Change Log for OSS Service Activation API version 1.2

OSS through Java™ Initiative

Andreas Ebbert, Nokia

SA-API-SPEC_change_log.1.2.1.doc

Executive Summary

This document summarizes the changes to the OSS Service Activation API (JSR 89) specification Version 1.2. The main purpose of this version is to

- Replace existing interfaces with functional equivalent Core Business Entities (CBE) interfaces,
- Use the OSS Common API (JSR 144) 1.2 specification
- Follow the new Draft of the OSS/J Design Guideline v1.2

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: http://bugs.sun.com/bugdatabase/index.jsp

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

Table of Contents

1 Preface

1.1 Objectives

This document lists all the changes that have been requested for the maintenance release v1.2 version of the OSS Service Activation API, JSR 89.

The changes have been collected through:

- Bug parade: Bug and Request For Evolution (RFE) submitted by Java developers
- OSS/J Architectural Board: The common API needs to reflect the necessary new architectural recommendation (new CBE, etc)

1.2 Audience

This document is used to start a Maintenance Release of the OSS Service Activation API JSR 89.

According to the JCP℠:

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_service_activation-1_0-fr-spec.zip: contains the Version 1.0 of the OSS Service Activation API, JSR 89,
http://java.sun.com/products/oss/start_download.html

1.5 Revision History

Date	Version	Author	State	Comments
2005-12-15	1.2.0	Andreas Ebbert	Initial Draft	• Initial Version for review in the Product Team
2006-01-13	1.2.1	Andreas Ebbert	Final	• Comments after Review in Product Team

2 Proposed changes

The following chapter will list the proposed changes for the maintenance release. In order to do that, the contextual diff format is used. The differences were compiled by the CVS web access from java.net.

Code 1: Contextual Diff Format

```
***************
*** <old version first line number>,<old version last line number> ****
 <old code that stayed the same>
!<old code that changed>
-<old code that was removed>

--- <new version first line number>,<new version last line number> ----
 <new code>
!<new code that is changed>
+<new code that is added>
```

2.1 Bug Database

2.1.1 Bug ID: 4947793 Invalid[1] Exception thrown by create method of SA Home interface

The JVTActivationHome interface incorrectly declared the EJBException in the throw clause of the create method.

Code 2: Changes in javax.oss.order.JVTActivationHome

```
***************
*** 37,42 ****


  public interface JVTActivationHome extends EJBHome {
!     public JVTActivationSession create() throws CreateException, EJBException,
RemoteException;
  }

--- 37,43 ----


  public interface JVTActivationHome extends EJBHome {
!       //aebbert: remove EJBException
!     public JVTActivationSession create() throws CreateException, RemoteException;
  }
```

2.2 Update to use OSS Common API 1.2 (JSR 144)

This change includes the usage of the ServiceValue definition from the Core Business Entities, as they are now defined in the OSS Common API, as well as other adaptations due to changes between the original release of JSR 144 and the maintenance release 1.2.

[1] Corrected spelling error in original bug report

2.2.1 JVTActivationSession

Changes in the `JVTActivationSession` session façade interface are needed because:

- Package change for ServiceValue from `javax.oss.service` to `javax.oss.cbe.service`
- The `javax.oss.IllegalArgumentException` (and others) was renamed to `javax.oss.OssIllegalArgumentException` to avoid confusion with the correspondent Exceptions in `java.lang` package.
- Javadoc tags and parameter names for better generation of the XML Schema for the XML over JMS integration profile

Code 3: Changes in javax.oss.order.JVTActivationSession

```
***************
*** 169,176 ****
   *
   * @see OrderValue
   * @see JVTActivationHome
!  * @author Stefan Vaillant, Nokia Networks
!  * @version 1.0
   */
  public interface JVTActivationSession extends javax.oss.JVTSession {

--- 169,177 ----
   *
   * @see OrderValue
   * @see JVTActivationHome
!  * @author Stefan Vaillant, Nokia Networks; Andreas Ebbert, Nokia Networks
!  * @version 1.1
!  * @ossj:jvtsession
   */
  public interface JVTActivationSession extends javax.oss.JVTSession {

***************
*** 211,221 ****
       * @return ServiceValue
       * @param serviceType A fully qualified type name.
       * @throws RemoteException
!      * @exception javax.oss.IllegalArgumentException if precondition violated:<br>
       * serviceType most be one of the strings returned by {@link #getServiceTypes}
       */
!    public javax.oss.service.ServiceValue makeServiceValue(String serviceType)
!    throws javax.oss.IllegalArgumentException, RemoteException;


     // Order Types
--- 212,222 ----
       * @return ServiceValue
       * @param serviceType A fully qualified type name.
       * @throws RemoteException
!      * @exception javax.oss.OssIllegalArgumentException if precondition violated:<br>
       * serviceType most be one of the strings returned by {@link #getServiceTypes}
       */
!    public javax.oss.cbe.service.ServiceValue makeServiceValue(String serviceType)
!    throws javax.oss.OssIllegalArgumentException, RemoteException;


     // Order Types
***************
```

```
*** 295,305 ****
      *
      * @param typeName The fully qualified type of the OrderValue that is to be created.
      * @return an Objects which is derived from the given typeName.
!     * @exception javax.oss.IllegalArgumentException if precondition violated: <br>
      * type must be one of the strings returned by {@link #getOrderTypes}
      */
      public OrderValue makeOrderValue(String typeName)
!     throws javax.oss.IllegalArgumentException, java.rmi.RemoteException;


    // State handling of orders
--- 296,306 ----
      *
      * @param typeName The fully qualified type of the OrderValue that is to be created.
      * @return an Objects which is derived from the given typeName.
!     * @exception javax.oss.OssIllegalArgumentException if precondition violated: <br>
      * type must be one of the strings returned by {@link #getOrderTypes}
      */
      public OrderValue makeOrderValue(String typeName)
!     throws javax.oss.OssIllegalArgumentException, java.rmi.RemoteException;


    // State handling of orders
***************
*** 343,362 ****
       *         An implementation may populate further attributes,
       *         especially the ones that were populated in <code>value</code>
       * </ul>
!      * @param value
       * @throws RemoteException
!      * @exception javax.oss.IllegalArgumentException if precondition violated: <br>
       * <code>value</code> must be a return value of {@link #makeOrderValue
makeOrderValue}
       * from this implementation bean instance.
!      * @exception javax.oss.IllegalAttributeValueException if precondition violated:
<br>
       * all populated attribute values must be in the value range
       * that is allowed be the implementation.
       * @exception javax.ejb.CreateException in case of an implementation internal
problem.
       * @return the server assigned key for that order
       */
!    public OrderKey createOrderByValue(OrderValue value)
!    throws   javax.oss.IllegalArgumentException,
!             javax.oss.IllegalAttributeValueException,
              javax.ejb.CreateException,
              java.rmi.RemoteException;

--- 344,363 ----
       *         An implementation may populate further attributes,
       *         especially the ones that were populated in <code>value</code>
       * </ul>
!      * @param orderValue
       * @throws RemoteException
!      * @exception javax.oss.OssIllegalArgumentException if precondition violated: <br>
       * <code>value</code> must be a return value of {@link #makeOrderValue
makeOrderValue}
       * from this implementation bean instance.
!      * @exception javax.oss.OssIllegalAttributeValueException if precondition violated:
<br>
       * all populated attribute values must be in the value range
       * that is allowed be the implementation.
       * @exception javax.ejb.CreateException in case of an implementation internal
problem.
       * @return the server assigned key for that order
       */
!    public OrderKey createOrderByValue(OrderValue orderValue)
!    throws   javax.oss.OssIllegalArgumentException,
!             javax.oss.OssIllegalAttributeValueException,
              javax.ejb.CreateException,
              java.rmi.RemoteException;
```

```
***************
*** 370,380 ****
      *      try {
      *          OrderKey k = createOrderByValue(values[i]);
      *          result[i] = new OrderKeyResultImpl(k, true, null);
!     *      } catch (javax.oss.IllegalAttributeValueException e) {
      *          result[i] = new OrderKeyResultImpl(null, false, e);
      *      } catch (CreateException ce) {
      *          result[i] = new OrderKeyResultImpl(null, false, ce);
!     *      } catch (javax.oss.IllegalArgumentException e) {
      *          result[i] = new OrderKeyResultImpl(null, false, e);
      *      }
      *  }
--- 371,381 ----
      *      try {
      *          OrderKey k = createOrderByValue(values[i]);
      *          result[i] = new OrderKeyResultImpl(k, true, null);
!     *      } catch (javax.oss.OssIllegalAttributeValueException e) {
      *          result[i] = new OrderKeyResultImpl(null, false, e);
      *      } catch (CreateException ce) {
      *          result[i] = new OrderKeyResultImpl(null, false, ce);
!     *      } catch (javax.oss.OssIllegalArgumentException e) {
      *          result[i] = new OrderKeyResultImpl(null, false, e);
      *      }
      *  }
***************
*** 382,389 ****
      *  return result;
      * </pre>
      */
!    public OrderKeyResult[] tryCreateOrdersByValues(OrderValue[] values)
!    throws   javax.oss.IllegalArgumentException,
              javax.ejb.CreateException,
              java.rmi.RemoteException;

--- 383,390 ----
      *  return result;
      * </pre>
      */
!    public OrderKeyResult[] tryCreateOrdersByValues(OrderValue[] orderValues)
!    throws   javax.oss.OssIllegalArgumentException,
              javax.ejb.CreateException,
              java.rmi.RemoteException;

***************
*** 402,423 ****
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().equals(RUNNING)</code>
      * </ul>
!     * @param key Key for an order.
      * @exception
!     *  javax.oss.IllegalStateException if precondition violated:
      *       <code>getOrder(key).getState().startsWith(NOT_STARTED)</code>
      * @exception
!     *  javax.oss.IllegalArgumentException if precondition violated:
      *       The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *       of this JVTActivationSession before.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      *
      */
!    public void startOrderByKey(OrderKey key)
!    throws   javax.oss.IllegalStateException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.IllegalArgumentException,
              RemoteException;

     /**
--- 403,424 ----
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
```

```
      * <li><code>msg.getObject().getCurrentState().equals(RUNNING)</code>
      * </ul>
!     * @param orderKey Key for an order.
      * @exception
!     *  javax.oss.OssIllegalStateException if precondition violated:
      *       <code>getOrder(key).getState().startsWith(NOT_STARTED)</code>
      * @exception
!     *  javax.oss.OssIllegalArgumentException if precondition violated:
      *       The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *       of this JVTActivationSession before.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      *
      */
!    public void startOrderByKey(OrderKey orderKey)
!    throws   javax.oss.OssIllegalStateException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.OssIllegalArgumentException,
              RemoteException;

     /**
***************
*** 430,438 ****
      *           startOrderByKey(orderKeys[i]);
      *       } catch (javax.ejb.ObjectNotFoundException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!     *       } catch (javax.oss.IllegalStateException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!     *       } catch (javax.oss.IllegalArgumentException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
      *       }
      *   }
--- 431,439 ----
      *           startOrderByKey(orderKeys[i]);
      *       } catch (javax.ejb.ObjectNotFoundException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!     *       } catch (javax.oss.OssIllegalStateException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!     *       } catch (javax.oss.OssIllegalArgumentException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
      *       }
      *   }
***************
*** 440,447 ****
      *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
      * </pre>
      */
!    public OrderKeyResult[] tryStartOrdersByKeys(OrderKey[] keys)
!    throws   javax.oss.IllegalArgumentException,
              RemoteException;

     /**
--- 441,448 ----
      *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
      * </pre>
      */
!    public OrderKeyResult[] tryStartOrdersByKeys(OrderKey[] orderKeys)
!    throws   javax.oss.OssIllegalArgumentException,
              RemoteException;

     /**
***************
*** 461,485 ****
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().startsWith(SUSPENDED)</code>
      * </ul>
!     * @param key Key for an order.
      * @exception
!     *  javax.oss.IllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(NOT_STARTED)</code>
      * @exception
```

```
!     *  javax.oss.IllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception
!     *  javax.oss.UnsupportedOperationException if precondition violated:
      *    <code>JVTActivationSessionOptionalOps.SUSPEND_ORDER_BY_KEY returned by
getSupportedOptionalOperations()</code>
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      */
!    public void suspendOrderByKey(OrderKey key)
!    throws javax.oss.IllegalStateException,
!           javax.oss.IllegalArgumentException,
            javax.ejb.ObjectNotFoundException,
!           javax.oss.UnsupportedOperationException,
            RemoteException;

     /**
--- 462,486 ----
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().startsWith(SUSPENDED)</code>
      * </ul>
!     * @param orderKey Key for an order.
      * @exception
!     *  javax.oss.OssIllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(NOT_STARTED)</code>
      * @exception
!     *  javax.oss.OssIllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception
!     *  javax.oss.OssUnsupportedOperationException if precondition violated:
      *    <code>JVTActivationSessionOptionalOps.SUSPEND_ORDER_BY_KEY returned by
getSupportedOptionalOperations()</code>
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      */
!    public void suspendOrderByKey(OrderKey orderKey)
!    throws javax.oss.OssIllegalStateException,
!           javax.oss.OssIllegalArgumentException,
            javax.ejb.ObjectNotFoundException,
!           javax.oss.OssUnsupportedOperationException,
            RemoteException;

     /**
***************
*** 498,522 ****
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().startsWith(RUNNING)</code>
      * </ul>
!     * @param key Key for an order.
      * @exception
!     *  javax.oss.IllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(SUSPENDED)</code>
      * @exception
!     *  javax.oss.IllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception
!     *  javax.oss.UnsupportedOperationException if precondition violated:
      *    <code>JVTActivationSessionOptionalOps.RESUME_ORDER_BY_KEY returned by
getSupportedOptionalOperations()</code>
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      */
!    public void resumeOrderByKey(OrderKey key)
!    throws javax.oss.IllegalStateException,
!           javax.oss.IllegalArgumentException,
            javax.ejb.ObjectNotFoundException,
```

```
!           javax.oss.UnsupportedOperationException,
            RemoteException;

     /**
--- 499,523 ----
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().startsWith(RUNNING)</code>
      * </ul>
!     * @param orderKey Key for an order.
      * @exception
!     *  javax.oss.OssIllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(SUSPENDED)</code>
      * @exception
!     *  javax.oss.OssIllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception
!     *  javax.oss.OssUnsupportedOperationException if precondition violated:
      *    <code>JVTActivationSessionOptionalOps.RESUME_ORDER_BY_KEY returned by
getSupportedOptionalOperations()</code>
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      */
!    public void resumeOrderByKey(OrderKey orderKey)
!    throws javax.oss.OssIllegalStateException,
!           javax.oss.OssIllegalArgumentException,
            javax.ejb.ObjectNotFoundException,
!           javax.oss.OssUnsupportedOperationException,
            RemoteException;

     /**
***************
*** 534,554 ****
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().startsWith(ABORTED)</code>
      * </ul>
!     * @param key Key for an order.
      * @exception
!     *  javax.oss.IllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(OPEN)</code>
      * @exception
!     *  javax.oss.IllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      */
!    public void abortOrderByKey (OrderKey key)
!    throws   javax.oss.IllegalStateException,
             javax.ejb.ObjectNotFoundException,
!            javax.oss.IllegalArgumentException,
             RemoteException;


--- 535,555 ----
      * <li><code>msg.getObject().getOrderKey().equals(key)</code>
      * <li><code>msg.getObject().getCurrentState().startsWith(ABORTED)</code>
      * </ul>
!     * @param orderKey Key for an order.
      * @exception
!     *  javax.oss.OssIllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(OPEN)</code>
      * @exception
!     *  javax.oss.OssIllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
      */
```

```
!    public void abortOrderByKey (OrderKey orderKey)
!    throws   javax.oss.OssIllegalStateException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.OssIllegalArgumentException,
              RemoteException;


***************
*** 561,571 ****
      *   for (int i=0; i&lt;orderKeys.length; i++) {
      *       try {
      *           abortOrderByKey(orderKeys[i]);
!     *       } catch (javax.oss.IllegalStateException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
      *       } catch (javax.ejb.ObjectNotFoundException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!     *       } catch (javax.oss.IllegalArgumentException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
      *       }
      *   }
--- 562,572 ----
      *   for (int i=0; i&lt;orderKeys.length; i++) {
      *       try {
      *           abortOrderByKey(orderKeys[i]);
!     *       } catch (javax.oss.OssIllegalStateException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
      *       } catch (javax.ejb.ObjectNotFoundException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!     *       } catch (javax.oss.OssIllegalArgumentException e) {
      *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
      *       }
      *   }
***************
*** 573,581 ****
      *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
      * </pre>
      */
!    public OrderKeyResult[] tryAbortOrdersByKeys (OrderKey[] keys)
!    throws   javax.oss.IllegalStateException,
!             javax.oss.IllegalArgumentException,
              RemoteException;

     /**
--- 574,582 ----
      *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
      * </pre>
      */
!    public OrderKeyResult[] tryAbortOrdersByKeys (OrderKey[] orderKeys)
!    throws   javax.oss.OssIllegalStateException,
!             javax.oss.OssIllegalArgumentException,
              RemoteException;

     /**
***************
*** 602,625 ****
      * <li><code>msg.getObject().getOrderValue() returns the order value</code>
      * </ul>
      * @exception
!     *  javax.oss.IllegalStateException if precondition violated:
      *         <code>getOrder(key).getState().startsWith(CLOSED)</code>
      * @exception
!     *  javax.oss.IllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception
!     *  javax.oss.UnsupportedOperationException if precondition violated:
      *    <code>JVTActivationSessionOptionalOps.REMOVE_ORDER_BY_KEY returned by
getSupportedOptionalOperations()</code>
       * @exception javax.ejb.RemoveException in case of an implementation internal
problem.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
```

```
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
       */
!     public void removeOrderByKey(OrderKey key)
!     throws  javax.oss.IllegalArgumentException,
!             javax.oss.IllegalStateException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.UnsupportedOperationException,
              javax.ejb.RemoveException,
              java.rmi.RemoteException;

--- 603,626 ----
      * <li><code>msg.getObject().getOrderValue() returns the order value</code>
      * </ul>
      * @exception
!     *  javax.oss.OssIllegalStateException if precondition violated:
      *        <code>getOrder(key).getState().startsWith(CLOSED)</code>
      * @exception
!     *  javax.oss.OssIllegalArgumentException if precondition violated:
      *   The <code>key</code> has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
      *      of this JVTActivationSession before.
      * @exception
!     *  javax.oss.OssUnsupportedOperationException if precondition violated:
      *    <code>JVTActivationSessionOptionalOps.REMOVE_ORDER_BY_KEY returned by
getSupportedOptionalOperations()</code>
       * @exception javax.ejb.RemoveException in case of an implementation internal
problem.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
      *    The <code>order.getOrderKey()</code> must refer to an existing object.
       */
!     public void removeOrderByKey(OrderKey orderKey)
!     throws  javax.oss.OssIllegalArgumentException,
!             javax.oss.OssIllegalStateException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.OssUnsupportedOperationException,
              javax.ejb.RemoveException,
              java.rmi.RemoteException;

***************
*** 635,645 ****
       *           removeOrderByKey(orderKeys[i]);
       *       } catch(javax.ejb.ObjectNotFoundException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!      *       } catch(javax.oss.IllegalStateException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!      *       } catch(javax.oss.IllegalArgumentException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!      *       } catch(javax.oss.UnsupportedOperationException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
       *       }
       *   }
--- 636,646 ----
       *           removeOrderByKey(orderKeys[i]);
       *       } catch(javax.ejb.ObjectNotFoundException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!      *       } catch(javax.oss.OssIllegalStateException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!      *       } catch(javax.oss.OssIllegalArgumentException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
!      *       } catch(javax.oss.OssUnsupportedOperationException e) {
       *           result.add(new OrderKeyResultImpl(orderKeys[i], false, e));
       *       }
       *   }
***************
*** 647,655 ****
       *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
       * </pre>
       */
!     public OrderKeyResult[] tryRemoveOrdersByKeys(OrderKey[] keys)
!     throws  javax.oss.IllegalArgumentException,
!             javax.oss.UnsupportedOperationException,
              javax.ejb.RemoveException,
```

```
             java.rmi.RemoteException;

--- 648,656 ----
       *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
       * </pre>
       */
!     public OrderKeyResult[] tryRemoveOrdersByKeys(OrderKey[] orderKeys)
!     throws  javax.oss.OssIllegalArgumentException,
!             javax.oss.OssUnsupportedOperationException,
              javax.ejb.RemoveException,
              java.rmi.RemoteException;

***************
*** 678,699 ****
       * <li><code>msg.getObject() instanceof OrderAttributeValueChangeEvent</code>
       * <li><code>msg.getObject().getNewOrderValue()</code> contains all the attributes
that have been changed.
       * </ul>
!      * @param order OrderValue.
!      * @exception javax.oss.IllegalArgumentException (Programming Error) if precondition
violated:
       * <ul><li><code> order.isPopulated(KEY)</code></li>
       *    <li>precondition of getOrder(order.getOrderKey()) must be true</li>
       * </ul>
       * @exception javax.ejb.ObjectNotFoundException if precondition violated:
       *   The <code>order.getOrderKey()</code> must refer to an existing object.
!      * @exception javax.oss.IllegalAttributeValueException if precondition violated:
       * all populated attribute values must be in the value range
       * that is allowed be the implementation.
!      * @exception javax.oss.IllegalStateException if precondition violated:
       * The order must be in the correct state, at least NOT_STARTED.
       * For all other states, the implemantation may throw this exception
       * For state CLOSED it must throw this exception.
!      * @exception javax.oss.SetException in case of an implementation internal problem.
!      * @exception javax.oss.ResyncRequiredException if precondition violated:
       *   If checkConcurrentAccess is set, then a the lastUpdateVersionNumber must be
correct:
       *   <pre>
       *   checkConcurrentAccess =>
--- 679,700 ----
       * <li><code>msg.getObject() instanceof OrderAttributeValueChangeEvent</code>
       * <li><code>msg.getObject().getNewOrderValue()</code> contains all the attributes
that have been changed.
       * </ul>
!      * @param orderValue OrderValue.
!      * @exception javax.oss.OssIllegalArgumentException (Programming Error) if
precondition violated:
       * <ul><li><code> order.isPopulated(KEY)</code></li>
       *    <li>precondition of getOrder(order.getOrderKey()) must be true</li>
       * </ul>
       * @exception javax.ejb.ObjectNotFoundException if precondition violated:
       *   The <code>order.getOrderKey()</code> must refer to an existing object.
!      * @exception javax.oss.OssIllegalAttributeValueException if precondition violated:
       * all populated attribute values must be in the value range
       * that is allowed be the implementation.
!      * @exception javax.oss.OssIllegalStateException if precondition violated:
       * The order must be in the correct state, at least NOT_STARTED.
       * For all other states, the implemantation may throw this exception
       * For state CLOSED it must throw this exception.
!      * @exception javax.oss.OssSetException in case of an implementation internal
problem.
!      * @exception javax.oss.OssResyncRequiredException if precondition violated:
       *   If checkConcurrentAccess is set, then a the lastUpdateVersionNumber must be
correct:
       *   <pre>
       *   checkConcurrentAccess =>
***************
*** 701,713 ****
       *             getOrder(order.getOrderKey()).getLastUpdateVersionNumber()
       *   </pre>
       */
!     public void setOrderByValue(OrderValue order, boolean checkConcurrentAccess)
```

```
!    throws   javax.oss.IllegalArgumentException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.IllegalAttributeValueException,
!             javax.oss.IllegalStateException,
!             javax.oss.SetException,
!             javax.oss.ResyncRequiredException,
              RemoteException;

     /**
--- 702,714 ----
       *             getOrder(order.getOrderKey()).getLastUpdateVersionNumber()
       *   </pre>
       */
!    public void setOrderByValue(OrderValue orderValue, boolean checkConcurrentAccess)
!    throws   javax.oss.OssIllegalArgumentException,
              javax.ejb.ObjectNotFoundException,
!             javax.oss.OssIllegalAttributeValueException,
!             javax.oss.OssIllegalStateException,
!             javax.oss.OssSetException,
!             javax.oss.OssResyncRequiredException,
              RemoteException;

     /**
***************
*** 721,746 ****
      *           setOrderByValue(values[i], checkAccess);
      *       } catch (ObjectNotFoundException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (javax.oss.ResyncRequiredException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (IllegalAttributeValueException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (javax.oss.IllegalStateException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (javax.oss.IllegalArgumentException e) {
      *           if (!values[i].isPopulated(OrderValue.KEY)) {
      *               result.add(new OrderKeyResultImpl(null, false, e));
      *           } else {
      *               result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false,
e));
      *           }
!     *       } catch (javax.oss.SetException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
      *       }
      *   }
      *
      *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
      * </pre>
!     * @exception javax.oss.IllegalArgumentException (Programming Error) if precondition
violated:
      * <ul><li>All values must be non null:<br>
      *        <code>values != null && forall v in values: v!= null</code>
      *    </li>
--- 722,747 ----
      *           setOrderByValue(values[i], checkAccess);
      *       } catch (ObjectNotFoundException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (javax.oss.OssResyncRequiredException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (OssIllegalAttributeValueException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (javax.oss.OssIllegalStateException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
!     *       } catch (javax.oss.OssIllegalArgumentException e) {
      *           if (!values[i].isPopulated(OrderValue.KEY)) {
      *               result.add(new OrderKeyResultImpl(null, false, e));
      *           } else {
      *               result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false,
e));
      *           }
!     *       } catch (javax.oss.OssSetException e) {
      *           result.add(new OrderKeyResultImpl(values[i].getOrderKey(), false, e));
```

```
      *       }
      *   }
      *
      *   return (OrderKeyResult[])result.toArray(new OrderKeyResultImpl[0]);
      * </pre>
!     * @exception javax.oss.OssIllegalArgumentException (Programming Error) if
precondition violated:
      * <ul><li>All values must be non null:<br>
      *        <code>values != null && forall v in values: v!= null</code>
      *    </li>
***************
*** 750,758 ****
      *    <li>All keys must be different</li>
      * </ul>
      */
!    public OrderKeyResult[] trySetOrdersByValues( OrderValue[] values, boolean
checkConcurrentAccess)
!    throws   javax.oss.IllegalArgumentException,
!             javax.oss.SetException,
             RemoteException;

     /**
--- 751,759 ----
      *    <li>All keys must be different</li>
      * </ul>
      */
!    public OrderKeyResult[] trySetOrdersByValues( OrderValue[] orderValues, boolean
checkConcurrentAccess)
!    throws   javax.oss.OssIllegalArgumentException,
!             javax.oss.OssSetException,
             RemoteException;

     /**
***************
*** 800,806 ****
      *   </pre>
      * </li>
      * </ul>
!     * @param key identifies the order.
      * @param attributeNames only extract part of the attributes.
      *        In this case this is null, all possible properties are extracted.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
--- 801,807 ----
      *   </pre>
      * </li>
      * </ul>
!     * @param orderKey identifies the order.
      * @param attributeNames only extract part of the attributes.
      *        In this case this is null, all possible properties are extracted.
      * @exception javax.ejb.ObjectNotFoundException if precondition violated:
***************
*** 808,814 ****
      * <li>The order as identified by the key must exist.
      * </li>
      * </ul>
!     * @exception javax.oss.IllegalArgumentException (Programming Error) if precondition
violated:
      * <ul>
      * <li>Key must be a valid key for this implementation:<br>
      *      The <code>key != null</code> and has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
--- 809,815 ----
      * <li>The order as identified by the key must exist.
      * </li>
      * </ul>
!     * @exception javax.oss.OssIllegalArgumentException (Programming Error) if
precondition violated:
      * <ul>
      * <li>Key must be a valid key for this implementation:<br>
      *      The <code>key != null</code> and has been returned by {@link
JVTActivationSession#createOrderByValue createOrderByValue}
***************
```

```
*** 825,832 ****
      * </li>
      * </ul>
      */
!     public OrderValue getOrderByKey(OrderKey key, String[] attributeNames)
!     throws   javax.oss.IllegalArgumentException,
               javax.ejb.ObjectNotFoundException,
               RemoteException;

--- 826,833 ----
      * </li>
      * </ul>
      */
!     public OrderValue getOrderByKey(OrderKey orderKey, String[] attributeNames)
!     throws   javax.oss.OssIllegalArgumentException,
               javax.ejb.ObjectNotFoundException,
               RemoteException;

***************
*** 835,841 ****
      * @see #getOrderByKey(OrderKey,String[])
      */
     public OrderValue getOrderByKeyAllAttr(OrderKey key)
!     throws   javax.oss.IllegalArgumentException,
               javax.ejb.ObjectNotFoundException,
               RemoteException;

--- 836,842 ----
      * @see #getOrderByKey(OrderKey,String[])
      */
     public OrderValue getOrderByKeyAllAttr(OrderKey key)
!     throws   javax.oss.OssIllegalArgumentException,
               javax.ejb.ObjectNotFoundException,
               RemoteException;

***************
*** 860,873 ****
      * More detailed semantics, including postconditions and exceptions,
      * are described in "getOrder".
      * @see #getOrderByKey(OrderKey, String[])
!     * @exception javax.oss.IllegalArgumentException (Programming Error) if precondition
violated:
      * <li>preconditions for attributeNames as in getOrder.</li>
      * <li>All keys must be non null:<br>
      *     <code>keys != null && forall k in keys: k!= null</code>
      * </li>
      */
!     public OrderValue[] getOrdersByKeys(OrderKey[] keys, String[] attributeNames)
!     throws   javax.oss.IllegalArgumentException,
               RemoteException;

     /**
--- 861,874 ----
      * More detailed semantics, including postconditions and exceptions,
      * are described in "getOrder".
      * @see #getOrderByKey(OrderKey, String[])
!     * @exception javax.oss.OssIllegalArgumentException (Programming Error) if
precondition violated:
      * <li>preconditions for attributeNames as in getOrder.</li>
      * <li>All keys must be non null:<br>
      *     <code>keys != null && forall k in keys: k!= null</code>
      * </li>
      */
!     public OrderValue[] getOrdersByKeys(OrderKey[] orderKeys, String[] attributeNames)
!     throws   javax.oss.OssIllegalArgumentException,
               RemoteException;

     /**
***************
*** 930,936 ****
      * </pre>
      * <p>
```

```
      *
!     * @exception javax.oss.IllegalArgumentException if precondition violated:
      * <ul>
      * <li>At least one template must be provided: <br>
      * <code>templates != null && templates.size() > 0</code>
--- 931,937 ----
      * </pre>
      * <p>
      *
!     * @exception javax.oss.OssIllegalArgumentException if precondition violated:
      * <ul>
      * <li>At least one template must be provided: <br>
      * <code>templates != null && templates.size() > 0</code>
***************
*** 940,952 ****
      * </li>
      * <li>attributeNames must fullfill the same conditions as in getOrder().</li>
      * </ul>
!     * @exception javax.oss.UnsupportedOperationException if precondition violated:
      *   <code>JVTActivationSessionOptionalOps.GET_ORDERS_BY_TEMPLATE returned by
getSupportedOptionalOperations()</code>
      */
     public OrderValueIterator getOrdersByTemplates( OrderValue[] templates, String[]
attributeNames)
!    throws   javax.oss.IllegalArgumentException,
              RemoteException,
!             javax.oss.UnsupportedOperationException;


     // Query Methods
--- 941,953 ----
      * </li>
      * <li>attributeNames must fullfill the same conditions as in getOrder().</li>
      * </ul>
!     * @exception javax.oss.OssUnsupportedOperationException if precondition violated:
      *   <code>JVTActivationSessionOptionalOps.GET_ORDERS_BY_TEMPLATE returned by
getSupportedOptionalOperations()</code>
      */
     public OrderValueIterator getOrdersByTemplates( OrderValue[] templates, String[]
attributeNames)
!    throws   javax.oss.OssIllegalArgumentException,
              RemoteException,
!             javax.oss.OssUnsupportedOperationException;


     // Query Methods
***************
*** 979,1013 ****
       * </ul>
      * @param queryName identifies the type of query.
     * @exception
!    *  javax.oss.IllegalArgumentException if precondition violated:
     *  <ul>
     *  <li>queryName returned by getQueryTypes()
     *  </ul>
     **/
     public javax.oss.QueryValue makeQueryValue(String queryName)
!    throws javax.oss.IllegalArgumentException, RemoteException;

     /**
      * Runs a (complex) query and returns the matching orders.
      *
!     * @param parameters must be one of the value objects returned by makeQueryValue.
      * @param attributeNames indicates which attributes should be populated in the
result.
      *        This is further explained in getOrder().
     * @exception
!    *  javax.oss.IllegalArgumentException if precondition violated:
     *  <ul>
     *  <li>For attributeNames, the conditions as listed in getOrder() apply.</li>
     *  <li>parameter has been retrieved by calling makeQueryValue()
     *  </ul>
```

```
     **/
!    public OrderValueIterator queryOrders( javax.oss.QueryValue parameters, String[]
attributeNames)
!    throws javax.oss.IllegalArgumentException, java.rmi.RemoteException;

     // Metadata Methods
     // -------------

      /**
!      * Gives information which methods will not throw UnsupportedOperationException.
       *
       * <p><b>Postcondition:</b>
       * <ul>
--- 980,1014 ----
       * </ul>
      * @param queryName identifies the type of query.
     * @exception
!    *  javax.oss.OssIllegalArgumentException if precondition violated:
     *  <ul>
     *  <li>queryName returned by getQueryTypes()
     *  </ul>
     **/
     public javax.oss.QueryValue makeQueryValue(String queryName)
!    throws javax.oss.OssIllegalArgumentException, RemoteException;

     /**
      * Runs a (complex) query and returns the matching orders.
      *
!     * @param queryValue must be one of the value objects returned by makeQueryValue.
      * @param attributeNames indicates which attributes should be populated in the
result.
      *        This is further explained in getOrder().
     * @exception
!    *  javax.oss.OssIllegalArgumentException if precondition violated:
     *  <ul>
     *  <li>For attributeNames, the conditions as listed in getOrder() apply.</li>
     *  <li>parameter has been retrieved by calling makeQueryValue()
     *  </ul>
     **/
!    public OrderValueIterator queryOrders( javax.oss.QueryValue queryValue, String[]
attributeNames)
!    throws javax.oss.OssIllegalArgumentException, java.rmi.RemoteException;

     // Metadata Methods
     // -------------

      /**
!      * Gives information which methods will not throw OssUnsupportedOperationException.
       *
       * <p><b>Postcondition:</b>
       * <ul>
***************
*** 1056,1068 ****
       * </ul>
      * @param eventType identifies the type of the event.
      * @exception
!     *  javax.oss.IllegalArgumentException if precondition violated:
      *  <ul>
      *  <li>eventType returned by getEventTypes()
      *  </ul>
       */
      public javax.oss.EventPropertyDescriptor getEventDescriptor(String eventType)
!     throws javax.oss.IllegalArgumentException , java.rmi.RemoteException;


  }
--- 1057,1069 ----
       * </ul>
      * @param eventType identifies the type of the event.
      * @exception
!     *  javax.oss.OssIllegalArgumentException if precondition violated:
      *  <ul>
```

```
      *  <li>eventType returned by getEventTypes()
      *  </ul>
       */
      public javax.oss.EventPropertyDescriptor getEventDescriptor(String eventType)
!     throws javax.oss.OssIllegalArgumentException , java.rmi.RemoteException;


  }
```

2.2.2 OrderValue

Changes in the `OrderValue` interface are only needed to use the CBE ServiceValue.

Code 4: Changes in javax.oss.order.OrderValue

```
***************
*** 19,29 ****

  package javax.oss.order;

! import javax.oss.service.*;
  import java.util.Date;

  /**
   * Value type interface for accessing orders.
   * <p>
   * Orders are stored inside the JVTActivationSession. A client can retrieve and
   * change orders indirectly by using the methods of an JVTActivationSession.
--- 19,32 ----

  package javax.oss.order;

!
! import javax.oss.cbe.service.ServiceValue;
! import javax.oss.cbe.service.ServiceKeyResult;
  import java.util.Date;

  /**
   * Value type interface for accessing orders.
+  *
   * <p>
   * Orders are stored inside the JVTActivationSession. A client can retrieve and
   * change orders indirectly by using the methods of an JVTActivationSession.
```

2.2.3 OrderValueIterator

The `OrderValueIterator` interface is now using the new exceptions from the OSS Common API.

Code 5: Changes in javax.oss.order.OrderValueIterator

```
***************
*** 93,99 ****
       * @return the array of OrderValues
       */
        public OrderValue[] getNextOrder(int howMany)
!       throws javax.oss.IllegalArgumentException, RemoteException;

  };
```

```
--- 93,99 ----
       * @return the array of OrderValues
       */
        public OrderValue[] getNextOrder(int howMany)
!       throws javax.oss.OssIllegalArgumentException, RemoteException;

  };
```

2.2.4 ServiceKey

For backwards compatibility the `ServiceKey` interface is extending the CBE `ServiceKey` interface.

Code 6: Changes in javax.oss.service.ServiceKey

```
***************
*** 18,27 ****
   */

  /**
!  * @author Stefan Vaillant, Nokia Networks
!  * @version 1.0
   */
  package javax.oss.service;

! public interface ServiceKey extends javax.oss.ManagedEntityKey {
  }
--- 18,29 ----
   */

  /**
!  * @author Stefan Vaillant, Nokia Networks; Andreas Ebbert, Nokia Networks
!  * @version 1.1
!  * @deprecated
!  * @ossj:managedentitykey
   */
  package javax.oss.service;

! public interface ServiceKey extends javax.oss.cbe.service.ServiceKey {
  }
```

2.2.5 ServiceKeyResult

For backwards compatibility the `ServiceKeyResult` interface is extending the CBE `ServiceKeyResult` interface.

Code 7: Changes in javax.oss.service.ServiceKeyResult

```
***************
*** 22,33 ****

  /**
   *
!  * @author Stefan Vaillant, Nokia Networks
!  * @version 1.0
!  */
! public interface ServiceKeyResult extends javax.oss.ManagedEntityKeyResult {
!
```

```
!     public ServiceKey getServiceKey();

  }

--- 22,32 ----

  /**
   *
!  * @author Stefan Vaillant, Nokia Networks; Andreas Ebbert, Nokia Networks
!  * @version 1.1
!  * @ossj:complexdata
!  */
! public interface ServiceKeyResult extends javax.oss.cbe.service.ServiceKeyResult {

  }
```

2.2.6 ServiceState

For backwards compatibility the `ServiceState` interface is extending the CBE `ServiceStatus` interface.

Code 8: Changes in javax.oss.service.ServiceState

```
***************
*** 1,35 ****
- /*
-  * Copyright (c) 2002
-  * Cisco Systems, Inc., Ericsson Radio Systems AB.,
-  * Motorola, Inc., NEC Corporation, Nokia Networks Oy,
-  * Nortel Networks Limited, Sun Microsystems, Inc.,
-  * Telcordia Technologies, Inc., Cygent, Inc.,
-  * Agilent Technologies, Inc., BEA Systems, Inc.,
-  * Digital Fairway Corporation, Orchestream Holdings plc.
-  *
-  * All rights reserved.  Use is subject to license terms.
-  *
-  * DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED
-  * CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY
-  * IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR FOR A
-  * PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT
-  * TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY
-  * INVALID.
-  */
-
  package javax.oss.service;

- import java.io.Serializable;
-
  /**
!   * String constants that define the predefined service states.
!  * @author Stefan Vaillant, Nokia Networks
!  * @version 1.0
!   */

- public interface ServiceState extends Serializable {
-
-    static final String ACTIVE             = "active";
-
-    static final String INACTIVE           = "inactive";
  }
--- 1,9 ----
  package javax.oss.service;

  /**
!  * @author aebbert
```

```
!   * @deprecated
!   */
! public interface ServiceState extends javax.oss.cbe.service.ServiceStatus {

  }
```

2.2.7 ServiceValue

For backwards compatibility the `ServiceValue` interface is extending the CBE `ServiceValue` interface.

Code 9: Changes in javax.oss.service.ServiceValue

```
***************
*** 20,86 ****
  package javax.oss.service;

  /** Represents the service object in the client.
!  * @author Stefan Vaillant, Nokia Networks
!  * @version 1.0
   */
! public interface ServiceValue extends javax.oss.ManagedEntityValue {
!
!     public static final String STATE = "state";  // attribute name
!     public static final String SUBSCRIBER_ID = "subscriberId";  // attribute name
!
!     /** Same as {@link javax.oss.ManagedEntityValue#setManagedEntityKey
setManagedEntityKey}.
!      * The key might be talken from the Service Inventory (not part of theis API).
!      * @param key identifying the service instance to modify.
!      */
!     public void setServiceKey(ServiceKey key);
!     /** See {@link javax.oss.ManagedEntityValue#getManagedEntityKey getManagedEntityKey}
!      * @return the key that this order will modify.
!      */
!     public ServiceKey getServiceKey();
!
!     /**
!      * Make a new key for this value object.
!      * This method is equivalent to {@link
javax.oss.ManagedEntityValue#makeManagedEntityKey makeManagedEntityKey}.
!      * The semantic of the make operation is equivalent to a null constructor for that
type.
!      * This means that this is not the way to create unique keys for a ServiceValue
object, the method
!      * just returns an implementation of the ServiceKey class.
!      * <p>
!      * <p><b>Postcondition:</b>
!      * <ul><li><code>result.getPrimarykey() == null</code></li>
!      *    <li><code>this instanceof Class.forName(result.getType())</code></li>
!      * </ul>
!      *
!      * @see #setServiceKey
!      */
!     public ServiceKey makeServiceKey();
!
!     /** Set the state the service should have after order has been executed.
!      * @param state see {@link javax.oss.service.ServiceState}
!      */
!     public void setState(String state);
!     /** See {@link #setState }
!      * @return the state of the service
!      */
!     public String getState();
!
!     /**
```

```
!      * Set the subscriber who is using this service.
!      * <p>
!      * This is an indication which subscriber or customer is using
!      * the service. The format of the string is not specified.
!      * However, it is expected that the string will be reference
!      * to a customer or subscriber database entry.
!      * <p>
!      * @see #getSubscriberId
!      * @param subscriber A string identifying the subscriber.
!      */
!     public void setSubscriberId(String subscriber);
!
!     /**
!      * Get the subscriber who is using this service.
!      * @see #setSubscriberId
!      * @return the subscriber Identification.
!      */
!     public String getSubscriberId();
!
  }
--- 20,29 ----
  package javax.oss.service;

  /** Represents the service object in the client.
!  * @author Stefan Vaillant, Nokia Networks; Andreas Ebbert, Nokia Networks
!  * @version 1.1
!  * @deprecated
!  * @ossj:managedentityvalue
   */
! public interface ServiceValue extends javax.oss.cbe.service.ServiceValue {
  }
```

2.3 XML Schema

As all OSS/J APIs the OSS Service Activation API also defines an XML Schema for the XML over JMS integration profile. Changes in this section were made to let the XML Schema generation tool produce a correct XML Schema.

Note that also some of the other interfaces listed in the previous section were changed for this reason (see all the @ossj tags in the javadocs). Additionally some interfaces only changed in the Javadoc section and hence are not listed in this document, since it does not affect the Java Value Type interface.

2.3.1 OrderKey

All changes in the `OrderKey` interface are for the generation of the XML Schema. Especially the addition of the methods

- `getPrimaryKey`
- `getType`
- `getApplicationDN`

- `getApplicationContext`

is a sacrifice to the XML Schema generation tool, which is used. They are the exact method same method declaration as in the extended interface `javax.oss.ManagedEntityKey`.

Code 10: Changes in javax.oss.order.OrderKey

```
***************
*** 21,32 ****

  /**
   * Value type interface: representing an OSS wide unique key to an order.
!  * <p>
!  * @author Stefan Vaillant, Nokia Networks
!  * @version 1.0
!  *
   */

  public interface OrderKey extends javax.oss.ManagedEntityKey {
  }

--- 21,43 ----

  /**
   * Value type interface: representing an OSS wide unique key to an order.
!  *
!  * @author Stefan Vaillant, Nokia Networks; Andreas Ebbert, Nokia Networks
!  * @version 1.1
!  * @ossj:managedentitykey
   */

  public interface OrderKey extends javax.oss.ManagedEntityKey {
+
+      /** @see javax.oss.ManagedEntityKey#getPrimaryKey
+       * @ossj:nillableField value=false
+       * @ossj:minOccursField value=1
+       */
+    public Object getPrimaryKey();
+    public String getType();
+    public String getApplicationDN();
+    public javax.oss.ApplicationContext getApplicationContext();
+
+   }
```