].

475 For an SCA service with a JMS binding, when a response message is sent as part of a request/response
476 MEP where the request message included a non-null **JMSReplyTo** destination, the SCA runtime MUST
477 send the response message to that destination [BJM60007].

478 For an SCA service with a JMS binding, when a response message is sent as part of a request/response
479 MEP where the request message included a null **JMSReplyTo** destination and the JMS binding includes
480 a **response/destination** element the SCA runtime MUST send the response message to that destination
481 [BJM60008].

Deleted: 1
Deleted: 21st January
Deleted: 2008.

482 For an SCA service with a JMS binding, when a response message is sent as part of a request/response
483 MEP where the request message included a null **JMSReplyTo** destination and the JMS binding does not
484 include a **response/destination** then an error SHOULD be raised by the SCA runtime [BJM60009].

485 For an SCA service with a JMS binding, when a response message is sent as part of a request/response
486 MEP the SCA runtime MUST set the correlation identifier in the JMS message that it creates for the
487 response as defined by the JMS binding's **@correlationScheme** attribute [BJM60010].

488 The use of request/response exchanges when using a bidirectional interface is described in section 6.4.

489 6.3 JMS User Properties

490 This protocol assigns specific behavior to JMS user properties:

- 491 • "**scaCallbackDestination**" holds the name of the JMS Destination to which callback messages are
492 sent.

493 6.4 Callbacks

494 Callbacks are SCA's way of representing bidirectional interfaces, where messages are sent in both
495 directions between a client and a service. A callback is the invocation of an operation on a service's
496 callback interface. A callback operation can be one-way or request/response. Messages that correspond
497 to one-way or request/response operations on a bidirectional interface use either the
498 **scaCallbackDestination** user property or the **JMSReplyTo** destination, or both, to identify the
499 destination to which messages are to be sent when operations are invoked on the callback interface. The
500 use of **JMSReplyTo** for this purpose is to enable interaction with non-SCA JMS applications, as
501 described below.

502 SCA runtimes MUST follow the behavior described in section 6.4 and its subsections when **binding.jms**
503 is used in both the forward and callback directions [BJM60018].

504 SCA runtimes can use different bindings for forward calls and callbacks, however the behavior and
505 requirements on messages is vendor-specific.

506 6.4.1 Invocation of operations on a bidirectional interface

507 **Error! Reference source not found.** [BJM60011].

508 For an SCA reference with a JMS binding and bidirectional interface, when a request message is sent the
509 SCA runtime MAY set the **JMSReplyTo** destination to the same value as the **scaCallbackDestination**
510 user property [BJM60012].

511 For an SCA reference with a JMS binding and bidirectional interface, when a request message is sent as
512 part of a request/response MEP, the SCA runtime MUST set the **JMSReplyTo** header in the message it
513 creates as described in section 6.2 [BJM60013].

514 For both one-way and request/response operations, the reference's callback service can be used to
515 identify the destination to which callback messages are to be sent.

516 For an SCA reference with a JMS binding and bidirectional interface, the SCA runtime MUST identify the
517 callback destination from the reference's callback service binding if present, or supply a suitable callback
518 destination if not present [BJM60014].

519 6.4.2 Invocation of operations on a callback interface

520 An SCA service with a callback interface can invoke operations on that callback interface by sending
521 messages to the destination identified by the **scaCallbackDestination** user property in a message that it
522 has received, the **JMSReplyTo** destination of a one-way message that it has received, or the destination
523 identified by the service's callback reference JMS binding.

524 For an SCA service with a JMS binding, the **callback destination** is identified as follows, in order of
525 priority:

- 526 • The **scaCallbackDestination** identified by an earlier request, if not null;

Deleted: When a request message is sent by a reference with a JMS binding for a request/response MEP, the SCA runtime MUST set a non-null value for the **JMSReplyTo** header in the JMS message it creates for the request. If the JMS binding has a **response** element with a **destination** defined, then the SCA runtime MUST use that destination for the **JMSReplyTo** header value, otherwise the SCA runtime MUST provide an appropriate destination on which to receive response messages. The SCA runtime MAY choose to receive the response message on the basis of its correlation ID as defined by the binding's **@correlationScheme** attribute, or use a unique destination for each response.¶
When a response message is sent by a service with a JMS binding for a request/ref ... [69]

Deleted: 7.4

Deleted: ¶
<#>"**scaConversation** ... [70]

Deleted: When a request message is sent by a re ... [71]

Formatted: Font color: Auto, English (U.S.)

Comment [SAJH4]: Is it OK to have a normative sta ... [72]

Deleted: 7.2

Deleted: if

Deleted: reference has a

Deleted: element with a JMS binding with a request ... [73]

Deleted: otherwise the SCA runtime MUST provide { ... [74]

Deleted: When a callback request message is sent by a

Deleted: for either a one-way or request/response MEP

Deleted: SCA runtime MUST send the

Formatted: Font: Italic

Deleted: request message to the JMS

Formatted: Font: Italic

Formatted

Deleted:

Deleted: 1

Deleted: 21st January

Deleted: 2008.

- 527 • the **JMSReplyTo** destination identified by an earlier one-way request, if not null;
- 528 • the request destination of the service's callback reference JMS binding, if specified.

529 **For an SCA service with a JMS binding, when a callback request message is sent for either a one-way or**
 530 **request/response MEP, the SCA runtime MUST send the callback request message to the callback**
 531 **destination. [BJM60015].**

532 **For an SCA service with a JMS binding, when a callback request message is sent and no callback**
 533 **destination can be identified then the SCA runtime SHOULD raise an error, and MUST throw an**
 534 **exception to the caller of the callback operation [BJM60016].**

535 **For an SCA service with a JMS binding, when a callback request message is sent the SCA runtime**
 536 **MUST set the *JMSReplyTo* destination and correlation identifier in the callback request message as**
 537 **defined in sections 6.1 or 6.2 as appropriate for the type of the callback operation invoked [BJM60017].**

538 6.4.3 Use of JMSReplyTo for callbacks for non-SCA JMS applications

539 When interacting with non-SCA JMS applications, the assembler can choose to model a
 540 request/response message exchange using a bidirectional interface. In this case it is likely that the non-
 541 SCA JMS application does not support the use of the **scaCallbackDestination** user property. To support
 542 this, for one-way messages the **JMSReplyTo** header can be used to identify the destination to be used to
 543 deliver callback messages, as described in sections Q and Q.

544

Deleted: .

Deleted: If no destination is identified then the SCA runtime SHOULD raise an error, and MUST throw an exception to the caller of the callback operation. ¶
 The SCA runtime MUST set the **JMSReplyTo** destination and correlation identifier in the callback request message as defined in sections 7.1 or 7.2 as appropriate for the type of the callback operation invoked. ¶

Deleted: 7.4.1

Deleted: 7.4.2

Deleted: <#>Conversations ¶
 A conversation is a sequence of operations between two parties that have a common context. The conversation can include a mixture of operations in either direction between the two parties, if the interface is also bidirectional. Interfaces are marked as conversational in order to ensure that the runtime manages the lifecycle of this context. Component implementation specifications define the manner in which the context that is associated with the conversation identifier is made available to component implementations. ¶

<#>Starting a conversation ¶
 A conversation is started when an operation is invoked on a conversational interface and there is no active conversation with the target of the invocation. When this happens the SCA runtime MUST supply an identifier for the conversation, if the client component has not already supplied an identifier, and the SCA runtime MUST set the **scaConversationStart** user property to this value in the JMS message that it sends for the request, and associate a new runtime context with this conversation identifier. ¶
 When a message is received that contains a value for the **scaConversationStart** user property, the SCA runtime MUST associate a new runtime context with the given conversation identifier. ¶
 The SCA runtime MAY include in the message that starts the conversation the **scaConversationMaxl**(... [75]

Deleted: 1

Deleted: 21st January

Deleted: 2008.

545 7 Examples

546 The following snippets show the *sca.composite* file for the *MyValueComposite* file containing the
547 *service* element for the *MyValueService* and a *reference* element for the *StockQuoteService*. Both the
548 service and the reference use a JMS binding.

549 7.1 Minimal Binding Example

550 The following example shows the JMS binding being used with no further attributes or elements. In this
551 case, it is left to the deployer to identify the resources to which the binding is connected.

```
552 <?xml version="1.0" encoding="ASCII"?>  
553 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"  
554           name="MyValueComposite">  
555  
556     <service name="MyValueService">  
557       <interface.java interface="services.myvalue.MyValueService"/>  
558       <binding.jms/>  
559     </service>  
560  
561     <reference name="StockQuoteService">  
562       <interface.java interface="services.stockquote.StockQuoteService"/>  
563       <binding.jms/>  
564     </reference>  
565 </composite>
```

Deleted: 200712

566 7.2 URI Binding Example

567 The following example shows the JMS binding using the *@uri* attribute to specify the connection type and
568 its information:

```
569 <?xml version="1.0" encoding="ASCII"?>  
570 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"  
571           name="MyValueComposite">  
572  
573     <service name="MyValueService">  
574       <interface.java interface="services.myvalue.MyValueService"/>  
575       <binding.jms uri="jms:MyValueServiceQueue?  
576                   activationSpecName=MyValueServiceAS&  
577                   ... "/>  
578     </service>  
579  
580     <reference name="StockQuoteService">  
581       <interface.java interface="services.stockquote.StockQuoteService"/>  
582       <binding.jms uri="jms:StockQuoteServiceQueue?  
583                   connectionFactoryName=StockQuoteServiceQCF&  
584                   deliveryMode=1&  
585                   ... "/>  
586     </reference>  
587 </composite>
```

Deleted: 200712

588 7.3 Binding with Existing Resources Example

589 The following example shows the JMS binding using existing resources:

```
590 <?xml version="1.0" encoding="ASCII"?>  
591 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"  
592           name="MyValueComposite">  
593  
594     <service name="MyValueService">
```

Deleted: 200712

Deleted: 1

Deleted: 21st January

Deleted: 2008.

```

595 <interface.java interface="services.myvalue.MyValueService"/>
596 <binding.jms>
597 <destination jndiName="MyValueServiceQ" create="never"/>
598 <activationSpec jndiName="MyValueServiceAS" create="never"/>
599 </binding.jms>
600 </service>
601 </composite>

```

Comment [SAJH5]: Is there a more realistic example of a JNDI name?

602 7.4 Resource Creation Example

603 The following example shows the JMS binding providing information to create JMS resources rather than
 604 using existing ones:

```

605 <?xml version="1.0" encoding="ASCII"?>
606 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"
607 <name="MyValueComposite">
608
609 <service name="MyValueService">
610 <interface.java interface="services.myvalue.MyValueService"/>
611 <binding.jms>
612 <destination jndiName="MyValueServiceQueue" create="always">
613 <property name="prop1" type="string">XYZ</property>
614 <property name="destName" type="string">MyValueDest</property>
615 </destination>
616 <activationSpec jndiName="MyValueServiceAS" create="always"/>
617 <resourceAdapter jndiName="com.example.JMSRA"/>
618 </binding.jms>
619 </service>
620
621 <reference name="StockQuoteService">
622 <interface.java interface="services.stockquote.StockQuoteService"/>
623 <binding.jms>
624 <destination jndiName="StockQuoteServiceQueue"/>
625 <connectionFactory jndiName="StockQuoteServiceQCF"/>
626 <resourceAdapter name="com.example.JMSRA"/>
627 </binding.jms>
628 </reference>
629 </composite>

```

Deleted: 200712

630 7.5 Request/Response Example

631 The following example shows the JMS binding using existing resources to support request/response
 632 operations. The service uses the **JMSReplyTo** destination to send response messages, and does not
 633 specify a response queue:

```

634 <?xml version="1.0" encoding="ASCII"?>
635 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"
636 <name="MyValueComposite">
637
638 <service name="MyValueService">
639 <interface.java interface="services.myvalue.MyValueService"/>
640 <binding.jms correlationScheme="sca:messageId">
641 <destination jndiName="MyValueServiceQ" create="never"/>
642 <activationSpec jndiName="MyValueServiceAS" create="never"/>
643 </binding.jms>
644 </service>
645
646 <reference name="StockQuoteService">
647 <interface.java interface="services.stockquote.StockQuoteService"/>
648 <binding.jms correlationScheme="sca:messageId">
649 <destination jndiName="StockQuoteServiceQueue"/>
650 <connectionFactory jndiName="StockQuoteServiceQCF"/>
651 <response>

```

Deleted: 200712

Deleted: messageId

Deleted: messageId

Deleted: 1

Deleted: 21st January

Deleted: 2008.

652
653
654
655
656
657

```
<destination jndiName="MyValueResponseQueue"/>
<activationSpec jndiName="MyValueResponseAS"/>
</response>
</binding.jms>
</reference>
</composite>
```

658 7.6 Use of Predefined Definitions Example

659 This example shows the case where there is common connection information shared by more than one
660 reference.

661 The common connection information is defined in a separate definitions file:

662
663
664
665
666
667
668
669

```
<?xml version="1.0" encoding="ASCII"?>
<definitions targetNamespace="http://acme.com"
  xmlns="http://docs.oasis-open.org/ns/opencsa/sca/2007903">
  <binding.jms name="StockQuoteService">
    <destination jndiName="StockQuoteServiceQueue" create="never"/>
    <connectionFactory jndiName="StockQuoteServiceQCF" create="never"/>
  </binding.jms>
</definitions>
```

Deleted: 200712

670 Any **binding.jms** element may then refer to that definition:

671
672
673
674
675
676
677
678
679

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"
  xmlns:acme="http://acme.com"
  name="MyValueComposite">
  <reference name="MyValueService">
    <interface.java interface="services.myvalue.MyValueService"/>
    <binding.jms requestConnection="acme:StockQuoteService"/>
  </reference>
</composite>
```

Deleted: 200712

680 7.7 Subscription with Selector Example

681 The following example shows how the JMS binding is used in order to consume messages from existing
682 JMS infrastructure. The JMS binding subscribes using selector:

683
684
685
686
687
688
689
690
691
692
693
694
695

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200903"
  name="MyValueComposite">
  <service name="MyValueService">
    <interface.java interface="services.myvalue.MyValueService"/>
    <binding.jms>
      <destination jndiName="MyValueServiceTopic" create="never"/>
      <connectionFactory jndiName="StockQuoteServiceTCF"
        create="never"/>
      <messageSelection selector="Price>1000"/>
    </binding.jms>
  </service>
</composite>
```

Deleted: 200712

Deleted: subscriptionHead
ers JMSSelector

696 7.8 Policy Set Example

697 A policy set defines the manner in which intents map to JMS binding properties. The following illustrates
698 an example of a policy set that defines values for the **@priority** attribute using the **"priority"** intent, and
699 also allows setting of a value for a user JMS property using the **"log"** intent.

700
701
702

```
<policySet name="JMSPolicy"
  provides="priority log"
  appliesTo="binding.jms">
```

Deleted: JMS

Deleted: 1

Deleted: 21st January

Deleted: 2008.

703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723

```
<intentMap provides="priority" default="medium">  
  <qualifier name="high">  
    <headers priority="9"/>  
  </qualifier>  
  <qualifier name="medium">  
    <headers priority="4"/>  
  </qualifier>  
  <qualifier name="low">  
    <headers priority="0"/>  
  </qualifier>  
</intentMap>  
  
<intentMap provides="log">  
  <qualifier>  
    <headers>  
      <property name="user_example_log">logged</property>  
    </headers>  
  </qualifier>  
</intentMap>  
</policySet>
```

Deleted: JMSPriority

Deleted: JMSPriority

Deleted: JMSPriority

724 Given this policy set, the intents can be required on a service or reference:

725
726
727
728
729
730
731

```
<reference name="StockQuoteService" requires="priority.high log">  
  <interface.java interface="services.stockquote.StockQuoteService"/>  
  <binding.jms>  
    <destination name="StockQuoteServiceQueue"/>  
    <connectionFactory name="StockQuoteServiceQCF"/>  
  </binding.jms>  
</reference>
```

Deleted: 1

Deleted: 21st January

Deleted: 2008.

732

8 Conformance

733

734

735

736

737

The XML schema pointed to by the RDDDL document, at the namespace URI, defined by this specification, are considered to be authoritative and take precedence over the XML schema defined in the appendix of this document. There are two categories of artifacts for which this specification defines conformance:

a) SCA JMS Binding XML Document

b) SCA Runtime

738

8.1 SCA JMS Binding XML Document

739

740

741

An SCA JMS Binding XML document is an SCA Composite Document, an SCA Definitions Document or an SCA ComponentType Document, as defined by the SCA Assembly Specification Section 13.1 , that uses the **binding:jms** element.

742

743

744

An SCA JMS Binding XML document MUST be a conformant SCA Composite Document, SCA Definitions Document or a SCA ComponentType Document, as defined by the SCA Assembly Specification , and MUST comply with all the applicable requirements specified in this specification.

745

8.2 SCA Runtime

746

747

An implementation that claims to conform to the requirements of an SCA Runtime defined in this specification has to meet the following conditions:

748

749

750

751

752

753

1. The implementation MUST comply with all statements in Appendix B: Conformance Items related to an SCA Runtime, notably all "MUST" statements have to be implemented
2. The implementation MUST conform to the SCA Assembly Model Specification Version 1.1 , and to the SCA Policy Framework Version 1.1 [SCA-Policy]
3. The implementation MUST reject an SCA JMS Binding XML Document that is not conformant per Section 8.1

Deleted: Any SCA runtime that claims to support this binding MUST abide by the requirements of this specification.¶

Deleted: available

Deleted: is

Deleted: takes

Deleted: Schema

Deleted: 1

Deleted: 21st January

Deleted: 2008.

754

A. JMS XML Binding Schema: sca-binding-jms.xsd

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright (C) OASIS (R) 2005, 2009. All Rights Reserved.
OASIS trademark, IPR and other policies apply. -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://docs.oasis-open.org/ns/opencsa/sca/200903"
xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200903"
elementFormDefault="qualified">

  <include schemaLocation="sca-core-1.1-cd03.xsd"/>

  <complexType name="JMSBinding">
    <complexContent>
      <extension base="sca:Binding">
        <sequence>
          <element name="destination" type="sca:JMSDestination"
minOccurs="0"/>
          <choice minOccurs="0" maxOccurs="1">
            <element name="connectionFactory"
type="sca:JMSConnectionFactory"/>
            <element name="activationSpec" type="sca:JMSActivationSpec"/>
          </choice>
          <element name="response" type="sca:JMSResponse" minOccurs="0"/>
          <element name="headers" type="sca:JMSHeaders" minOccurs="0"/>
          <element name="messageSelection" type="sca:JMSMessageSelection"
minOccurs="0"/>
          <element name="resourceAdapter" type="sca:JMSResourceAdapter"
minOccurs="0"/>
          <element name="operationProperties"
type="sca:JMSOperationProperties"
minOccurs="0" maxOccurs="unbounded"/>
          <any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="correlationScheme" type="QName"
default="sca:messageId"/>
        <attribute name="initialContextFactory" type="anyURI"/>
        <attribute name="jndiURL" type="anyURI"/>
        <attribute name="requestConnection" type="QName"/>
        <attribute name="responseConnection" type="QName"/>
        <attribute name="operationProperties" type="QName"/>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="JMSCreateResource">
    <restriction base="string">
      <enumeration value="always"/>
      <enumeration value="never"/>
      <enumeration value="ifNotExist"/>
    </restriction>
  </simpleType>

  <complexType name="JMSDestination">
    <sequence>
      <element name="property" type="sca:BindingProperty"
minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="jndiName" type="anyURI" use="required"/>
  </complexType>

```

Deleted: Within this specification, the following conformance targets are used:¶
 <#>XML document elements and attributes, including **binding.jms** and its children, and **bindingType**¶
 <#>The SCA runtime – this refers to the implementation that provides the functionality to support the SCA specifications, including that specific to the JMS binding as well as other SCA capabilities¶
 <#>JMS objects, including Destinations, ConnectionFactories and ActivationSpecs¶
 <#>WSDL documents¶
 <#>**JMS Binding Schema**¶

Deleted: (c)

Deleted: 2006, 2008

Deleted: 200712 "

Deleted: 200712

Deleted:
 <sequence> ¶

<element name="destination" type="sca:JMSDestination"/> ¶

Deleted:

Deleted:
 </sequence> ¶

<sequence> ¶

<element name="destination" type="sca:JMSDestination" minOccurs="0"/> ¶

Deleted: ¶

</sequence>

Deleted: subscriptionHeaders "

Deleted:
 type="sca:JMSSubscriptionHeaders"¶

Deleted: ="

Deleted: "

Deleted: messageId

Deleted: ifnotexist

Deleted: 1

Deleted: 21st January

Deleted: 2008.

```

815 <attribute name="type" use="optional" default="queue">
816 <simpleType>
817 <restriction base="string">
818 <enumeration value="queue"/>
819 <enumeration value="topic"/>
820 </restriction>
821 </simpleType>
822 </attribute>
823 <attribute name="create" type="sca:JMSCreateResource"
824 use="optional" default="ifNotExist"/>
825 </complexType>
826
827 <complexType name="JMSConnectionFactory">
828 <sequence>
829 <element name="property" type="sca:BindingProperty"
830 minOccurs="0" maxOccurs="unbounded"/>
831 </sequence>
832 <attribute name="jndiName" type="anyURI" use="required"/>
833 <attribute name="create" type="sca:JMSCreateResource"
834 use="optional" default="ifNotExist"/>
835 </complexType>
836
837 <complexType name="JMSActivationSpec">
838 <sequence>
839 <element name="property" type="sca:BindingProperty"
840 minOccurs="0" maxOccurs="unbounded"/>
841 </sequence>
842 <attribute name="jndiName" type="anyURI" use="required"/>
843 <attribute name="create" type="sca:JMSCreateResource"
844 use="optional" default="ifNotExist"/>
845 </complexType>
846
847 <complexType name="JMSResponse">
848 <sequence>
849 <element name="wireFormat" type="sca:WireFormatType" minOccurs="0"/>
850 <element name="destination" type="sca:JMSDestination" minOccurs="0"/>
851 <choice minOccurs="0">
852 <element name="connectionFactory" type="sca:JMSConnectionFactory"/>
853 <element name="activationSpec" type="sca:JMSActivationSpec"/>
854 </choice>
855 </sequence>
856 </complexType>
857
858 <complexType name="JMSHeaders">
859 <sequence>
860 <element name="property" type="sca:BindingProperty"
861 minOccurs="0" maxOccurs="unbounded"/>
862 </sequence>
863 <attribute name="type" type="string"/>
864 <attribute name="deliveryMode">
865 <simpleType>
866 <restriction base="string">
867 <enumeration value="persistent"/>
868 <enumeration value="nonpersistent"/>
869 </restriction>
870 </simpleType>
871 </attribute>
872 <attribute name="timeToLive" type="long"/>
873 <attribute name="priority">
874 <simpleType>
875 <restriction base="string">
876 <enumeration value="0"/>
877 <enumeration value="1"/>
878 <enumeration value="2"/>

```

Deleted: ifnotexist

Deleted: ifnotexist

Deleted: ifnotexist

Deleted: JMSType

Deleted: JMSDeliveryMode

Deleted: PERSISTENT

Deleted: NON_PERSISTENT

Deleted: JMSTimeToLive

Deleted: JMSPriority

Deleted: 1

Deleted: 21st January

Deleted: 2008.

```

879         <enumeration value="3"/>
880         <enumeration value="4"/>
881         <enumeration value="5"/>
882         <enumeration value="6"/>
883         <enumeration value="7"/>
884         <enumeration value="8"/>
885         <enumeration value="9"/>
886     </restriction>
887 </simpleType>
888 </attribute>
889 </complexType>
890
891 <complexType name="JMSMessageSelection">
892     <sequence>
893         <element name="property" type="sca:BindingProperty"
894             minOccurs="0" maxOccurs="unbounded"/>
895     </sequence>
896     <attribute name="selector" type="string"/>
897 </complexType>
898
899 <complexType name="JMSResourceAdapter">
900     <sequence>
901         <element name="property" type="sca:BindingProperty"
902             minOccurs="0" maxOccurs="unbounded"/>
903     </sequence>
904     <attribute name="name" type="string" use="required"/>
905 </complexType>
906
907 <complexType name="JMSOperationProperties">
908     <sequence>
909         <element name="property" type="sca:BindingProperty"
910             minOccurs="0" maxOccurs="unbounded"/>
911         <element name="headers" type="sca:JMSHeaders"/>
912     </sequence>
913     <attribute name="name" type="string" use="required"/>
914     <attribute name="nativeOperation" type="string"/>
915 </complexType>
916
917 <complexType name="BindingProperty">
918     <simpleContent>
919         <extension base="string">
920             <attribute name="name" type="NMTOKEN"/>
921             <attribute name="type" type="string" use="optional"
922                 default="xs:string"/>
923         </extension>
924     </simpleContent>
925 </complexType>
926
927 <element name="binding.jms" type="sca:JMSBinding"
928     substitutionGroup="sca:binding"/>
929
930 <element name="wireFormat.jmsDefault" type="sca:WireFormatType"
931     substitutionGroup="sca:wireFormat"/>
932
933 <element name="operationSelector.jmsDefault" type="sca:OperationSelectorType"
934     substitutionGroup="sca:operationSelector"/>
935 </schema>

```

Deleted: JMSSubscriptionH
eaders

Deleted: JMSSelector

Deleted: jmsdefault

Deleted: jmsdefault

Deleted: 1

Deleted: 21st January

Deleted: 2008.

936

B. Conformance Items

937

This section contains a list of conformance items for the SCA JMS Binding specification.

Conformance ID	Description
[BJM30001]	The value of the <i>@uri</i> attribute MUST have the format defined by the IETF URI Scheme for Java™ Message Service 1.0 [IETFJMS]
[BJM30002]	When the <i>@uri</i> attribute is specified, the SCA runtime MUST raise an error if the referenced resources do not already exist
[BJM30003]	If the value of the <i>@correlationScheme</i> attribute is " <i>sca:messageID</i> " the SCA runtime MUST set the correlation ID of replies to the message ID of the corresponding request
[BJM30004]	If the value of the <i>@correlationScheme</i> attribute is " <i>sca:correlationID</i> " the SCA runtime MUST set the correlation ID of replies to the correlation ID of the corresponding request
[BJM30005]	If the value of the <i>@correlationScheme</i> attribute is " <i>sca:none</i> " the SCA runtime MUST NOT set the correlation ID
[BJM30006]	SCA runtimes MAY allow other values of the <i>@correlationScheme</i> attribute to indicate other correlation schemes
[BJM30007]	If the <i>@requestConnection</i> attribute is specified, the <i>binding.jms</i> element MUST NOT contain a <i>destination</i> , <i>connectionFactory</i> , <i>activationSpec</i> or <i>resourceAdapter</i> element
[BJM30008]	If the <i>@responseConnection</i> attribute is specified, the <i>binding.jms</i> element MUST NOT contain a <i>response</i> element
[BJM30009]	If the <i>@operationProperties</i> attribute is specified, the <i>binding.jms</i> element MUST NOT contain an <i>operationProperties</i> element
[BJM30010]	Whatever the value of the <i>destination/@type</i> attribute, the runtime MUST ensure a single response is delivered for request/response operations
[BJM30011]	If the <i>@create</i> attribute value for a destination, connectionFactory or activationSpec element is " <i>always</i> " then the <i>@jndiName</i> attribute is optional; if the resource cannot be created at the specified location then the SCA runtime MUST raise an error
[BJM30012]	If the <i>@create</i> attribute value for a destination, connectionFactory or activationSpec element is " <i>ifNotExist</i> " then the <i>@jndiName</i> attribute MUST specify the location of the possibly existing resource
[BJM30013]	If the destination, connectionFactory or activationSpec does not exist at the location identified by the <i>@jndiName</i> attribute, but cannot be created there then the SCA runtime MUST raise an error
[BJM30014]	If the destination, connectionFactory or activationSpec's <i>@jndiName</i> attribute refers to an existing resource that is not a JMS Destination of the appropriate type, a JMS connection factory or a JMS activation spec respectively then the SCA runtime MUST raise an error
[BJM30015]	If the <i>@create</i> attribute value for a destination, connectionFactory or

Deleted: 1

Deleted: 21st January

Deleted: 2008.

	activationSpec element is " never " then the @jndiName attribute MUST specify the location of the existing resource
[BJM30016]	If the destination, connection factory or activation spec is not present at the location identified by the @jndiName attribute, or the location refers to a resource of an incorrect type then the SCA runtime MUST raise an error
[BJM30017]	A binding.jms element MUST NOT include both a connectionFactory element and an activationSpec element
[BJM30018]	When the connectionFactory element is present, then the destination MUST be defined either by the destination element or the @uri attribute
[BJM30019]	If the activationSpec element is present and the destination is also specified via a destination element or the @uri attribute then it MUST refer to the same JMS destination as the activationSpec
[BJM30020]	The activationSpec element MUST NOT be present when the binding is being used for an SCA reference
[BJM30021]	A response element MUST NOT include both a connectionFactory element and an activationSpec element
[BJM30022]	If a response/destination and response/activationSpec element are both specified they MUST refer to the same JMS destination
[BJM30023]	The response/activationSpec element MUST NOT be present when the binding is being used for an SCA service
[BJM30024]	The SCA runtime MUST set JMS headers in messages that it creates to the values specified by the headers element unless overridden for the operation being invoked.
[BJM30025]	If the @uri attribute includes values for the type, delivery mode, time to live or priority properties then the @uri values are used and the headers and operationProperties/headers @type, @deliveryMode, @timeToLive or @priority attributes are ignored
[BJM30026]	For each header/properties element the SCA runtime MUST set the named JMS user property to the given value in messages it creates unless overridden for the operation being invoked
[BJM30027]	If the @uri attribute includes a value for the message selector then the @uri value is used and the messageSelection/@selector attribute is ignored
[BJM30028]	SCA runtimes MAY place restrictions on the properties of the resource adapter Java bean that can be set using the resourceAdapter element
[BJM30029]	The value of the operationProperties/@selectedOperation attribute MUST be unique across the containing binding.jms element
[BJM30030]	The SCA runtime SHOULD make the operationProperties element corresponding to the selectedOperation available to the wireFormat implementation
[BJM30031]	The resourceAdapter element MUST be present when JMS resources are to be created for a JMS provider that implements the JCA 1.5 specification [JCA15], and is ignored otherwise
[BJM30032]	The SCA runtime MUST set JMS headers in messages it creates when the operation identified by the operationProperties/@name attribute is invoked to

Deleted: 1

Deleted: 21st January

Deleted: 2008.

	the values specified by the corresponding operationProperties/headers element
[BJM30033]	For each operationProperties/headers/property element the SCA runtime MUST set the named JMS user property to the given value in messages it creates when the operation identified by the operationProperties/@name attribute is invoked
[BJM30034]	When the @uri attribute is specified, the destination element MUST NOT be present
[BJM30035]	An SCA runtime MUST use the values specified in the @uri attribute in preference to corresponding attributes and elements in the binding
[BJM30036]	The binding.jms element MUST conform to the XML schema defined in sca-binding-jms.xsd
[BJM40001]	The SCA runtime MUST support the default JMS wire format and operation selector behavior, and MAY provide additional means to override it
[BJM40002]	If no operationSelector element is specified then SCA runtimes MUST use operationSelector.jmsDefault as the default
[BJM40003]	When using the default wire format to send request messages, if there is a single parameter and the interface includes more than one operation, the SCA runtime MUST set the JMS user property " scaOperationName " to the name of the operation being invoked
[BJM40004]	If no wireFormat element is specified in a JMS binding then SCA runtimes MUST use wireFormat.jmsDefault as the default
[BJM40005]	When using the default wire format an SCA runtime MUST be able to receive both JMS text and bytes messages
[BJM40006]	When using the default wire format an SCA runtime MUST send either a JMS text or a JMS bytes message
[BJM40007]	When using the default wire format an SCA runtime MAY provide additional configuration to allow selection between JMS text or bytes messages to be sent
[BJM40008]	When a binding.jms element specifies the operationSelector.jmsDefault element, the SCA runtime MUST use the default operation selection algorithm to determine the selected operation
[BJM40009]	When a binding.jms element specifies the wireFormat.jmsDefault element, the SCA runtime MUST use the default wire format
[BJM60001]	For an SCA reference with a JMS binding, when a request message is sent as part of a one-way MEP, the SCA runtime SHOULD NOT set the JMSReplyTo destination header in the JMS message that it creates, regardless of whether the JMS binding has a response element with a destination defined
[BJM60002]	For an SCA service with a JMS binding, when a request message is received as part of a one-way MEP, the SCA runtime MUST ignore the JMSReplyTo destination header in the JMS message, and not raise an error
[BJM60003]	For an SCA reference with a JMS binding, when a request message is sent as part of a request/response MEP, the SCA runtime MUST set a non-null value for the JMSReplyTo header in the JMS message it creates for the request

Deleted: 1

Deleted: 21st January

Deleted: 2008.

[BJM60004]	For an SCA reference with a JMS binding, when a request message is sent as part of a request/response MEP, and the JMS binding has a response element with a destination defined, then the SCA runtime MUST use that destination for the JMSReplyTo header in the JMS message it creates for the request
[BJM60005]	For an SCA reference with a JMS binding, when a request message is sent as part of a request/response MEP, and the JMS binding does not have a response element with a destination defined, the SCA runtime MUST provide an appropriate destination on which to receive response messages and use that destination for the JMSReplyTo header in the JMS message it creates for the request
[BJM60006]	For an SCA reference with a JMS binding, the SCA runtime MAY choose to receive response messages on the basis of their correlation ID as defined by the binding's @correlationScheme attribute, or use a unique destination for each response
[BJM60007]	For an SCA service with a JMS binding, when a response message is sent as part of a request/response MEP where the request message included a non-null JMSReplyTo destination, the SCA runtime MUST send the response message to that destination
[BJM60008]	For an SCA service with a JMS binding, when a response message is sent as part of a request/response MEP where the request message included a null JMSReplyTo destination and the JMS binding includes a response/destination element the SCA runtime MUST send the response message to that destination
[BJM60009]	For an SCA service with a JMS binding, when a response message is sent as part of a request/response MEP where the request message included a null JMSReplyTo destination and the JMS binding does not include a response/destination then an error SHOULD be raised by the SCA runtime
[BJM60010]	For an SCA service with a JMS binding, when a response message is sent as part of a request/response MEP the SCA runtime MUST set the correlation identifier in the JMS message that it creates for the response as defined by the JMS binding's @correlationScheme attribute
[BJM60011]	For an SCA reference with a JMS binding and a bidirectional interface, when a request message is sent the SCA runtime MUST set the destination to which callback messages are to be sent as the value of the scaCallbackDestination user property in the message it creates
[BJM60012]	For an SCA reference with a JMS binding and bidirectional interface, when a request message is sent the SCA runtime MAY set the JMSReplyTo destination to the same value as the scaCallbackDestination user property
[BJM60013]	For an SCA reference with a JMS binding and bidirectional interface, when a request message is sent as part of a request/response MEP, the SCA runtime MUST set the JMSReplyTo header in the message it creates as described in section 6.2
[BJM60014]	For an SCA reference with a JMS binding and bidirectional interface, the SCA runtime MUST identify the callback destination from the reference's callback service binding if present, or supply a suitable callback destination if not present
[BJM60015]	For an SCA service with a JMS binding, when a callback request message is

Deleted: 1

Deleted: 21st January

Deleted: 2008.

	sent for either a one-way or request/response MEP, the SCA runtime MUST send the callback request message to the callback destination.
[BJM60016]	For an SCA service with a JMS binding, when a callback request message is sent and no callback destination can be identified then the SCA runtime SHOULD raise an error, and MUST throw an exception to the caller of the callback operation
[BJM60017]	For an SCA service with a JMS binding, when a callback request message is sent the SCA runtime MUST set the JMSReplyTo destination and correlation identifier in the callback request message as defined in sections 6.1 or 6.2 as appropriate for the type of the callback operation invoked
[BJM60018]	SCA runtimes MUST follow the behavior described in section 6.4 and its subsections when binding.jms is used in both the forward and callback directions

938

Deleted: 1
 Deleted: 21st January
 Deleted: 2008.

939

C. Acknowledgements

940 The following individuals have participated in the creation of this specification and are gratefully
941 acknowledged:

942 **Participants:**

<u>Participant Name</u>	<u>Affiliation</u>
Bryan Aupperle	IBM
Ron Barack	SAP AG
Michael Beisiegel	IBM
Henning Blohm	SAP AG
David Booz	IBM
Martin Chapman	Oracle Corporation
Jean-Sebastien Delfino	IBM
Laurent Domenech	TIBCO Software Inc.
Jacques Durand	Fujitsu Limited
Mike Edwards	IBM
Billy Feng	Primeton Technologies, Inc.
Nimish Hathalia	TIBCO Software Inc.
Simon Holdsworth	IBM
Eric Johnson	Software Inc.
Uday Joshi	Oracle Corporation
Khanderao Kand	Oracle Corporation
Anish Karmarkar	Oracle Corporation
Nickolaos Kavantzias	Oracle Corporation
Mark Little	Red Hat
Ashok Malhotra	Oracle Corporation
Jim Marino	Individual
Jeff Mischkinsky	Oracle Corporation
Dale Moberg	Axway Software
Simon Nash	Individual
Sanjay Patil	SAP AG
Plamen Pavlov	SAP AG
Peter Peshev	SAP AG
Piotr Przybylski	IBM
Luciano Resende	IBM
Tom Rutt	Fujitsu Limited
Vladimir Savchenko	SAP AG
Scott Vorthmann	TIBCO Software Inc.
Tim Watson	Oracle Corporation
Owen Williams	Avaya, Inc.
Prasad Yendluri	Software AG, Inc.

Deleted: 1

Deleted: 21st January

Deleted: 2008.

Deleted: DisplayText canr
DisplayText cannot span
¶
<#>Non-Normative Text¶

Deleted: 1
Deleted: 21st January
Deleted: 2008.

Deleted: 1
Deleted: 21st January
Deleted: 2008.

945
946
947

D. Revision History

[optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
1	2007-09-25	Anish Karmarkar	Applied the OASIS template + related changes to the Submission
2	2008-03-12	Simon Holdsworth	Updated text for RFC2119 conformance Updates to resolve following issues: BINDINGS-1 BINDINGS-5 BINDINGS-6 BINDINGS-12 BINDINGS-14 BINDINGS-18 BINDINGS-26 Applied updates discussed at Bindings TC meeting of 27 th March
3	2008-06-19	Simon Holdsworth	* Applied most of the editorial changes from Eric Johnson's review
cd01	2008-08-01	Simon Holdsworth	Updates to resolve following issues: BINDINGS-13 (JMS part) BINDINGS-20 (complete) BINDINGS-30 (JMS part) BINDINGS-32 (JMS part) BINDINGS-33 (complete) BINDINGS-34 (complete) BINDINGS-35 (complete) BINDINGS-38 (JMS part)
cd01-rev1	2008-10-16	Simon Holdsworth	Updated text for RFC2119 conformance throughout Updates to resolve following issues: BINDINGS-41 BINDINGS-46 BINDINGS-47
cd01-rev2	2008-12-01	Simon Holdsworth	Added comments identifying those updates that relate to RFC2119 language (issue 52)
cd01-rev3	2008-12-02	Simon Holdsworth	Final RFC2119 language updates BINDINGS-52
cd01-rev4	2009-01-09	Simon Holdsworth	Updates to resolve following issues:

Deleted: 1

Deleted: 21st January

Deleted: 2008.

			BINDINGS-7 BINDINGS-31 BINDINGS-40 BINDINGS-42 BINDINGS-44 BINDINGS-50
cd02	2009-02-16	Simon Holdsworth	Rename and editorial updates
cd02-rev1	2009-05-22	Simon Holdsworth	Updates to resolve issue BINDINGS-62 (conformance statement numbering) Updated assembly namespace to 200903 Fixed errors in schema
cd02-rev2	2009-05-22	Simon Holdsworth	Updates to resolve following issues: BINDINGS-39 BINDINGS-59 BINDINGS-65 BINDINGS-66 BINDINGS-67 BINDINGS-68 BINDINGS-70 BINDINGS-71
cd02-rev3	2009-06-18	Simon Holdsworth	Editorial concerns addressed Added acknowledgements appendix
cd02-rev4	2009-06-19	Simon Holdsworth	Updates to resolve following issues BINDINGS-74 Some editorial updates Fixed normative statement missed in application of BINDINGS-67

948

Deleted: 1
Deleted: 21st January
Deleted: 2008.

Page 4: [1] Change **Unknown**

Field Code Changed

Page 4: [1] Change **Unknown**

Field Code Changed

Page 4: [2] Change **Unknown**

Field Code Changed

Page 4: [2] Change **Unknown**

Field Code Changed

Page 4: [3] Change **Unknown**

Field Code Changed

Page 4: [3] Change **Unknown**

Field Code Changed

Page 4: [4] Change **Unknown**

Field Code Changed

Page 4: [4] Change **Unknown**

Field Code Changed

Page 4: [5] Change **Unknown**

Field Code Changed

Page 4: [6] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[2.....Messaging Bindings](#)

Page 4: [7] Formatted **Simon Holdsworth** **19/06/2009 15:18:00**

TOC 2, Tabs: Not at 0.85 cm

Page 4: [8] Change **Unknown**

Field Code Changed

Page 4: [9] Change **Unknown**

Field Code Changed

Page 4: [10] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[3 JMS Binding Schema.....](#)

Page 4: [11] Change **Unknown**

Field Code Changed

Page 4: [12] Change **Unknown**

Field Code Changed

Page 4: [13] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

4 [Operation Selectors and Wire Formats](#).....

Page 4: [14] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [15] Formatted	Simon Holdsworth	19/06/2009 15:18:00
-------------------------------	-------------------------	----------------------------

TOC 1,TOC 1 Char,TOC 1 Char1 Char,TOC 1 Char Char Char, Tabs: 0.85 cm, Left

Page 4: [16] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [17] Deleted	Simon Holdsworth	19/06/2009 15:18:00
-----------------------------	-------------------------	----------------------------

4.1 [Default Operation Selection](#).....

Page 4: [18] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [19] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [20] Deleted	Simon Holdsworth	19/06/2009 15:18:00
-----------------------------	-------------------------	----------------------------

[2](#)

Page 4: [20] Deleted	Simon Holdsworth	19/06/2009 15:18:00
-----------------------------	-------------------------	----------------------------

[Wire Format](#).....

Page 4: [21] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [22] Formatted	Simon Holdsworth	19/06/2009 15:18:00
-------------------------------	-------------------------	----------------------------

TOC 2, Tabs: Not at 0.85 cm

Page 4: [23] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [24] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [25] Formatted	Simon Holdsworth	19/06/2009 15:18:00
-------------------------------	-------------------------	----------------------------

TOC 3, Tabs: Not at 0.85 cm

Page 4: [26] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [27] Deleted	Simon Holdsworth	19/06/2009 15:18:00
-----------------------------	-------------------------	----------------------------

[6](#).....[Message Exchange Patterns](#)

Page 4: [28] Change	Unknown
----------------------------	----------------

Field Code Changed

Page 4: [29] Formatted **Simon Holdsworth** **19/06/2009 15:18:00**

TOC 1,TOC 1 Char,TOC 1 Char1 Char,TOC 1 Char Char Char, Tabs: 0.85 cm, Left

Page 4: [30] Change **Unknown**

Field Code Changed

Page 4: [31] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.1 One-way message exchange \(no Callbacks\)](#)

Page 4: [32] Change **Unknown**

Field Code Changed

Page 4: [33] Change **Unknown**

Field Code Changed

Page 4: [34] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.2 Request/response message exchange \(no Callbacks\)](#)

Page 4: [35] Change **Unknown**

Field Code Changed

Page 4: [36] Change **Unknown**

Field Code Changed

Page 4: [37] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.3 JMS User Properties](#)

Page 4: [38] Change **Unknown**

Field Code Changed

Page 4: [39] Change **Unknown**

Field Code Changed

Page 4: [40] Change **Unknown**

Field Code Changed

Page 4: [41] Formatted **Simon Holdsworth** **19/06/2009 15:18:00**

TOC 2

Page 4: [42] Change **Unknown**

Field Code Changed

Page 4: [43] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.4.1 Invocation of operations on a bidirectional interface](#)

Page 4: [44] Change **Unknown**

Field Code Changed

Page 4: [45] Change **Unknown**

Field Code Changed

Page 4: [46] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.4.2 Invocation of operations on a callback interface](#)

Page 4: [47] Change **Unknown**

Field Code Changed

Page 4: [48] Change **Unknown**

Field Code Changed

Page 4: [49] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.4.3 Use of JMSReplyTo for callbacks for non-SCA JMS applications](#)

Page 4: [50] Change **Unknown**

Field Code Changed

Page 4: [51] Formatted **Simon Holdsworth** **19/06/2009 15:18:00**

TOC 3

Page 4: [52] Change **Unknown**

Field Code Changed

Page 4: [53] Change **Unknown**

Field Code Changed

Page 4: [54] Change **Unknown**

Field Code Changed

Page 4: [55] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.5.1 Starting a conversation](#)

Page 4: [56] Change **Unknown**

Field Code Changed

Page 4: [57] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

[6.5.2 Continuing a conversation](#)

Page 4: [58] Formatted **Simon Holdsworth** **19/06/2009 15:18:00**

TOC 1,TOC 1 Char,TOC 1 Char1 Char,TOC 1 Char Char Char, Tabs: 0.85 cm, Left

Page 4: [59] Change **Unknown**

Field Code Changed

Page 4: [60] Change **Unknown**

Field Code Changed

Page 4: [61] Formatted **Simon Holdsworth** **19/06/2009 15:18:00**

TOC 2

Page 1: [62] Deleted **Simon Holdsworth** **16/02/2009 10:52:00**

Page 1: [62] Deleted	Simon Holdsworth	16/02/2009 10:53:00
-----------------------------	-------------------------	----------------------------

21st January

Page 9: [63] Deleted	Simon Holdsworth	19/06/2009 15:18:00
-----------------------------	-------------------------	----------------------------

- . Possible values for the **@correlationScheme** attribute are "**sca:MessageID**" (the default) where the SCA runtime MUST set the correlation ID of replies to the message ID of the corresponding request; "**sca:CorrelationID**" where the SCA runtime MUST set the correlation ID of replies to the correlation ID of the corresponding request, and "**sca:None**" which indicates that the SCA runtime MUST NOT set the correlation ID. SCA runtimes MAY allow other values to indicate other correlation schemes

Page 11: [64] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

that the SCA runtime MUST set to the given values for all operations.

Page 11: [65] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

JMSType, @JMSDeliveryMode, @JMSTimeToLive, @JMSPriority

Page 11: [66] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

- . The value of the **@uri** attribute MUST NOT include values for these properties if they are specified using these attributes.

Page 11: [67] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

JMSDeliveryMode are "**PERSISTENT**" and "**NON_PERSISTENT**"

Page 11: [68] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

The value of the **@uri** attribute MUST NOT include values for this property if it is specified using this attribute

Page 18: [69] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

When a request message is sent by a reference with a JMS binding for a request/response MEP, the SCA runtime MUST set a non-null value for the **JMSReplyTo** header in the JMS message it creates for the request. If the JMS binding has a **response** element with a **destination** defined, then the SCA runtime MUST use that destination for the **JMSReplyTo** header value, otherwise the SCA runtime MUST provide an appropriate destination on which to receive response messages. The SCA runtime MAY choose to receive the response message on the basis of its correlation ID as defined by the binding's **@correlationScheme** attribute, or use a unique destination for each response.

When a response message is sent by a service with a JMS binding for a request/response MEP, the SCA runtime MUST send the response message to the destination identified by the request message's **JMSReplyTo** header value if it is not null, otherwise the SCA runtime MUST send the response message to the destination identified by the JMS binding's **response** element if specified. If there is no destination defined by either means then an error SHOULD be raised by the SCA runtime. The SCA runtime MUST set the correlation identifier in the JMS message that it creates for the response as defined by the JMS binding's **@correlationScheme** attribute.

Page 18: [70] Deleted	Simon Holdsworth	19/06/2009 15:18:00
------------------------------	-------------------------	----------------------------

"**scaConversationStart**" indicates that a conversation is to be started, its value is the identifier for the conversation.

"**scaConversationMaxIdleTime**" defines the maximum time that should be allowed between operations in the conversation.

"**scaConversationId**" holds the identifier for the conversation.

Page 18: [71] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

When a request message is sent by a reference with a JMS binding for a one-way MEP with a bidirectional interface, the SCA runtime MUST set the destination to which callback messages are to be sent as the value of the **scaCallbackDestination** user property in the message it creates. The SCA runtime MAY also set the **JMSReplyTo** destination to this value.

When a request message is sent by a reference with a JMS binding for a request/response MEP with a bidirectional interface, the SCA runtime MUST set the **scaCallbackDestination** user property in the message it creates to identify the destination from which it will read callback messages. The SCA runtime MUST set the **JMSReplyTo** header in the message it creates as described in section

Page 18: [72] Comment [SAJH4] **Simon Holdsworth** **20/04/2009 14:54:00**

Is it OK to have a normative statement refer to a section in the doc?

Page 18: [73] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

element with a JMS binding with a request destination, then the SCA runtime MUST use that destination as the one

Page 18: [74] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

, otherwise the SCA runtime MUST provide an appropriate destination for this purpose.

Page 19: [75] Deleted **Simon Holdsworth** **19/06/2009 15:18:00**

Conversations

A conversation is a sequence of operations between two parties that have a common context. The conversation can include a mixture of operations in either direction between the two parties, if the interface is also bidirectional. Interfaces are marked as conversational in order to ensure that the runtime manages the lifecycle of this context. Component implementation specifications define the manner in which the context that is associated with the conversation identifier is made available to component implementations.

Starting a conversation

A conversation is started when an operation is invoked on a conversational interface and there is no active conversation with the target of the invocation. When this happens the SCA runtime MUST supply an identifier for the conversation, if the client component has not already supplied an identifier, and the SCA runtime MUST set the **scaConversationStart** user property to this value in the JMS message that it sends for the request, and associate a new runtime context with this conversation identifier.

When a message is received that contains a value for the **scaConversationStart** user property, the SCA runtime MUST associate a new runtime context with the given conversation identifier.

The SCA runtime MAY include in the message that starts the conversation the **scaConversationMaxIdleTime** user property; if this value is not present the SCA runtime MUST derive the maximum idle time for the conversation by subtracting the current time from the value of the **JMSExpiration** property, unless the **JMSExpiration** property value is zero, in which case the maximum idle time is unlimited.

The SCA runtime MUST consider operations invoked on or by other parties to be outside of a conversation with a given party, and MUST use different conversation identifiers if those operations are conversational.

Continuing a conversation

When creating messages for subsequent operations between the sender and receiver that are part of this conversation, the SCA runtime MUST include the ***scaConversationId*** user property in the JMS message, set to the conversation identifier. The SCA runtime MAY also include an updated value of the ***scaConversationMaxIdleTime*** property. Once a conversation has been started, the SCA runtime MUST use the initial value of the ***scaCallbackDestination*** user property for all messages in the conversation, and MUST ignore the value of the ***scaCallbackDestination*** user property in subsequent messages in the same conversation.

The SCA runtime MUST deal with messages received either containing a conversation identifier that does not correspond to a started conversation, or containing the ***scaConversationStart*** user property with a conversation identifier that matches an active conversation, by raising an error, and MUST NOT deliver such messages.

Ending a conversation

When an operation is invoked by either party that is marked as “***endsConversation***”, or the maximum idle time is exceeded, then the SCA runtime MUST discard the conversation identifier and associated context after the operation has been processed. The idle time is defined as the amount of time since the SCA runtime last completed processing of an operation that is part of the conversation. There may be times when one party ends the conversation before the other does. In that case if one party does invoke an operation on the other, the SCA runtime MUST NOT deliver the message and SHOULD raise an error.

The SCA runtime MAY reuse conversation identifiers. In particular, the SCA runtime does not have to guarantee unique conversation identifiers and does not have to be able to identify an ended conversation indefinitely, although it MAY do so for some period after the conversation ends. Due to the long-running nature of conversations, the SCA runtime SHOULD ensure conversation context is available across server restarts, although it MAY choose to treat a server restart as implicitly ending the conversation.