# RFC 30 - XML Format for Bundle Management Information

Members Only, Draft
<Document Number Here>

14 Pages

## Abstract

Management Systems need to access information about the configurable properties of a bundle. Not all management systems will want to use the Java interfaces defined in the Metatype Specification [2]. As a possible solution, this RFC proposes an xml vocabulary for describing bundle configuration information and a way for a bundle to point to the location of its configuration information.

# 0 Document Information

## 0.1 Table of Contents

## 0.2 Status

This document specifies an xml vocabulary for describing bundle configuration information for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

## 0.3 Acknowledgement

Anders Joelson, Kaspar Helldén, and Gunnar Ekolin all reviewed and gave valuable input to this document.

## 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 | 08/27/01 | Initial revision.<br>Per Gustafson,<br>perg@gatespace.com. |

# 1 Introduction

# 2 Motivation and Rationale

# 3 Technical Discussion

## 3.1 Solution Overview

The proposed solution consists of an xml vocabulary for describing bundle configuration information and a bundle manifest header that is an url pointing to an xml document conforming to the xml vocabulary.

We have to come up with a name for this header.

The vocabulary is defined in 4 separate dtd:s. The reason for this is to allow parts of the bundle configuration information to be put in separate documents. For instance you might want to keep configuration descriptions for different languages separately.

The four DTD:s are:

- The Bundle Configuration Information DTD – This dtd defines the root element for the Bundle Configuration Information (the bundleConfigurationInformation element) and will be used as DOCTYPE for the xml document pointed to by the manifest header. It includes the other three dtd:s.

- The Configuration Definition DTD – This dtd defines the xml vocabulary for defining a configuration in terms of properties, their types, etc. This dtd is intended to represent the typing information in the ObjectClassDefinition and AttributeDefinition interfaces defined in the Metatype Specification [2].

- The Configuration Description DTD – This dtd defines the xml vocabulary for providing descriptions of configurations intended for configuration management user interfaces. This dtd is intended to represent the localizable information related to user interfaces in the ObjectClassDefinition and AttributeDefinition interfaces defined in the Metatype Specification [2].

- The Configuration Dictionary DTD – This dtd defines an xml vocabulary intended for representing configuration dictionaries with keys and values. It doesn’t correspond to any interface in the Metatype Specification [2], instead its purpose is to allow default configurations to  be included in the bundle that can be extracted by management systems and entered into the CM.

The information of the MetaTypeProvider interface defined in the Metatype Specification [2] is available by reading the configuration definitions and the configuration descriptions and merging the information.

## 3.2 The Bundle Configuration Information DTD

```
<!-- DTD for Bundle Configuration Information -->

<!-- This dtd defines a root element for bundle configuration information.    -->
<!-- The intended usage is for a bundle to have a manifest header with an url -->
<!-- that points to an xml document using this dtd.                           -->
<!-- Note that this document can either be located in the bundle itself or    -->
<!-- the url can point to some external location.                             -->
```

```
<!-- Include configurationDefinition element from ConfigurationDefinition DTD -->
<!-- The configurationDefinition element defines the configuration properties -->
<!-- for a given configuration: the keys, the types, etc.                     -->
<!ENTITY % definition PUBLIC "TODO: Decide on an ID"
                             "http://www.osgi.org/ConfigurationDefiniton.dtd">
%definition;

<!-- Include configurationDescription element from ConfigurationDescription DTD -->
<!-- The configurationDescription element provides a localized description of a -->
<!-- given configuration intended to be used for user interfaces.               -->
<!ENTITY % description PUBLIC "TODO: Decide on an ID"
                              "http://www.osgi.org/ConfigurationDescription.dtd">
%description;

<!-- Include configurationDictionary element from ConfigurationDictionary DTD   -->
<!-- The configurationDictionary element provides a format to represent actual  -->
<!-- configuration dictionaries in xml.                                         -->
<!ENTITY % dictionary PUBLIC "TODO: Decide on an ID"
                             "http://www.osgi.org/ConfigurationDictionary.dtd">
%dictionary;

<!-- The bundleConfigurationInformation element is the root element -->
<!-- for bundle configuration information.                          -->
<!-- It collects configuration definitions, descriptions, and       -->
<!-- default configurations.                                        -->
<!ELEMENT bundleConfigurationInformation
(configurationDefinitions?,configurationDescriptions?,defaultConfigurations?) >

<!-- The configurationDefinitions element holds the configurationDefinition   -->
<!-- elements for a bundle, both those defined inline and those included from -->
<!-- separate documents (See include element).                                -->
<!ELEMENT configurationDefinitions
          (configurationDefinition|include)* >

<!-- The configurationDescriptions element holds the configurationDescription  -->
<!-- elements for a bundle, both those defined inline and those included from -->
<!-- separate documents (See include element).                                -->
<!ELEMENT configurationDescriptions
          (configurationDescription|include)* >

<!-- The defaultConfigurations element holds a number of configurations -->
<!-- that can be extracted by management systems and used as default    -->
<!-- configurations for the services provided by the bundle.            -->
<!ELEMENT defaultConfigurations (configuration|factoryConfiguration)* >

<!-- The configuration element represents a configuration intended for      -->
```

```
<!-- a ManagedService.                                                      -->
<!-- The pid attribute matches the pid of the ManagedService registered     -->
<!-- by the bundle.                                                         -->
<!-- The configuration parameters represented by a configurationDictionary  -->
<!-- can either be provided inline or in a separate xml document (See       -->
<!-- include element).                                                      -->
<!ELEMENT configuration (configurationDictionary|include) >
<!ATTLIST configuration
          pid         CDATA    #REQUIRED >

<!-- The factoryConfigurations element lists configurations intended for a  -->
<!-- ManagedServiceFactory.                                                 -->
<!-- The pid attribute matches the pid of the ManagedServiceFactory         -->
<!-- registered by the bundle.                                              -->
<!-- The configuration parameters for each configuration are represented by -->
<!-- configurationDictionary elements that can either be provided inline or -->
<!-- in a separate xml documents (See include element).                     -->
<!ELEMENT factoryConfiguration (configurationDictionary|include)* >
<!ATTLIST factoryConfiguration
          pid         CDATA    #REQUIRED >

<!-- The include element contains an url that points to a separate document     -->
<!-- that contains xml conforming to the context of the include tag.            -->
<!-- For instance an include within a configurationDefinitions tag would point  -->
<!-- to an xml document based on the ConfigurationDefintion.dtd.                -->
<!ELEMENT include (EMPTY) >
<!ATTLIST include
          url         CDATA    #REQUIRED >
```

## 3.3 The Configuration Definition DTD

```
<!-- DTD for Configuration Definitions -->

<!-- The allowed types. -->
<!ENTITY  % TYPE "(STRING|INTEGER|LONG|FLOAT|DOUBLE|BYTE|SHORT|
                 BIGINTEGER|BIGDECIMAL|CHARACTER|BOOLEAN)">

<!-- The configurationDefinition element defines the configuration properties   -->
<!-- for a given configuration: the keys, the types, etc.                       -->
<!-- The pid attribute matches the pid of the ManagedService(Factory) for which -->
<!-- this configuration definition applies.                                     -->
<!-- The factory attribute is true if this configuration definition applies to  -->
<!-- a ManagedServiceFactory (meaning multiple configuration instances can be   -->
<!-- created) and false if it applies to a ManagedService (a singleton).        -->
<!-- The oid attribute can hold an object identifier (See chapter 10 Metatype   -->
<!-- Specification, in the OSGi 2.0 specification). If it is left out, the oid  -->
<!-- is assumed to be the same as the pid.                                      -->
<!ELEMENT configurationDefinition (propertyDefinition)* >
```

```
<!ATTLIST configurationDefinition
          pid     CDATA         #REQUIRED
          factory (true|false)  "false"
          oid     CDATA         #IMPLIED  >

<!-- The propertyDefinition element defines a configuration property for a  -->
<!-- configuration.                                                         -->
<!-- The key attribute matches the key that will appear in the Dictionary   -->
<!-- received by the ManagedService(Factory) the configuration applies to.  -->
<!-- The type attribute holds the leaf type of this property. The actual    -->
<!-- (Java) type appearing in the Dictionary also takes the cardinality     -->
<!-- into account.                                                          -->
<!-- For instance a type="STRING" can depending on cardinality result in    -->
<!-- String, Vector of String, String[], String[][], etc.                   -->
<!-- The cardinality attribute is a comma-separated list of Integer values. -->
<!-- This is an extension to the Metatype Specification 1.0.                -->
<!-- The semantics for the individual values are the same as in the         -->
<!-- specification with the addition that two or more values in the list     -->
<!-- means a multi-dimensional type.                                        -->
<!-- For instance: cardinality="10,4" means that the property is an array   -->
<!-- of max length 10 of arrays of max length 4.                            -->
<!-- The minlength is a comma-separated list of positive Integer values.    -->
<!-- It is an optional attribute and defines the minimum length of arrays   -->
<!-- or vectors as defined by the cardinality attribute.                    -->
<!-- The required attribute tells if this property is required or not in    -->
<!-- the configuration.                                                     -->
<!ELEMENT propertyDefinition (options?,default?) >
<!ATTLIST propertyDefinition
          key         CDATA         #REQUIRED
          type        %TYPE;        #REQUIRED
          cardinality CDATA         #REQUIRED
          minlength   CDATA         #IMPLIED
          required    (true|false)  "true"    >

<!-- The options element holds a list of value elements limiting the allowed  -->
<!-- values of the type indicated by the type attribute of the containing     -->
<!-- propertyDefinition.                                                      -->
<!ELEMENT options (value)* >

<!-- The default element holds a list of default values for the property.     -->
<!-- Note that this element won't make sense if the property is               -->
<!-- multi-dimensional.                                                       -->
<!ELEMENT default (value)* >

<!-- The value element contains a string representing a value of the type    -->
<!-- indicated by the type attribute of the containing propertyDefinition.   -->
<!ELEMENT value (#PCDATA)* >
```

### 3.4 The Configuration Description DTD

```
<!-- DTD for Configuration Descriptions -->

<!-- Root element for a Configuration Description.                              -->
<!-- This element serves to collect description data for a configuration to be  -->
<!-- to be used in configuration management user interfaces.                    -->
<!-- The pid attribute matches the pid of the ManagedService(Factory) for which -->
<!-- this configuration description applies.                                    -->
<!-- The locale attribute must have one of the values defined in the Locale     -->
<!-- class and tells which locale this description applies to.                  -->
<!ELEMENT configurationDescription (description,icon*,propertyDescription*) >
<!ATTLIST configurationDescription
          pid         CDATA     #REQUIRED
          locale      CDATA     #REQUIRED >

<!-- The icon element points to an url holding an icon for the configuration. -->
<!-- The url attribute holds an url that can be used to retrieve the icon.    -->
<!-- The size attribute optionally hints a size of the icon pointed to by     -->
<!-- the url.                                                                 -->
<!ELEMENT icon (EMPTY) >
<!ATTLIST icon
          url         CDATA #REQUIRED
          size        CDATA #IMPLIED  >

<!-- The propertyDescription element describes a single property in a      -->
<!-- configuration.                                                        -->
<!-- The key attribute matches the key attribute of the propertyDefinition -->
<!-- for this property (See ConfigurationDefinition DTD).                  -->
<!ELEMENT propertyDescription (name?,description?,helptext?,options?,headers?) >
<!ATTLIST propertyDescription
          key          CDATA     #REQUIRED >

<!-- The name element contains a localized user-friendly name for the property. -->
<!ELEMENT name (#PCDATA)* >

<!-- The description element contains a localized short description of the  -->
<!-- property.                                                              -->
<!ELEMENT description (#PCDATA)* >

<!-- The helptext element holds a localized longer description of the property. -->
<!ELEMENT helptext (#PCDATA)* >

<!-- The options element collects a number of localized labels for  -->
<!-- the allowed values of the property.                            -->
<!ELEMENT options(option)* >

<!-- The option element contains a localized label for one of the -->
```

```
<!-- allowed values of the property.                              -->
<!-- The value attribute matches a value from the corresponding   -->
<!-- property definition (See ConfigurationDefinition DTD).       -->
<!ELEMENT option (#PCDATA)* >
<!ATTLIST option
          value   CDATA     #REQUIRED

<!-- The headers element holds a list of headers that can be used as  -->
<!-- column headers in the case of properties that hold 2 dimensional -->
<!-- values.                                                          -->
<!-- The order of the header elements correspond to the order of the  -->
<!-- columns.                                                         -->
<!ELEMENT headers (header)* >

<!-- The header element holds a single column header.                 -->
<!ELEMENT header (#PCDATA)* >
```

## 3.5 The Configuration Dictionary DTD

```
<!-- DTD for Configuration Dictionaries -->

<!-- Allowed Non-primitive java types. -->
<!ENTITY  % OBJECTTYPE "String|Integer|Long|Float|Double|Byte|Short|
                       BigInteger|BigDecimal|Character|Boolean" >

<!-- Allowed primitive java types. -->
<!ENTITY  % PRIMITIVETYPE "long|int|short|char|byte|double|float|boolean" >

<!-- Allowed types. -->
<!ENTITY  % TYPE "%PRIMITIVETYPE;|%OBJECTTYPE;" >

<!-- Root element of a configuration dictionary. -->
<!ELEMENT configurationDictionary (property)* >

<!-- A property in the dictionary.                                      -->
<!-- The key attribute holds the key to the value for this property.    -->
<!-- The type attribute holds the type of the value for this property.  -->
<!-- It can be any of the values defined by OBJECTTYPE or if the        -->
<!-- last value in the list in containertypes is array it can also be   -->
<!-- any of the PRIMITIVETYPE values.                                   -->
<!-- The containertypes attribute holds a comma separated list of the   -->
<!-- values array and vector and tells what containers should be used   -->
<!-- at the different levels if the value of the property is            -->
<!-- multi-dimensional.                                                 -->
<!-- For instance a String[][] would be:                                -->
<!--          type="String"                                             -->
```

```
<!--          containertypes="array,array"                              -->

<!ELEMENT property  (value|vector|array) >
<!ATTLIST property
          key             CDATA   #REQUIRED
          type            %TYPE;  #REQUIRED
          containertypes  CDATA   #IMPLIED >

<!-- The value element contains a string representation of a value of  -->
<!-- the type indicated by the containing property element.           -->
<!ELEMENT value     (#PCDATA)* >

<!-- The vector element serves to group a number of value, array or other -->
<!-- vector elements to represent the elements of a Vector.               -->
<!ELEMENT vector    (value*|array*|vector*) >

<!-- The array element serves to group a number of value, vector or other -->
<!-- array elements to represent the elements of an array.                -->
<!ELEMENT array     (value*|array*|vector*) >
```

## 3.6 Example XML document

```
<bundleConfigurationInformation>
  <configurationDefinitions>
    <configurationDefinition pid="org.osgi.nursery.factory.EXAMPLE"
                             factory="true" >
      <propertyDefinition key="org.osgi.nursery.property3" type="BOOLEAN" >
        <default>
          <value>true</value>
        </default>
      </propertyDefinition>
      <propertyDefinition key="org.osgi.nursery.property4" type="SHORT"
                          cardinality="-10" minlength="1" >
        <options>
          <value>0</value>
          <value>1</value>
          <value>2</value>
        </options>
        <default>
          <value>0</value>
        </default>
      </propertyDefinition>
    </configurationDefinition>
    <configurationDefinition pid="org.osgi.nursery.service.EXAMPLE" >
      <propertyDefinition key="org.osgi.nursery.property0" type="INTEGER" >
        <default>
          <value>0</value>
        </default>
      </propertyDefinition>
      <propertyDefinition key="org.osgi.nursery.property1"
                          type="INTEGER" required="false">
        <options>
          <value>0</value>
          <value>1</value>
          <value>2</value>
        </options>
        <default>
          <value>0</value>
        </default>
      </propertyDefinition>
      <propertyDefinition key="org.osgi.nursery.property2"
                          type="STRING" cardinality="10,4" minlength="0,4"/>
    </configurationDefinition>
    <include url="http://nursery.osgi.org/exampleDefinition.xml"/>
  </configurationDefinitions>
  <configurationDescriptions>
    <configurationDescription pid="org.osgi.nursery.factory.EXAMPLE"
                              locale="en_us">
      <description>Example factory service.</description>
      <propertyDescription key="org.osgi.nursery.property3">
        <name>Example Property</name>
        <description>This property is just an example property.</description>
      </propertyDescription>
      <propertyDescription key="org.osgi.nursery.property4">
        <name>Options Example Property</name>
        <description>This property illustrates the use of option labels.
        </description>
        <options>
          <option value="0">red</option>
          <option value="1">green</option>
          <option value="2">blue</option>
        </options>
      </propertyDescription>
    </configurationDescription>
    <configurationDescription pid="org.osgi.nursery.service.EXAMPLE"
                              locale="en_us">
      <description>Example service.</description>
      <icon url="http://nursery.osgi.com/icon.gif" size="16"/>
      <propertyDescription key="org.osgi.nursery.property0">
        <name>Example Property</name>
        <description>This property is just an example property.</description>
      </propertyDescription>
      <propertyDescription key="org.osgi.nursery.property1">
        <name>Options Example Property</name>
        <description>This property illustrates the use of option labels.
        </description>
        <options>
          <option value="0">low</option>
          <option value="1">medium</option>
          <option value="2">high</option>
        </options>
      </propertyDescription>
      <propertyDescription key="org.osgi.nursery.property2">
        <name>Nested Array Property</name>
        <description>This property illustrates the use of column headers.
        </description>
        <headers>
          <header>Column 1</header>
          <header>Column 2</header>
          <header>Column 3</header>
          <header>Column 4</header>
        </headers>
      </propertyDescription>
    </configurationDescription>
    <include url="http://nursery.osgi.org/exampleDescription.xml"/>
  </configurationDescriptions>
  <defaultConfiguration>
    <configuration pid ="org.osgi.nursery.service.EXAMPLE">
      <configurationDictionary>
        <property key="org.osgi.nursery.property0" type="Integer">
          <value>42</value>
        </property>
        <property key="org.osgi.nursery.property2" type="String"
                  containertypes="array,array">
          <array>
            <array>
              <value>these</value>
              <value>are</value>
              <value>array</value>
              <value>values</value>
            </array>
            <array>
              <value>so</value>
              <value>are</value>
              <value>these</value>
              <value>values</value>
            </array>
          </array>
        </property>
      </configurationDictionary>
    </configuration>
    <factoryConfiguration pid ="org.osgi.nursery.factory.EXAMPLE">
```

```
      <configurationDictionary>
        <property key="org.osgi.nursery.property3" type="Boolean">
          <value>false</value>
        </property>
        <property key="org.osgi.nursery.property4" type="Short"
                  containertypes="vector">
          <vector>
            <value>1</value>
            <value>0</value>
            <value>1</value>
            <value>2</value>
          </vector>
        </property>
      </configurationDictionary>
      <include url="http://nursery.osgi.org/exampleDictionary.xml"/>
    </factoryConfiguration>
  </defaultConfiguration>
</bundleConfigurationInformation>
```

# 4 Security Considerations

# 5 Document Support

## 5.1 References

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

[2]. ??, MetaType Specification version 1.0, ?? 2001

## 5.2 Author's Address

| | |
|---|---|
| Name | Per Gustafson |
| Company | Gatespace AB |
| Address | Stora Badhusgatan 18-20, SE-41121 Gothenburg, Sweden |
| Voice | +46 31 743 98 23 |
| e-mail | perg@gatespace.com |

## 5.3 Acronyms and Abbreviations

## 5.4 End of Document