RFC 56 - Manifest Header For EE Designation

Confidential, Draft

8 Pages

Abstract

The OSGi supports multiple exection environments. This has the consequence that the Framework and management systems must be able to detect for what execution environment a bundle has been written. This RFC defines a manifest header, a syntax for this header, and a number of values for existing EEs.

0 Document Information

0.1 Table of Contents

0.2 Status

This document specifies the Manifest header for the EE designation for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

All Page Within This Box

0.3 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.`

.

0.4 Revision History

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

Revision	Date	Comments
Initial	MAY 20 2002	Peter.Kriens@aQute.se Initial
	JUN 24 2002	Comments from Köln meeting: Removed provided EE and changed the names used in J2ME
	AUG 2 2002	Minor modifications

1 Introduction

This RFP was requested by CPEG, it is in preparation for the SP-R3 release.

2 Application Domain

An OSGi Execution Environment (EE) is a fixed set of methods that must be available in an implementation of a Service Platform. An EE is defined by the OSGi, another standardization/specification body, or a vendor. The OSGi has defined the following EE's:

All Page Within This Box

- RFC 26 – Based on JCP Foundation Profile: http://membercvs.osgi.org/rfcs/rfc0026
- RFC 39 – A small EE: http://membercvs.osgi.org/rfcs/rfc0039

The purpose of an EE is to make the requirements on the runtime environment explicit so that a bundle can be written against such a runtime environment.

EE's are defined on method level. A cleaner approach would have been to decompose an EE on package level but this is unfortunately impossible due to inter-package dependencies. Removing a package must also remove any method in other packages that use objects of class that are members of the removed package.

The OSGi RFC 39 small execution environment is a proper subset of the RFC 26 Foundation Profile EE. However, this will not always be the case. For example, the CLDC profile contains javax.microedition classes that are not part of the J2SE and J2EE profiles.

3 Problem Description

The problem discussed in this RFC has two aspects:

- A management system must be able to compute the feasibility to run a bundle on a Service Platform. This requires knowledge of what methods must be available on the Service Platform.
- A Service Platform must be able to detect that a bundle can actually run with the available classes.

There is thus a need for a description of what the required EE(s) is/are for a bundle before it can run on a service platform.

4 Requirements

4.1 Goal

Define a simple mechanism to designate the required EE(s) for a Service Platform

4.2 Basic Requirements

- Support multiple EE's. Rationale: a bundle could run on multiple EE's that are not a proper subset.

All Page Within This Box

- The Framework implementation must not be required to know about subset/superset relations of EE's.
- Provide rules for compatibility with 2.0 and 3.0 bundles.
- Define naming rules how standard bodies and vendors can develop EE's without conflicting.
- Define the header values for RFC 26 and RFC 39.
- Simplify the common case when a bundle only requires a single EE.
- Use existing standards when possible

5 Technical Solution

5.1 EE Manifest Header

This RFC proposes to introduce a new manifest header: Bundle-RequireExecutionEnvironment. The syntax for this header is:

```
Bundle-RequireExecutionEnvironment := ee [ ',' ee ]

ee := NAME

NAME := <may not contain whitespace or comma>
```

Comparison is not case sensitive.

The Bundle-RequiredExecutionEnvironment header sets the required execution environment for the bundle. The service platform may run this bundle if any of the EE's named in this header matches one of the EE's it implements. This implies that a bundle must only refer to methods that are included in **all** the listed EE's.

Execution environments should be prefixed by the definer's reversed domain name, except for EE's defined by OSGi and SUN's JCP.

```
Bundle-RequiredExecutionEnvironment: se.aQute.Tiny-1.2

Bundle-RequiredExecutionEnvironment: CDC-1.0/Foundation-1.0
```

5.2 Backward compatibility and default rule

Existing bundles will not contain a Bundle-RequiredExecutionEnvironment header and the Framework must thus take a default action. The existing situation is that a bundle is installed regardless of the EE. This will also be the default rule for SP-R3 frameworks.

All Page Within This Box

5.3 Installation

The Bundle-RequireExecutionEnvironment header indicates a pre-requisite to the Framework. If a Bundle with this header is installed, the Framework must verify that the listed execution environments match the available execution environments during the installation of the bundle. When the pre-requisite cannot be fulfilled, the Framework must throw a BundleException during installation with an appropriate.

5.4 Introspection for EE's

The Framework must provide a system property via BundleContext.getProperty(). The name of this property is org.osgi.framework.executionenvironment. This will contain a comma separated list of EE's that are compatible with the framework. Spaces are allowed in this list, like the Package-Import and Package-Export headers.

The format of an entry in this list is the same as in the manifest header Bundle-RequiredExecutionEnvironment.

5.5 Names for existing EE's

There already exists a large number of execution environments from SUN/JCP. The value for the EE header must obviously be compatible with these specifications. Currently, SUN only specifies properties for configurations and profiles that are part of J2ME. An EE is defined as a configuration and a profile. In J2ME, these are 2 different headers/properties. The properties are:

```
microedition.configuration

microedition.profile
```

The EE name will consists of the configuration name + / + the profile name.This means that for example MIDP NG has an EE name of *CLDC-1.0/MIDP-2.0*. The following table contains a number of examples.

RFC 26 Foundation	CDC-1.0/Foundation-1.0
RFC 39 Small EE	OSGi/Minimum-1.0
Java Embedded	JavaEmbedded-1.2
Java Card	JavaCard
MIDP	CLDC-1.0/MIDP-1.0
Personal Java	PersonalJava-1.2.8
Java 2 EE	J2EE-1.2
Java 2 SE	J2SE-1.3

All Page Within This Box

6 Considered Alternatives

No alternatives were considered.

7 Security Considerations

No security issues are considered.

8 Document Support

8.1 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]. J2ME, http://java.sun.com/j2me/

[4]. CDC, http://java.sun.com/products/cdc/

[5]. CLDC device configuration, http://java.sun.com/products/cldc/

8.2 Author's Address

Name	Peter Kriens
Company	aQute
Address	Finnasandsvägen 22

All Page Within This Box

Voice	+46 30039800
e-mail	Peter.Kriens@aQute.se

8.3 Acronyms and Abbreviations

8.4 End of Document