

Oracle Utilities Application Framework

API Reference Notes

Release 4.x

E54104-03

August 2016

E54104-03

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Table of Contents

Introduction	5
What are the Public APIs?.....	5
What are <i>Not</i> APIs?.....	5
Changes and Deprecations.....	6
OUAF Release 4.0.0.0	6
com.splwg.base.domain.common.algorithm.Algorithm	6
getActiveVersion	6
com.splwg.base.support.algorithm.AbstractAlgorithmComponent.setAlgorithm	6
com.splwg.base.support.algorithm.AbstractAlgorithmComponent.setParametersFrom	6
OUAF Release 4.0.2.0	7
com.splwg.base.support.FrameworkSession.clear	7
com.splwg.base.support.FrameworkSession.flush	7
com.splwg.base.domain.common.maintenanceObject	7
MaintenanceObjectInformationAlgorithmSpot.setEntity	7
java.util.HashSet Behavior Change	7
OUAF Release 4.1.0.0	8
Deprecation: com.splwg.base.api.batch.AbstractThreadWorker.rollbackToSavePoint	8
Deprecation: com.splwg.base.api.batch.AbstractThreadWorker.setSavePoint.....	8
OUAF Release 4.2.0.0	8
com.splwg.base.api.batch.AbstractThreadWorker.getActiveBatchThread	8
com.splwg.base.support.context.FrameworkSession	9
getConnection	9
OUAF Release 4.2.0.1	9
Bug Fix: com.splwg.base.support.context.HibernateCacheHelper	9
flushAndClear.....	9
Bug Fix: com.splwg.base.api.PreparedStatement (Constructor).....	10
OUAF Release 4.2.0.2	10

Bug Fix: com.splwg.base.domain.common.masterConfiguration.....	10
MasterConfiguration_Id (Constructor).....	10
Bug Fix: com.splwg.base.api.PreparedStatement (Constructor).....	10
Bug Fix: com.splwg.base.domain.common.businessObject.	10
MessageRepository.businessObjectStatusOptionReplacedBy	10
OUAF Release 4.3.0.0.0	10
com.splwg.base.domain.common.businessObject. BusinessObjectEnterStatusAlgorithmSpot	10
com.splwg.base.domain.common.businessObject. BusinessObjectExitStatusAlgorithmSpot	11
Alta Look and Feel	11
Bug Fix: JS API – handleExpand/handleCollapse Methods in xslInclude.js.....	11
OUAF Release 4.3.0.2.0	12
Removed Two-decimal Money Limitation	12

Chapter 1

Introduction

This document describes recommended Oracle Utilities Application Framework (OUAF) API guidelines and any notable changes, deprecations, and bug fixes between versions.

What are the Public APIs?

The following packages are designated as public APIs. This code is sanctioned to be called by application customization layers, and consequently, these methods and classes will not likely change across releases. This sanctioned the packages include the name “api” and “domain”. In OUAF these are sanctioned packages:

- `com.splwg.base.api.*`
- `com.splwg.base.domain.*`

The same pattern can be applied to the application layer code. In the case of Customer Care & Billing, the package name would change from "base" to "ccb".

What are *Not* APIs?

Avoid calling other packages *not* mentioned above. The code is not designed to be called outside of OUAF code. For example, avoid the following packages:

- `com.splwg.base.support.*`
- `com.splwg.base.web.*`
- `com.splwg.base.xai.*`

These are often specialized calls intended to be used within the Application Framework. Calling specialized methods may, for example, change processing behavior by introducing unexpected events or skipping optimizations.

Chapter 2

Changes and Deprecations

This section contains notable changes and deprecations and may include public, non-public, and changes caused by other 3rd party libraries or programs.

OUAF Release 4.0.0.0

com.splwg.base.domain.common.algorithm.Algorithm. getActiveVersion

New Method:

com.splwg.base.domain.common.algorithm.AlgorithmComponentCache.getAlgorithmInfo().getActiveVersionInfo

Change Description: This call was changed to improve performance. This new method returns an instance of com.splwg.base.api.businessObject.AlgorithmInfo, which can return the active version's information. Alternatively, com.splwg.base.domain.common.algorithm.AlgorithmComponentCache can directly be used to instantiate any AlgorithmComponent implementation.

com.splwg.base.support.algorithm.AbstractAlgorithmComponent. setAlgorithm

Change Description: This method now requires a com.splwg.base.api.businessObject.AlgorithmInfo, a new object introduced to improve performance by caching the Algorithm information.

com.splwg.base.support.algorithm.AbstractAlgorithmComponent. setParametersFrom

Change Description: This method now requires a com.splwg.base.api.businessObject.AlgorithmInfo, a new object introduced to improve performance by caching the Algorithm information.

OUAF Release 4.0.2.0

`com.splwg.base.support.FrameworkSession.clear`

Change Description: This new method requires a string to describe the reason the clear method was called. It was added to aid in diagnosing performance issues.

Note: This is a non-public API.

`com.splwg.base.support.FrameworkSession.flush`

Change Description: This new method requires a string to describe the reason the clear method was called. It was added to aid in diagnosing performance issues.

Note: This is a non-public API.

`com.splwg.base.domain.common.maintenanceObject.MaintenanceObjectInformationAlgorithmSpot.setEntity`

Change Description: This method was changed from `setEntity(BusinessEntity<?> entity)` to `setEntity(BusinessEntity entity)`. The change was needed to support Java 6 (and may possibly have been a JDK 6 bug).

java.util.HashSet Behavior Change

Change Description: A behavior change in Java 6 exposed an issue that may require code changes. The iteration order of the HashSet changed. While HashSet does not guarantee any iteration order, there have been instances of code that relied on the old behavior.

In Hibernate, a given table/column can only participate in one property mapping. If a table/column is included in more than one property mapping (based on the defined constraints), the artifact generator must determine which property mapping to use. In prior versions of the artifact generator, based on Java version 5, if two constraints went to the same table and used some of the same fields, the preference order was determined based on the order that the constraints were returned from a database query. In OUAF 4.x, which is based on Java 6, that order comes out differently, resulting in a different property mapping that broke the product upgrade. This is a rare situation, but whenever it arises, the property mapping must come out the same as before the upgrade to avoid breaking the upgraded product.

A check was added to the artifact generator to always deterministically identify the property mapping. If the artifact generator cannot make a definitive decision, this results in the following error:

Constraints should differ in at least some way.

To address the error, the developer should decide which constraint should be followed and which should be "ignored".

Choose a field that is in the constraint to be ignored that is not in the other constraint, and add it to the annotation for the entity implementation class. For example,

```
* @BusinessEntity (cobolPersistence = true, tableName = CI_RC, ignoredFKFields =  
{ "STEP_RC_SEQ" },
```

OUAF Release 4.1.0.0

Deprecation:

com.splwg.base.api.batch.AbstractThreadWorker.rollbackToSavePoint

New Method: Use the `com.splwg.base.api.SavepointExecutable` class (see next item, `AbstractThreadWorker.setSavePoint`).

Deprecation:

com.splwg.base.api.batch.AbstractThreadWorker.setSavePoint

New Method: Use the `com.splwg.base.api.SavepointExecutable` class.

Code example:

```
//Before Code:
doSomething1();
doSomething2();
try() {
    setSavepoint("BeforeUpdate");
    doSomething3();
} catch (Exception e) {
    rollbackToSavepoint("BeforeUpdate");
    doSomething4();
}

//After code:
doSomething1();
doSomething2();
SavepointExecutable executable = new SavepointExecutable() {
    @Override
    protected void execute() {
        doSomething3();
    }
};
SavepointResult result = executable.doIt(savepointName);
if (result.hasError()) {
    doSomething4();
}
```

Note: This method does not support proper nesting of savepoints.

OUAF Release 4.2.0.0

com.splwg.base.api.batch.AbstractThreadWorker.getActiveBatchThread

New Method: `com.splwg.base.api.batch.AbstractThreadWorker.getBatchThreadMBean`

Change Description: The method name was changed to `getBatchThreadMBean`. This was a name change only.

com.splwg.base.support.context.FrameworkSession. getConnection

Change Description: The upgrade of hibernate changed the connection handling semantics wherein the unit of work is passed in to hibernate rather than obtaining a connection.

Note: This object is intended to be used internally. More appropriate APIs should be called. For example, if the purpose of retrieving the connection is to create a PreparedStatement, use `com.splwg.base.api.GenericBusinessObject.createPreparedStatement` instead.

Code Example 1 (preferred):

```
//Before Code:
FrameworkSession session = (FrameworkSession)SessionHolder.getSession();
Connection connection = FrameworkSession.getConnection(session);

// work with the connection
connection.prepareStatement();
...

//After code:
PreparedStatement query = createPreparedStatement(sql...);
```

Code Example 2 (uses internal APIs):

```
//Before Code:
FrameworkSession session = (FrameworkSession)SessionHolder.getSession();
Connection connection = FrameworkSession.getConnection(session);

// work with the connection
connection.prepareStatement();
...

//After code:
((FrameworkSession) getSession()).doWork(new ConnectionWork() {

    @Override
    public void execute(Connection connection) throws SQLException {

        // work with the connection
        connection.prepareStatement();
        ...
    }
});
```

OUAF Release 4.2.0.1

Bug Fix: com.splwg.base.support.context.HibernateCacheHelper. flushAndClear

Bug: The OUAF 2.2 method was not implemented in the initial OUAF Release 4.0.0.0. It has since been added with patch 16580225.

Note: This is a specialized call intended for a specific case in which a COBOL program is needed to clear the hibernate cache because it could not issue commits. It is not recommended for general use. Patch 7447321 contains information about this method.

Bug Fix: com.splwg.base.api.PreparedStatement (Constructor)

Bug: This method was removed in OUAF Release 4.2.0.1 and was restored with patch 17793307.

OUAF Release 4.2.0.2

Bug Fix: com.splwg.base.domain.common.masterConfiguration.MasterConfiguration_Id (Constructor)

Bug: The method signature was changed in OUAF Release 4.2.0.2 and was restored with patch 18347676.

Bug Fix: com.splwg.base.api.PreparedStatement (Constructor)

Bug: This method was removed in OUAF Release 4.2.0.1 and was restored with patch 17802274.

Bug Fix: com.splwg.base.domain.common.businessObject.MessageRepository.businessObjectStatusOptionReplacedBy

Bug: This method was removed in OUAF Release 4.2.0.1 and was restored with patch 18277216.

OUAF Release 4.3.0.0.0

Bug: This change was made to pass down the action details to BusinessObjectEnterStatusAlgorithmSpot and BusinessObjectExitStatusAlgorithmSpot.

17230588 - EXPOSE ACTION IN BO STATUS PLUG-IN SPOTS

com.splwg.base.domain.common.businessObject.BusinessObjectEnterStatusAlgorithmSpot

New Method:

```
/**
 * Set the Action of Business Object
 */
void setAction(BusinessObjectActionLookup boAction);
```

Change Description: The method is added to the plugin spot to pass down the action information to all implementation classes.

Note: As the spot is changed, all implementation classes should implement the method. Implementation can be left blank if the algorithm does not need the action information.

Code Example 1 (preferred):

```
//Before Code:
    There was no setAction() method.

//After code:
    public void setAction(BusinessObjectActionLookup boAction) {
    // null
    }
```

com.splwg.base.domain.common.businessObject. BusinessObjectExitStatusAlgorithmSpot

New Method:

```
/**
 * Set the Action of Business Object
 */
void setAction(BusinessObjectActionLookup boAction);
```

Change Description: The method is added to the plugin spot to pass down the action information to all implementation classes.

Note: As the spot is changed, all implementation classes should implement the method. The implementation can be left blank if the algorithm does not need the action information.

Code Example 1 (preferred):

```
//Before Code:
    There was no setAction() method.

//After code:
    public void setAction(BusinessObjectActionLookup boAction) {
        // null
    }
```

Alta Look and Feel

As a part of the new UI, OUAF now supports browsers in standards mode; for example, Microsoft Internet Explorer (MSIE) is supported in standards mode and *not* in compatibility mode. This may change the behavior of various UI elements such as HTML, JavaScript, and CSS, due to the underlying execution mode change in the browsers.

Previously, for example, `firstChild` on a certain DOM element would return the first child node of the element in the DOM but would ignore text nodes such as new lines, blank spaces, etc. Now, text nodes are no longer ignored, and `firstChild` will return the first child of the element in the DOM, which may be a text node.

For example, for the following XML snippet,

```
<root>
<foo>bar</foo>
</root>
```

`firstChild` on the `<root>` node will return a text node in MSIE (running in standards mode) in the new OUAF version, while in earlier versions it would return `<foo>`.

OUAF provides a cross-browser API with abstract browser variations to return consistent objects. The API can be found in the OUAF source file, `common.js`; refer to the file for additional details.

Bug Fix: JS API – `handleExpand/handleCollapse` Methods in `xslInclude.js`

Bug: `handleExpand()` and `handleCollapse()` functions in `xslInclude.js` used to expand/collapse a zone. Prior to Framework 4.3, these functions used to expected one or two arguments to expand/collapse a zone. However, since the release of Framework 4.3, these functions do not work as expected with a single argument. They now accept only two arguments, e.g., sequence id of the zone and user event object, to work as expected. The API change was a part of overall standardization and adoption of best practices for the JS layer to enable cross-browser support. Refer to patch 23745289 for details.

OUAF Release 4.3.0.2.0

Removed Two-decimal Money Limitation

Bug: A change was made to allow for 0, 1, 3, or more decimal places for money objects. Code that is incorrectly comparing number values using the Java “equals(),” method must be changed to use the compareTo() or isEqualTo() methods. This code might have worked in the past due to the hard-coding of two decimal places. A money value of “2” would be returned as “2.00”. This is no longer true, since the decimal places for money are now based on metadata, and returns the proper number of digits.

21788123 – INVOICING – CURRENCY / DECIMAL ISSUES

Incorrect Code Example

```
assertEquals(algComp.getTotalAmount(), new Money(adjustmentAmount, new  
Currency_Id("USD")));
```

Correct Code:

```
assertEquals(algComp.getTotalAmount().compareTo(new Money(adjustmentAmount, new  
Currency_Id("USD"))), 0);
```