OSGi™

OPEN SERVICES GATEWAY INITIATIVE

RFC 61 - Bundle Lookup Based on Location

Confidential, Draft

5 Pages

Abstract

Many of the new OSGi services use location as opposed to Bundle id as a handle to a bundle installed on the platform. The location is a convenient way to identify a bundle since it uniquely identifies installed bundles and it can be known a-priori. This means that information about a bundle can be setup before the bundle is installed and kept after it has been removed.

0 Document Information

0.1 Table of Contents

0.2 Status

This document specifies an **API** for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

0.3 Acknowledgment

The authors would like to acknowledge the letters L and B without whom this RFC would not be possible. We also must recognize the tremendous contribution of the number 6.

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 Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997..

`Source code is shown in this typeface.`

0.5 Revision History

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

Revision	Date	Comments
Initial	02/12/03	Initial Draft Benjamin Reed, Core Platform Expert Group, OSGi. breed@almaden.ibm.com.

1 Introduction

There are two ways to uniquely identify a bundle in a framework. The first is the bundle identifier. This identifier is a nonce that is assigned to a bundle when it is installed. Since it is a nonce, it uniquely identifies an installed bundle and will not be assigned to another bundle even after the bundle has been uninstalled.

The bundle location is the second way to identify a bundle. Only a single bundle on the framework can have a given location. It is assigned by the caller when a bundle is installed using the `BundleContext.install()` method. One advantage of using location over the identifier is that the location is known a-priori and it can be reused after a bundle with a given location has been uninstalled.

Bundle references are obtained from the framework using the `BundleContext.getBundles()` method. This method returns an array of all the installed bundles. A bundle programmer can iterate through the array to search for a bundle with a specific identifier or location.

Because the bundle identifier is often used as a persistent handle to a bundle, the `BundleContext.getBundle(long id)` method was added to ease the lookup of a particular `Bundle` reference given a bundle id. Lately, services have been using the location as a way to identify a bundle, but the corresponding `BundleContexet.getBundle(String location)` method is missing. This RFC proposes to remedy this situation by adding such a method.

2 Technical Solution

We propose adding a new `getBundle` method to the `BundleContext` interface. This new method would take a `String` that would specify the location of the desired bundle and would return a `Bundle` object representing that bundle or `null`, if no such bundle exists. Specifically, here is the signature of the new `getBundle` method:

```
public Bundle getBundle(String location);
```

Note, that this method will be in addition to the `getBundle(long id)` method that currently exists in `BundleContext`.

3 Considered Alternatives

It should be noted that, when discussing the first release of the framework, there was a proposal put forth to put the `Bundle` objects in the service registry. This would have obviated the need for any of the `getBundle*` methods on `BundleContext` and allowed for much more powerful searches on bundles by registering properties, such as the location, identifier, and header information, in the service properties of each bundle.

Adoption of such a proposal would have made this current RFC unnecessary. We did not choose to adopt the proposal, and the methods were added to `BundleContext` to obtain references to Bundles.

4 Security Considerations

This RFC does not enable functionality that was not previously available. It just makes a specific task more convenient, so it does not affect the security of the framework.

5 Document Support

5.1 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

5.2 Author's Address

Name	Benjamin Reed
Company	IBM
Address	650 Harry Road
Voice	408 927-1811
e-mail	breed@almaden.ibm.com

5.3 Acronyms and Abbreviations

5.4 End of Document