



Management CIM Model

Confidential, Draft B
RFC 19

12 Pages

Abstract

Modelling the OSGi Bundle and its life cycle in CIM enables CIM-based remote management of gateways. Such a model, and its application to remote management, is proposed.

Copyright © The Open Services Gateway Initiative (2000). All Rights Reserved. This information contained within this document is the property of OSGi and its use and disclosure are restricted.

Implementation of certain elements of the Open Services Gateway Initiative (OSGi) Specification 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 OSGi). OSGi 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.

This document and the information contained herein are provided on an "AS IS" basis and OSGi DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL OSGi BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All Company, brand and product names may be trademarks that are the sole property of their respective owners.

The above notice and this paragraph must be included on all copies of this document that are made.

0 Document Information

0.1 Table of Contents

0 Document Information.....	2
0.1 Table of Contents.....	2
0.2 Status	2
0.3 Acknowledgement.....	2
0.4 Terminology and Document Conventions.....	3
0.5 Revision History	3
1 Introduction	3
2 Motivation and Rationale	4
3 Technical Discussion	4
3.1 Introduction	4
3.2 Bundle Life Cycle	4
3.3 Bundle Dependencies	6
4 Security Considerations.....	7
5 Appendix.....	7
5.1 Appendix A: OSGi_Core 1.0 MOF	7
6 Document Support.....	12
6.1 References.....	12
6.2 Author's Address.....	12
6.3 Acronyms and Abbreviations	12
6.4 End of Document	12

0.2 Status

This document specifies a CIM model of the OSGi Bundle and its life cycle for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

0.3 Acknowledgement

The author wishes to thank Michael Beauregard and Zach Scott for their invaluable CIM prototyping and modeling work.

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 [1].

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	Feb 21 2001	Initial version. Ted Goddard, Wind River. ted.Goddard@windriver.com.
Draft B	Mar 8 2001	Added MOF appendix, renamed Java_Package to OSGi_JavaPackage, and changed OSGi_Bundle.ClassPath type to string[].

1 Introduction

The Common Information Model (CIM) [2] provides a comprehensive schema for modeling hardware and software systems. Several commercially available management systems from a variety of vendors use this model and its associated protocols for managing large numbers of computer systems ranging from network infrastructure (routers, switches) to servers (file servers, computational clusters) and desktop machines.

2 Motivation and Rationale

This document presents a model of the OSGi Bundle and its life-cycle with the goal of enabling CIM-based management of OSGi gateways. This is the most basic aspect of the OSGi gateway that must be modeled so that it can be managed by a CIM-based management system.

3 Technical Discussion

3.1 Introduction

In order to manage the software load of an OSGi gateway, it is necessary to be able to manage the life cycles and relationships of the bundles on that gateway. To achieve this with CIM-based systems, we have modeled the OSGi bundle in the context of CIM, software actions that manipulate a bundle's life cycle, and dependency relationships between bundles. This is organized into the following two sections for presentation purposes.

3.2 Bundle Life Cycle

Figure 1. illustrates the proposed CIM model of bundles and management actions on bundles. Please refer to Appendix A for the MOF representation of this model.

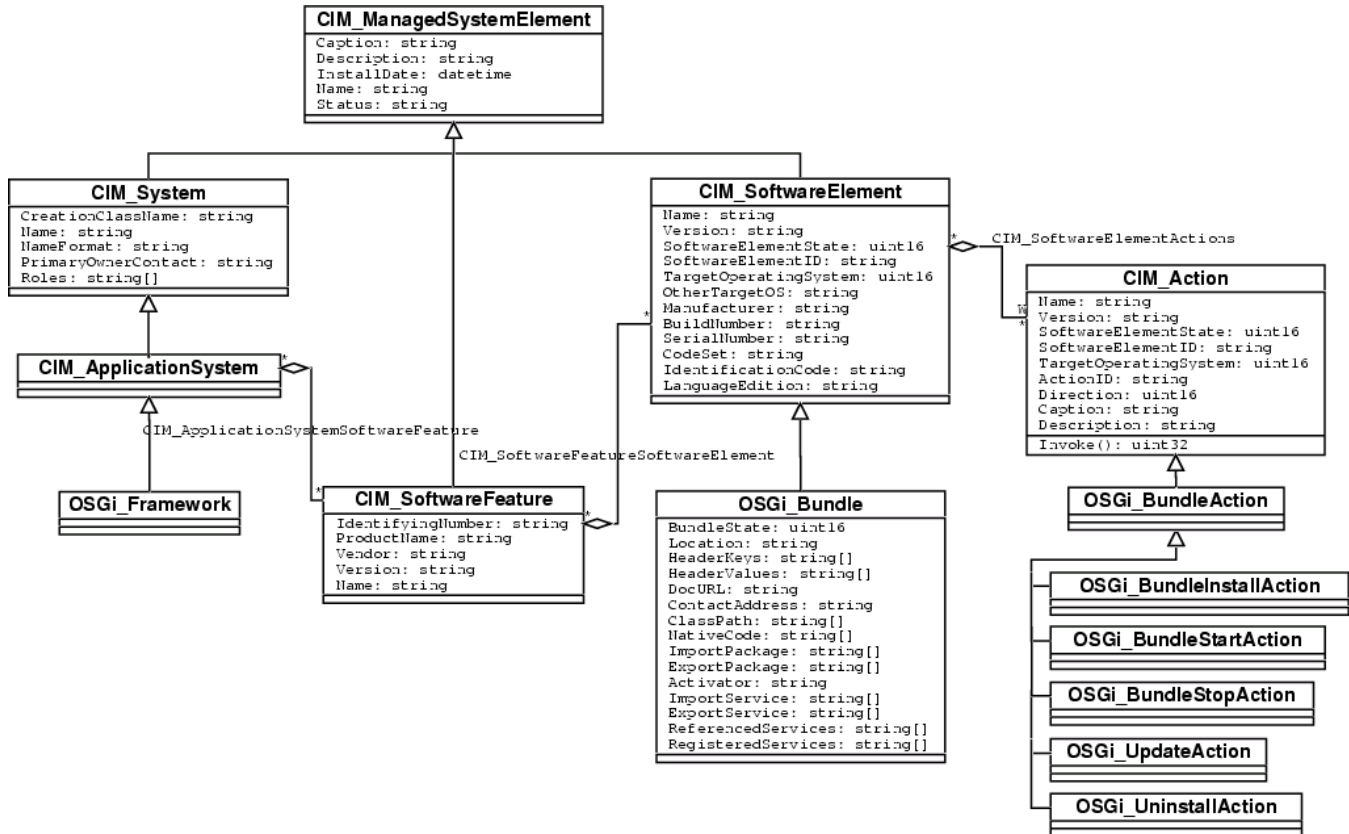


Figure 1

This model is intended to be used as follows: an instantiation of `OSGi_Bundle` need only be initialized with `Location` and `BundleState` meaningfully set. From there, a sequence of `OSGi_BundleInstallAction` and `OSGi_BundleStartAction` will lead to the subsequent installation and execution of the bundle. Remaining properties in `OSGi_Bundle` shall be set once the deployment is complete. Bundles present in the framework can be queried for and categorized by their `BundleState`. Thus, `OSGi_Framework` does not play as strong a role in this model as `org.osgi.framework.BundleContext` plays in the Java APIs, but does serve to aggregate its associated bundles.

Most of the `OSGi_Bundle` properties are directly extracted from the bundle's manifest headers (with parsing into string arrays for convenience in some cases) but a few warrant more explanation. The `HeaderKeys` and `HeaderValues` properties are string arrays of the same size with a model correspondence between them. Thus, the header value corresponding to a given header key is obtained from the `HeaderValues` array at the same index as the key in the `HeaderKeys` array. `ImportService` and `ExportService` are convenience properties directly extracted from the manifest headers. The `ReferencedServices` and `RegisteredServices` properties contain respectively the current services in use by the bundle (as in the Java API `Bundle.getServicesInUse()`) and The current services registered by the bundle (as in the Java API `Bundle.getRegisteredServices()`). These properties are only valid for a bundle at runtime.

3.3 Bundle Dependencies

The above model allows bundles to be managed, but such management would be “blind” -- without regard to the consequences on surrounding bundles. Management of this nature may be acceptable for a developer console, but it is not suitable for a management server with many clients tended over low bandwidth connections.

Figure 2. presents a model that allows the relationships between bundles to be represented. Please refer to Appendix A for the MOF representation of this model. The most interesting aspect of bundle inter-relationship is the fact that bundles do not really depend on other bundles; they depend on the packages contained within those bundles (although `OSGi_BundleBundleDependency` is included to capture runtime bundle relationships). To model the static dependency through packages, we introduce `OSGi_JavaPackage` together with `OSGi_BundlePackageDependency` (to indicate which bundles need which packages) and `OSGi_BundleExportsPackage` (to indicate which bundles provide which packages). It is then up to the management system to analyze these dependencies and exports to determine the effects of its future management actions. It may desirable to revise this schema in the future if a class corresponding to `OSGi_JavaPackage` is defined in a general Java CIM schema.

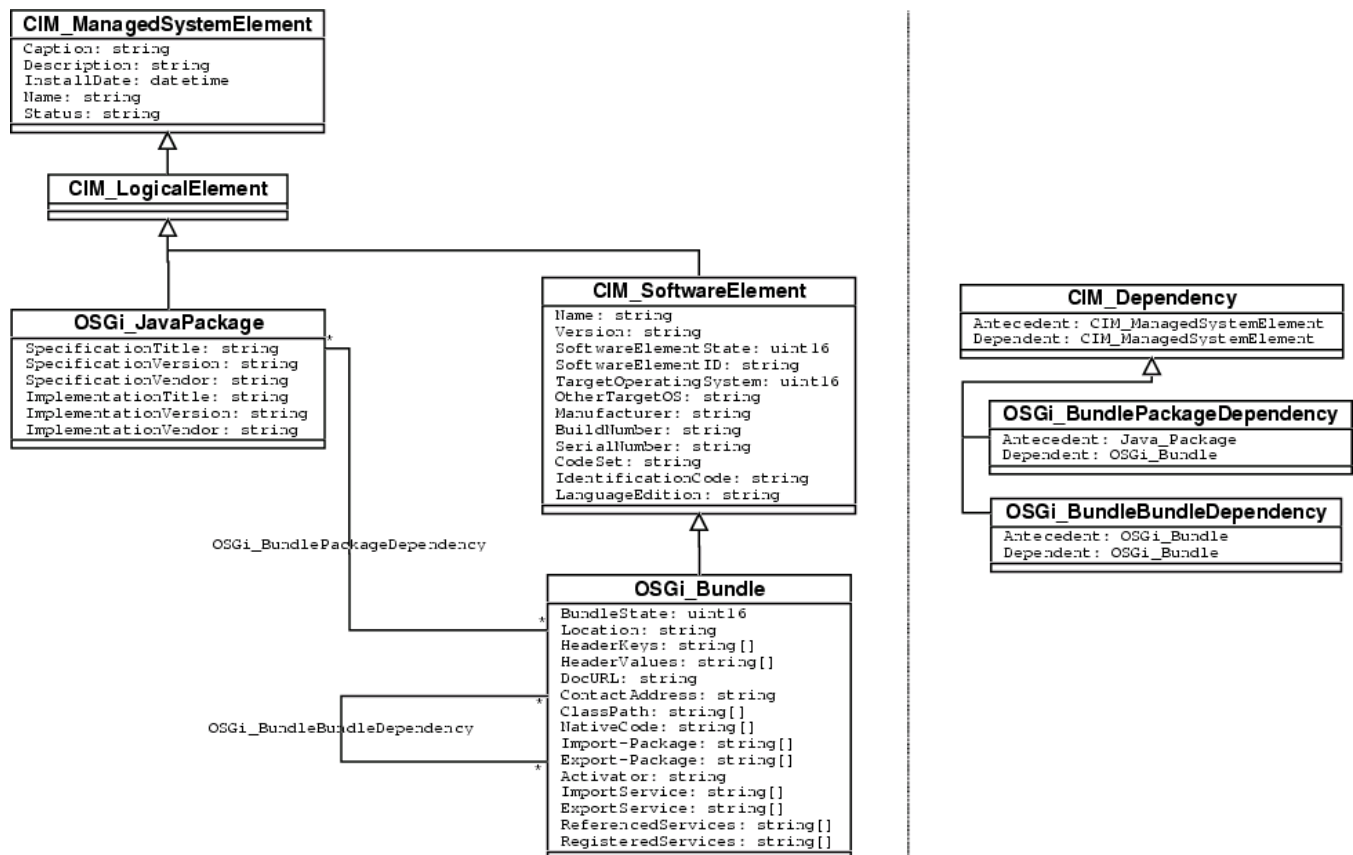


Figure 2

4 Security Considerations

The model itself does not have any security considerations, however, the application of this model for purposes of remote management of gateways, must be considered very carefully.

5 Appendix

5.1 Appendix A: OSGi_Core 1.0 MOF

```
// =====
// Title:      OSGi Core MOF specification 1.0
// Filename:    OSGi_Core10.mof
// Version:     1.0
// Release:     0
// Date:        03/06/2001
// Description: The object classes below are listed in an order that
//              avoids forward references. Required objects, defined
//              by other working groups, are omitted.
// =====
// Author:      Wind River Systems
// =====

#pragma locale ("en_US")

// =====
// OSGi_Framework
// =====
[Description (
  "The OSGi_Framework class is used to represent an OSGi "
  "framework. The aggregation of Bundles for a particular "
  "framework can be located using the "
  "CIM_ApplicationSystemSoftwareFeature to find the "
  "CIM_SoftwareFeature elements associated with the Bundles." )]
class OSGi_Framework:CIM_ApplicationSystem
{
};
```



```
// =====
//   OSGi_Bundle
// =====
[Description (" The OSGi_Bundle class is derived from "
"CIM_SoftwareElement for the purpose of representing OSGi Bundles. "
"An OSGi_Bundle object captures the management details of a Bundle "
"in one of six states characterized by the "
"BundleState property, which is associated with the SoftwareElementState "
"property of CIM_SoftwareElement.  ") ]
class OSGi_Bundle:CIM_SoftwareElement
{
    [key, Description (
        " The BundleState is defined in this model to identify various "
        "states of a Bundle's life cycle. BundleState is associated "
        "with SoftwareElementState as indicated by the permitted values.\n"
        " - A Bundle in the UNINSTALLED state must be in SoftwareElementState "
        "Deployable.\n. An UNINSTALLED Bundle may be known to the system "
        "by location only."
        " - A Bundle in the INSTALLED state must be in SoftwareElementState "
        "Installable. Bundles in the INSTALLED state cannot move to "
        "the SoftwareElementState Executable until all of their "
        "dependencies are satisfied, in which case they move to the "
        "RESOLVED state\n"
        " - A Bundle in the RESOLVED state must be in SoftwareElementState "
        "Executable. The RESOLVED state indicates that all of a Bundle's "
        "dependencies are satisfied, and it may be started.\n"
        " - A Bundle in the STARTING state must be in SoftwareElementState "
        "Running. STARTING is a transient state preceeding ACTIVE.\n"
        " - A Bundle in the STOPPING state must be in SoftwareElementState "
        "Running. STOPPING is a transient state preceeding RESOLVED on "
        "a Bundle being stopped.\n"
        " - A Bundle in the ACTIVE state must be in SoftwareElementState "
        "Running. Started Bundles are ACTIVE until they are stopped.\n") ,
        Values {"UNINSTALLED", "INSTALLED", "RESOLVED",
        "STARTING", "STOPPING", "ACTIVE"} ]
    uint16 BundleState ;
    [key, maxlen (1024) , Description (
        "The URL for the Bundle's jar file.") ]
    string Location ;
    [Description (
        "The names in the Bundle's jar file manifest headers."),
        ModelCorrespondence {"OSGi_Bundle.HeaderValues"} ]
    string HeaderKeys[] ;
    [Description (
        "The values in the Bundle's jar file manifest headers."),
        ModelCorrespondence {"OSGi_Bundle.HeaderKeys"} ]
    string HeaderValues[] ;
    [maxlen (1024) , Description (
        "The URL for the Bundle's documentation.") ]
    string DocURL ;
    [Description ("A contact address for reporting problems with "
        "the bundle.") ]
    string ContactAddress ;
    [Description (
        "The /-separated paths to jar files within the Bundle's jar file "
        "that are to be placed on the Bundle's classpath.") ]
    string ClassPath[] ;
```




```
[Description (
    "The native code clauses for the Bundle's contained native "
    "code libraries.") ]
string NativeCode[] ;
[Description (
    "The collection of packages imported by the Bundle.") ]
string ImportPackage[] ;
[Description (
    "The collection of packages exported by the Bundle.") ]
string ExportPackage[] ;
[Description (
    "The Class name of the Bundle's Activator.") ]
string Activator ;
[Description (
    "The collection of services that the Bundle is expected to use, as "
    "declared in its manifest.") ]
string ImportService[] ;
[Description (
    "The collection of services that the Bundle is expected to register, "
    "as declared in its manifest.") ]
string ExportService[] ;
[Description (
    "The collection of services actually in use by the Bundle.") ]
string ReferencedServices[] ;
[Description (
    "The collection of services actually registered by the Bundle.") ]
string RegisteredServices[] ;
};

// =====
//   OSGi_BundleAction
// =====
[Abstract, Description (
    "The OSGi_BundleAction is an abstract class that is used for "
    "actions on OSGi_Bundle objects.") ]
class OSGi_BundleAction:CIM_Action
{
};

// =====
//   OSGi_BundleInstallAction
// =====
[Description (
    "The OSGi_BundleInstallAction installs the Bundle from its "
    "specified location.") ]
class OSGi_BundleInstallAction:OSGi_BundleAction
{
};

// =====
//   OSGi_BundleStartAction
// =====
[Description (
    "The OSGi_BundleStartAction starts the associated Bundle.") ]
class OSGi_BundleStartAction:OSGi_BundleAction
{
};
```



```
// =====
//   OSGi_BundleStopAction
// =====
    [Description (
        "The OSGi_BundleStopAction stops the associated Bundle.") ]
class OSGi_BundleStopAction:OSGi_BundleAction
{
};

// =====
//   OSGi_BundleUpdateAction
// =====
    [Description (
        "The OSGi_BundleUpdateAction updates the Bundle from its "
        "specified location.") ]
class OSGi_BundleUpdateAction:OSGi_BundleAction
{
};

// =====
//   OSGi_BundleUninstallAction
// =====
    [Description (
        "The OSGi_BundleUninstallAction uninstalls the associated Bundle.") ]
class OSGi_BundleUninstallAction:OSGi_BundleAction
{
};

// =====
//   OSGi_JavaPackage
// =====
    [Description (
        "The OSGi_JavaPackage class is used to represent Java "
        "packages as modeled specifically for OSGi purposes." )]
class OSGi_JavaPackage:CIM_LogicalElement
{
    [Description (
        "Title of the specification") ]
    string SpecificationTitle ;
    [Description (
        "Version number of the specification in major.minor.micro format "
        "where these are defined as follows:\n "
        "Major version numbers identify significant functional changes.\n"
        "Minor version numbers identify smaller extensions to the functionality.\n"
        "Micro versions are even finer grained versions.\n"
        "These version numbers are ordered with larger numbers specifying "
        "additions to the specification.") ]
    string SpecificationVersion ;
    [Description (
        "Vendors company or organization") ]
    string SpecificationVendor ;
    [Description (
        "Title of the package") ]
    string ImplementationTitle ;
};
```



```
[Description (
  "Package implementation version number. "
  "Unlike version numbers for specifications, version information for "
  "implementations cannot be used to identify the package as being "
  "backward compatible with earlier versions. Package version numbers "
  "are present to identify differences between the specification and the "
  "implementation, i.e. bugs. New versions of implementations are "
  "specifically produced to remove (undesirable or incorrect) behavior "
  "and thus are intended not to be backward compatible. Therefore, "
  "package version strings can have any unique value and can only be "
  "compared for equality. ") ]
string ImplementationVersion ;
[Description (
  "Vendors company or organization") ]
string ImplementationVendor ;
};

// =====
//   OSGi_BundlePackageDependency
// =====
[Description (
  "The OSGi_BundlePackageDependency represents the dependency of "
  "an OSGi Bundle on a Java package."
  " The cardinality of this association is many-to-many. A Bundle may "
  "depend on more than one Package, and more than one Bundle "
  "may depend on a given Package.") ]
class OSGi_BundlePackageDependency:CIM_Dependency
{
  [Min (0) , Max (NULL), Override ("Antecedent") ]
  OSGi_JavaPackage REF Antecedent;
  [Min (0) , Max (NULL), Override ("Dependent") ]
  OSGi_Bundle REF Dependent;
};

// =====
//   OSGi_BundleBundleDependency
// =====
[Description (
  "The OSGi_BundleBundleDependency represents the dependency of "
  "one OSGi Bundle upon another (especially a runtime "
  "relationship based on the Java packages exported by the antecedent "
  "bundle). "
  " The cardinality of this association is many-to-many. A Bundle may "
  "depend on more than one Bundle, and more than one Bundle "
  "may depend on a given Bundle.") ]
class OSGi_BundleBundleDependency:CIM_Dependency
{
  [Min (0) , Max (NULL), Override ("Antecedent") ]
  OSGi_Bundle REF Antecedent;
  [Min (0) , Max (NULL), Override ("Dependent") ]
  OSGi_Bundle REF Dependent;
};
```

6 Document Support

6.1 References

- [1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.
- [2]. CIM specifications <http://www.dmtf.org/spec/cims.html>

6.2 Author's Address

Name	Ted Goddard
Company	Wind River Systems Inc.
Address	#180, 6815-8 th St NE, Calgary, Alberta, Canada
Voice	(403) 730-7590
e-mail	Ted.Goddard@windriver.com

6.3 Acronyms and Abbreviations

6.4 End of Document