

Java EE Integration

Java EE Integration

This page is used to track the status of implementation of SCA Java EE Integration Spec v1.0 [1] and Tuscany integration with Geronimo.

Latest update

The latest Geronimo Tuscany plugin we have can be installed on Geronimo v2.1.3. The plugin supports SCA reference and property injection in JSP, Servlet, Servlet Filters, Listeners, JSF Managed Beans, Session EJB and Message-Driven EJB. For instructions on installing the plugin and running sample applications, follow [this link](#).

Deploying JEE archives with SCA

Deployment Options

- non-enhanced
- enhanced
 - deployable composite
 - application composite
 - contribution explicit deployment
 - contribution implicit deployment

If an archive is not marked as "contribution explicit deployment" or "contribution implicit deployment" it is assumed to be included within, or otherwise reference by, an SCA contribution. See "Archive Location Scenarios" for ways in which JEE archive can be positioned with respect to SCA contribution.

Deployment Scenarios

- 1 WAR non-enhanced
- 2 WAR enhanced
- 3 WAR enhanced, deployable composite, contribution explicit deployment
- 4 SCAZIP contribution
 - 6 WAR enhanced, application composite, contribution implicit deployment
- 5 WAR enhanced, application composite, contribution explicit deployment
- 6 WAR enhanced, application composite, contribution implicit deployment
- *
- 7 EJBJAR non-enhanced
- 8 EJBJAR enhanced
- 9 EJBJAR enhanced, deployable composite, contribution explicit deployment
- 10 SCAZIP contribution
 - 12 EJBJAR enhanced, application composite, contribution implicit deployment
- 11 EJBJAR enhanced, application composite, contribution explicit deployment
- 12 EJBJAR enhanced, application composite, contribution implicit deployment
- *
- 13 EAR non-enhanced
 - WAR non-enhanced
 - JAR non-enhanced
- 14 EAR implied-enhanced
 - WAR enhanced
 - JAR enhanced
- 15 EAR implied-enhanced
 - WAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - JAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
- *
- 16 EAR enhanced
 - WAR non-enhanced
 - JAR non-enhanced
- 17 EAR enhanced
 - WAR enhanced
 - JAR enhanced
- 18 EAR enhanced
 - WAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - JAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
- *
- 19 EAR enhanced, deployable composite, not deployed
 - WAR non-enhanced
 - JAR non-enhanced
- 20 EAR enhanced, deployable composite, not deployed
 - WAR enhanced
 - JAR enhanced
- 21 EAR enhanced, deployable composite, not deployed
 - WAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE

- JAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
- *
 - 22 EAR enhanced, deployable composite, contribution explicit deployment
 - WAR non-enhanced
 - JAR non-enhanced
 - 23 EAR enhanced, deployable composite, contribution explicit deployment
 - WAR enhanced
 - JAR enhanced
 - 24 EAR enhanced, deployable composite, contribution explicit deployment
 - WAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - JAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - *
 - 25 EAR enhanced, application composite, contribution explicit deployment
 - WAR non-enhanced
 - JAR non-enhanced
 - 26 EAR enhanced, application composite, contribution explicit deployment
 - WAR enhanced
 - JAR enhanced
 - 27 EAR enhanced, application composite, contribution explicit deployment
 - WAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - JAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - *
 - 28 EAR enhanced, application composite, contribution implicit deployment
 - WAR non-enhanced
 - JAR non-enhanced
 - 29 EAR enhanced, application composite, contribution implicit deployment
 - WAR enhanced
 - JAR enhanced
 - 30 EAR enhanced, application composite, contribution implicit deployment
 - WAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - JAR enhanced, application composite >>>> IGNORES APPLICATION COMPOSITE
 - 31 SCAJAR contribution
 - References external 1 WAR non-enhanced
 - 32 SCAJAR contribution
 - References external 2 WAR enhanced
 - 33 SCAJAR contribution
 - References external 6 WAR enhanced, application composite
 - 34 NOT USED
 - *
 - 35 SCAJAR contribution
 - References external 13 EAR non-enhanced
 - 36 SCAJAR contribution
 - References external 14 EAR implied-enhanced
 - 37 SCAJAR contribution
 - References external 15 EAR implied-enhanced
 - *
 - 38 SCAJAR contribution
 - References external 16 EAR enhanced
 - 39 SCAJAR contribution
 - References external 17 EAR enhanced
 - 40 SCAJAR contribution
 - References external 18 EAR enhanced
 - 41 SCAJAR contribution
 - References external 29 EAR enhanced, application composite
 - 42 SCAJAR contribution
 - References external 28 EAR enhanced, application composite
 - 43 SCAJAR contribution
 - References external 30 EAR enhanced, application composite
 - *
 - 44 SCAJAR contribution
 - References external 7 EJBjar non-enhanced
 - 45 SCAJAR contribution
 - References external 8 EJBjar enhanced
 - 46 SCAJAR contribution
 - References external 12 EJBjar enhanced, application composite
 - 47 SCAZIP contribution
 - Include 13 EAR non-enhanced
 - 48 SCAZIP contribution
 - Include 14 EAR implied-enhanced
 - 49 SCAZIP contribution
 - Include 15 EAR implied-enhanced
 - *
 - 50 SCAZIP contribution
 - Include 16 EAR enhanced
 - 51 SCAZIP contribution
 - Include 17 EAR enhanced

- 52 SCAZIP contribution
 - Include 18 EAR enhanced
 - *
- 53 SCAZIP contribution
 - Include 28 EAR enhanced, application composite
- 54 SCAZIP contribution
 - Include 29 EAR enhanced, application composite
- 55 SCAZIP contribution
 - Include 30 EAR enhanced, application composite

Scenario Groups

Group	Scenarios	Comments
1	2, 3, 5, 6	
2	8, 9, 11, 12	
3	14, 15, 17, 18, 20, 21	
4	23, 24, 26, 27, 29, 30	
5	1, 7, 13, 16, 19	
6	22, 25, 28	

Archive location scenarios

When using implementation.jee archive="xyz" what does xyz mean? The artifact location described by these scenarios could equally apply to stand alone war and EJB jar files. Section 1 of the spec includes the following statement "This specification defines the integration of SCA and Java EE within the context of a Java EE application, the use of Java EE components as service component implementations, and the deployment of Java EE archives either within or as SCA contributions. "

A - EAR is external resource (external)

Not defined in Spec.

General pattern for accessing external EAR fles.

```
<component name="xyz">
<implementation.jee archive="file://c:/temp/my.ear"/>
</component>
```

Or, when EAR is available in the JEE container

```
<component name="xyz">
<implementation.jee archive="logical ear name"/>
</component>
```

B - EAR is artifact in same contribution (nested)

Spec : The archive attribute specifies a relative path to the Java EE archive that serves as implementation artifact

```
<component name="xyz">
<implementation.jee archive="../my.ear"/>
</component>
```

C - EAR is another contribution (other)

Not defined in Spec.

Not clear how the reference would be formed

```
<component name="xyz">
<implementation.jee archive="contribution name"/>
</component>
```

D - EAR is artifact in another contribution (othernested)

Not defined in Spec.

Again not clear how the reference would be formed but it would seem consistent to exploit the contribution export/import mechanism

```
<component name="xyz">
<implementation.jee archive="contribution name/ear name?"/>
</component>
```

E - EAR is a self contained contribution (contrib)

Spec : If EAR is a contribution @archive attribute can be left unspecified, and the archive defaults to be the archive of the contribution itself.

```
<component name="xyz">
<implementation.jee/>
</component>
```

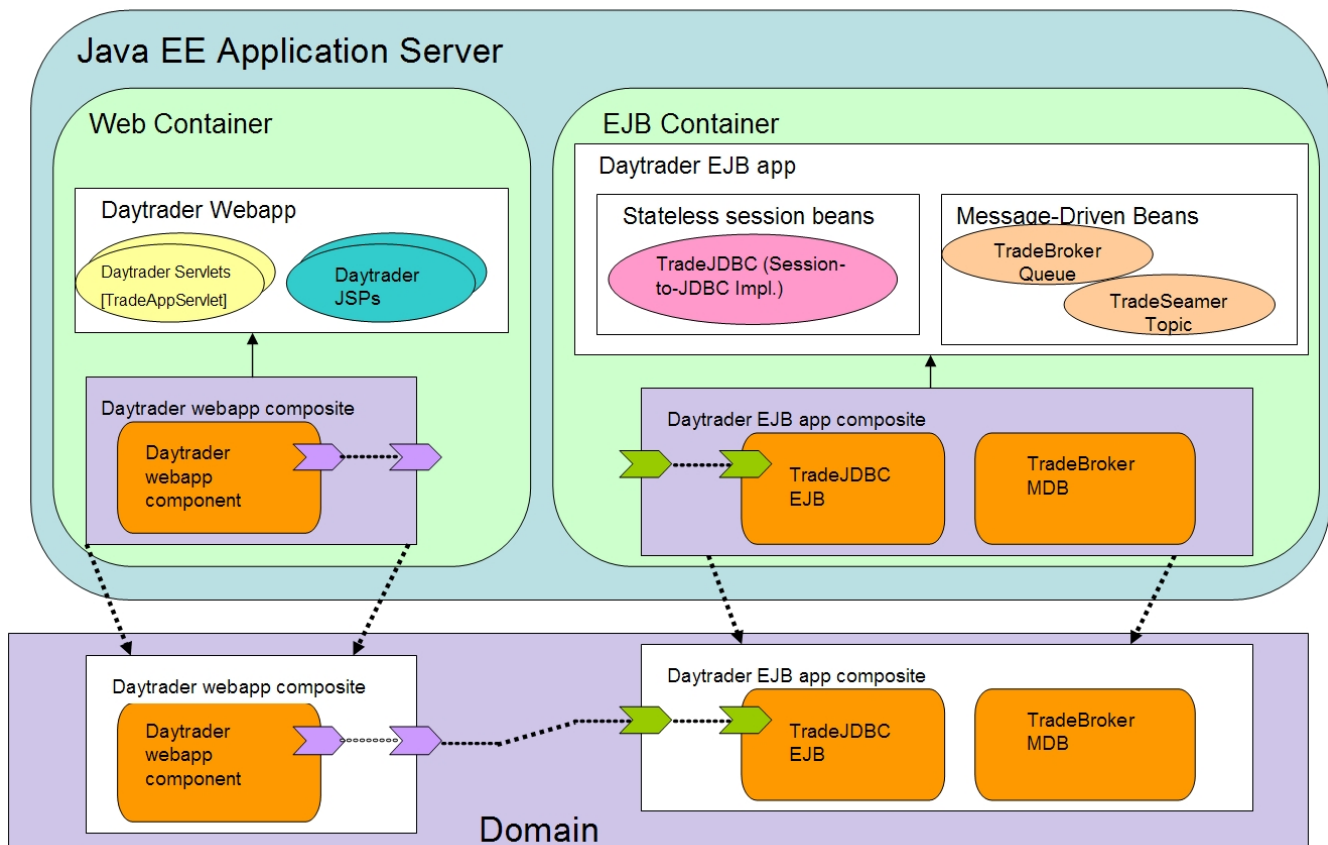
High level functions to be supported

1. Recognize JEE application as an SCA composite by default without any support for annotation or wiring
Scenario covered: Tuscany will create a composite based on JEE component model. This will enable JEE application to be used by other components in the SCA domain or call other components.
2. local assembly and the application composite covered in 1c, 2c, and 3c in the table below
 - support wiring across JEE applications
 - support using SCA components within a JEE application
Scenario covered: Enables a JEE component to participate in the assembly which can include JEE and none-JEE services. Wiring gets defined in the application composite xml file.
1. support SCA annotations in JEE components
See 1a and 2b below, so things like reference and property injection into EJBs, Servlets etc
Scenario covered: It enables a JEE component to be recognized as an SCA component to participate in a composition. Without SCA annotation, JEE component would still participate in a composition, but we would continue using JEE way to define services, references and properties. You would loose the ability to have none-JEE bindings, for example web20 bindings or to consume none-jee services.
2. support implementation.jee, implementation.ejb, and implementation.web for coarse-grained wiring
ie a reference in a JSP in a webapp war wired to an EJB in a separate EAR and a BPEL process deploy as a non-JEE SCA contributions
or a reference in a non-JEE SCA component wired to JEE artifacts
3. support SCA policies such as transaction and security within the JEE container
4. ability to deploy Non-JEE artifact SCA contributions and have them run in the container context
ie an SCA contribution containing a SCA service using binding.ws would use the container http infrastructure
5. support using the JEE container infrastructure where appropriate
ie the various Tuscany host-xxx modules, a service with binding.ws uses the existing HTTP connector, binding.jms uses the messaging provider, and support for the various QoS aspects using JEE capabilities

Java EE Integration features in detail

Feature	Section(s) in Java EE Spec v1.0	Priority	Map to Daytrader	Comments	Status
1. WAR	5.4.5			implementation.web	
a) Web application to consume services; @Reference/@EJB resulting in references.	5.4.1	1	Daytrader web application	Inject SCA Service in web component	Prototype code in the Tuscany runtime-tomcat module that supports injecting references into Servlets, Filters and Event listeners using the @Reference annotation.
b) WAR with componentType side file	5.4.3	2		Some of component type information comes from a component type file META-INF/web.componentType	
c) web.composite as a composite implementation	6.1.3	4			
d) JSP tag libraries	5.4.4	1	Daytrader web application	Inject SCA Service in web component	Taglib to include the SCA tags and define the reference field with the reference tag.
2. EJB Applications	5.1.6, 5.2.4			implementation.ejb	
a) EJB application to provide services exposed into SCA Domain	5.1, 5.1.1	1	TradeBean	Business interfaces of session beans as services	
b) EJB application to consume services; @Reference/@EJB resulting in references	5.1.2, 5.1.3, 5.1.4, 5.2.1, 5.2.2, 5.2.3	1	TradeBean	Remote EJB references and references to SCA services	
c) An EJB application's ejb-jar. composite to be used as composite implementation	6.1.3				
d) EJB with a componentType side file	5.1.5	3		Some of component type information comes from a component type file META-INF/<bean-name>.componentType	
e) Services by Stateful Session Beans as conversational services	5.1.9	2			
f) @OneWay	5.1.10	2	TradeBrokerMDB	Non-blocking service operations	
g) @Callback	5.1.11	2	TradeBrokerMDB	callback service	
3. Java EE Archives (SCA-enhanced)					
a) META-INF/sca-contribution.xml	6.1.1			Contribution to SCA Domain	
b) META-INF/sca-contribution-generated.xml	6.1.1				
c) Application composite	6.1.3				
	6.1.2				

d) Local assembly: Additional implementation types, binding and wiring capabilities					
e) Domain-level assembly	6.1.4				
4. Java EE Applications as implementation	7			implementation.jee	
a) Pure Web App as implementation	7.1.2	1	Daytrader web application		Code to model WAR as an SCA contribution available in Tuscany contribution-jee module
b) Pure EJB App as implementation	7.1.1	1	Daytrader ejb application		Code to model EJB jar as an SCA contribution available in Tuscany contribution-jee module
c) Pure JEE App as implementation	7.1.3	1			Code to model EAR as an SCA contribution available in Tuscany contribution-jee module
d) SCA-enhanced Web App as implementation w includeDefaults	7.2				
e) SCA-enhanced EJB App as implementation w includeDefaults	7.2				
f) SCA-enhanced JEE App as implementation w includeDefaults	7.2				
5. Leveraging Tuscany/SCA					
a) Web2.0 clients					
b) Expose JEE components with JSONRPC bindings					
6. Transactions	5.3			Mapping EJB Transaction Demarcation to SCA Transaction Policies	
7. Miscellaneous					
a) Import and Export of SCA artifacts	6.1.5			As described in the assembly spec	
b) Resolution of WSDL and XSD artifacts	6.1.6			Ref: OASIS SML Catalogs 1.1 spec	



References

- [1] http://www.osoa.org/download/attachments/35/SCA_JAVAAEE_Integration_V100.pdf?version=1
- [2] <http://cwiki.apache.org/GMOxDOC21/daytrader.html>