RFP 129 Versioning External Specifications

Draft

5 Pages

Abstract

Through the work of the Enterprise Expert Group, the OSGi is integrating with specifications that it does not control. A particular pain point is how these OSGi specifications handle the versioning of the external specifications that these OSGi specifications require. This RFP will identify the issues and requirements that surround the general problem OSGi faces when integrating with these specifications.

0 Document Information

0.1 Table of Contents

0.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 6.1.

`Source code is shown in this typeface.`

0.3 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	*01/25/10*	*Created the initial version of the document, spelled out first cut at requirements, first cut at use cases.* *Hal Hildebrand, Oracle, hal.hildebrand@oracle.com*

1 Introduction

The OSGi Enterprise Expert Group has a mandate to integrate the Java EE specifications into the OSGi. As such, the specifications that result from this process must refer to the external Java EE specifications. These specifications are not controlled by the OSGi, and these specifications have a preexisting notion of versioning and define the compatibility between these versions the users of these specifications may expect. The OSGi and the EEG, in particular, need a standard policy dictating how these external specifications are accommodated.

2 Application Domain

When the OSGi creates a specification which references other specifications, the version number of these external specifications are not controlled by the OSGi. Consequently, there is an inherent versioning problem that every integration specification must resolve, as the users of these specifications must be able to understand how to correctly use the versions of these specifications when creating and maintaining the imports of their bundles. These external specifications are, in most of the existing specifications, pre-existing specifications that have their own notion of how these specifications are to be versioned and the compatibility guarantees between versions. Consequently, OSGi has no control over the accepted versions defined by these specifications.

One of the main goals of a specification is to be understandable and unambiguous. Consequently, the specifications the OSGi produces cannot remain silent about the versions of the external specifications the specification refers to. Further, each specification produced by the OSGi should not use a different strategy, rather a single strategy should be used by all specifications which refer to external specifications. In that way, a consistent policy is created which will create uniform expectations across the specification products.

2.1 Terminology + Abbreviations

3 Problem Description

The OSGi has a strong notion of versioning which has been a part of the specification since the beginning of the specification. Versioning is rightly seen as a critical piece of the modularity infrastructure and a large part of the specification is devoted to the semantics around the run time resolution of versioned Java packages. Building on the long experience of the OSGi with versioning, the OSGi has guidelines regarding best practice of package versioning, and there is a strong commitment within the OSGi organization to use these best practices in the creation and maintenance of specifications produced within OSGi.

There is, however, an intrinsic problem when the OSGi specification is an integration of an external specification. These specifications are – by definition – controlled by OSGi. Rather, a new specification is being created with the explicit purpose of integrating a specification from another organization into the OSGi environment. In the case of the specifications being produced by the Enterprise Expert Group, these external specifications have a long history stretching back over a decade. These specifications are in wide spread use throughout the industry and they have been through several major revisions. They also have their own strong notion of how their specifications are versioned and what the compatibility rules are between versions of their specifications. The basic problem is that these specifications do not follow the versioning recommendations of the OSGi and consequently pose a source of potentially serious confusion for the OSGi when producing these integration specifications.

4 Use Cases

4.1 Using the RFC 66 Web Container

A programer has just been given the project of integrating create a Web Application Bundle for an RFC 66 compliant Web Container. The programmer needs to know which versions of the Servlet specification to use in the import statements in the bundles that result from the integration. The programmer also would like to know how to formulate these version strings such that they do not break when backward compatible versions of the Servlet specification are supplied by the runtime environment as well as excluding versions that will not be compatibile with Servlet version that the programmer is compiling against.

4.2 Using the RFC 143 JPA Integration

In the course of developing a WAB, the developer needs to integrate with persistent data and so naturally turns to the RFC 143 for guidance as to how. The developer does not know what the runtime version will be provided for the JPA, but knows that version 2.0 of the specification is backward compatible with version 1.1. The developer also wants to be assured that future versions of the specification will not be used by the bundles until these bundles are tested against future versions of the specification.

5 Requirements

6 Document Support

6.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

6.2 Author's Address

Name	Hal Hildebrand
Company	Oracle
Address	500 Oracle Pkwy, Redwood City, CA, USA
Voice	+1 650 506 2055
e-mail	hal.hildebrand@oracle.com

6.3 End of Document