Change Log for OSS Common API version 1.5

OSS through Java™ Initiative

Vincent Perrot, Sun Microsystems, Inc.

COM-API-SPEC_change_log.1.5.1.doc

Executive Summary

This document summarizes the changes to the OSS Common API (JSR 144) specification Version 1.5. The main purpose of this version is

- Update and fix issues related to the Core Business Entities (CBE),
- Include feedback from other OSS/J API extending JSR 144

However, since maintenance release to the specification was taking place, additional modifications to the previously existing Java Value Type interface were also incorporated. All these modifications are coming from the Web Bug tracking system at:

- https://jsr144-public.dev.java.net
- https://jsr144-private.dev.java.net (dedicated to OSS/J Members)

There are two lists of changes:

- "proposed" changes are those modifications that are included in OSS Common API version 1.5.
- "deferred" changes are those modifications that are not included in OSS Common API version 1.5, whether for time reasons or because it was considered that the changes were too significant.

The detailed description of changes in this document is principally of interest to people implementing the OSS Common API specification.

Table of Contents

1 Preface

1.1 Objectives

This document lists all the changes that have been requested for the maintenance release v1.5 version of the OSS Common API, JSR 144.

The changes have been collected through:

- Java.net Issue Tracker: Bug and Request For Evolution (RFE) submitted by Java developers

1.2 Audience

This document is used to start a Maintenance Release of the OSS Common API JSR 144.

According to the JCP^{SM} :

The Maintenance Lead (ML) will arrange to have all change items placed into the PROPOSED section of the Change Log (this document) and then send a request to the PMO to initiate a Maintenance Review. The PMO will make a public announcement and begin the review.

1.3 Approval and Distribution

The ML may choose to modify one or more of the proposed changes based on comments received during review.

1.4 Related Information

oss_common-1_4-mr-spec.zip: contains the Version 1.4 of the OSS common API, JSR 144, http://www.tmforum.org/ossj/downloads/jsr144

The backward compatibility may not be guarantied. The deprecation mechanism of java will be applied every time possible.

“*Deprecated* means this method is still usable, but you should not use it. It will gradually be phased out. There is a new method to do the same thing. Deprecated methods are marked with a special JavaDoc comment.”

See addition information about java deprecation at:

http://java.sun.com/j2se/1.3/docs/guide/misc/deprecation/deprecation.html

1.5 Revision History

Date	Version	Author	State	Comments
Dec 2006	1.5.1	Vincent Perrot, Sun Microsystems	Initial Draft	• Collection of private and public issues from java.net projects
				•

2 Summary of changes

Public domain

Issue #6: OSS/J Design Guidelines should provide some unified strategy to deal with dynamic introduction of types
Issue #12: Extends CBEManagedEntity definition to allow characteristic description (SID)
Issue #13: Update RequestValue definition
Issue #14: Add description to EntitySpecification definition
Issue #15: Add Business interaction components into bi package
Issue #16: Priority in RequestValue shall be a int instead of String
Issue #17: Invalid datatype State modeling generating invalid XSD
Issue #18: need to add an attribute for origininating ManagedEntity to AlarmValue
Issue #19: javax.oss.cbe.datatypes.LifeCycleState.INACTIVEPLANNED is not correct

Private domain

Issue #60: UrbanPropertyAddress attribute list invalid
Issue #61: Adding DistChannelProdOffer to CBE
Issue #62: rename attribut interactionState to InteractionStatus
Issue #64: Invalid request processing

3 Proposed changes

Details on the fix implementations are available in Issue tracker.

3.1 Public domain

3.1.1 Issue #6: OSS/J Design Guidelines should provide some unified strategy to deal with dynamic introduction of types

Presently OSS/J assumes that the data model used by an implementation is static, i.e. known at compile time. However, practically all significant implementations need some sort of solution to introduce new types on runtime, and since OSS/J does not provide any sort of guidelines for doing this, every implementation deals with this in their own way. This undermines the very purpose of a standard, because the solutions to dynamic introduction of new types typically push all crucial functionality into extendable parts of the OSS/J interfaces (such as queryInventory and updateInventory of OSS/J Inventory) while the standard functionality (such as template searches and set methods of OSS/J Inventory) remains practically unusable (yet mandatory from standard's point of view). The result is that no genuine generality is achieved; any client that wishes to use any of these implementations must know about the details of that particular implementation in order to be able to interact with it meaningfully.

VP: This have been partially addressed I guess by the fix to Issue #12 below.

3.1.2 Issue #12: Extends CBEManagedEntity definition to allow characteristic description (SID)

The following method and components needs to be added to the basic CBEMnagaedENtity in order to aloow Characterisc addition.

Characteristic is an abstract base class that is used to define the essential characteristics (attributes, methods, constraints, and relationships) of a managed entity.

CharacteristicSpecification specifies the name and type of a characteristic. A Characteristic is an attribute that can be added to an entity without the need to modify the interface for it.

CBEManagedEntity shall also contains methods to "characterized" the entity:

- CharacteristicSpecification[] getCharacterizedBy() throws java.lang.IllegalStateException
- set and make...

3.1.3 Issue #13: Update RequestValue definition

The requestValue denition shall be update in order to fit with OrderMnaagememt

(JSR264) requirements:

- add requestedCompletionDate attribute
- add clientId attribute

3.1.4 Issue #14: Add description to EntitySpecification definition

To align with the SID the description field shall be added to the EntitySpecification definition.

3.1.5 Issue #15: Add Business interaction components into bi package

The following description shall be added to the bi package:

- businessInteractionLocation
- businessInteractionRelationShip
- BusinessInteractionSpecification
- Request Specification
- BusinessInteractionBulk

Still need to wait from OM expert group reveiw on proposd changes are other components may also need to be added.

3.1.6 Issue #16: Priority in RequestValue shall be a int instead of String

Change the priority type to int in RequestValue.

3.1.7 Issue #17: Invalid datatype State modeling generating invalid XSD

The State object in java is declared as a class with attribute and methods. Tools model it as a complex type following the guide lines.

State enums in java inherit from State and declare only static values (enum format) being handled. Thus generated as XML substition group by tools relating to simple XML type.

So there is a type mismatch when substition group reference the complex type State in datatypes.

The propsed fix:

- Rename the current State object as StateHandler.
- define a new State Object being of the enum type and containing only the static SEPARATOR definition.

Accepted...Other Design guidelines have also been reviewed in order to fix the remaining issues with enums:

- add abstract="true" to the definition to limit the possible values to only the enum items
- handle duplicated names with different definitions in the same xsd: add entity name in postfix or prefix (for example)
- support of array of enum: use a standard complex type of baseEnum....

3.1.8 Issue #18: need to add an attribute for origininating ManagedEntity to AlarmValue

In the conversion from the old QoS API to the new FM API, the AlarmInfo object was dropped. In dropping this object (which was the correct thing to do) the only object that tied an AlarmValue instance to the managed entity that originated the Alarm was lost.

The JSR 263 EG would like to have a new "originatingManagedEntityKey" and an "originatingManagedEntityOID" added to the AlarmValue object.

3.1.9 Issue #19: javax.oss.cbe.datatypes.LifeCycleState.INACTIVEPLANNED is not correct

In the source file of the javax.oss.cbe.datatypes.LifeCycleState interface there are missing dquotes.

I think it should look like this:

```
  public final static java.lang.String INACTIVEPLANNED =

             "INACTIVE"

           + State.SEPARATOR

           + "PLANNED";

// an not like this

"INACTIVE + State.SEPARATOR+ PLANNED";

      ^            ^
```

The source file is also part of / copied into the ri distribution

Thanks, will fix this in the next release. The name will be also harmonized in order to use

- o uppercase for name
- o lower case for the value (no space)

So impact:

- AdministrativeState

- OperationalState
- BusinessInteractionState
- TroubleTicketState
- RequestSate

3.2 Private domain

3.2.1 Issue #60: UrbanPropertyAddress attribute list invalid

The UrbanPropertyAddress attributes in the SID VI are:

Attributes

- streetNrFirst
- streetNrFirstSuffix
- streetNrLast
- streetNrLastSuffix
- streetName
- streetType
- streetSuffix
- locality
- postcode

This definitions shall replace the current one.

- Need the addition of GeographicAddress entity
- UrbanPropertyAddress is also an extension of GeographicAddress
- Need also to add the “country” attribute

3.2.2 Issue #61: Adding DistChannelProdOffer to CBE

A new entity, DistChannelProdOffer, has been added to the SID. This CR is for its addition to the CBE to keep them in parallel.

The new entity is an association class that modifies the N..N relationship between a DistributionChannel and a GeographicArea.

A DistChannelProdOffer can be associated with any number of ProductOfferings, and a ProductOffering can be associated with any number of DistChannelProdOffers.

The only required attribute for DistChannelProdOffer is the optional validFor defining the period of time during which the DistributionChannel can distribute Products in the GeographicArea.

In addition to DistChannelProdOffer (Association + attribute validFor)

The following components will be added to anticipate next SID version based on content of Document: GB922_Addendum_3_V7dot0bv0[1].0.1.doc:

DistributionChannel (attribute name)

3.2.3 Issue #62: rename attribut interactionState to InteractionStatus

Even if State and status are synonym, SID is continue to think that status in this case is more appropriated.

So the naming convention applied in this CBE implementation still apply to the type definition. (use state for string base enum and status for int based enum).

So the method shall become string getInteractionStatus.

3.2.4 Issue #64: Invalid request processing

Below if the proposal on how to handle invalid incoming XML Requests, this is also submitted to the DG as a proposal.

The following rules apply to situation where an invalid XML Request or a Request that does not contain any XML at all is received by an OSS/J Server implementation:

IF : the replyToQueue is not provided on the incoming Request, log an error locally (on the server) and ignore the Request.

IF :

· the replyToQueue is provided on the incoming Request, and

· it is possible to determine the top level XML element from the contents of the JMS message, and

· this top level element is a valid top level element (so it corresponds to the [operationName] of one of the supported operations)

*_THEN_* : sent an IllegalArgumentException to the specified replyToQueue and include the JMSCorrelationID as message property. This IllegalArgumentException must be contained in the [operationName]Exception wrapper.

IF :

· the replyToQueue is provided on the incoming Request,and

· it is NOT possible to determine the top level XML element from the contents of the JMS message (for example the message is not in XML format at all)

THEN : sent an RemoteException to the specified

replyToQueue and include the JMSCorrelationID as message property.

This RemoteException must be wrapped in a OssInvalidXMLRequestException wrapper (since no operationName can be determined).

The XML Schema proposal for the OssInvalidXMLRequestException is:

<element name="ossInvalidXMLRequestException">

<annotation>

<documentation>

This exception container is defined to cover for situations where the server receives a 'garbage' request and cannot determine the operation name from the request. Because no operation name can be determined it is not possible to return the remoteException as part of the [operationName]Exception wrapper that is defined for each operation on the

XML based integration profiles. This OssInvalidXMLRequestException type is intended to cover that gap in the specifications.

```
      </documentation>

    </annotation>

    <complexType>

        <choice>

          <element name="remoteException"
type="common:RemoteException"/>

        </choice>

      </complexType>

  </element>
```