OSGi Vehicle Device Profile

Confidential, Draft

12 Pages

Abstract

OSGi Vehicle Device Profile specifies a minimum set of services that must be implemented in a compliant vehicle terminal.

0 Document Information

0.1 Table of Contents

0.2 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].

All Page Within This Box

Deleted: 12

Deleted: July 19, 2005

Source code is shown in this typeface.

0.3 Revision History

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

Revision	Date	Comments
Initial	Mar 22, 2005	Initial version Olivier Pavé
Second	June 23, 2005	Olivier Pavé. • Change document title to OSGi Vehicle Device Profile; • Remove the split between fundamental and optional services
Third	October 6, 2005	Olivier Pavé Declarative Services added to the Vehicle profile as approved in the last face to face meeting

1 Introduction

OSGi Vehicle Device Profile is a set of OSGi services that MUST be implemented by a compliant Vehicle Device. The purpose of such profile is to avoid market fragmentation and provide a solid minimum platform for application developers. In the future, this set may be extended; therefore, the versioning of profiles is introduced. This document defines the 1.0 version of OSGi Vehicle Device Profile.

As long as the profile is defined as a list of service or similarly high-level features, verify compliance to the profile is easy using the OSGi compliance program; it is enough to compare the list of supported services with the list provided in this RFC.

All Page Within This Box

2 Application Domain

The profile defined in this RFC is applicable to any Vehicle systems.

3 Problem Description

The OSGi Service Platform Specification becomes very rich and has a larger scope. In this context, the specification is now decomposed in several documents:

- Platform Implementer Specification

- Bundle Developer Guide

- Execution Environment

- Collection of data sheets of each services and utility classes

- Mobile Device Specification

- Vehicle Device Specification

The purpose of this RFC is to define the content of this last specification.

4 Requirements

The main requirement is to define and share a common view on OSGi services that must be implemented by a Vehicle Device.

The OSGi Vehicle Device Profile is defined by two lists of OSGi services:

- Fundamental Services

All Page Within This Box

These services MUST BE implemented by a compliant Vehicle Device.

- Utility Classes

    These classes MUST BE provided by a compliant Vehicle Device.

5 Technical Solution

The following tables will list the services that an OSGi Vehicle Device Profile 1.0 compliant device MUST implement. They also contain references to the detailed definition of the services.

In OSGi, the services themselves are versioned. For each service, the version included in OSGi Release 4 specification should be referenced by OSGi Vehicle Device Profile 1.0.

NOTE: RFC references should be replaced with a reference to the corresponding section of the OSGi specification, when it is compiles. RFC references are used for the time being.

NOTE: There are some RFCs defining modifications to existing services. Whenever an exiting service is referred, the reference is meant to target the new, modified version of the service, i.e. the one that will be included in R4. These references in this document are pointing to the corresponding chapter of R3 specification. This needs to be updated in the R4 spec.

5.1 Fundamental Services

The services listed in this section are required in an Vehicle Device:

Service	Notes	Reference
OSGi Framework	This is mandatory part of all OSGi implementations. *NOTE: Framework is updated by a number of RFCs for R4. There will be a single Framework specification in R4 which implies that all of the following RFCs must be included:* • *RFC-59 – Service Filter Changes in R4* • *RFC-70 – Bundle Class Loading Changes* • *RFC-71 – Framework API Changes* • *RFC-72 – Native Code Changes*	Chapter 4 of [3]

All Page Within This Box

Deleted: 12

Deleted: July 19, 2005

Service	Notes	Reference
	• *RFC-73 – Permission Update* • *RFC-74 – Manifest Localization* • *RFC-79 – Framework Modularization* `org.osgi.framework`	
Package Admin Service	Bundles can export packages to other bundles. This exporting creates a dependency between the bundle exporting a package and the bundle using the package. When the exporting bundle is uninstalled or updated, a decision must be taken regarding any shared packages. The Package Admin service provides an interface to let the Management Agent make this decision. `org.osgi.service.packageadmin`	Chapter 5 of [3]
Permission Admin Service	Permission Admin Service enables Management Agent to manage policy. `org.osgi.service.permissionadmin`	Chapter 7 of [3]
Log Service	The Log Service provides a general-purpose message logger for the OSGi Service Platform. It consists of two services, one for logging information and another for retrieving current or previously recorded log information. `org.osgi.service.log`	Chapter 9 of [3]
Configuration Admin Service	The Configuration Admin service allows an Operator to set the configuration information of deployed bundles. `org.osgi.service.cm`	Chapter 10 of [3]
User Admin Service	User Admin implements a user database, which stores logon credentials (username and password) and enable the creation of user groups. `org.osgi.service.useradmin`	Chapter 12 of [3]
Preferences Service	Service used to save some data persistently dedicated to a user or to the system. `org.osgi.service.prefs`	Chapter 15 of [3]
Power Manager Service	Power Manager handles system power state changes and maintain a consistency with device power states `org.osgi.service.power`	RFC-96

All Page Within This Box

Service	Notes	Reference
Start Level Service	Enables ordering startup/shutdown of bundles, and controlling activity level of the system. `org.osgi.service.startlevel`	Chapter 6 of [3]
URL Handlers	This specification defines how to register new URL schemes and how to convert content-typed java.io.InputStream objects to specific Java objects. `org.osgi.service.packageadmin`	Chapter 8 of [3]
Device Access	The Device Access specification supports the coordination of automatic detection and attachment of existing devices on an OSGi Service Platform, facilitates hot-plugging and – unplugging of new devices, and downloads and installs device drivers on demand. `org.osgi.service.device`	Chapter 11 of [3]
IO Connector Service	IO Connector Service specification adopts the Java 2 Micro Edition (J2ME) javax.microedition.io packages as a basic communications infrastructure. In J2ME, this API is also called the Connector framework. A key aspect of this framework is that the connection is configured by a single string, the URI. `org.osgi.service.io`	Chapter 13 of [3]
Http Service	An OSGi Service Platform normally provides users with access to services on the Internet and other networks. This access allows users to remotely retrieve information from, and send control to, services in an OSGi Service Platform using a standard web browser. `org.osgi.service.http`	Chapter 14 of [3]
Wire Admin Service	The Wire Admin service is an administrative service that is used to control a wiring topology in the OSGi Service Platform. It is intended to be used by user interfaces or management programs that control the wiring of services in an OSGi Service Platform. `org.osgi.service.wireadmin`	Chapter 16 of [3]
XML Parser Service	This specification addresses how the classes defined in JAXP can be used in an OSGi Service Platform. It defines how: • Implementations of XML parsers can become available to other bundles	Chapter 17 of [3]

All Page Within This Box

Service	Notes	Reference
	• Bundles can find a suitable parser • A standard parser in a JAR can be transformed to a bundle `org.osgi.util.xml`	
Metatype Service	The Metatype specification defines interfaces that allow bundle developers to describe attribute types in a computer readable form using so-called metadata. `org.osgi.service.metatype`	Chapter 18 of [3] RFC-62
Metatype2 Service	The Metatype2 specification defines interfaces that allow bundle developers to describe complex attribute and actions types in a computer readable form using so-called metadata. `org.osgi.service.metatype2`	RFC-69
Event Admin Service	Defines a generic event mechanism, which extends the even mechanism defined in the framework. `org.osgi.service.event`	RFC-97
Control Unit Admin Service	Control unit is an abstraction of device capabilities defined by actions and state variables. Control Unit Admin manages the creation/deletion and activation of control units. `org.osgi.service.cu.*`	RFC-82
Diagnostic	Diagnostic defines additional interfaces to control unit to publish and invoke diagnostic commands. `org.osgi.service.cu.diag`	RFC-77
Declarative Services	Currently a bundle has to be active in order to offer a service(s) to other bundles. An active bundle consumes system resources such as ClassLoaders, loaded classes, created objects, etc. It is desirable to minimize the amount of resources consumed by a bundle, when a service(s) offered by it is not being used. This RFC describes a declarative service model, which allows a bundle to delay instantiating the service object until they are needed, thus minimizing the resource consumption at any point of time. `org.osgi.service.component`	RFC-80

5.2 Utility Classes

The classes listed in this section are considered useful and crucial for the application and/or service developers; therefore, they were included in the Profile.

Service	Notes	Reference
Service Tracker	ServiceTracker is a utility class, which makes tracking the registration, modification, and deregistration of services much easier. `org.osgi.util.tracker`	Chapter 19 of [3]
Measurement and State	The Measurement class is a utility that provides a consistent way of handling a diverse range of measurements for bundle developers. `org.osgi.util.measurement`	Chapter 20 of [3]
Position	The Position class is a utility that provides a consistent way of handling positions. `org.osgi.util.position`	Chapter 21 of [3]

5.3 Specification Content Summary

Here is a summary of the VEG Specification content.

Services / Classes	Version	Services	Utilities
Framework	1.3		
Package Admin Service	1.2		
Permission Admin Service	1.1		
Log Service	1.2		
Configuration Admin Service	1.2		
User Admin Service	1.0		
Preferences Service	1.1		
Power Manager Service	1.0		
Start Level Service	1.0		
URL Handler	1.0		

All Page Within This Box

Device Access	1.1		
IO Connector Service	1.0		
Http Service	1.1		
Wire Admin Service	1.0		
XML Parser Service	1.0		
Metatype Service	1.1		
Metatype2 Service	1.0		
Event Admin Service	1.0		
Control Unit Admin Service	1.0		
Diagnostic	1.0		
Declarative Services	1.0		
Service Tracker	1.2		
Measurement & State	1.0		
Position	1.0		

5.4 Profile versioning

OSGi Vehicle Device Profile is likely to be extended in the future; therefore, version of the profiles is necessary. This specification defines the 1.0 version.

The profile version numbers should follow the <major>.<minor> scheme but no particular semantics is associated with the major and minor components.

6 Considered Alternatives

The following services were also considered for inclusion but not approved for version 1.0 of the OSGi Vehicle Device Profile:

Service	Notes	Reference
Initial Provisioning	This specification defines how the Management Agent can make its way to the Service Platform, and gives a structured view of the problems and their corresponding resolution methods. `org.osgi.service.provisioning`	Chapter 26 of [3]

7 Security Considerations

None.

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].   OSGi R3 specification

8.2 Author's Address

Name	Olivier Pavé
Company	Siemens VDO Automotive
Address	Bâtiment Alpha 80, route des lucioles – BP 305 06906 Sophia-Antipolis Cedex

All Page Within This Box

| Voice | +33 (0) 492 381 129 |
| e-mail | olivier.pave@siemens.com |

8.3 Acronyms and Abbreviations

8.4 End of Document