One Pager: Java EE Service Engine in GlassFish V3

Table of Contents

1. Introduction

The Sun Java EE Engine is a JSR 208 compliant Java Business Integration (JBI) runtime component that connects Java EE web services to JBI components. It is essentially designed to act as a bridge between the application server and the JBI runtime environment. The complete details of Sun Java EE Engine is found at http://java.sun.com/developer/technicalArticles/J2EE/sunjavaee_engine

1.1. Project/Component Working Name

Sun Java EE Engine (aka Java EE Service Engine) in GlassFish V3.

1.2. Name(s) and e-mail address of Document Author(s)/Supplier

Bhavanishankara Sapaliga (bhavanishankar@dev.java.net)
Mohit Gupta (mohit2602@dev.java.net)

1.3. Date of This Document

1.3. Date of This Document

23/03/09

2. Project Summary

2.1. Project Description

Mudularizing Java EE Service Engine to make it runnable in GlassFish V3. This project does NOT cover integrating OpenESB with GlassFish V3.

2.2. Risks and Assumptions

This project is dependent on the timely delivery of JAX-WS RI, JSR 109 implementation, Deployment modules for V3.

3. Problem Summary

3.1. Problem Area

This project aims to bridge GlassFish V3 with a JSR 208 JBI container/runtime. Java EE component that is deployed as a Servlet or EJB web service can be accessed from any JBI component . Also, the Java EE component can access the services exposed by the JBI container.

3.2. Justification

Java EE components can readily leverage JBI services and vice versa.

4. Technical Description

4.1. Details

Java EE Service Engine is an existing V2 component, so the work involved is mainly in making it modular for V3. Tentatively, Java EE Service Engine will be delivered as :

(a) Core module which consists of the core runtime of Java EE Service Engine. This module will be in STOPPED state until a JBI container starts it.

(b) Application listener module, which collects the maintains the web service endpoint details. This module is required to make these informations available when the core module is started at a later point of time.

(c) JBI bootstrap and lifecycle module. This module needs to be installed only after a JBI container is installed. This module implements the bootstrap and lifecycle methods defined in JSR 208 specification. The core module is started (and stopped) via this module.

Mode of delivery : The modules (a) and (b) should be available as part of standard GlassFish V3 installation. The module (c) should be made available publicly (eg., via maven repository/update center) so that it can be installed while installing a JBI container.

4.2. Bug/RFE Number(s)

Yet to be filed at glassfish/V3/jbi category.

4.3. In Scope

4.4. Out of Scope

This project does NOT cover integrating OpenESB with GlassFish V3.

4.5. Interfaces

Modularizing Java EE Service Engine will not affect its already exposed interfaces.

It will be fully obliged to JSR 208 specification, as in GlassFish V2.

4.5.1 Exported interfaces

Interface	*Stability*	*Exporting Project: Name, Specification or other Link.*	*Comments*
META-INF/jbi.xml for making a Java EE application as JSR 208 service unit.	Stable		META-INF/jbi.xml is an additional entry in the standard .ear, .war or .jar deployment archive. META-INF/jbi.xml lists the web service endpoints which are exposed to the JBI container. META-INF/jbi.xml also lists the JBI services that can be accessed from the Java EE application.

4.5.2 Imported interfaces

Interface	*Stability*	*Exporting Project: Name, Specification or other Link.*	*Comments*
com.sun.xml.ws.api.**	Standard	metro.dev.java.net	
javax.jbi.**	Standard	JSR 208	
org.glassfish.deployment.client.** com.sun.enterprise.deployment.**	Internal	GF V3 Deployment module	
org.glassfish.webservices.**	Internal	GlassFish V3 jsr109-impl	
javax.wsdl.** com.ibm.wsdl.**	Standard	WSDL shared library from open-jbi-components.dev.java.net	

4.6. Doc Impact

Developer's guide (chapter 6 - JBI runtime section)

4.7. Admin/Config Impact

4.8. HA Impact

4.9. I18N/L10N Impact

4.10. Packaging & Delivery

As described in section 4.1.

4.11. Security Impact

4.12. Compatibility Impact

The following private and deprecated interfaces will be removed in V3:

(a) Support of jbi-enabled flag in sun-web.xml. This was to enable the web service client to be used as a JBI consumer.

(b) Support of jbi-enabled flag in domain.xml. This was to enable the web service to be exposed as a JBI endpoint.

4.13. Dependencies

5. Reference Documents

http://java.sun.com/developer/technicalArticles/J2EE/sunjavaee_engine

https://open-esb.dev.java.net/kb/60/ep-javaee-se.html

http://weblogs.java.net/blog/bhavanishankar/archive/2007/01/servicemix_on_g_1.html

6. Schedule

6.1. Projected Availability

Inlined with GlassFish V3 schedule.