



RFP 63 OSGi Framework extensions for Tools RFP

Confidential, Draft

8 Pages

Abstract

This RFP enumerates the requirements for tool interoperability amongst tools from multiple vendors.

Copyright © IBM Corp. 2004.

This contribution is made to the OSGi Alliance as MEMBER LICENSED MATERIALS pursuant to the terms of the OSGi membership agreement and specifically the license rights and warranty disclaimers as set forth in Sections 3.2 and 12.1, respectively.

All company, brand and product names contained within this document may be trademarks that are the sole property of the respective owners.

The above notice must be included on all copies of this document that are made.

0 Document Information

0.1 Table of Contents

0 Document Information	1
0.1 Table of Contents	2
0.2 Status	2
0.3 Acknowledgement	2
0.4 Terminology and Document Conventions	2
0.5 Revision History	3
1 Introduction	3
2 Application Domain	3
3 Problem Description	4
4 Requirements	6
5 Document Support	8
5.1 References	8
5.2 Author's Address	8
5.3 Acronyms and Abbreviations	8
5.4 End of Document	8

0.2 Status

This document suggests the following improvement/change/enhancement/extension to the OSGi specification for the Open Services Gateway Initiative, and requests discussion. Distribution of this document is unlimited within OSGi.

0.3 Acknowledgement

0.4 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 [1].

Source code is shown in this typeface.

All Page Within This Doc

0.5 Revision History

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

Revision	Date	Comments
Initial	Apr 08, 2004	Started document. Mark Rogalski, IBM, Rogalski@us.ibm.com
DGB	Apr 28, 2004	Minor text edit and reorder in 5.3.2 Dave Bevis, IBM, Bevisdg@uk.ibm.com
DGB	May 07 2004	Clarification texts in 4.2 , 4.4, 4.5 description, addition of 4.2 precondition Dave Bevis, IBM, Bevisdg@uk.ibm.com

1 Introduction

There are some portions of the OSGi specification which leave open to implementers the interfaces between various components. Since each vendor implements these interfaces differently, tools from different vendors are not interoperable. This RFP is being written to address this problem.

2 Application Domain

There are multiple vendors that have their own OSGi Framework implementations and tools. Because tools from different vendors are not interoperable, users must select just one vendor's runtime and tools to work with.

All Page Within This Doc

3 Problem Description

OSGi did not specify the interfaces between various tools components or between tools and the OSGi framework runtime, (i.e. how bundles are given to a bundle server, how bundles are installed in to a framework runtime, etc.). Therefore, these interfaces have been implemented in different ways. Thus, tools from different vendors are not interoperable and this limits the ability of developers to mix and match components.

An open specification of tool/framework interfaces will enable developers to obtain best of breed functionality or use tools they are familiar with for interacting with different frameworks.

4 Use Cases

4.1 Bundle Development

Use Case	U1. Bundle Development
Actor(s)	Developer
Preconditions	<ul style="list-style-type: none">Workbench tool is runningDeveloper adds an import statement to source or manifest which can not be satisfied locally.
Description	The Developer is developing an OSGi bundle and adds an API call or import statement for a package that the workbench tool does not know about (not in Java build path). The workbench tool communicates with bundle servers to discover a bundle which can satisfy API call or import statement. The discovered bundle is transferred to the local machine and added to the Java build path. Bundles contain enough information to enable most advanced editing features such as signature verification, type-ahead, but not JavaDoc.
Exceptions	<ul style="list-style-type: none">The required dependency bundles can not be obtained.
Post conditions	<ul style="list-style-type: none">Bundle is built.

All Page Within This Doc

4.2 Bundle Build

Use Case	U2. Bundle Build
Actor(s)	Developer
Preconditions	<ul style="list-style-type: none"> Workbench tool is running Workbench tool has a list of bundle servers Bundle source and manifest are completed and error free
Description	The Developer saves a changed source or manifest file. The workbench tool compiles the Java source and zips the bundle. If the manifest calls for a "self-contained" bundle, the workbench tool queries bundle servers to obtain the required dependency bundles and adds them to the bundle.
Exceptions	<ul style="list-style-type: none"> The required dependency bundles can not be obtained.
Post conditions	<ul style="list-style-type: none"> Bundle is built.

4.3 Bundle Submit

Use Case	U3. Bundle Submit
Actor(s)	Developer
Preconditions	<ul style="list-style-type: none"> Workbench tool is running Bundle Server is running
Description	The Developer submits a bundle to a remote bundle server. Developer right clicks on bundle project and chooses Bundle Server/Submit. A dialog is displayed allowing choice of bundle server. Developer chooses server and clicks OK. Status response is returned to the caller.
Exceptions	<ul style="list-style-type: none"> The bundle is not ready to be submitted. The chosen bundle server can not be accessed. The bundle server rejects the submission.
Post conditions	<ul style="list-style-type: none"> Bundle is available on bundle server. Status is returned to the caller.

4.4 Bundle Install

Use Case	U4. Bundle Install
Actor(s)	Developer
Preconditions	<ul style="list-style-type: none"> Bundle server is running OSGi runtime is running
Description	The Developer installs a bundle to a running OSGi runtime. Developer right clicks on bundle in bundle server available bundle list and chooses Install. A dialog is displayed allowing choice of OSGi runtime. Developer chooses runtime and clicks OK. Workbench tool communicates with target runtime to inform it to install the specified bundle. Status response is returned to the caller.
Exceptions	<ul style="list-style-type: none"> Can't communicate with chosen OSGi runtime. The OSGi runtime rejects the install.

Post conditions	<ul style="list-style-type: none">• Bundle is installed in the OSGi runtime.• Status is returned to the caller.
-----------------	--

4.5 Bundle Update

Use Case	U5. Bundle Update
Actor(s)	Developer
Preconditions	<ul style="list-style-type: none">• Bundle server is running• OSGi runtime is running
Description	The Developer updates an existing bundle in an OSGi runtime. Developer right clicks on bundle name in a list of installed bundles for a target runtime, and chooses Update. The workbench tool communicates with the target runtime to inform it to update the specified bundle. The runtime requests bundle from bundle server that the bundle was originally installed from. Status response is returned to the caller.
Exceptions	<ul style="list-style-type: none">• Can't communicate with chosen OSGi runtime.• The original bundle server can not be accessed.• The OSGi runtime rejects the update.
Post conditions	<ul style="list-style-type: none">• Bundle is updated in the OSGi runtime.• Status is returned to the caller.

5 Requirements

5.1 OSGi bundle server related interactions

5.1.1 A development workbench tool must be able to connect to a selected internal or external OSGi bundle server through a URL.

REQ-FET-01-01 MUST be able to connect to bundle server via a URL

5.1.2 A development workbench tool must be able to query a bundle server

REQ-FET-01-02 MUST be able to list bundles available on the server

REQ-FET-01-03 SHOULD be able to obtain bundle details (such as version number) from the server

REQ-FET-01-04 MUST be able to query server for a bundle which provides specified package or service

REQ-FET-01-05 MUST be able to request delivery of a specified bundle

5.1.3 A development workbench tool must be able to manage a selected internal or external OSGi bundle

server.

REQ-FET-01-06 MUST be able to add bundles (submit) to the server

REQ-FET-01-07 MUST be able to update bundles on the server

REQ-FET-01-08 MUST be able to remove bundles from the server

5.2 OSGi framework related interactions

5.2.1 A development workbench tool must be able to connect to a selected internal or external OSGi framework runtime through a URL.

REQ-FET-02-01 MUST be able to connect to device emulator and/or a real device

5.2.2 A development workbench tool must be able to query a selected internal or external OSGi framework runtime.

REQ-FET-02-02 SHOULD be able to list installed bundles in the framework

REQ-FET-02-03 SHOULD be able to obtain bundle details/status from the framework

5.2.3 A development workbench tool must be able to manage a selected internal or external OSGi framework runtime.

REQ-FET-02-04 MUST be able to Install, uninstall, update, start and stop bundles in the framework

5.3 Build process related interactions

5.3.1 A development workbench tool must be able to build new applications/services/libraries via importing and/or making references to other bundles available through various bundle servers.

REQ-FET-03-01 Workbench tool MUST be able to satisfy import dependencies at development time using bundle discovery via communication with bundle servers

5.3.2 A development workbench tool must be able to package the application/service/shared library as an OSGi bundle. The tool could select appropriate bundles from a repository for a particular target environment. This concept is that instead of dependency checking and package completion being completed always at the client, some packages may be "prebuilt assemblies of bundles" and these prebuilt packages may be deployed to a client as being self contained, thus mitigating the need of a client to always go to a bundle server for bundle requirements it does not have locally

REQ-FET-03-02 Workbench MUST be able to produce a self-contained package whereby all bundle dependencies are included within the bundle itself.

REQ-FET-03-03 Workbench tool MUST be able to produce a sparse bundle for which dependencies are resolved at bundle installation. The framework is responsible for retrieval and installation of bundle dependencies which are not already installed on the client.

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	
Company	
Address	
Voice	
e-mail	

6.3 Acronyms and Abbreviations

6.4 End of Document