



RFC 198 - System Bundle and Framework Hooks

Final

11 Pages

Abstract

The OSGi Core specification version 4.2 introduced service hooks and version 4.3 introduced bundle, resolver and weaving hooks. The service, bundle and resolver hooks together may be used to implement a scoping model for isolating groups of bundles within the same framework. These framework hooks can control the visibility of services, bundles and generic capabilities for the system. The current specification is silent on the ability to isolate or hide services, bundles and capabilities from the system bundle. This implies that framework hooks are able to hide services, bundles and capabilities from the system bundle.

This RFC specifies the behavior of framework hooks when acting upon the system bundle.

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.....	5
2 Application Domain.....	5
3 Problem Description.....	5
4 Requirements.....	6
4.1 Service Find Hooks.....	6
4.2 Service Event Hooks.....	7
4.3 Bundle Find Hooks.....	7
4.4 Bundle Event Hooks.....	7
4.5 Bundle Collision Hooks.....	8
5 Technical Solution.....	8
5.1 Bundle Event Hooks.....	8
5.1.1 System Bundle Listeners.....	8
5.2 Bundle Find Hooks.....	8

5.2.1 System Bundle Context.....	8
5.3 Service Event Listener Hooks.....	9
5.3.1 System Service Listeners.....	9
5.4 Service Find Hooks.....	9
5.4.1 System Bundle Context.....	9
5.5 Bundle Collision Hooks.....	9
5.5.1 System Bundle Context.....	9
6 Data Transfer Objects.....	10
7 Considered Alternatives.....	10
8 Security Considerations.....	10
9 Document Support.....	11
9.1 References.....	11
9.2 Author's Address.....	11
9.3 Acronyms and Abbreviations.....	11
9.4 End of Document.....	11

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	Mar 21 2013	Thomas Watson, IBM, tjwatson@us.ibm.com Initial version
	Nov 12 2013	Thomas Watson Changes to not ignore bundle collision hooks for the system bundle.
Final	7 Feb 2014	Final version for voting.

1 Introduction

The OSGi Core specification version 4.2 introduced service hooks and version 4.3 introduced bundle, resolver and weaving hooks. The service, bundle and resolver hooks together may be used to implement a scoping model for isolating groups of bundles within the same framework. These framework hooks can control the visibility of services, bundles and generic capabilities for the system. The current specification is silent on the ability to isolate or hide services, bundles and capabilities from the system bundle. This implies that framework hooks are able to hide services, bundles and capabilities from the system bundle.

This RFC specifies the behavior of framework hooks when acting upon the system bundle. In particular this design specifies that framework hooks cannot affect the outcome when the client bundle is the system bundle.

2 Application Domain

The basic framework provides complete visibility for any bundle to any other bundle, service or capability. In certain use cases it is important to provide a notion of isolation or scope to a group of bundles. The Enterprise specification has defined subsystems as a standard way to isolate groups of bundles.

In environments that include an isolation model, such as subsystems, it can be difficult to get a view of the complete framework. The isolation model may provide introspective API for getting a view of the complete system, but this requires intimate knowledge of each isolation model. Also with the introduction of framework DTOs there needs to be a standard way to obtain access to all bundles, services and capabilities.

3 Problem Description

The behavior of framework hooks when acting upon the system bundle is not consistent across several framework implementations.

- Equinox currently gives the system bundle visibility to find all bundles, but does nothing special for other hook operations

- Prosyst has an option to not call hooks when applied against listeners registered with system bundle context
- Felix does not do anything different for the system bundle (TJW - If I recall correctly)

This introduces issues for use cases that need access to all bundles, services and capabilities. For example, populating a FrameworkDTO is not possible in a standard way without knowledge of the possible isolation models. It also forces extenders that want to support extending bundles in isolation models (such as subsystems) to be bound to API provided by the isolation model.

It also seems inappropriate to hide things from the system bundle. The system bundle is the framework and should be able to have a full view of the complete system of bundles, services and capabilities. This RFC defines a solution that will not allow hooks to influence the visibility of bundles, services and capabilities from the system bundle. This will allow the system bundle to be used in the use cases mentioned above to gain full view of the bundles, services and capabilities in the system.

Providing a full view of the bundles, services and capabilities in the system, regardless of the presents of an isolation model, is not enough for the extender usecase. Extenders that support extending bundles across isolated boundaries must consider the visibility of the resources from the perspective of the extendee bundles. There are two ways to achieve this:

- By using the extendee bundle's BundleContext to listen and get to the required resources when extending the bundle. For example, DS and Blueprint extenders should use the extendee bundles BundleContext to listen to service events and to get ServiceReference objects. This ensures that these resources are retrieved using the perspective of the extendee bundle and will properly drive the required framework hooks that implement the isolation model (i.e. a subsystems implementation)
- Using the system bundle context to listen and get the required resources when extending the bundle. For example, DS and Blueprint extendees can use the system bundle context to listen to the service events and get ServiceReference object and then use the BundleContext of the extendee bundle to get the actual service object using a ServiceReference. But the extender would also have an additional responsibility to drive the service find hooks registered with the framework to ensure the extendee bundle really does have visibility to the service. One way to do this would be to call one of the find methods on the extendee bundle context using a service filter that matches on the service id. This is a similar responsibility current extenders have if they do not use the extendee bundle's BundleContext to check for permissions and class space consistency (assignability) for the extendee bundles.

4 Requirements

4.1 Service Find Hooks

1. The framework service find hooks must not influence the outcome of the system bundle's BundleContext find service reference methods: all getServiceReference[s] methods.

2. The framework service find hooks must be invoked for all service find operations on the system bundle's BundleContext: all getServiceReference[s] methods.
3. The Collection<ServiceReference> references param passed to the service find hook must have the same mutable characteristics as when service find operations are done using a normal BundleContext (i.e. only allowed to shrink the collection). Removing a service reference from the references collection will cause the service reference to no longer be contained in the service references collection for service find hooks called later.

4.2 Service Event Hooks

4. The framework service event hooks must not influence the delivery of service events to service listeners that are added using the system bundle's BundleContext.
5. The framework service event hooks must be invoked for all service events before delivery to the service listeners added to the system bundle's BundleContext.
6. The Collection<BundleContext> contexts param and the Map<BundleContext, Collection<ListenerInfo>> listeners param passed to the service event hooks must contain the system bundle's BundleContext if at least one service listener has been added to the system bundle's BundleContext.
7. The system bundle's BundleContext must be able to be removed from the contexts collection or the listeners map passed to the service event hooks. Removal of the system bundle's BundleContext from the contexts collection or listeners map will cause the system bundle's BundleContext to no longer be contained in the contexts collection or listeners map for service event hooks called later.

4.3 Bundle Find Hooks

8. The framework bundle find hooks must not influence the outcome of the system bundle's BundleContext find bundle methods: all getBundle[s] methods.
9. The framework bundle find hooks must be invoked for all bundle find operations on the system bundle's BundleContext: all getBundle[s] methods.
10. The Collection<Bundle> bundles param passed to the bundle find hook must have the same mutable characteristics as when bundle find operations are done using a normal BundleContext (i.e. only allowed to shrink the collection). Removing a bundle from the bundles collection will cause the bundle to no longer be contained in the bundles collection for bundle find hooks called later.

4.4 Bundle Event Hooks

11. The framework bundle event hooks must not influence the delivery of bundle events to BundleListeners that are added using the system bundle's BundleContext.
12. The framework bundle event hooks must be invoked for all bundle events before delivery to the bundle listeners added to the system bundle's BundleContext.
13. The Collection<BundleContext> contexts param passed to the bundle event hooks must contain the BundleContext of the system bundle if at least one bundle listener is added to the system bundle's BundleContext.
14. The system bundle's BundleContext must be able to be removed from the contexts collection passed to the bundle event hooks. Removal of the system bundle's BundleContext from the contexts collection will

cause the system bundle's BundleContext no longer be contained in the contexts collection for bundle event hooks called later.

4.5 Bundle Collision Hooks

15. The framework bundle collision hooks must be able to influence the outcome when the system bundle's BundleContext is used to install a bundle: all installBundle methods.
16. The framework bundle collision hooks must be invoked for all bundle install operations on the system bundle's BundleContext: all installBundle methods.
17. The Collection<Bundle> collisionCandidates param passed to the bundle collision hook must have the same mutable characteristics as when bundle install operations are done using a normal BundleContext (i.e. only allowed to shrink the collection). Removing a bundle from the collisionCandidates collection will cause the bundle to no longer be contained in the collisionCandidates collection for bundle collision hooks called later.

5 Technical Solution

5.1 Bundle Event Hooks

The following section is added as a subsection of R5 section 54.3 Bundle Event Hooks

5.1.1 System Bundle Listeners

Bundle Listeners may be registered with the system bundle's Bundle Context. If at least one Bundle Listener is registered with the system bundle's Bundle Context then the system bundle's BundleContext must be contained in the shrinkable collection passed to the event(BundleEvent,Collection) bundle event hook method. Just like other Bundle Context objects contained in the shrinkable collection, the system bundle's Bundle Context may be removed. If the system bundle's Bundle Context is removed by a bundle event hook then the Bundle Context will not be contained in the collection for the next bundle event hooks called. Unlike other Bundle Context objects, if the system bundle's Bundle Context is removed from the shrinkable collection then the bundle event is still delivered to the Bundle Listeners added to the system bundle's Bundle Context.

5.2 Bundle Find Hooks

The following section is added as a subsection of R5 section 54.4 Bundle Find Hooks

5.2.1 System Bundle Context

When the getBundle(long) or getBundles() methods is called using the system bundle's Bundle Context then the bundle find hooks are called in the same way the hooks are called when a normal Bundle Context is used. The system bundle's Bundle Context along with the shrinkable candidate bundles collection is passed to the find(BundleContext context, Collection<Bundle> bundles) bundle find hook method. Bundle find hooks may remove bundles from the shrinkable collection. If a bundle is removed by a bundle find hook then the bundle will not be contained in the collection for the next bundle find hook called. Unlike other Bundle Context objects, if the

system bundle's Bundle Context is used to find bundles then the framework ignores the bundle removals performed by the bundle find hooks and allows the bundles removed to remain visible in the find results.

The bundle find hook is also used during an install operation. When the system bundle's Bundle Context is used to install a bundle then the bundle find hooks are called in the same way the hooks are called when a normal Bundle Context is used. Unlike other Bundle Context objects, if the system bundle's Bundle Context is used to install bundles then the framework ignores the bundle removals performed by the bundle find hooks. This implies that if the system bundle's Bundle Context is used to install a bundle at a location where an existing bundle is installed then the existing bundle is always returned.

5.3 Service Event Listener Hooks

The following section is added as a subsection of the R5 section 55.4 Event Listener Hook. The specification does not have a separate section describing the deprecated service event hook and this RFC also makes no separate section discussing the deprecated service event hook:

5.3.1 System Service Listeners

Service Listeners may be registered with the system bundle's Bundle Context. If at least one Service Listener is registered with the system bundle's Bundle Context then the system bundle's BundleContext must be contained in the keys of the shrinkable map passed to the event(ServiceEvent event, Map<BundleContext, Collection<ListenerInfo>> listeners) service event listener hook method. Just like other BundleContext objects keys contained in the shrinkable map, the system bundle's Bundle Context may be removed. If the system bundle's Bundle Context is removed by a service event listener hook then the Bundle Context will not be contained in the map for the next service event listener hooks called. Unlike other Bundle Context objects, if the system bundle's Bundle Context is removed from the shrinkable map then the service event is still delivered to the Service Listeners added to the system bundle's Bundle Context.

5.4 Service Find Hooks

The following section is added as a subsection of R5 section 55.5 Find Hook

5.4.1 System Bundle Context

When one of the getServiceReference or getServiceReferences methods is called using the system bundle's Bundle Context then the service find hooks are called in the same way the hooks are called when a normal Bundle Context is used. The system bundle's Bundle Context along with the shrinkable candidate service references collection is passed to the find(BundleContext context, String name, String filter, boolean allServices, Collection<ServiceReference<?>> references) service find hook method. Service find hooks may remove service references from the shrinkable collection. If a service reference is removed by a service find hook then the service reference will not be contained in the collection for the next service find hook called. Unlike other Bundle Context objects, if the system bundle's Bundle Context is used to find services then the framework ignores the service reference removals performed by the service find hooks and allows the service references removed to remain visible in the find results.

5.5 Bundle Collision Hooks

The following section is added as a subsection of R5 section 54.5 Bundle Collision Hook

5.5.1 System Bundle Context

When the system bundle's Bundle Context is used to install a bundle then the collision hooks are called in the same way the hooks are called when a normal Bundle Context is used. The system bundle along with the shrinkable candidate bundles collection is passed to the filterCollisions(int operationType, Bundle target, Collection<Bundle> collisionCandidates) bundle collision hook method. Bundle collision hooks may remove bundles from the shrinkable collection. If a bundle is removed by a bundle collision hook then the bundle will not

be contained in the collection for the next bundle collision hook called. Unlike the other bundle and service hooks, if the system bundle's Bundle Context is used to install a bundle then the framework must not ignore the bundle removals performed by the bundle collision hooks. This implies that the system bundle's Bundle Context can be used to install the same bundle multiple times when the *org.osgi.framework.bsnversion* launching property is set to 'managed'.

6 Data Transfer Objects

None needed.

7 Considered Alternatives

1. Considered not calling the framework hooks altogether when the system bundle is the target. Ultimately it was decided that we should still allow the hooks to observe the operations which use the system bundle's BundleContext.
2. Considered allowing a hook to remove objects from the shrinkable collections/maps but then reset the collections/maps for the next hook in the call chain. This seems like an unnecessary burden and overhead for calling out to the hooks.

8 Security Considerations

No additional security concerns. A security manager remains the correct way to secure a system. Framework hooks are not designed to be a lightweight security mechanism.

9 Document Support

9.1 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

9.2 Author's Address

Name	
Company	
Address	
Voice	
e-mail	

9.3 Acronyms and Abbreviations

9.4 End of Document