



Bulk Configuration Updates

Draft

9 Pages

*Text in Red is here to help you. Delete it when you have followed the instructions.
The <RFC Title> can be set from the File>Properties:User Defined menu. To update it onscreen, press F9. To update all of the fields in the document Select All (CTRL-A), then hit F9. Set the release level by selecting one from: Draft, Final Draft, Release. The date is set automatically when the document is saved.*

Abstract

When several configurations have related updates, processing all the updates at once can considerably reduce churn. This behavior can be implemented by a Coordination. The configuration management agent that updates the configuration starts a Coordination, and the consumer of the updates such as DS registers as a participant in the Coordination and tracks the configuration events received during the Coordination. After completion the consumer can process all the updates in a suitable order. Labeling the Coordination with a well known name enables consumers to detect that the Coordination is actually involved in configuration. ManagedService and ManagedServiceFactory can be registered with a similar (or identical) well known name so config admin itself can implement this behavior on their behalf.

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.....	5
3 Problem Description.....	5
4 Requirements.....	5
5 Technical Solution.....	5
6 Data Transfer Objects.....	6
7 Javadoc.....	6
8 Considered Alternatives.....	6

9 Security Considerations.....7**10 Document Support.....7**

10.1 References.....7

10.2 Author's Address.....7

10.3 Acronyms and Abbreviations.....7

10.4 End of Document.....7

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 10.1.

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	06 06 2016	David Jencks, IBM, djencks@us.ibm.com
Update from cpeg call	06 09 2016	David Jencks, IBM, djencks@us.ibm.com

1 Introduction

Configurations can be interdependent in several ways. For instance, DS components can now consume multiple pids. Also, configurations can reference each other in various ways either because the configure linked components or as a (poor) way to represent tree-like configuration information. In these cases, it can considerably reduce churn if all related updates can be persisted in config admin before any configuration events are processed.

2 Application Domain

~~This section should be copied from the appropriate RFP(s). It is repeated here so it can be extended while the RFC authors learn more subtle details.~~ Configuration management agents and configuration consumers wishing to participate in bulk configuration updates.

3 Problem Description

There are several situations in which a configuration management agent may wish to indicate that a set of configuration updates are “one thing” and in which a configuration consumer such as SCR may wish to collect all relevant configuration changes before taking action.

4 Requirements

- C0010 – A configuration management agent must be able to indicate that a set of configuration updates should be treated as a unit of work. This must not need to wait for any asynchronous event delivery for the configuration updates in the unit of work.
- C0020 – A configuration consumer must be able to detect whether a configuration update is part of a larger unit of work and detect when the unit of work is complete so that all updates done as part of the unit can be processed at once.
- C0030 – ManagedService and ManagedServiceFactory instances should be able to be notified of configuration changes part of a unit of work at the end of the unit of work

5 Technical Solution

~~1. A configuration management agent may start a thread local Coordination and mark it as “for configurations” using a well-known variable name. It can complete the coordination when it’s done with supplying configuration changes. No additional delays should be introduced for this coordination.~~

1. Config Admin will examine the thread of incoming requests from configuration management agents for an existing Coordination. If such a coordination is found, Config Admin will regard register as a participant. The actions upon completion are the same whether the coordination succeeds or fails. Similarly participants in related coordinations should take the same action whether the coordination succeeds or fails. No additional delays should be introduced for an incoming Coordination.

2. SynchronousConfigurationListeners can participate directly in this coordination if present if Config Admin chooses to deliver these events on the incoming thread. Otherwise Config Admin must create thread local Coordinations for such delivery threads and complete these coordinations when the incoming coordination completes.

3. A marker interface Delayed must be implemented by ManagedService and ManagedServiceFactory instances that wish to be notified at the end of a coordination rather than during the coordination. Config Admin will do an instanceof check to detect this: the interface does not need to be exposed as an object class.

~~3. Config admin tracks the configuration updates occurring within such a Coordination and creates one or more coordinations attached to the threads it emits configuration events on. These coordinations should also be marked with the well-known variable name.~~

4. When Config Admin detects an incoming coordination, it will create “related” coordinations for each thread used to deliver configuration events and update delayed ManagedService and delayed ManagedServiceFactory updates and deletions for configuration changes made within the scope of the incoming coordination.

~~54. Once all the events and delayed ManagedService and delayed ManagedServiceFactory notifications for configuration updates done in the original incoming Coordination are complete have been sent (and the methods returned) config admin can complete all of it’s the Coordinations related to the original one.~~

6. As a consequence of the thread local Coordinations that may be created by Config Admin, different threads must be used for notifications originating in different (concurrent) Coordinations or outside any Coordination. No ordering is specified in regards to concurrent updates of the same Configuration within different Coordinations.

~~5. Configuration listeners can participate in any coordination with the well-known variable name and should return promptly. They can do the actual work when the coordination completes.~~

~~6. In order to prevent unneeded delays from configuration listeners that aren’t coordination aware, ConfigurationListeners that are coordination aware should register with a well-known service property. Config admin does not need to supply a thread local coordination for non-aware listeners.~~

~~7. ManagedService[Factory] services that wish to be notified on completion of the (config admin, not original) coordination must register using a well-known service property.~~

~~I don’t understand coordinations well enough to know if it would be safe for config admin to attach the same coordination to all threads it uses for configuration events or if it needs a coordination per thread. As long as~~

~~config-admin waits until all configuration event calls return, I don't see a problem. Possibly if some configuration listener terminates the coordination it could cause chaos on other threads that had started nested coordinations (even though this would be extremely bad practice, as the idea is to wait to do work until the coordination completes).~~

6 Data Transfer Objects

No DTOs seem appropriate.

7 Javadoc

~~Javadoc needed for marker interface Delayed. Please include Javadoc of any new APIs here, once the design has matured. Instructions on how to export Javadoc for inclusion in the RFC can be found here: <https://www.osgi.org/members/RFC/Javadoc>~~

8 Considered Alternatives

~~For posterity, record the design alternatives that were considered but rejected along with the reason for rejection. This is especially important for external/earlier solutions that were deemed not applicable:~~

- ~~1. Coordinations must be created for each involved thread as although it is possible to push a Coordination onto a particular thread it is not possible to push a single Coordination onto two threads at the same time.~~

- ~~2. ConfigurationListeners cannot reasonably indicate (e.g. with a service property) that they are "Coordination aware" as they may call arbitrary other code and cannot know if the other code is coordination aware.~~

- ~~3. Marking coordinations as "related to configuration" defeats the general purpose of coordinations. Management agents wishing to not participate in existing thread local Coordinations can pop them off the thread temporarily or do their work on a separate thread.~~

4. A marker interface for delayed ManagedService and ManagedServiceFactory instances is more appropriate than a service property as this is not a configurable aspect of the implementation.

9 Security Considerations

~~Description of all known vulnerabilities this may either introduce or address as well as scenarios of how the weaknesses could be circumvented.~~None known.

10 Document Support

10.1 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

*Add references simply by adding new items. You can then cross-refer to them by chosing <Insert><Cross Reference><Numbered Item> and then selecting the paragraph. **STATIC REFERENCES (I.E. BODGED) ARE NOT ACCEPTABLE, SOMEONE WILL HAVE TO UPDATE THEM LATER, SO DO IT PROPERLY NOW.***

10.2 Author's Address

Name	David Jencks
Company	IBM
Address	
Voice	
e-mail	djencks@us.ibm.com

10.3 Acronyms and Abbreviations

10.4 End of Document
