

RFP 127 OBR Hosting

Draft

9 Pages

Abstract

The OSGi Alliance hosts a public OSGi Bundle Repository. Users can browse this repository on the web, leave comments, and ask to resolve a set of bundles. The current software is becoming overloaded and in general, better software is needed. This RFP collects the requirements for a hosting company to provide the OSGi OBR.

The RFP will be used to select a company that can host OBR.

Copyright © aQute 2009.

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..... | 2 |
| 0.1 Table of Contents..... | 2 |
| 0.2 Terminology and Document Conventions..... | 2 |
| 0.3 Revision History..... | 3 |
| 1 Introduction..... | 3 |
| 2 Application Domain..... | 3 |
| 2.1 Terminology + Abbreviations..... | 5 |
| 3 Problem Description..... | 5 |
| 4 Use Cases..... | 6 |
| 4.1 Architect..... | 6 |
| 4.2 End User..... | 6 |
| 4.3 Runtime..... | 6 |
| 5 Requirements..... | 6 |
| 5.1 Repository..... | 6 |
| 5.2 Browsing..... | 6 |
| 5.3 Content Provisioning..... | 7 |
| 5.4 Security..... | 7 |
| 5.5 Legal..... | 7 |
| 6 Document Support..... | 8 |
| 6.1 References..... | 8 |
| 6.2 Author's Address..... | 8 |
| 6.3 End of Document..... | 9 |

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 | AUG 26 2009 | Initial, Peter Kriens |

1 Introduction

The OSGi Alliance has been hosting a bundle repository since 2005. This repository was hosted on the OSGi web server and it obtained its content from a number of members and open source organizations. The software that hosts the OBR has been developed as a marketing effort. Now the need for repositories is increasing, it becomes more clear what features are missing and there is also a load issue.

This report investigates the needs the OSGi Alliance has for a public repository. During the early discussions it turned out that there are a number of companies willing to host the OSGi repository. This document can be used by these companies to provide an offering so that the board can select the most appropriate company.

2 Application Domain

Repositories have become an integral part of today's developers work environment. The advent of the internet makes it possible to download software artifacts on the fly. This model was made highly popular by Maven, a build tool developed and Ivy, a dependency manager. Both built in Apache. These tools were developed as an answer to the increasing number of open source software projects that commercial projects started to depend on. Where a late 1980 application had maybe 4 or 5 external libraries, today's projects easily have hundreds of external dependencies.

Maven repositories host the artifacts together with the pom.xml. This pom.xml contains information about the artifact and its dependencies. Dependencies are on JARs that are also hosted in Maven repositories. Traversing these dependencies allows a maven build to download all necessary artifacts and put them on the class path, without much manual intervention.

Maven repositories are very useful in standard Java development. However, the OSGi Alliance developed a comprehensive set of specifications that provide a component system. The dependencies between these components in the current set of specifications are not defined on artifacts (bundles), but are normally specified on packages, the existing Java modules. This indirection from package to bundle does not map naturally to maven repositories.

Additionally, it became clear in the OSGi Alliance that code dependencies do not always map well to other, real life, dependencies. The OSGi extender model, for example, has a natural dependency from the extender (the to be extended bundle) to the extender, but there does not have to be a code dependency. A bundle could require a certain window system, the availability of a certificate, the presence of a high resolution screen, a specific execution environment. The list is endless.

In 2005, the OSGi Alliance published a draft RFC and, as an exception, made this RFC available to the public. This RFC was based on an enumeration of artifacts that had capabilities and requirements. A capability is a set of properties with their values, and a requirement is a filter on a capability. The RFC defined an API, which was implemented by Apache Felix and an XML interchange format.

Adopting OSGi is not always easy because there are limited bundles readily available, especially in 2005. providing a free, public repository was seen as a good tool to increase OSGi adoption. Such a repository could create a market place for bundles, taking advantage of Metcalfe's law. This law states that a network becomes more valuable with more connections. The availability of bundles would drive more people to provide bundles.

Therefore, RFC 112 was used to create a website hosted by the OSGi Alliance that aggregated the result of Eclipse, Makewave, aQute, ProSyst, Paremus, and the Spring Source repository. At that time, there was no standard software available so OSGi marketing enabled the development of a prototype. Since 2005, this prototype has been available on the OSGi website.

The website OBR provides a rudimentary user interface for browsing the repository. Search criteria can be entered and these words are searched in the metadata available from the repository. The search result can be used to download the artifact. However, it is also possible to resolve dependencies. For each artifact, it is possible to comment it. Commenting is very rarely used though.

The web site can be used through a web browser, or programmatically.

Though the site works, one of the key problems is the fact that most bundles do not have proper metadata. Very few bundles have for example a description of their purpose.

In 2008, Springsource announced that they had ported a large number of bundles to OSGi and made them available via a website and maven.

In 2008, Paremus spent a significant effort in building OSGi tools that took OBR as the standard. They connected OBR to Ivy, an ant based dependency manager. They hosted an OBR based on Spring source's repository because SpringSource had not yet made an OBR index available.

In 2009, Sonatype became an OSGi member. Sonatype is the driving force behind Maven and is a primary player in the Java build tools. Most major Java projects today are using Maven. In 2006 the Felix project created a plugin to create bundles based on bnd, which made it quite easy to generate the manifest. This plugin is wildly popular with OSGi developers. In a recent move, Sonatype started to work on maven 3, basing it on OSGi. This will likely make it much easier for developers to create bundles by default.

In the period preceding this RFC, Oracle has taken the pen again of RFC 112 OBR. However, some member companies in the OSGi objected to start with an RFC and demanded the creation of an RFP to investigate the requirements around OBR. Though RFC 112 established itself as a de-facto standard for OSGi bundles, it is not sure what will happen with this this RFC.

2.1 Terminology + Abbreviations

3 Problem Description

The current incarnation of the OBR requires replacement because it is becoming old and needs additional features. One of the mission statements of OSGi is to foster the eco systems. Repositories are becoming increasingly important in the OSGi world. Though Maven repositories are very popular, there is a clear lack of a bundle repository that can handle the additional metadata from bundles.

Such repository could provide the much needed central market place for bundles. Though in the current market access to these bundles must be free, it should be possible in the future to create monetary incentives for bundle providers. It is not clear how the business case for this “bundle shop” could look like, but options are advertising revenue, sales, subscriptions, sponsoring, or a combination.

Though repositories solve the reliable problem distribution of artifacts and their dependencies, any user of a repository still has work to do. Current repositories do not provide any guarantee about what they deliver regarding licensing, patents, pedigree, quality, integrity, and over time the availability. For smaller companies, most of these issues are ignored but larger and more careful corporations must handle these issues themselves. Though these costs are often hidden inside an organization, they can be substantial. Providing guaranteed (and thus signed) artifacts might be a potential extension of the repository.

Repositories are at the core of future developments. It is foreseen that they will move up in the design chain and become very relevant very early during architectural design while remaining a crucial aspect of development, testing, and deployment.

The OSGi does not have the expertise nor the desire to run a solid OBR. This RFP therefore seeks a company that is willing:

- To host an OBR repository on behalf of the OSGi Alliance, and
- Work with the OSGi Alliance to develop further repository based solutions

4 Use Cases

4.1 Architect

An architect must design a system for his company. After an initial study he finds that he needs a document database, a web framework, PDF generation, a rules engine, and persistence. He goes to the OBR website and searches for bundles providing these functions. Using the resolver he finds the dependencies and selects the set of bundles he needs. These bundles are then synchronized with his local repository.

4.2 End User

A developer is looking for a specific bundle. He goes through the website and finds it, together with possible extensions.

4.3 Runtime

A user installs an application in a framework through an OBR application. The input are the bundle symbolic name of the bundles and optionally their versions. The OBR application contacts the repository and together they calculate the dependency graph for that framework. The proper bundles are then installed and started. The repository to be contacted is likely local and synchronized with a remote repository.

5 Requirements

5.1 Repository

- RE0001 – There must be a repository with bundles and other resources like certificates, translations, sound files, etc., as long as they can be represented as a byte stream.
- RE0002 – The RFC 112 OBR concepts like requirements and capabilities must be supported
- RE0003 – If the OSGi changes this format, then these changes must be supported in the future
- RE0004 – It must be possible to synchronize or mirror the repository by authorized parties. Authorization is in joint agreement by OSGi and the hosting company.
- RE0005 – The hosting company must provide adequate bandwidth to prevent complaints

5.2 Browsing

- BR0001 – There must be a web page provide human access to the repository

- BR0002 – It must be possible to search on bundle symbolic name, keywords in the description, category, version, exported and imported packages, license, and other relevant fields.
- BR0003 – For a given bundle, it must be possible to find a set of bundles that satisfy its dependencies and download these as a whole.
- BR0004 – It must be possible to provide comments and a rating on a bundle
- BR0005 – For read only access, users must not be required to login
- BR0006 – It must be possible to download the selected bundles directly from a web page
- BR0007 – The style of the website must follow the OSGi style

5.3 Content Provisioning

- CP0001 – Providers of bundles must be able to electronically submit bundles via the web or a program.
- CP0002 – The bundle metadata must be contained in the bundle
- CP0003 – The OBR data about the bundles must be generated from the bundle alone
- CP0004 – Content providers must not be able to withdraw submissions
- CP0005 – Content providers must be able to deprecate submissions
- CP0006 – Content providers must be able to put remarks and notifications on submissions
- CP0007 – Content providers must be able to provide a URL to an OBR repository for synchronization. However, the hosting company must not trust the meta data and reindex the bundles.
- CP0008 – Bundles must be verified for consistency with the OSGi specifications. That is, header syntax, proper values, etc. At minimum the verifications that a tool like bnd can do.
- CP0009 – Rejected bundles must be notified to their submitter
- CP0010 – Submitted bundles must have a properly set Bundle-License header.

5.4 Security

- SE0001 – Browsers are not required to authenticate
- SE0002 – Commenters require simple authentication like openid or via an email roundtrip
- SE0003 – Submitters of content must be strongly authenticated
- SE0004 – Bundles part of the repositories must be signed

5.5 Legal

The legal issues will be fine tuned in a contract. However, the following requirements sketch the current thinking in lay men's terms.

- LE0001 – Bundles submitted to the OSGi repository are owned by OSGi
- LE0002 – If the agreement with the OBR hosting company is terminated, the hosting company will provide the full content of the OBR to the OSGi Alliance as of the date of termination
- LE0003 – The hosting company is allowed to use its role in marketing its company
- LE0004- The hosting company may show its name and logo on the website pages for the OBR under the condition that it is adamantly clear the OSGi is the primary party.
- LE0005 – All DNS names used for OBR must be under full control of the OSGi Alliance
- LE0006 – The hosting company will be responsible for the day to day operations
- LE0007 – The hosting company is responsible for high availability of the repository
- LE0008 – It must be made clear to submitters of content, browsers and other users of the repository that the OSGi Alliance will not have any liability.

6 Document Support

6.1 References

- [1]. RFC 112 OBR
- [2]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

6.2 Author's Address

| | |
|---------|------------------------|
| Name | Peter Kriens |
| Company | aQute |
| Address | 9c, Avenue St. Drezery |
| Voice | +33467542167 |
| e-mail | Peter.Kriens@aQute.biz |

6.3 End of Document