

Change Log for OSS Common API version 1.4

OSS through Java™ Initiative

Vincent Perrot, Sun Microsystems, Inc.

COM-API-SPEC_change_log.1.4.4.doc

Copyright © 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Executive Summary

This document summarizes the changes to the OSS Common API (JSR 144) specification Version 1.4. 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 1.4.
- "deferred" changes are those modifications that are not included in OSS Common API version 1.4, 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

Executive Summary	2
Table of Contents	3
1 Preface	4
1.1 Objectives	4
1.2 Audience	4
1.3 Approval and Distribution	4
1.4 Related Information	4
1.5 Revision History	5
2 Summary of changes	6
3 Proposed changes	7
3.1 Public domain	7
3.1.1 Issue #6: OSS/J Design Guidelines should provide some unified strategy to deal with dynamic introduction of types	7
3.1.2 Issue #7: Add (SID) Customer definition to CBE	7
3.1.3 Issue #8: [Spec] Invalid WSDL	8
3.1.4 Issue #9: Improve JVTSession definition by splitting the Remote and the business declarations	8
3.2 Private domain	9
3.2.1 Issue #38: Use generation tools to produce, at least, CBE interface definition	9
3.2.2 Issue #42: Invalid zip package for Part 5	9
3.2.3 Issue #43: Customer and Customer SLA Additions to CBE	9
3.2.4 Issue #44: Invalid Quantity XML definition in datatypes schema	12
3.2.5 Issue #49: Fix the Chapter 3.1.4 content (replace query by upv)	12
3.2.6 Issue #54: Add SID Product Catalog entities (Pricing API)	12
3.2.7 Issue #57: Add MonthlySchedule and AnnualSchedule CBE/schedule package	13
3.2.8 Remove Deprecated interfaces/Methods	13

1 Preface

1.1 Objectives

This document lists all the changes that have been requested for the maintenance release v1.4 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 JCPSM:

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_3-mr-spec.zip: contains the Version 1.3 of the OSS common API, JSR 144, http://java.sun.com/products/oss/start_download.html

The backward compatibility may not be guaranteed. 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
May 2006	1.4.1 to 1.4.3	Vincent Perrot, Sun Microsystems	Initial Draft	<ul style="list-style-type: none">• Add Customer and CustomerSLA def + ProductCatalog stuff...• Issue with TTValue...• Issue Monthly and Annual schedule
June 2006	1.4.4	Vincent Perrot, Sun Microsystems	Submitted for MREL	<ul style="list-style-type: none">•
				<ul style="list-style-type: none">•

2 Summary of changes

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
- 3.1.2 Issue #7: Add (SID) Customer definition to CBE (duplicate issue#43)
- 3.1.3 Issue #8: [Spec] Invalid WSDL
- 3.1.4 Issue #9: Improve JVTSession definition by splitting the Remote and the business declarations

3.2 *Private domain*

- 3.2.1 Issue #38: Use generation tools to produce, at least, CBE interface definition
- 3.2.2 Issue #42: Invalid zip package for Part 5
- 3.2.3 Issue #43: Customer and Customer SLA Additions to CBE
- 3.2.4 Issue #44: Invalid Quantity XML definition in datatypes schema
- 3.2.5 Issue #49: Fix the Chapter 3.1.4 content (replace query by upv)
- 3.2.6 Issue #54: Add SID Product Catalog entities (Pricing API)
- 3.2.7 Issue #57: Add MonthlySchedule and AnnualSchedule CBE/schedule package
- 3.2.8 Remove Deprecated interfaces/Methods

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.

3.1.2 Issue #7: Add (SID) Customer definition to CBE

Note: This is redundant with private domain Issue #43 below.

We're using the OSS/J SA and TT specs at a customer and noticed that the TT spec has defined its own customer type that is not based on any of the CBE types. This is of course an issue for the TT spec lead, we contacted him and he confirmed that the next TT spec will be based on CBE.

However, in the current CBE specs (v1.2) there is no Customer type defined. According to SID, Customer should be an extension of PartyRole (which is defined in CBE). Two other types that we're missing in the CBE specs are Individual and Organization (both extend Party in SID). IndividualName and OrganizationName are defined in CBE, but Organization and Individual are missing.

I would like to request to include Customer, Individual and Organization to an upcoming version of CBE.

The case we're dealing with in our project requires a Customer type and instead of defining our own it would be much better if it was based on the SID/CBE specs. Of course we can define the Customer type ourselves and

stay inline with SID, but if at some point in time we need to integrate with a system that also defined it's own interpretation of SID Customer it's likely that there will be conflicts. If CBE would include SID Customer, Individual, Organization it more likely that integration will go smoothly. Of course we realize that each company may define it own customer that extends from the CBE Customer type, but at least they would all extend from the same base Customer type.

Note: IndividualName and OrganizationName definitions are already present in the javax.oss.cbe.party package and aligned with the SID.

3.1.3 Issue #8: [Spec] Invalid WSDL

The WSDL provided with the spec contains the following errors:

Line 32:

the <import...

shall be replace by:

<xsd:import...

3.1.4 Issue #9: Improve JVTSession definition by splitting the Remote and the business declarations

The current JVTSession definition (same for JVTLocalSession) directly extends EJBObject (or EJBLocalObject).

The proposal is to split the business declaration from the Remote interface definition as follow:

```
public interface JVTSession extends EJBObject,
JVTSessionRemoteBusiness {

}
```

```
public interface JVTLocalSession extends EJBLocalObject,
JVTLocalSessionBusiness {

}
```

and JVTSessionBusiness will get all the declarations (from the old 1.3 JVTSession) and the same for local definition.

Using these new declarations, the Bean implementation will be able to implement the JVTSessionBusiness (or local).

This will also follow the latest (IDE and App Server) way to create new (JEE5) EJBs.

3.2 Private domain

3.2.1 Issue #38: Use generation tools to produce, at least, CBE interface definition

To insure object creation consistency, and simplify maintenance, at least the CBE definitions shall be generated (for all integration profiles).

3.2.2 Issue #42: Invalid zip package for Part 5

The WSDL zip package shall follow the web service naming convention for the path and for the file name.

shall be under wsdl/Common/v1-3

and named: see DG 1.2...

3.2.3 Issue #43: Customer and Customer SLA Additions to CBE

Attachement includes full SID Phase VI XMI 1.0

Find below the definitions aligned with the SID phase 6 info model:

CustomerValue definition:

```

/*
Copyright 2002-2006 The Members of the OSS through Java(TM) Initiative.
All rights reserved. Use is subject to license terms.
*/
package javax.oss.cbe.customer;
/**
 *      Public interface definition for Customer
 *
 *      A person or organization that buys products and services from the enterprise or
receives free offers or services. This is modeled as a Party playing the role of Customer.
A Customer is a type of PartyRole. Customers can also be other service providers who
resell the enterprises products, other service providers that lease the enterprise's
resources for utilization by the other service provider's products and services, and so
forth.
 *
 *      See SID model phase VI
 *
 */
public interface CustomerValue
    extends javax.oss.cbe.party.PartyRoleValue {
    public static final String VALUE_TYPE = "javax.oss.cbe.customer.CustomerValue";
    public static final String CUSTOMER_KEY = "customerKey";
}

```

```

public final static String CUSTOMER_STATUS = "customerStatus";
public final static String CUSTOMER_RANK = "customerRank";

/**
 * Deep copy of this object
 */
public Object clone();
/**
 *
 * Attribute getter for customerStatus
 *
 *** Automatically created by Tigerstripe workbench from user defined attribute:
Customer.customerStatus ***
 *
 *
 *
 * @return the value of the customerStatus field.
 * @throws java.lang.IllegalStateException - if the attribute is not populated.
 *
 */
public java.lang.String getCustomerStatus()
throws
    java.lang.IllegalStateException
    ;

/**
 * Attribute setter for customerStatus
 *
 * The current condition of a customer, such as active, inactive, prospective.
 * @param value - the value to use to set the customerStatus attribute.
 * @throws java.lang.IllegalArgumentException - Is thrown to report that a bad
argument was provided to the method.
 *
 */
public void setCustomerStatus( java.lang.String value )
throws java.lang.IllegalArgumentException;
/**
 *
 * Attribute getter for customerRank
 *
 *** Automatically created by Tigerstripe workbench from user defined attribute:
Customer.customerRank ***
 *
 *
 *
 * @return the value of the customerRank field.
 * @throws java.lang.IllegalStateException - if the attribute is not populated.
 *
 */
public java.lang.String getCustomerRank()
throws
    java.lang.IllegalStateException
    ;

/**
 * Attribute setter for customerRank
 *
 * Degree of importance relative to other customers.
 * @param value - the value to use to set the customerRank attribute.
 * @throws java.lang.IllegalArgumentException - Is thrown to report that a bad
argument was provided to the method.
 *
 */
public void setCustomerRank( java.lang.String value )
throws java.lang.IllegalArgumentException;

/**
 * Gets the key for this value object
 *
 * @return the key for this value object
 * @throws java.lang.IllegalStateException if no key was populated in this

```

```

    *    value object.
    */
    public javax.oss.cbe.customer.CustomerKey getCustomerKey()
    throws java.lang.IllegalStateException;

    /**
     * Sets the key for this value object
     *
     * @param key - the key to set on this value object.
     * @throws java.lang.IllegalArgumentException, if the key is not a valid
     *         key for this value object.
     */
    public void setCustomerKey( javax.oss.cbe.customer.CustomerKey key )
    throws java.lang.IllegalArgumentException;

    /**
     * Factory method for CustomerKey
     *
     * @return a new instance of a blank CustomerKey
     */
    public javax.oss.cbe.customer.CustomerKey makeCustomerKey();
}

```

CustomerServiceLevelAgreementValue definition:

```

/*
Copyright 2002-2006 The Members of the OSS through Java(TM) Initiative.
All rights reserved. Use is subject to license terms.
*/
package javax.oss.cbe.customer;
/**
 *    Public interface definition for CustomerServiceLevelAgreement
 *
 *    A service level agreement (SLA) is a type of agreement that represents a formal
    negotiated agreement between two parties designed to create a common understanding about
    products, services, priorities, responsibilities, and so forth. The SLA is a set of
    appropriate procedures and targets formally or informally agreed between parties in order
    to achieve and maintain specified Quality of Service.
    */
    public interface CustomerServiceLevelAgreementValue
    extends javax.oss.cbe.sla.ServiceLevelAgreementValue {
        public static final String VALUE_TYPE =
"javax.oss.cbe.customer.CustomerServiceLevelAgreementValue";
        public static final String CUSTOMER_SERVICE_LEVEL_AGREEMENT_KEY =
"customerServiceLevelAgreementKey";

        /**
         * Deep copy of this object
         */
        public Object clone();

        /**
         * Gets the key for this value object
         *
         * @return the key for this value object
         * @throws java.lang.IllegalStateException if no key was populated in this
         *         value object.
         */
        public javax.oss.cbe.customer.CustomerServiceLevelAgreementKey
getCustomerServiceLevelAgreementKey()

```

```

throws java.lang.IllegalStateException;

/**
 * Sets the key for this value object
 *
 * @param key - the key to set on this value object.
 * @throws java.lang.IllegalArgumentException, if the key is not a valid
 *         key for this value object.
 */
public void setCustomerServiceLevelAgreementKey(
    javax.oss.cbe.customer.CustomerServiceLevelAgreementKey key )
    throws java.lang.IllegalArgumentException;

/**
 * Factory method for CustomerServiceLevelAgreementKey
 *
 * @return a new instance of a blank CustomerServiceLevelAgreementKey
 */
public javax.oss.cbe.customer.CustomerServiceLevelAgreementKey
    makeCustomerServiceLevelAgreementKey();
}

```

3.2.4 Issue #44: Invalid Quantity XML definition in datatypes schema

The quantity definition of in the

xmlns:cbedatatypes-v1-3="http://ossj.org/xml/Common-CBEDatatypes/v1-3

is using "decimal " instead of "decimal" (remove the trailing space)

It may be a generator issue or a Quantity.java error....

3.2.5 Issue #49: Fix the Chapter 3.1.4 content (replace query by upv)

In Chapter 3.1.4, the text:

"For any NamedQueryValue definition, the corresponding NamedQueryResponse shall be defined using the following naming conventiton: "

shall be repalced by

"For any UpdateprocedureValue definition, the corresponding UpdateProcedureResponse shall be defined using the following naming conventiton: "

3.2.6 Issue #54: Add SID Product Catalog entities (Pricing API)

Add Product Catalog entities (Product Catalog section and related figures(30-32).

Definitions available from:

GB922_Addendum_3_V7.0c-1.doc

Note: `javax.oss.cbe.product.productoffering.ProductCatalogValue` already exists in CBE:

The following fields will be added to it:

```
public java.lang.String type;

public java.lang.String id;

public javax.oss.cbe.datatypes.TimePeriod validFor;
```

Add also `ProductCatalogSpecification`, `ProductCatalogCharacteristic` (Id, name Presence), `ProductCatalogCharacteristicValue` (value, valueType, valueFrom, valueTo, validFor)

3.2.7 Issue #57: Add `MonthlySchedule` and `AnnualSchedule` CBE/schedule package

`MonthlySchedule` and `AnnualSchedule` CBEs under the CBE schedule package.

3.2.8 Remove Deprecated interfaces/Methods

Deprecated Interfaces	
javax.oss.cbe.alarm.AlarmAckState	<i>replaced by AlarmAckStatus</i>
javax.oss.cbe.alarm.AlarmConfig	<i>Redundant with AlarmValue definition</i>
javax.oss.cbe.datatypes.LifeCycleStatus	<i>replaced by LifeCycleState</i>
javax.oss.cbe.product.ProductStatus	<i>replaced by ProductState</i>
javax.oss.cbe.resource.ResourceStatus	<i>replaced by ResourceState</i>
javax.oss.cbe.service.ServiceStatus	<i>replaced by ServiceState</i>

Deprecated Fields

[javax.oss.cbe.measurement.PerformanceMonitorState.ACTIVE_OFF_DUTY](#)
replaced by *ACTIVE.OFFDUTY*

[javax.oss.cbe.measurement.PerformanceMonitorState.ACTIVE_ON_DUTY](#)
replaced by *ACTIVE.ONDUTY*

[javax.oss.cbe.alarm.AlarmValue.BACKED_UP_STATUS](#)
replaced by *BACKED_UP*

[javax.oss.cbe.bi.BusinessInteractionValue.INTERACTION_STATUS](#)
replaced by *INTERACTION_STATE*

[javax.oss.cbe.product.ProductSpecificationValue.LIFE_CYCLE_STATUS](#)
replace by *LIFE_CYCLE_STATE*

[javax.oss.cbe.product.ProductValue.PRODUCT_STATUS](#)
replaced by *PRODUCT_STATE*

[javax.oss.cbe.party.PartyRoleValue.STATUS](#)
replaced by *STATE*

[javax.oss.cbe.product.productoffering.ProductOfferingValue.STATUS](#)
replaced by *STATE*

Deprecated Methods

[javax.oss.cbe.alarm.AlarmValue.getBackedUpStatus\(\)](#)
replaced by *isBackedUp()*

[javax.oss.cbe.bi.BusinessInteractionValue.getInteractionStatus\(\)](#)
replaced by *getInteractionState*

[javax.oss.cbe.product.ProductSpecificationValue.getLifeCycleStatus\(\)](#)
replaced by *getLifeCycleState()*

[javax.oss.cbe.service.ServiceValue.getMandatory\(\)](#)
replaced by *isMandatory()*

[javax.oss.cbe.report.ReportInfoIterator.getNext\(int\)](#)
replaced by *getNextReportInfos()*

[javax.oss.cbe.report.ReportData.getPerformanceMonitorReport\(\)](#)
replaced by *getReport()*

[javax.oss.cbe.product.ProductValue.getProductStatus\(\)](#)
replaced by *String getProductState()*

[javax.oss.cbe.party.PartyRoleValue.getStatus\(\)](#)
replaced by *getState()*

[javax.oss.cbe.product.productoffering.ProductOfferingValue.getStatus\(\)](#)
replaced by *getState()*

[javax.oss.cbe.alarm.AlarmValue.setBackedUpStatus\(boolean\)](#)
replaced by *setBackedUp(boolean value)*

<u>javax.oss.cbe.bi.BusinessInteractionValue.setInteractionStatus(String)</u> <i>replaced by setInteractionState()</i>
<u>javax.oss.cbe.product.ProductSpecificationValue.setLifeCycleStatus(int)</u> <i>replaced by setLifeCycleState(String)</i>
<u>javax.oss.cbe.product.ProductValue.setProductStatus(int)</u> <i>replaced by setProductState(String)</i>
<u>javax.oss.cbe.party.PartyRoleValue.setStatus(String)</u> <i>replaced by setState(String)</i>
<u>javax.oss.cbe.product.productoffering.ProductOfferingValue.setStatus(String)</u> <i>replaced by setState(String)</i>