

## **Change Log for OSS Common API version 1.5**

### **OSS through Java™ Initiative**

Vincent Perrot, Sun Microsystems, Inc.

COM-API-SPEC\_change\_log.1.5.4.doc

Copyright © 2007 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 1.5. The main purpose of this version is

- Deliver a simplified and consistent (across profiles and APIs) set of entity definitions refocusing on the main CBE principles as described in the document:  
“How Core Business Entities Integrate with the Common API and Relate to the SID”  
<http://www.tmforum.org/browse.aspx?catid=2951&linkID=6391&docID=4779>
- Include feedback from other OSS/J APIs extending JSR 144
- Fix known issue from the java, XML or WS profiles definitions.

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

<b>Executive Summary.....</b>	<b>2</b>
<b>Table of Contents.....</b>	<b>3</b>
<b>1 Preface.....</b>	<b>4</b>
1.1 Objectives.....	4
1.2 Audience.....	4
1.3 Approval and Distribution.....	4
1.4 Related Information.....	4
1.5 Revision History.....	4
<b>2 Summary of changes.....</b>	<b>6</b>
<b>3 Proposed changes.....</b>	<b>7</b>
3.1 Private domain.....	7
3.1.1 Issue #84: Simplify and apply OSSJ requirements for Summer release 2007.....	7
3.1.2 Issue #60: UrbanPropertyAddress attribute list invalid.....	9
3.1.3 Issue #62: rename attribut interactionState to InteractionStatus.....	9
3.1.4 Issue #64: Invalid request processing.....	10
3.1.5 Issue #65: Remove instance methods from Alarm.....	11
3.1.6 Issue #66: AckStatus and Interaction Record in AlarmValue.....	12
3.1.7 Issue #67: ClearStatus and Interaction Record in AlarmValue.....	13
3.1.8 Issue #68: Invalid Integer MAX_VALUE in Alarm and Core XSD.....	13
3.1.9 Issue #69: Add primaryKey field in ManagedEntityKey XML schema.....	13
3.1.10 Issue #70: Add support for characteristicis to CBEManagedEntitySpecificationKey.....	14
3.1.11 Issue #73: Use String for ProbableCause attribute definition in BaseAlarm.....	15
3.1.12 Issue #77: Optional attributes in Alarm.....	17
3.1.13 Issue #80: Change alarmSpecificInfo to an array.....	18
3.1.14 Issue #88: Change Alarm to BaseAlarm.....	18
3.1.15 Issue #94: Enum Substitution group generate only 2 levels of inheritance.....	18
3.1.16 Issue #109: Value instead of name shall be used in XML Schema for enums.....	19
3.1.17 Issue #118: BaseEventType in XML Schema should define managedObjectClass and managedObjectInstance optional....	19
3.2 Public domain.....	20
3.2.1 Issue #6: OSS/J Design Guidelines should provide some unified strategy to deal with dynamic introduction of types.....	20
3.2.2 Issue #12: Extends CBEManagedEntity definition to allow characteristic description (SID).....	21
3.2.3 Issue #14: Add description to EntitySpecification definition.....	22
3.2.4 Issue #15: Add Business interaction components into bi package.....	22
3.2.5 Issue #17: Invalid datatype State modeling generating invalid XSD.....	22
3.2.6 Issue #18: need to add an attribute for originating ManagedEntity to AlarmValue.....	23
3.2.7 Issue #19: javax.oss.cbe.datatypes.LifeCycleState.INACTIVEPLANNED is not correct.....	23

# 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<sup>SM</sup>:

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 guaranteed. The deprecation mechanism of java will be applied every time possible.

## 1.5 Revision History

Date	Version	Author	State	Comments
Dec 2006	1.5.1	Vincent Perrot, Sun Microsystems	Initial Draft	<ul style="list-style-type: none"> <li>• Collection of private and public issues from java.net projects</li> </ul>

April 2007	1.5.3	Vincent Perrot, Sun Microsystems	After Summer release decisions	<ul style="list-style-type: none"> <li>Simplify CBE definition See issue 84</li> </ul>
June 2007	1.5.4	Vincent Perrot, Sun Microsystems	Submitted for Review	

## 2 Summary of changes

### *Private domain.....*

- Issue #84: Simplify and apply OSSJ requirements for Summer release 2007.....
- Issue #60: UrbanPropertyAddress attribute list invalid.....
- Issue #62: rename attribut interactionState to InteractionStatus.....
- Issue #64: Invalid request processing.....
- Issue #65: Remove instance methods from Alarm.....
- Issue #66: AckStatus and Interaction Record in AlarmValue.....
- Issue #67: ClearStatus and Interaction Record in AlarmValue.....
- Issue #68: Invalid Integer MAX\_VALUE in Alarm and Core XSD.....
- Issue #69: Add primaryKey field in ManagedEntityKey XML schema.....
- Issue #70: Add support for characteristicis to CBEManagedEntitySpecificationKey.....
- Issue #73: Use String for ProbableCause attribute definition in BaseAlarm.....
- Issue #77: Optional attributes in Alarm.....
- Issue #80: Change alarmSpecificInfo to an array
- Issue #88: Change Alarm to BaseAlarm.....
- Issue #94: Enum Substitution group generate only 2 levels of inheritance.....
- Issue #109: Value instead of name shall be used in XML Schema for enums
- Issue #118: BaseEventType in XML Schema should define managedObjectClass and managedObjectInstance optional

### *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 #14: Add description to EntitySpecification definition.....
- Issue #15: Add Business interaction components into bi package.....
- 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

## 3 Proposed changes

Details on the fix implementations are available in Issue tracker.

### 3.1 Private domain

#### 3.1.1 Issue #84: Simplify and apply OSSJ requirements for Summer release 2007

Modifications are detailed in documents and files section:

COM-API-CBE\_changes.0.1.xls

[https://jsr144-private.dev.java.net/files/documents/3868/53313/COM-API-CBE\\_changes.0.1.xls](https://jsr144-private.dev.java.net/files/documents/3868/53313/COM-API-CBE_changes.0.1.xls)

<i>Current Package names</i>	<i>Proposed change</i>	<i>Progress</i>	<i>Usage</i>	<i>Description</i>
javax.oss.cbe	none	N/A	generic	The CBE Core package contains the base en
javax.oss.cbe.agreement	1. get the definition from javax.oss.ce.sla to align with SID.	1, 2 done	SQM only	One form of business interaction in which se
javax.oss.cbe.alarm	1. rename package javax.oss.shared.alarm (alarm is not defined in SID)	1, 2, 3 done	FM + SQM	The CBE Alarm package defines entities an immediate or potential negative impact on t
javax.oss.cbe.bi	1. keep only BI, BII,BILocation,BIRole,BIRelationShip	1, 2 done	TT + SQM + OM	A BusinessInteraction is an arrangement, co PartyRoles, ResourceRoles, or CustomerAcc
javax.oss.cbe.customer	1. keep only Customer definition 2. Move the rest (CustomerSLA defs) to	1, 2 done	generic + SQM	Customers are at the center of any enterpris
javax.oss.cbe.datatypes	none	N/A	generic	The CBE DataTypes package contains comm
javax.oss.cbe.location	none	N/A	generic	The CBE Location package contains the obj
javax.oss.cbe.measurement	1. removed as not SID and PM not starte	done	PM + SQM	The CBE Measurement package contains me information.
javax.oss.cbe.party	none	N/A	generic	The CBE Party package contains the repres
javax.oss.cbe.policy	1. Move to Pricing	done	Pricing	The Policy domain defines Policy entities th entities in other domains.
javax.oss.cbe.product	none	N/A	generic	The CBE Product package defines a basic th
javax.oss.cbe.product.productoffering	1. keep only PO, POSpec, Pcatalog 2. Move the rest to Pricing	1, 2 done	generic + Pricing	This package contains only the base interfac
javax.oss.cbe.product.productofferingp	1. Move to Pricing	done	Pricing	This package contains only the base interfac
javax.oss.cbe.report	none	N/A	generic	The CBE Report package contains entity de
javax.oss.cbe.resource	none	N/A	generic	The CBE Resource package defines a basic th
javax.oss.cbe.schedule	1. remove package (not SID, no func		PM? +	The CBE Schedule package specifies entities
javax.oss.cbe.service	1. keep only Service, ServiceSpec, ServiceState definitions	1, 2 done	generic + SQM	The CBE Service package defines a basic the
javax.oss.cbe.sla	1. move definitions to javax.oss.cbe.agreement (align to SID)	done		The CBE SLA (Service Level Agreement) p agreement between two parties.
javax.oss.cbe.trouble	1. Move to TT	done	TT	This package contains the definitions for T
javax.oss.cbe.user	none	N/A	generic	
javax.oss.util	1. move to FM	done	FM	Contains miscellaneous utility classes.

The following list of issues will be closed due to the move of the corresponding objects outside the scope of the JSR 144:

Issue #61: Adding DistChannelProdOffer to CBE

Issue #74: New attributes needed for Alarm Entity

Issue #78: Add description in InteractionRecord

Issue #79: Remove attributeChanges from Alarm

Issue #81: Alarm should not be extensible

Issue #87: Static ITEMS attribute missing on javax.oss.cbe.bi.RequestValue



### 3.1.2 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

### 3.1.3 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.

### 3.1.4 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>

```

### 3.1.5 Issue #65: Remove instance methods from Alarm

From thr mail I sent to the JSR 263 EG:

For all actions on alarms (ack...) done from the Monitor session, there needs to be a test in the RI that the alarm is not delegated and a specific exception to indicate that the alarm is delegated. I propose to add an OssAlarmDelegatedException for all actions.

However, it raises a problem with having the ack and clear actions in the AlarmEntity as it would mean adding this exception and that would put in CBE a dependency on the FM API. If we agree on this exception, then the actions should be moved back to the FM API.

So all instance actions on the CBE Alarm entity should be set to false.

----- Additional comments from Spec leader -----

the current list of "instance" methods in BaseAlarm are:

- get, remove (simplebykey)
- clear, acknowledge, unclear, comment, unacknowledge (simple)

All will be set to false.

Note: This kind of operation on "instance" definitions could be done at anytime. They just only need to not impact the AlarmaValue definition.

### 3.1.6 Issue #66: AckStatus and Interaction Record in AlarmValue

Ack state : today the ack interaction record is used to indicate that an alarm has been acknowledged. Now that we can ack/unack an alarm, this is not sufficient. We need the AckState to be present in the Alarm Value and the ackInteractionRecord to become a chain (multiplicity \*) to keep track of all acks and unacks. The records will be kept in order: ack, then unack...

----- Additional comments from Spec leader -----

To align with update applied to issue #84 the alarm package will get the following definitions:

- AlarmAckStatus.java
- AlarmClearStatus.java
- AlarmSpecificInfo.java
- AlarmType.java
- BaseAlarmKey.java
- BaseAlarmKeyResult.java
- BaseAlarmKeyResultIterator.java
- BaseAlarmValue.java
- BaseAlarmValueIterator.java
- PerceivedSeverity.java

With BaseAlarm attributes:

- attribute name alarmChangedTime
- attribute name alarmRaisedTime
- attribute name alarmType
- attribute name perceivedSeverity
- attribute name probableCause
- attribute name specificProblem
- attribute name systemDN
- attribute name alarmSpecificInfo

- attribute name originatingManagedEntityKey
- attribute name alarmAckStatus
- attribute name alarmClearStatus
- attribute name additionalText
- attribute name originatingManagedEntityOid

### 3.1.7 Issue #67: ClearStatus and Interaction Record in AlarmValue

Today the clearing interaction record is used to indicate that an alarm has been cleared. Now that we can clear/unclear an alarm, this is not sufficient. We need an AlarmClearState with values Cleared and Uncleared (default).

The AlarmClearState needs to be added in the Alarm Value and the learingInteractionRecord to become a chain ([]) to keep track      ○ today the interaction record is used to indicate that an alarm has been acknowledged. Now that we can ack/unack an alarm, this is not sufficient. We need the AckState to be present in the Alarm Value and the ackInteractionRecord to become a chain ([]) to keep track of all clear/ unclear actions. The records will be kept in order: clear, then unclear, then second clear, then second unclear...

Changes needed:

1) In javax.oss.shared.alarm, create a new datatype AlarmClearStatus, value:

CLEARED, UNCLEARED

2) In the BaseAlarm Entity add a new attribute alarmClearStatus of type AlarmClearStatus.

### 3.1.8 Issue #68: Invalid Integer MAX\_VALUE in Alarm and Core XSD

Integer.MAX\_VALUE by -1 when needed.

Cardinality and AlarmType

### 3.1.9 Issue #69: Add primaryKey field in ManagedEntityKey XML schema

The schema for ManagedEntityKey shall be:

```
<complexType name="ManagedEntityKey">
  <annotation>
    <documentation>A <CODE>ManagedEntityKey</CODE> is a
unique identifier for a
    <CODE>ManagedEntityValue</CODE>.
    <p>
    The Type, ApplicationDN and ApplicationContext
are included in <CODE>ManagedEntityKey</CODE>, because
a <CODE>primaryKey</CODE> is only unique within an
application instance as given by the ApplicationDN running
in a given naming system as given by the ApplicationContext
```

```

URL.</documentation>
  </annotation>
  <sequence>
    <element name="applicationContext" minOccurs="0"
type="co-v1-5:ApplicationContext" />
    <element name="applicationDN" minOccurs="0" type="string" />
    <element name="type" type="string" />
    <element name="primaryKey" type="anyType" minOccurs="1"
nillable="false" />
  </sequence>
</complexType>

```

----- Additional comments from Spec leader -----

All Spec leaders agree to add `primaryKey` field in `ManagedEntityKey` XML definition.

This goes also with an update of the MEV definition (following the proposal):

The "key" field will be also added in the `ManagedEntityValue` definition.

```

<complexType name="ManagedEntityValue" abstract="true">
  <annotation>
  </annotation>
  <sequence>
    <element name="lastUpdateVersionNumber" minOccurs="0" type="long" />
    <element name="key" type="co-v1-5:ManagedEntityKey" minOccurs="0"/>
  </sequence>
</complexType>

```

### 3.1.10 Issue #70: Add support for characteristics to CBEManagedEntitySpecificationKey

During yesterday's conf call David, Andreas and myself came up with the following solution. We would like your view on this either before or after your vacations, because if the approach is incorrect it will trigger quite some rework in the RI and TCK.

The approach we will be taking is that we will use `describingEntityType` on the `RequestSpecificationKey`. The client will use this client to instantiate the XML structure, `describingEntityType` must be the name of the `RequestValue` type to be instantiated.

Assumption: The value of `describingEntityType` must identify a `RequestValue` subtype that is in the same namespace as the `RequestSpecificationKey`.

Using the above is not enough to inform the client which dynamic attributes / Characteristics belong to the `RequestValue` subtype. Therefore we propose to extend the `CBEManagedEntitySpecificationKey` interface to include an `is` getters/setter for Characteristics that define which Characteristics are supported for the corresponding `CBEManagedEntityValue` (identified by `describingEntityType`).

Since this is a change on the common api I will log a issue for this in thejsr144 issue tracker. Hope we can still fix this in 1.5 cause otherwise it willnot be possible to include support for Characteristics in the OM 1.0 release.

Since Characteristics are not included in the currentCBEMAnagedEntitySpecificationKey definition we cannot implement support for Characteristics at this moment in the RI and TCK. What we can do is implement the use of the describingEntityType attribute for the RequestSpecificationKeys in the RI and TCK.

----- Additional comments from Spec leader -----

Duplicated of Issue #12 of the public Domain.

### 3.1.11Issue #73: Use String for ProbableCause attribute definition in BaseAlarm

analyzed the latest 3GPP rel 6 documents to check the differences with the latest common 1.5 version. I stick to release 6 as it is the goal of the FM API to align with 3GPP release 6.

I use the document 3GPP TS 32.111-3 V6.6.0 (2006-06), Part 3: Alarm Integration Reference Point, Corba Solution Set as it provides a nice enumeration with codes.

Probable Causes with different texts between JSR 144 and 3GPP

Code	JSR 144	3GPP
303	BANDWIDTH_REDUCTION	BANDWIDTH_REDUCED
566	CALL_ESTABLISHMENT_ERROR	CONNECTION_ESTABLISHMENT_ERROR
307	CONFIGURATION_OR_CUSTOMIZING_ERROR	CONFIGURATION_OR_CUSTOMIZATION_ERROR
153	CORRUPT_DATA_M3100	CORRUPT_DATA
3	DEGRADED_SIGNAL_M3100	DEGRADED_SIGNAL
118	ENCLOSURE_DOOR_OPEN_M3100	ENCLOSURE_DOOR_OPEN
54	EXTERNAL_DEVICE_PROBLEM	EXTERNAL_IF_DEVICE_PROBLEM
5	FRAMING_ERROR_M3100	FRAMING_ERROR
326	LEAK_DETECTION	LEAK_DETECTED
56	MULTIPLEXER_PROBLEM_M3100	MULTIPLEXER_PROBLEM
58	POWER_PROBLEM_M3100	POWER_PROBLEM
59	PROCESSOR_PROBLEM_M3100	PROCESSOR_PROBLEM
113	PUMP_FAILURE M3100	PUMP_FAILURE

61	RECEIVER_FAILURE_M3100	RECEIVER_FAILURE
67	TRANSMITTER_FAILURE_M3100	TRANSMISTTER_FAILURE

I don't see any reason for some probable causes having the suffix \_M3100 while a lot of M3100 probable causes don't have it. The text of the probable causes should be aligned to the 3GPP wording.

Probable Causes to be removed from JSR 144

Value JSR 144 name reason

10	TRANSMISSION_ERROR	duplicated probable cause
----	--------------------	---------------------------

Probable Causes to be added in JSR 144

```

COMMUNICATIONS_RECEIVE_FAILURE = 17;
COMMUNICATIONS_TRANSMIT_FAILURE = 18;
MODULATION_FAILURE = 19;
DEMODULATION_FAILURE = 20;
REAL_TIME_CLOCK_FAILURE = 70;
PROTECTION_MECHANISM_FAILURE = 81;
PROTECTING_RESOURCE_FAILURE = 82;
INTRUSION_DETECTION = 126;
EXTERNAL_POINT_FAILURE = 136;
LOSS_OF_REAL_TIME = 157;
REINITIALIZED = 158;
EXCESSIVE_ERROR_RATE = 203;
AUTHENTICATION_FAILURE = 701;
BREACH_OF_CONFIDENTIALITY = 702;
CABLE_TAMPER = 703;
DELAYED_INFORMATION = 704;
DENIAL_OF_SERVICE = 705;
DUPLICATE_INFORMATION = 706;
INFORMATION_MISSING = 707;
INFORMATION_MODIFICATION_DETECTED = 708;
INFORMATION_OUT_OF_SEQUENCE = 709;
KEY_EXPIRED = 711;
NON_REPUDIATION_FAILURE = 712;
OUT_OF_HOURS_ACTIVITY = 713;
OUT_OF_SERVICE = 714;
PROCEDURAL_ERROR = 715;
UNAUTHORISED_ACCESS_ATTEMPT = 716;
UNEXPECTED_INFORMATION = 717;
UNSPECIFIED_REASON = 718;

```

----- Additional comments from Spec leader -----

The ProbalClause attribute in BaseAlarm definition will simply be declared as a String type.

As the Value of 3GPP probable causes are model specific they will move to the extension of the FM API.

But As this attribute definition is shared by FM and SQM the attribute shall still be defined in the baseAlarm.



The enum ProbableCause have been removed from the ceb defs.

### 3.1.12 Issue #77: Optional attributes in Alarm

After a review of the alarm attributes and a number of them should be optional.

Here is the list:

ADDITIONAL_TEXT	OPTIONAL
ALARM_CHANGED_TIME	OPTIONAL
ACKNOWLEDGEMENT_INTERACTION_RECORD	MANDATORY
CLEARING_INTERACTION_RECORD	OPTIONAL
ALARM_RAISED_TIME	MANDATORY
ALARM_TYPE	MANDATORY
ATTRIBUTE_CHANGES	OPTIONAL
BACKED_UP	OPTIONAL
BACK_UP_OBJECT	OPTIONAL
COMMENTS	OPTIONAL
CORRELATED_NOTIFICATIONS	OPTIONAL
NOTIFICATION_ID	OPTIONAL
PERCEIVED_SEVERITY	MANDATORY
PROBABLE_CAUSE	MANDATORY
PROPOSED_REPAIR_ACTIONS	OPTIONAL
SPECIFIC_PROBLEM	OPTIONAL
SYSTEM_DN	OPTIONAL
ALARM_SPECIFIC_INFO	OPTIONAL
ORIGINATING_MANAGED_ENTITY_KEY	MANDATORY
ORIGINATING_MANAGED_ENTITY_OID	MANDATORY
ALARM_ACK_STATUS	MANDATORY
ALARM_CLEAR_STATUS	MANDATORY

The 3 new security alarm attributes are all optional.

----- Additional comments from Spec Leader -----

ADDITIONAL_TEXT	OPTIONAL
ALARM_ACK_STATUS	MANDATORY
ALARM_CHANGED_TIME	OPTIONAL
ALARM_CLEAR_STATUS	MANDATORY
ALARM_RAISED_TIME	MANDATORY
ALARM_SPECIFIC_INFO	OPTIONAL
ALARM_TYPE	MANDATORY
ORIGINATING_MANAGED_ENTITY_KEY	MANDATORY
ORIGINATING_MANAGED_ENTITY_OID	MANDATORY
PERCEIVED_SEVERITY	MANDATORY
PROBABLE_CAUSE	MANDATORY
SPECIFIC_PROBLEM	OPTIONAL
SYSTEM_DN	OPTIONAL

Only apply to BaseAlarm

### 3.1.13 Issue #80: Change alarmSpecificInfo to an array

Following the discussion in FM EG, it is likely that an alarm would have more than 1 specificInfo field. So it is proposed to change the alarmSpecificInfo field to make it an array (multiplicity \*) of AlarmSpecificInfo.

### 3.1.14 Issue #88: Change Alarm to BaseAlarm

It is a very bad idea to use the same class name in different packages. As the usage of the "alarm" world makes more sense in the FM scope, the basic definition present in Common API will be named BaseAlarm.

### 3.1.15 Issue #94: Enum Substitution group generate only 2 levels of inheritance

The enum substitution group are currently partially generated in XML schemas.

Example:

level3 extends level2 extends level1

Is only generated:

```
<element name="baseMyLevel3Attribute" type="string"/>
<element name="myLevel3Attribute" type="ns-PartialSubstitutionGroup:level3"
  substitutionGroup="ns-PartialSubstitutionGroup:baseMyLevel3Attribute" />
<element name="myLevel3Attributelevel2"
  type="ns-PartialSubstitutionGroup:level2"
  substitutionGroup="ns-PartialSubstitutionGroup:baseMyLevel3Attribute"/>
```

Is missing:

```
<element name="myLevel3Attributelevel1"
  type="ns-PartialSubstitutionGroup:level1"
  substitutionGroup="ns-PartialSubstitutionGroup:baseMyLevel3Attribute"/>
```

----- Additional comments from Spec Leader -----

Additional other issues regarding enum definitions have also been submitted and resolved for this maintenance release.

Here is the list:

Issue 97, 98, 99, 105, 106, 109 (below) and 112

In order to keep this change log readable, consider that all enum type definition and usage have been impacted in this release.

### 3.1.16 Issue #109: Value instead of name shall be used in XML Schema for enums

For example in Schema enum shall be represented by there value instead of their name.

```
simpleType name="ReportMode"
  base="int"
  enumeration value="0"
  enumeration value="1"
  etc
```

```
instead of String
  enumeration value="EVENT_MULTIPLE"
  etc
```

### 3.1.17 Issue #118: BaseEventType in XML Schema should define managedObjectClass and managedObjectInstance optional

aseEventType is currently defined as:

```

<!--=====Events=====-->
<complexType name="BaseEventType" abstract="true">
  <annotation>
    <documentation>Base Event</documentation>
  </annotation>
  <sequence>
    <element name="applicationDN" type="string"/>
    <element name="eventTime" type="dateTime"/>
    <element name="managedObjectClass" type="string"/>
    <element name="managedObjectInstance" type="string"/>
  </sequence>
</complexType>

```

managedObjectClass and managedObjectInstance should be optional / minOccurs=0,  
as they are not applicable to all events.

----- Additional comments from Spec Leader -----

BaseEventType will be updated as follow:

```

<!--=====Events=====-->
<complexType name="BaseEventType" abstract="true">
  <annotation>
    <documentation>Base Event</documentation>
  </annotation>
  <sequence>
    <element name="applicationDN" type="string"/>
    <element name="eventTime" type="dateTime"/>
    <element name="managedObjectClass" type="string" minOccurs="0"/>
    <element name="managedObjectInstance" type="string" minOccurs="0"/>
  </sequence>
</complexType>

```

## 3.2 Public domain

### 3.2.1 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.

----- Additional comments from Spec Leader -----

This issue is addressed by the resolution of the issue #12 of the Public domain.

### 3.2.2 Issue #12: Extends CBManagedEntity definition to allow characteristic description (SID)

The following method and components needs to be added to the basic CBEMnagaedEntity in order to allow 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.

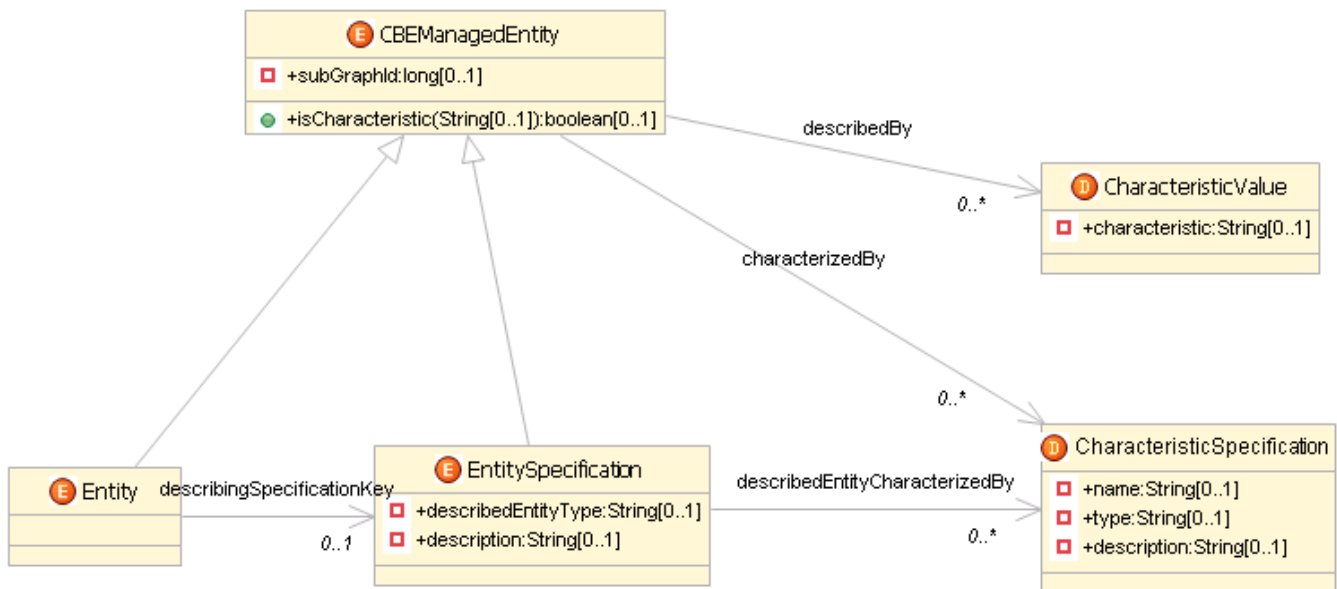
CharacteristicSpecification specifies the name and type of a characteristic. A Characteristic is an attribute that can be added to an entity without the need to modify the interface for it.

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

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

----- Additional comments from Spec Leader -----

Schema proposal:



### 3.2.3 Issue #14: Add description to EntitySpecification definition

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

### 3.2.4 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 review on proposed changes are other components may also need to be added.

----- Additional comments from Spec Leader -----

For the Summer Release the CBE have been simplified and reorganized following the main OSSJ / CBE principle:

CBE Present in Common only when shared by multiple JSRs, or generic, and model independent.

See also issue #84 at [https://jsr144-private.dev.java.net/issues/show\\_bug.cgi?id=84](https://jsr144-private.dev.java.net/issues/show_bug.cgi?id=84)

Only the following will be added:

- BusinessInteractionLocation
- BusinessInteractionRelationship
- BusinessInteractionSpecification

### 3.2.5 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 enumerations in java inherit from `javax.oss.cbe.datatypes.State`. They declare only static values (enumeration format) being handled. Thus generated as XML substitution groups by tools relating to simple XML type.

So there is a type mismatch when substitution groups reference the complex type State in datatypes.

The proposed 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....

### 3.2.6 Issue #18: need to add an attribute for originating 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.

### 3.2.7 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

----- Additional comments from Spec Leader -----

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

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

So impact:

- AdministrativeState
- OperationalState
- BusinessInteractionState
- TroubleTicketState
- RequestSate