

1 Introduction

The OSGi™ Alliance was founded in March 1999. Its mission is to create open specifications for the network delivery of managed services to local networks and devices. The OSGi organization is the leading standard for next-generation Internet services to homes, cars, small offices, and other environments.

The OSGi service platform specification delivers an open, common architecture for service providers, developers, software vendors, gateway operators and equipment vendors to develop, deploy and manage services in a coordinated fashion. It enables an entirely new category of smart devices due to its flexible and managed deployment of services. The primary targets for the OSGi specifications are set top boxes, service gateways, cable modems, consumer electronics, PCs, industrial computers, cars and more. These devices that implement the OSGi specifications will enable service providers like telcos, cable operators, utilities, and others to deliver differentiated and valuable services over their networks.

This is the fourth release of the OSGi service platform specification developed by representatives from OSGi member companies. The OSGi Service Platform Release 4 mostly extends the existing APIs into new areas. The few modifications to existing APIs are backward compatible so that applications for previous releases should run unmodified on release 4 Frameworks. The built-in version management mechanisms allow bundles written for the new release to adapt to the old Framework implementations, if necessary.

1.1 Reader Level

This specification is written for the following audiences:

- Application developers
- Framework and system service developers (system developers)
- Architects

This specification assumes that the reader has at least one year of practical experience in writing Java programs. Experience with embedded systems and server environments is a plus. Application developers must be aware that the OSGi environment is significantly more dynamic than traditional desktop or server environments.

System developers require a *very* deep understanding of Java. At least three years of Java coding experience in a system environment is recommended. A Framework implementation will use areas of Java that are not normally encountered in traditional applications. Detailed understanding is required of class loaders, garbage collection, Java 2 security, and Java native library loading.

Architects should focus on the introduction of each subject. This introduction contains a general overview of the subject, the requirements that influenced its design, and a short description of its operation as well as the entities that are used. The introductory sections require knowledge of Java concepts like classes and interfaces, but should not require coding experience.

Most of these specifications are equally applicable to application developers and system developers.

1.2 Version Information

This document specifies [2] *OSGi Service Platform, Release 4*. This specification is backward compatible to releases 3.

Components in this specification have their own specification-version, independent of the OSGi Service Platform, Release 4 specification. The following table summarizes the packages and specification-versions for the different subjects.

Table 1.1 Packages and versions

Item	Package	Version	Opt
Framework	org.osgi.framework	Version 1.3	
Package Admin Service Specification	org.osgi.service.packageadmin	Version 1.2	
Conditional Permission Admin Specification	org.osgi.service.condpermissionadmin	Version 1.0	Yes
Permission Admin Service Specification	org.osgi.service.permissionadmin	Version 1.2	
3 Mobile Management Tree		Version 1.0	
101 Log Service Specification	org.osgi.service.log	Version 1.3	
104 Configuration Admin Service Specification	org.osgi.service.cm	Version 1.2	
105 Metatype Service Specification	org.osgi.service.metatype	Version 1.1	
109 IO Connector Service Specification	org.osgi.service.io	Version 1.0	Yes
112 Declarative Services Specification	org.osgi.service.component	Version 1.0	
113 Event Admin Service Specification	org.osgi.service.event	Version 1.0	
114 Deployment Admin Specification	org.osgi.service.deploymentadmin org.osgi.service.deploymentadmin spi	Version 1.0	
115 Auto Configuration Specification		Version 1.0	
116 Application Admin Service Specification	org.osgi.service.application	Version 1.0	

Table 1.1 Packages and versions

Item	Package	Version	Opt
117 DMT Admin Service Specification	info.dmtree info.dmtree.spi info.dmtree.security info.dmtree.notification info.dmtree.notification.spi info.dmtree.registry	Version 1.0	
118 Mobile Conditions Specification	org.osgi.util.gsm org.osgi.util.mobile	Version 1.0	Yes
119 Monitor Admin Service Specification	org.osgi.service.monitor	Version 1.0	Yes
120 Foreign Application Access Specification	org.osgi.application	Version 1.0	
701 Service Tracker Specification	org.osgi.util.tracker	Version 1.3	
702 XML Parser Service Specification	org.osgi.util.xml	Version 1.0	Yes

When a component is represented in a bundle, a specification-version is needed in the declaration of the Import-Package or Export-Package manifest headers.

1.3 Non Functional Requirements

1.3.1 Framework Optionality

The OSGi Release 4 Core specification on which the Mobile Specifications are based defines a number of aspects as optional:

- Fragments
- Require-Bundle

This specification inherits the optionality. Compliant implementations are not required to implement these optionalities, however, when they do, they must conform strictly to the Core specification. This implies that bundle developers cannot rely on the presence of these features in a compliant device. Bundles that use any of the optional features must not be allowed to install on a device that does not implement these features.

1.3.2 Security Formats and Algorithms

Required certificate formats:

- [3] *X.509 Certificates*

Required Digest Algorithms:

- [4] *Secure Hash Algorithm 1*
- [5] *RFC 1321 The MD5 Message-Digest Algorithm*

Required Signing Algorithm

- [7] *RSA*

1.3.3 Device Management

- Node name lengths of 32 bytes or more must be supported. This length is defined upon the unescaped, UTF-8 encoded name.
- URI lengths of 255 bytes or more must be supported. This length is based on the same encoding as previous bullet.
- The minimum number of segments that must be supported in a URI is 16.

1.3.4 Execution Environment

This specification requires a Profile that meets the [9] *OSGi Minimum Execution Environment Version 1.1*.

1.3.5 Configuration Admin

Property key names used in the Configuration Admin service must be less or equal than 32 bytes when encoded in UTF-8 to prevent from being mangled in the DMT Admin.

1.4 References

- [1] *Bradner, S., Key words for use in RFCs to Indicate Requirement Levels*
<http://www.ietf.org/rfc/rfc2119.txt>, March 1997.
- [2] *OSGi Service Platform, Release 4*
<http://www.osgi.org/download>
- [3] *X.509 Certificates*
<http://www.ietf.org/rfc/rfc2459.txt>
- [4] *Secure Hash Algorithm 1*
<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf>
- [5] *RFC 1321 The MD5 Message-Digest Algorithm*
<http://www.ietf.org/rfc/rfc1321.txt>
- [6] *DSA*
<http://www.itl.nist.gov/fipspubs/fip186.htm>
- [7] *RSA*
<http://www.ietf.org/rfc/rfc2313.txt> which is superseded by
<http://www.ietf.org/rfc/rfc2437.txt>
- [8] *Public Key Cryptography Standard #7*
<http://www.rsasecurity.com/rsalabs/node.asp?id=2129>
- [9] *OSGi Minimum Execution Environment Version 1.1*
<http://www.osgi.org/download>
- [10] *Open Mobile Alliance*
http://www.openmobilealliance.org/release_program/index.html