



OSGiTM
Alliance

RFP 162 - Standardize Managed Bean Registration

Draft

7 Pages

Abstract

A standard is needed in OSGi to provide a mechanism for registering *Managed Beans* with an Mbean Server.

0 Document Information

0.1 License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

The OSGi Alliance hereby grants you a limited copyright license to copy and display this document (the "Distribution") in any medium without fee or royalty. This Distribution license is exclusively for the purpose of reviewing and providing feedback to the OSGi Alliance. You agree not to modify the Distribution in any way and further agree to not participate in any way in the making of derivative works thereof, other than as a necessary result of reviewing and providing feedback to the Distribution. You also agree to cause this notice, along with the accompanying consent, to be included on all copies (or portions thereof) of the Distribution. The OSGi Alliance also grants you a perpetual, non-exclusive, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights, to create and/or distribute an implementation of the Distribution that: (i) fully implements the Distribution including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the OSGi Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the OSGi Name Space other than those required and authorized by the Distribution. An implementation that does not satisfy limitations (i)-(ii) is not considered an implementation of the Distribution, does not receive the benefits of this license, and must not be described as an implementation of the Distribution. "OSGi Name Space" shall mean the public class or interface declarations whose names begin with "org.osgi" or any recognized successors or replacements thereof. The OSGi Alliance expressly reserves all rights not granted pursuant to these limited copyright licenses including termination of the license at will at any time.

EXCEPT FOR THE LIMITED COPYRIGHT LICENSES GRANTED ABOVE, THE OSGi ALLIANCE DOES NOT GRANT, EITHER EXPRESSLY OR IMPLIEDLY, A LICENSE TO ANY INTELLECTUAL PROPERTY IT, OR ANY THIRD PARTIES, OWN OR CONTROL. Title to the copyright in the Distribution will at all times remain with the OSGi Alliance. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted therein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

THE DISTRIBUTION IS PROVIDED "AS IS," AND THE OSGi ALLIANCE (INCLUDING ANY THIRD PARTIES THAT HAVE CONTRIBUTED TO THE DISTRIBUTION) MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DISTRIBUTION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

NEITHER THE OSGi ALLIANCE NOR ANY THIRD PARTY WILL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE DISTRIBUTION.

Implementation of certain elements of this Distribution may be subject to third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a member of the OSGi Alliance). The OSGi Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

The Distribution is a draft. As a result, the final product may change substantially by the time of final publication, and you are cautioned against relying on the content of this Distribution. You are encouraged to update any implementation of the Distribution if and when such Distribution becomes a final specification.

The OSGi Alliance is willing to receive input, suggestions and other feedback ("Feedback") on the Distribution. By providing such Feedback to the OSGi Alliance, you grant to the OSGi Alliance and all its Members a non-exclusive, non-transferable,

worldwide, perpetual, irrevocable, royalty-free copyright license to copy, publish, license, modify, sublicense or otherwise distribute and exploit your Feedback for any purpose. Likewise, if incorporation of your Feedback would cause an implementation of the Distribution, including as it may be modified, amended, or published at any point in the future ("Future Specification"), to necessarily infringe a patent or patent application that you own or control, you hereby commit to grant to all implementers of such Distribution or Future Specification an irrevocable, worldwide, sublicenseable, royalty free license under such patent or patent application to make, have made, use, sell, offer for sale, import and export products or services that implement such Distribution or Future Specification. You warrant that (a) to the best of your knowledge you have the right to provide this Feedback, and if you are providing Feedback on behalf of a company, you have the rights to provide Feedback on behalf of your company; (b) the Feedback is not confidential to you and does not violate the copyright or trade secret interests of another; and (c) to the best of your knowledge, use of the Feedback would not cause an implementation of the Distribution or a Future Specification to necessarily infringe any third-party patent or patent application known to you. You also acknowledge that the OSGi Alliance is not required to incorporate your Feedback into any version of the Distribution or a Future Specification.

I HEREBY ACKNOWLEDGE AND AGREE TO THE TERMS AND CONDITIONS DELINEATED ABOVE.

0.2 Trademarks

OSGi™ is a trademark, registered trademark, or service mark of the OSGi Alliance in the US and other countries. Java is a trademark, registered trademark, or service mark of Oracle Corporation in the US and other countries. All other trademarks, registered trademarks, or service marks used in this document are the property of their respective owners and are hereby recognized.

0.3 Feedback

This document can be downloaded from the OSGi Alliance design repository at <https://github.com/osgi/design> The public can provide feedback about this document by opening a bug at <https://www.osgi.org/bugzilla/>.

0.4 Table of Contents

0 Document Information.....	2
0.1 License.....	2
0.2 Trademarks.....	3
0.3 Feedback.....	3
0.4 Table of Contents.....	3
0.5 Terminology and Document Conventions.....	4
0.6 Revision History.....	4
1 Introduction.....	4
2 Application Domain.....	4
2.1 Terminology + Abbreviations.....	5
3 Problem Description.....	5
4 Use Cases.....	5
5 Requirements.....	6
6 Document Support.....	6
6.1 References.....	6
6.2 Author's Address.....	6
6.3 End of Document.....	6

0.5 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.

0.6 Revision History

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

Revision	Date	Comments
Initial	01/10/14	Initial Draft Thomas Watson, IBM, tjwatson@us.ibm.com

1 Introduction

The Java Management Extensions is the standard API specification for providing a management interface to J2SE and JEE applications. The JMX specification defines the design patterns, APIs, services and architecture for application and network management and monitoring in the Java programming language. The need to administer, monitor and manage a container is now recognized as a prerequisite in the enterprise software domain.

OSGi already defines a specification for a JMX Management Model to provide standard MBean interfaces that can represent the OSGi Framework and many of the standard services to any compliant JMX implementation. There is no OSGi specification for how the individual MBean implementations are registered with an *MBean Server*. This is left as an implementation detail for the implementation of the various standard MBean interfaces as well as any other entity running in an OSGi Framework which wants to provide an MBean.

2 Application Domain

2.1 Terminology + Abbreviations

2.2 Apache Aries Whiteboard Pattern

Apache Aries provides a whiteboard pattern for registering MBean implementations with an MBean Server [3].

The JMX Whiteboard support bundle allows registration of JMX MBeans through the OSGi service registry. As a prerequisite the Apache Aries JMX Whiteboard Bundle has to be installed.

Once this is installed, register your services with the `jmx.objectname` service registration property. This is either a JMX `ObjectName` or a single String. The value is used as the JMX Object Name to register the MBean. If the property is of some other type a message is logged and the MBean is still registered if the service implements the `MBeanRegistration` interface and the implementation provides the actual JMX Object Name to use for the registration. (TJW – I'm not sure I like the use of `ObjectName` as a service property since it is a domain specific object)

The name of the service as which the MBean service is registered is ignored by the Whiteboard. (TJW - I assume by name they mean `objectName`).

The service object, though, must be a valid JMX MBean: It has to either implement the `DynamicMBean` interface or implement an interface whose name is the object's (simple) class name with the MBean suffix. (TJW – I'm not sure why this is specified here since I assume an object can also implement an interface with `@MXBean` annotation.

3 Problem Description

4 Use Cases

4.1 Delay MBean Server creation

In many environments there is no need to use JMX to communicate with a system until a period of time after the system has been running and in many cases for the whole lifetime of the system instance. In such cases it is desirable to avoid the creating of the MBean Server until the first JMX request is made. If all MBean implementations eagerly register themselves with an MBean Server then it is likely the MBean Server is going to be created early during system bring up. This can lead to unnecessary overhead.

4.2 Configuration of MBean Server

In some environments it may be necessary to be able to configure and influence the MBean Server used to register MBean with. In such a scenario each MBean implementation needs a common way of obtaining the MBean Server and registering themselves with it. The common way of obtaining the MBean Server is by calling `java.lang.management.ManagementFactory.getPlatformMBeanServer()`. But in some environments this may not be the correct MBean Server to use.

4.3 Dynamic MBean Implementations

The OSGi environment is dynamic in nature. Many entities which are good candidates for MBean representations are dynamically available within the OSGi environment. For example, any service objects which need MBean representations. MBean implementations which represent a dynamically available object should be automatically unregistered from the MBean Server when the dynamic object backing the MBean implementation becomes unavailable (e.g. unregistering a service object). Also, any MBean implementations which are registered by a bundle should be unregistered from the MBean Server when the bundle is stopped.

5 Requirements

- REQ 010 – Must provide a mechanism for registering MBean and MXBean implementations with one or more MBean Servers
- REQ 020 – Must allow for lazy creation of the MBean Servers which are used to register the MBean implementations
- REQ 030 – Must allow MBean implementations to be registered with an MBean Server without imposing any policy or configuration choices of the MBean Server on the MBean implementation.
- REQ 040 – Must provide a mechanism for unregistering MBean implementations from all MBean Servers they are registered with.

- REQ 050 – Must track registration of MBeans from bundles and automatically unregister the MBeans from the MBean Servers they are registered with when a bundle is stopped.

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
- [3]. <http://aries.apache.org/modules/jmx.html>

6.2 Author's Address

Name	
Company	
Address	
Voice	
e-mail	

6.3 End of Document