Service Component Architecture EJB Session Bean Binding Specification Version 1.1

Committee Draft 01

2 September 2009

Specification URIs:
This Version:
http://docs.oasis-open.org/opencsa/sca-j/sca-ejbbinding-1.1-spec-cd01.html
http://docs.oasis-open.org/opencsa/sca-j/sca-ejbbinding-1.1-spec-cd01.doc
http://docs.oasis-open.org/opencsa/sca-j/sca-ejbbinding-1.1-spec-cd01.pdf (Authoritative)

Previous Version:
N/A

Latest Version:
http://docs.oasis-open.org/opencsa/sca-j/sca-ejbbinding-1.1-spec.html
http://docs.oasis-open.org/opencsa/sca-j/sca-ejbbinding-1.1-spec.doc
http://docs.oasis-open.org/opencsa/sca-j/sca-ejbbinding-1.1-spec.pdf (Authoritative)

Technical Committee:
OASIS Service Component Architecture / J (SCA-J) TC

Chair(s):
David Booz, IBM
Mark Combellack, Avaya

Editor(s):
David Booz, IBM
Anish Karmarkar, Oracle

Related work:

This specification replaces or supercedes:

- Service Component Architecture EJB Session Bean Binding Specification Version 1.00, February 22 2007

This specification is related to:

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

Declared XML Namespace(s):

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

Abstract:

This document explains the SCA EJB session bean binding. It describes how to integrate a previously deployed session bean into an SCA assembly, and how to expose SCA services to clients which use the EJB programming model.

Copyright © OASIS® 2005, 2009. All Rights Reserved.

Status:

This document was last revised or approved by the OASIS Service Component Architecture / J (SCA-J) 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-j/.

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-j/ipr.php.

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

Notices

Copyright © OASIS® 2005, 2009. All Rights Reserved.

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 name "OASIS" is a 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.

Table of Contents

Copyright © OASIS® 2005, 2009. All Rights Reserved.

1 Introduction

EJB session beans are a common technology used to implement business services. The ability to integrate SCA with session bean based services is useful because it preserves the investment incurred during the creation of those business services, while enabling the enterprise to embrace the newer SCA technology in incremental steps. The simplest form of integration is to simply enable SCA components to invoke session beans as SCA services. There is also a need to expose SCA services such that they are consumable by programmers skilled in the EJB programming model. This enables existing session bean assets to be enhanced to exploit newly deployed SCA services without the EJB programmers having to learn a new programming model.

This document explains the EJB SCA binding. This proposal describes how to integrate a previously deployed stateless session bean into an SCA assembly, and how to expose SCA services to clients which use the EJB programming model.

The EJB Session Bean binding enables:

- SCA developers to treat previously deployed stateless session beans as SCA services, by wiring them into an SCA assembly (SCA reference).
- SCA service deployers to expose a SCA service as a stateless session bean for consumption by Java EE applications.

Stateful session beans are out of scope for this specification. The terms 'session bean' and 'stateless session bean' are interchangeable for the purpose of this specification.

The use of EJBs and EJB modules as SCA component implementations is beyond the scope of this specification and is described in the Java EE integration specification [SCAJEE]. The following diagram shows the use of the EJB SCA binding on both SCA services and references.

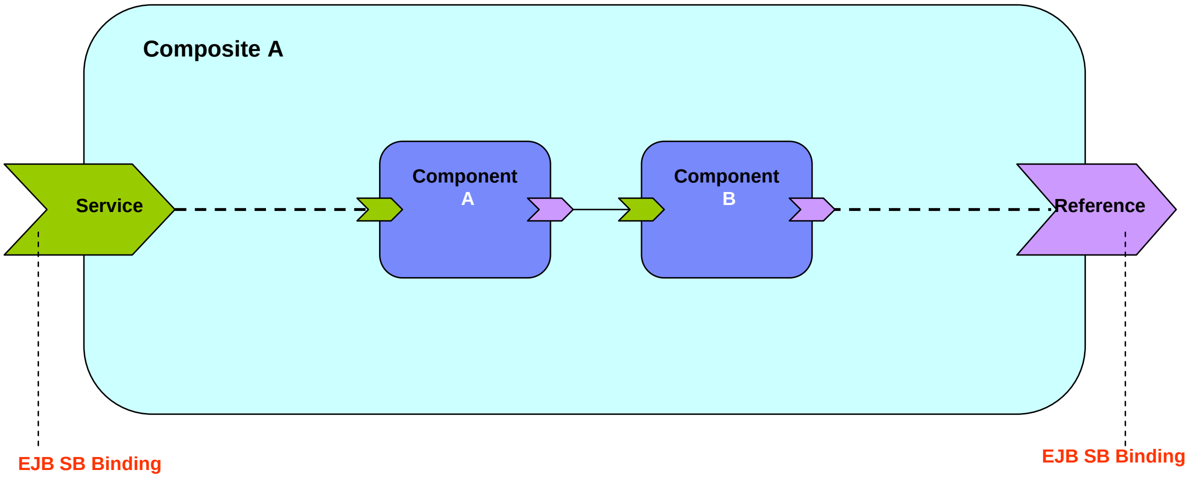


Figure 1: EJB Binding used on SCA Services and References

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]**.

Copyright © OASIS® 2005, 2009. All Rights Reserved.

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.

[SCAJEE] SCA Java EE Implementation Specification,
http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications

[EJB] Enterprise JavaBeans Specification,
http://java.sun.com/products/ejb/docs.html

[CORBA] CORBA Naming Service Specification,
http://www.omg.org/docs/formal/04-10-03.pdf

[ASSEMBLY] OASIS Committee Draft 03, SCA Assembly Model Specification Version 1.1, March 2009.
http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec-cd03.pdf

[JAVACAA] OASIS Committee Draft 03, Service Component Architecture SCA-J Common Annotations and APIs Specification Version 1.1, May 2009
http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec-cd03.pdf

[POLICY] OASIS Committee Draft 02, SCA Policy Framework Specification Version 1.1, February 2009
http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-spec-cd02.pdf

1.3 Non-Normative References

TBD TBD

Copyright © OASIS® 2005, 2009. All Rights Reserved.

2 Session bean binding schema

The EJB session bean binding element is defined by the following pseudo-schema.

```
<binding.ejb homeInterface="NCName"?
             ejb-link-name="string"?
             ejb-version="EJB2 or EJB3"?
             name="NCName"?
             policySets="sca:listOfQNames"?
             requires="sca:listOfQNames"?
             uri="anyURI"?>
    <wireFormat ... />?
    <operationSelector ... />?
</binding.ejb>
```

- ***/binding.ejb/@homeInterface : NCName (0..1)*** - The homeInterface attribute of the EJB binding is the session bean's home interface, and is used when exposing SCA services as EJB 2.x session beans. For <binding.ejb/>, if @ejb-version="EJB2", then @homeInterface MUST be specified and MUST have a value that is the fully qualified package name of the Java interface class of the EJB's home interface. [BSB20001]

- ***/binding.ejb/@ejb-link-name : string (0..1)*** - The ejb-link-name attribute provides a means for integrating EJB reference resolution with SCA. When used on a binding for an SCA reference, it allows a SCA client to bind to an EJB that is packaged in the same Java EE EAR file as the SCA client. When used on an SCA service binding, it exposes an <ejb-link/> target for Java EE clients that want to use Java EE assembly to wire to the SCA service. This attribute is functionally equivalent to using the <ejb-link/> subelement of the <ejb-ref/> element in an EJB deployment descriptor. The value of this attribute is supplied by an application assembler, and is in the form as specified by the Java EE specification [SCAJEE] (i.e. <jar-name>#<ejb-name>). When <binding.ejb/> applies to an SCA reference, if @ejb-link-name attribute is specified it MUST contain the value of an <ejb-link/> target packaged within the same Java EE EAR file as the SCA component containing the SCA reference. [BSB20002]
  When <binding.ejb/> applies to an SCA service, if @ejb-link-name attribute is specified, it MUST contain a value in the form "<jar-name>#<ejb-name>" which MUST be unique amongst the <ejb-link/> targets contained within the same Java EE EAR file as the SCA component containing the SCA service. [BSB20003]

- ***/binding.ejb/@ejb-version : VersionValue (0..1)*** – The ejb-version attribute is used to indicate the EJB client view exposed by the EJB binding when used on an SCA service. This attribute has no meaning when used on an SCA reference. The value 'EJB2' indicates the desire to expose an EJB 2.x client view. The value 'EJB3' indicates the desire to expose an EJB 3.0 client view. The default value is 'EJB3'. When <binding.ejb/> applies to an SCA service and the @ejb-version attribute is set to 'EJB2', the SCA Runtime MUST support invocation of the SCA service using the EJB 2.x client view as specified in the Java EE specification [SCAJEE]. [BSB20004] When <binding.ejb/> applies to an SCA service and the @ejb-version attribute is set to 'EJB3', the SCA Runtime MUST support invocation of the SCA service using the EJB 3.x client view as specified in the Java EE specification [SCAJEE]. [BSB20005]

- ***/binding.ejb/@name : NCName (0..1)*** – As defined in the SCA Assembly Specification [ASSEMBLY]

- ***/binding.ejb/@requires : QName (0..1)*** – A list of policy intents as defined in the SCA Policy Framework Specification [POLICY]

- ***/binding.ejb/@policySets : QName (0..1)*** – A list of policy sets as defined in the SCA Policy Framework Specification [POLICY]

The base SCA binding schema provides an attribute called **uri**, that is used to denote the URI of an endpoint. In the context of the SCA EJB binding, the **uri** attribute is defined as follows:

- ***/binding.ejb/@uri : anyURI (0..1)*** – Specifies the URI of a session bean endpoint. For EJB 2.x, this is the endpoint of the session home. For EJB 3.x, this is the endpoint of the session bean. The value of the @uri attribute MUST take the form of an Object URL as specified in the CORBA Services specification [CORBA]. [BSB20006] This is a standard URI form for referring to remotable CORBA objects. Briefly, the corbaname URI format looks like this:

  - corbaname:iiop:<hostName>:<port>/<key string>#<path to home>

  Typically, a corbaname URI doesn't include all these components. The following example shows a corbaname URI that uses the default ORB configuration to find an EJB home at ejb/MyHome in the JNDI directory:

  - corbaname:rir:#ejb/MyHome

  Other forms of URI specification are admissible when interoperability is of no concern.

/binding.ejb/wireFormat – As defined in the SCA Assembly Specification [ASSEMBLY]. This specification does not define any new wireFormat elements.

- ***/binding.ejb/operationSelector*** – As defined in the SCA Assembly Specification [ASSEMBLY]. This specification does not define any new operationSelector elements.

When <binding.ejb/> applies to an SCA reference, the @uri and @ejb-link-name attributes MUST NOT be specified together in the same binding configuration. [BSB20007]

The <binding.ejb/> element MUST conform to the XML schema defined in the sca-binding-ejb.xsd. [BSB20008]

The implementation MUST reject a SCA Session Bean Binding XML Document that is not conformant per Section 9.1. [BSB20009]

2.1 Additional binding configuration data

SCA runtime implementations can provide additional metadata that is associated with an EJB binding. This is done by providing extension points in the schema; refer to Appendix B: EJB Binding Schema for the locations of these extension points.

Copyright © OASIS® 2005, 2009. All Rights Reserved.

3 Interface Mapping

When used with the EJB binding, an SCA runtime MUST ensure that an SCA service or reference interface is compatible with a session bean interface, according to the rules defined in the section “Compatibility of Interfaces used for SCA Services & References with EJB Session Bean Interfaces”. [BSB30001]

3.1 Compatibility of Interfaces used for SCA Services & References with EJB Session Bean Interfaces

This section defines the compatibility of the interface used by an SCA reference with the interface provided by an EJB, when the SCA reference is wired to the EJB. It also defines the compatibility of the interface used by an EJB reference with the interface of an SCA service when the EJB reference is wired to the SCA service.

If an SCA reference is wired to an EJB remote session bean interface, the SCA reference interface is compatible if it is remotable.
If an SCA reference is wired to an EJB local session bean interface, the SCA reference interface is compatible if it is local.

The interface used by an SCA reference which is wired to a session bean is a compatible subset [ASSEMBLY] of the interface used by the session bean. In particular, the interface used by the SCA reference can omit any methods inherited from EJBObject or EJBLocalObject that appear in the session bean interface.

The interface used by an SCA service which is wired to by an EJB reference is a compatible superset [ASSEMBLY] of the interface used by the EJB reference. In particular, the interface used by the SCA service can omit any methods inherited from EJBObject or EJBLocalObject that appear in the EJB reference interface.

Compatibility for an individual method is defined by the SCA Assembly Model Specification [ASSEMBLY], and can be stated simply as compatibility of the signature. That is, the method name, input types, output types, and faults are identical.

The interface used by an SCA service or reference can be an SCA business interface or an EJB 3.0 remote or local interface.

3.2 EJBObject and EJBLocalObject Interfaces

The interfaces exposed from EJB 2.X beans inherit from either EJBObject or EJBLocalObject. EJBObject and EJBLocalObject contain methods directed toward the management of bean instances, meaning that the exposed 2.X interfaces mix business and infrastructure methods in a way that makes them poorly suited for use as an SCA business interface. However, EJB 2.X beans developed using the “Business Interface Pattern” will already have an interface that is a suitable SCA business interface. An EJB 2.x session bean interface itself MUST NOT be used as the interface of an SCA reference. [BSB30002]

Section 6.1 describes the behavior associated with each inherited method when <binding.ejb/> is used on an SCA service.

4 SCA Reference Binding

When used on an SCA reference, the EJB binding specifies the means for connecting an SCA component to a previously deployed or co-deployed session bean.

The SCA reference interface used with the EJB binding can be either a remote or local interface. SCA deployment logic and the binding implementation will introspect the SCA reference interface class to determine whether it is local or remote. If an SCA component needs to access both the local and remote interface of a session bean, then this can be modeled in SCA assembly through two SCA references, one with a local interface and one with a remote interface.

The following example shows a reference binding using a corbaname URI:

```
<reference name="CandidateCheck">
   <interface.java interface="com.app.jobbank.CandidateCheck"/>
   <binding.ejb uri="corbaname:rir:#ejb/CandidateCheckHome"/>
</reference >
```

The specific **uri** would be supplied prior to the completion of deployment.

The following example is a reference binding using an ejb-link.

```
<reference name="CandidateCheck">
 <interface.java interface="com.app.jobbank.CandidateChk"/>
 <binding.ejb ejb-link-name="candidateEJB.jar#CandidateChk"/>
</reference >
```

4.1 Exception Handling

Exception handling for interactions with session beans has been specified in chapter 14 of the EJB 3 specification [EJB] and in Chapter 18 of the EJB 2.1 specification [EJB]. The EJB [EJB] specifications define non-business exceptions that can be thrown to the EJB client. When <binding.ejb/> applies to an SCA reference, the SCA Runtime MUST wrap non-business exceptions in a ServiceRuntimeException that is thrown to the client [JAVACAA]. [BSB40001]

5 Packaging

There is no requirement to package the session bean home interface or client stubs with an SCA component that uses the Session bean binding. The EJB Session Bean binding implementation can dynamically lookup, create and invoke the bean without the usual EJB client classes.

Copyright © OASIS® 2005, 2009. All Rights Reserved.

6 SCA Service Binding

When used on an SCA service, the EJB SCA binding causes the SCA service to be exposed as a session bean. This enables a client that is using the EJB programming model to call the SCA service using its native programming model.

The ***/binding.ejb/@homeInterface*** attribute is used to indicate the Session Home interface that an EJB client will use to bootstrap itself with the SCA service, just as it would with any other session bean. When <binding.ejb/> applies to an SCA service, the Java interface class specified on the @homeInterface attribute MUST have one create method [EJB]. [BSB60001]

The following is an example of a service using the EJB binding.

```
<service name="JobBank">
   <interface.java interface="com.app.jobbank.JobBankService"/>
   <binding.ejb
      uri="corbaname:rir:#ejb/JobBankServiceHome"
      homeInterface="com.app.jobbank.JobBankServiceHome"
      ejb-link-name="jobbankEJB.jar#JobBankComponent"/>
</service>
```

A corresponding local home interface com.app.jobbank.JobBankServiceHome looks like this:

```
package com.app.jobbank;

import javax.ejb.CreateException;
import javax.ejb.EJBLocalHome;

public interface JobBankServiceHome extends EJBLocalHome {
    JobBankService create()  throws CreateException;
}
```

Similarly, the remote home interface can be formulated by extending javax.ejb.EJBHome and making sure to declare a RemoteException:

```
package com.app.jobbank;

import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;

public interface JobBankServiceHome extends EJBHome {
    JobBankService create()  throws CreateException, RemoteException;
}
```

Copyright © OASIS® 2005, 2009. All Rights Reserved.

In the corbaname used in this example, the first part of the URI (up to the #) would logically be supplied by the target deployment environment. See the SCA Assembly Model Specification [ASSEMBLY] for a discussion of base URIs provided by an SCA domain configuration. The remainder of the name would be provided prior to completion of deployment. The example above shows the URI that a client would use after deployment. Prior to deployment, an assembler or developer can specify only the last portion of the URI (i.e. everything following the #).

The SCA service interface used with the EJB binding can be either a remote or local interface. SCA deployment logic and the binding implementation will introspect the interface class to determine whether it is local or remote. If an SCA component needs to be exposed as both a local and remote session bean, this can be modeled in SCA through two SCA services, one with the local interface and one with the remote interface.

When used on an SCA service binding, **ejb-link-name** and **uri** are NOT mutually exclusive. They each provide a means for wiring to the SCA service depending on the locality of the client EJB reference. For example, an SCA service packaged with an JEE EJB application could be exposed for consumption by local EJB clients (using the ejb-link-name element) and remote EJB clients (using the uri).

From the perspective of an EJB client (local and remote), SCA services that are exposed as session beans are not distinguishable from ordinary session beans. When <binding.ejb/> applies to an SCA service and @ejb-version is set to ‘EJB2’, the binding implementation MUST implement the methods from the EJBObject or EJBLocalObject interface. [BSB60002]

Specifically, this means that a local client will be able to reference the SCA service as a session bean using ejb-(local)-ref declarations in the appropriate locations and by issuing JNDI lookups or relying on dependency injection mechanisms. If the SCA service is exposed as EJB 2.x session bean, by virtue of a home interface specification, the client needs to be aware of the EJB 2.x home interface contract.

Similarly remote EJB clients are expected to be able to consume SCA services that are exposed as session beans just as they are able to consume ordinary session beans.

6.1 Handling methods from EJBObject and EJBLocalObject

This section describes the SCA specific behavior of the methods that EJB 2.X service bindings inherit from the EJBObject and EJBLocalObject interfaces.

Method	Behavior
isIdentical	Tests whether the SCA component, which the binding exposes, is the same instance as the one exposed by the specified object.
getEJBHome getEJBLocalHome	Returns an implementation of the interface specified as ***/binding.ejb/@homeInterface.*** The instance can be used to create or remove bean instances.

7 Callbacks

The SCA Assembly Model Specification [ASSEMBLY] defines the callback feature which enables asynchronous interactions between two SCA components. This specification does not support the callback feature. However, implementations can choose to support the callback feature, in conjunction with this binding, by creating extensions to this specification.

8 EJB Session Bean Binding bindingType

The bindingType for the Session Bean binding is defined as follows:

```
    <bindingType type="binding.ejb" alwaysProvides="EJB"/>
```

The EJB intent is defined in the SCA Policy Specification [POLICY] document in the section entitled "Miscellaneous Intents".

9 Conformance

The XML schema pointed to by the RDDL 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 EJB Session Bean Binding XML Document

b) SCA Runtime

9.1 SCA EJB Session Bean Binding XML Document

An SCA EJB Session Bean Binding XML document is an SCA Composite Document, or an SCA ComponentType Document, as defined by the SCA Assembly Model Specification [ASSEMBLY], that uses the <binding.ejb> element.

An SCA EJB Session Bean Binding XML document MUST be a conformant SCA Composite Document or a SCA ComponentType Document, as defined by the SCA Assembly Model Specification [ASSEMBLY], and MUST comply with all statements in Appendix C: Conformance Items related to elements and attributes in an SCA EJB Session Bean Binding XML document, notably all "MUST" statements have to be implemented.

9.2 SCA Runtime

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

1. The implementation MUST comply with all statements in Appendix C: Conformance Items related to an SCA Runtime.
2. The implementation MUST conform to the SCA Assembly Model Specification Version 1.1 [ASSEMBLY] and to the SCA Policy Framework Version 1.1 [POLICY].

Copyright © OASIS® 2005, 2009. All Rights Reserved.

A. Use cases

The following use cases provide some examples of the usage of the SCA EJB Session Bean binding.

A.1 Consuming an Existing EJB SOA Service

An SCA service is developed that needs to call a business service which is already deployed and running in a Java EE server. The SCA service will be deployed into an SCA runtime somewhere in the enterprise that is not necessarily a Java EE runtime. The business service was implemented as a session bean. The SCA component defines a SCA reference to the business service, and the deployer attaches an EJB binding to the SCA reference. In this use case, the EJB remote interface is the business interface.

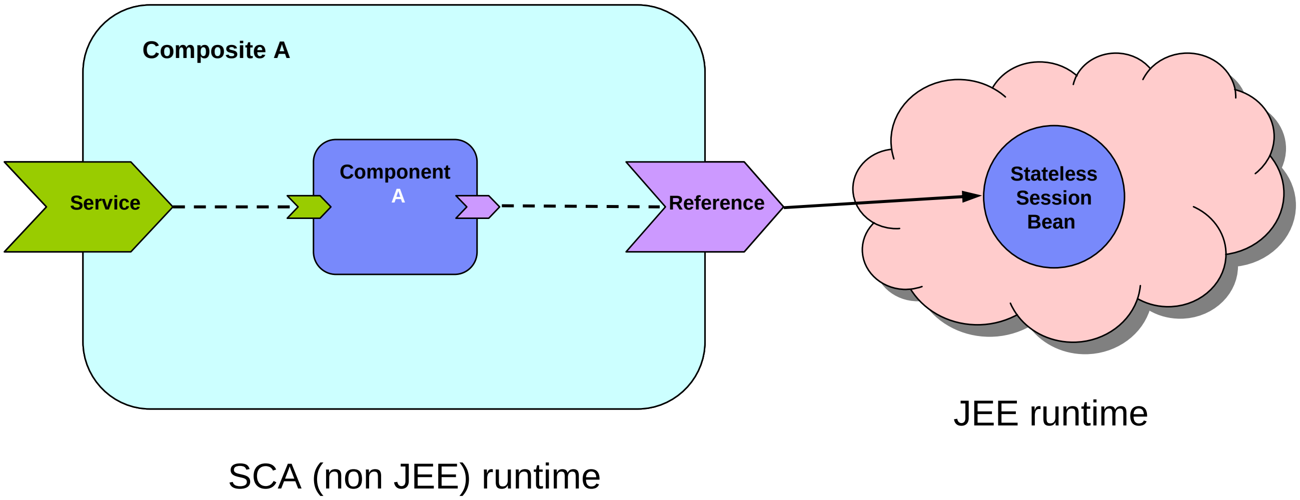


Figure 2: SCA Reference invoking EJB Session Bean

The reference in the deployed sca.composite file looks like this:

```
<reference name="CandidateCheck">
  <interface.java interface="com.app.jobbank.CandidateChk"/>
  <binding.ejb uri="corbaname:rir:#ejb/CandidateChkHome"/>
</reference >
```

A.2 Exposing an SCA Service with an EJB SCA Binding

An SCA service is developed that will be called from a Java EE environment. The Java EE programmer doesn't know the SCA programming model and therefore wants to use the Java EE programming model that he knows in order to invoke the SCA service (i.e. new initialContext(), nc.lookup(), etc.). In this case, the SCA service has to be deployed into a runtime that is capable of supporting the EJB binding. Note that deployment of this SCA service can result in the generation and deployment of a session bean, along with its home interface. This aspect is significantly different from the previous use case.

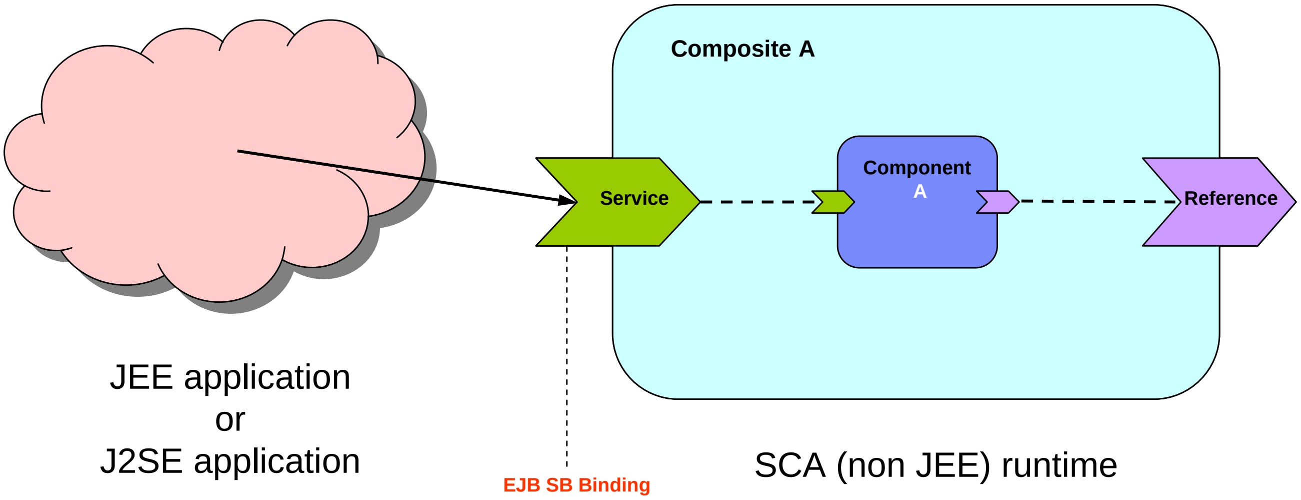


Figure 3: SCA Service accessed as an EJB Session Bean

Since the client will use the standard Java EE programming model, the client needs to know the home interface of the SCA service. The service in the SCA composite file will look like this:

```
<service name="CompanyInfo">
 <interface.java interface="com.app.jobbank.CompanyInfo"/>
 <binding.ejb uri="corbaname:rir:#ejb/CompanyInfoHome"
   homeInterface="com.app.jobbank.CompanyInfoHome"
   ejb-version="EJB2"/>
 <reference>CompanyInfoComponent/CompanyInfo</reference>
</service>
```

The client code as per the standard Java EE programming model looks like this:

```
Context initialContext = new InitialContext(env);
CompanyInfoHome companyInfoHome= (CompanyInfoHome)
     initialContext.lookup("corbaname:rir:#ejb/CompanyInfoHome");

CompanyInfo companyInfo = companyInfoHome.create();
companyInfo.getCompanyInfo("ACME Corp");
```

A.3 Consuming Existing Local EJB SOA Services

This use case is similar to the use case in section 3.1, except that the SCA service is going to be deployed into a Java EE capable JVM, and it is the same JVM as the EJB service. In this use case, the EJB's local interface is used as the business interface.

Note that the SCA client could also use the EJB remote interface. If an SCA component wanted to access both the local and remote interface, then it would declare 2 SCA references (one with the local interface, one with the remote interface).

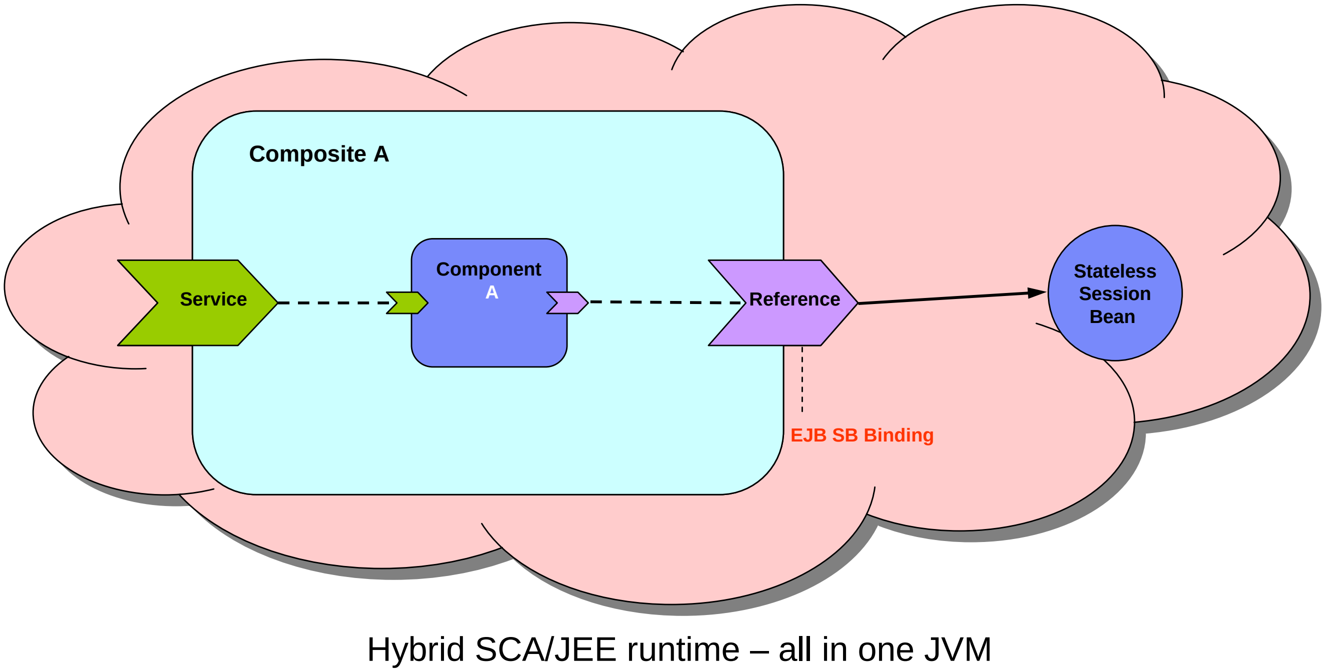


Figure 4: SCA reference consuming a Local EJB service

The example below shows the usage of a local interface in the reference definition.

```
<reference name="CandidateCheck">
 <interface.java interface="com.app.jobbank.CandidateCheckLocal"/>
 <binding.ejb
      uri="corbaname:rir:#ejb/CandidateCheckHome"/>
</reference>
```

A.4 Exposing an SCA Service with a Local SLSB SCA Binding

This use case is similar to the use case in section 3.2, except that the SCA service is going to be deployed into the same JVM as the client. This use case allows for the possibility that the SCA service is exposed as a local EJB interface. Note that deployment of this SCA service will effectively result in the generation and deployment of a session bean with a local interface and a local home interface.

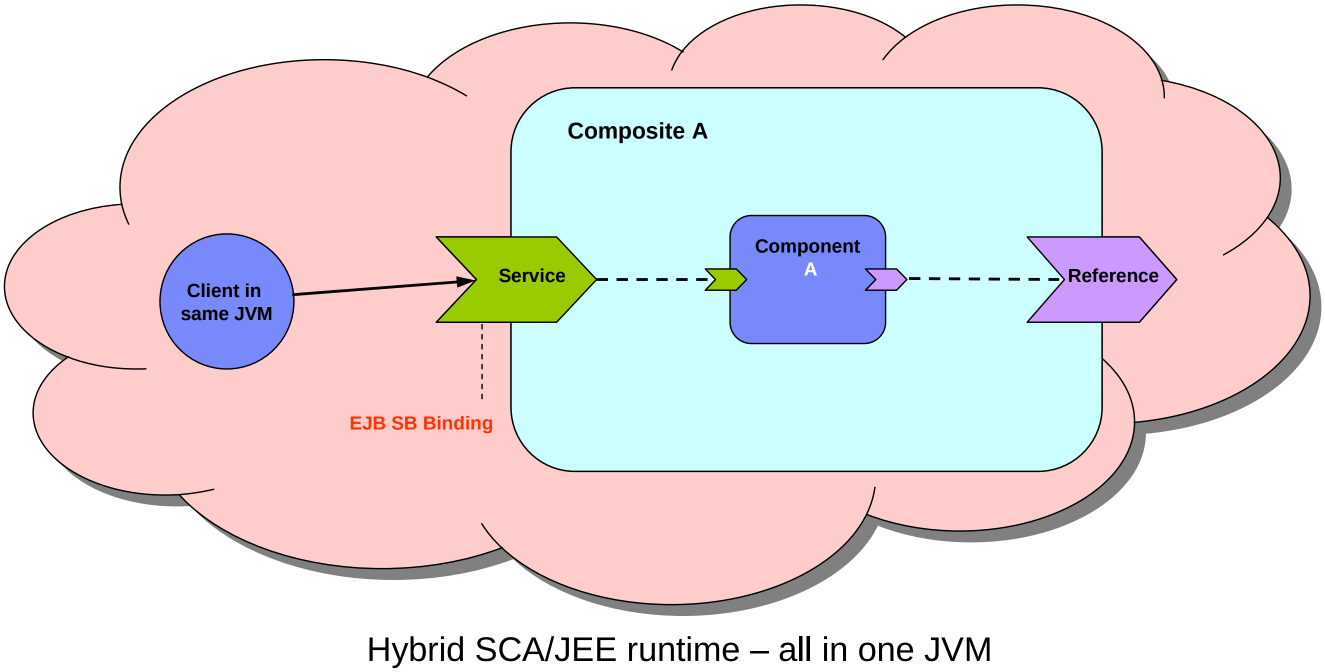


Figure 5: SCA Service exposed as a Local session bean

The following is an example:

```
<service name="CompanyInfo">
 <interface.java interface="com.app.jobbank.CompanyInfoLocal"/>
 <binding.ejb uri="corbaname:rir#ejb/CompanyInfoHome"
     homeInterface="com.app.jobbank.CompanyInfoLocalHome"/>
 <reference>CompanyInfoComponent/CompanyInfo</reference>
</service>
```

A.5 Consuming an EJB Service inside a Java EE EAR file

This use case is similar to sections 3.1 and 3.3, except that the SCA service is going to be packaged inside a Java EE EAR file. By packaging it in this way, the SCA reference binding can be configured as if it were an <ejb-ref> with the <ejb-link> subelement.

The following is an example of the SCA reference binding.

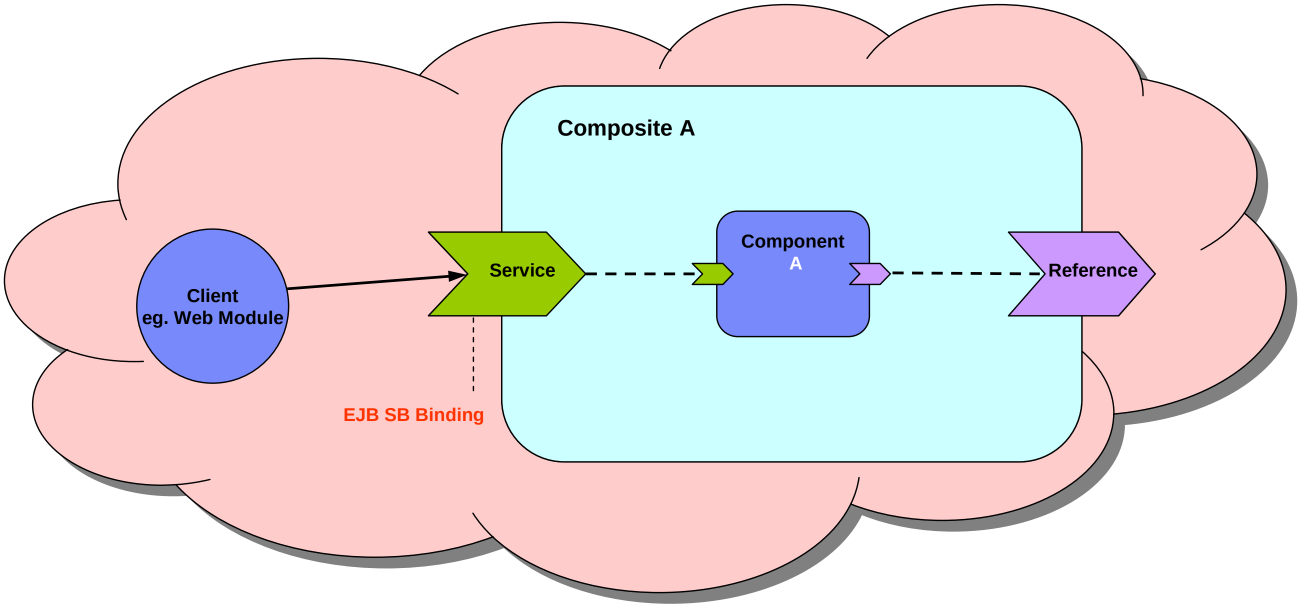
```
<reference name="CandidateCheck">
 <interface.java interface="com.app.jobbank.CandidateChk"/>
 <binding.ejb ejb-link-name="candidateEJB.jar#CandidateChk"/>
</reference >
```

The following is an <ejb-ref/> snippet that is functionally equivalent to the SCA reference above.

```
<ejb-ref>
 <ejb-ref-name>CandidateCheck</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <home>com.app.jobbank.CandidateChkHome</home>
 <remote>com.app.jobbank.CandidateChk</remote>
 <ejb-link>candidateEJB.jar#CandidateChk</ejb-link>
</ejb-ref>
```

A.6 Exposing an SCA Service inside a Java EE EAR file

This use case is similar to sections 3.2 and 3.4, except that the SCA service is going to be deployed inside a Java EE EAR file so that it can be referenced by an EJB client, using the EJB assembly model.



Caller and SCA Composite within one EAR file

Figure 6: SCA Service with client within one EAR file

The following is an example of the SCA service binding.

```
<service name="CompanyInfo">
 <interface.java interface="com.app.jobbank.CompanyInfo"/>
 <binding.ejb
   homeInterface="com.app.jobbank.CompanyInfoHome"
   ejb-link-name="companyInfoEJB.jar#CompanyInfoComponent"/>
 <reference>CompanyInfoComponent/CompanyInfo</reference>
</service>
```

The following is an example of an EJB deployment descriptor created by the client that is wired to the SCA Service binding.

```
<ejb-ref>
 <ejb-ref-name>ejb/CompanyInfo</ejb-ref-name>
 <ejb-ref-type>Session</ejb-ref-type>
 <home>com.app.jobbank.CompanyInfoHome</home>
 <remote>com.app.jobbank.CompanyInfo</remote>
 <ejb-link>companyInfoEJB.jar#CompanyInfoComponent</ejb-link>
</ejb-ref>
```

Note: There is a variant of this use case that needs to be considered. If the SCA service is in the same EJB module as the client, then the ejb-link specified by the client does not have to include the EJB module jar name.

B. EJB binding schema

```
<?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"
        xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200903"
        targetNamespace="http://docs.oasis-open.org/ns/opencsa/sca/200903"
        elementFormDefault="qualified">

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

    <element name="binding.ejb" type="sca:EJBSessionBeanBinding"
             substitutionGroup="sca:binding" />

    <simpleType name="VersionValue">
        <restriction base="string">
            <enumeration value="EJB2"/>
            <enumeration value="EJB3"/>
        </restriction>
    </simpleType>

    <complexType name="EJBSessionBeanBinding">
        <complexContent>
            <extension base="sca:Binding">
                <sequence>
                    <any namespace="##other" processContents="lax"
                         minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
                <attribute name="homeInterface" type="NCName"
                        use="optional"/>
                <attribute name="ejb-link-name" type="string"
                        use="optional"/>
                <attribute name="ejb-version" type="sca:VersionValue"
                        use="optional" default="EJB3"/>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

C. Conformance Items

This section contains a list of conformance items for the SCA EJB Session Bean Binding specification.

Conformance ID	Description
[BSB20001]	For <binding.ejb/>, if @ejb-version="EJB2", then @homeInterface MUST be specified and MUST have a value that is the fully qualified package name of the Java interface class of the EJB's home interface.
[BSB20002]	When <binding.ejb/> applies to an SCA reference, if @ejb-link-name attribute is specified it MUST contain the value of an <ejb-link/> target packaged within the same Java EE EAR file as the SCA component containing the SCA reference.
[BSB20003]	When <binding.ejb/> applies to an SCA service, if @ejb-link-name attribute is specified, it MUST contain a value in the form "<jar-name>#<ejb-name>" which MUST be unique amongst the <ejb-link/> targets contained within the same Java EE EAR file as the SCA component containing the SCA service.
[BSB20004]	When <binding.ejb/> applies to an SCA service and the @ejb-version attribute is set to ‘EJB2’, the SCA Runtime MUST support invocation of the SCA service using the EJB 2.x client view as specified in the Java EE specification [SCAJEE].
[BSB20005]	When <binding.ejb/> applies to an SCA service and the @ejb-version attribute is set to ‘EJB3’, the SCA Runtime MUST support invocation of the SCA service using the EJB 3.x client view as specified in the Java EE specification [SCAJEE].
[BSB20006]	The value of the @uri attribute MUST take the form of an Object URL as specified in the CORBA Services specification [CORBA].
[BSB20007]	When <binding.ejb/> applies to an SCA reference, the @uri and @ejb-link-name attributes MUST NOT be specified together in the same binding configuration.
[BSB20008]	The <binding.ejb/> element MUST conform to the XML schema defined in the sca-binding-ejb.xsd.
[BSB20009]	The implementation MUST reject a SCA Session Bean Binding XML Document that is not conformant per Section 9.1.
[BSB30001]	When used with the EJB binding, an SCA runtime MUST ensure that an SCA service or reference interface is compatible with a session bean interface, according to the rules defined in the section “Compatibility of Interfaces used for SCA Services & References with EJB Session Bean Interfaces”.
[BSB30002]	An EJB 2.x session bean interface itself MUST NOT be used as the interface of an SCA reference.
[BSB40001]	The EJB [EJB] specifications define non-business exceptions that can be thrown to the EJB client. When <binding.ejb/> applies to an SCA reference, the SCA Runtime MUST wrap non-business exceptions in a ServiceRuntimeException that is thrown to the client [JAVACAA].

[BSB60001]	When <binding.ejb/> applies to an SCA service, the Java interface class specified on the @homeInterface attribute MUST have one create method [EJB].
[BSB60002]	When <binding.ejb/> applies to an SCA service and @ejb-version is set to ‘EJB2’, the binding implementation MUST implement the methods from the EJBObject or EJBLocalObject interface.

Copyright © OASIS® 2005, 2009. All Rights Reserved.

D. Acknowledgements

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

Participants:

Participant Name	Affiliation
Bryan Aupperle	IBM
Ron Barack	SAP AG
Michael Beisiegel	IBM
Henning Blohm	SAP AG
David Booz	IBM
Martin Chapman	Oracle Corporation
Graham Charters	IBM
Shih-Chang Chen	Oracle Corporation
Chris Cheng	Primeton Technologies, Inc.
Vamsavardhana Reddy Chillakuru	IBM
Roberto Chinnici	Sun Microsystems
Pyounguk Cho	Oracle Corporation
Eric Clairambault	IBM
Mark Combellack	Avaya, Inc.
Jean-Sebastien Delfino	IBM
Mike Edwards	IBM
Raymond Feng	IBM
Bo Ji	Primeton Technologies, Inc.
Uday Joshi	Oracle Corporation
Anish Karmarkar	Oracle Corporation
Michael Keith	Oracle Corporation
Rainer Kerth	SAP AG
Meeraj Kunnumpurath	Individual
Simon Laws	IBM
Yang Lei	IBM
Mark Little	Red Hat
Ashok Malhotra	Oracle Corporation
Jim Marino	Individual
Jeff Mischkinsky	Oracle Corporation
Sriram Narasimhan	TIBCO Software Inc.
Simon Nash	Individual
Sanjay Patil	SAP AG
Plamen Pavlov	SAP AG
Peter Peshev	SAP AG
Ramkumar Ramalingam	IBM
Luciano Resende	IBM
Michael Rowley	Active Endpoints, Inc.
Vladimir Savchenko	SAP AG
Pradeep Simha	TIBCO Software Inc.
Raghav Srinivasan	Oracle Corporation
Scott Vorthmann	TIBCO Software Inc.

Copyright © OASIS® 2005, 2009. All Rights Reserved.

Feng Wang	Primeton Technologies, Inc.
Robin Yang	Primeton Technologies, Inc.

E. Revision History

[optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
1	2007-09-26	Anish Karmarkar	Applied the OASIS template + related changes to the Submission
2	2007-10-04	David Booz	Issue 5: Ending a conversation should invoke the remove method of EJBObject or EJBLocalObject.
wd02	2007-11-02	David Booz	Applied OSOA Errata
wd03	2009-06-04	David Booz	Editorial upgrade of namespaces, attribute descriptions, etc Applied Issues 86, 140
wd04	2009-07-20	David Booz	Applied 24, 122, 118
wd05	2009-08-14	David Booz	Applied 107, 170
cd01	2009-09-02	David Booz	Creation of CD01