

**Merative Social Program Management**  
**Apache Commons Library updates for 7.0.11 iFix11**

## Note

Before using this information and the product it supports, read the information in [Notices \(on page xii\)](#)

# Edition

This edition applies to Merative Social Program Management 7.0.11.0 iFix 11.

© **Merative US L.P. 2023**

# Chapter 1. Updating the Apache Commons Lang and Apache Commons Configuration libraries

## Context of the change

The Apache Commons Lang library has been updated from version 2.6 to 3.12.0, and the Apache Commons Configuration library has been updated from version 1.3 to 2.8.0 in Merative Social Program Management.

Apache Commons Lang provides a host of helper utilities for the `java.lang` API, notably string manipulation methods, basic numerical methods, object reflection, concurrency, creation and serialization and system properties. Additionally, it contains basic enhancements to `java.util.Date`, and to a series of utilities that are dedicated to help with building methods, such as `hashCode`, `toString`, and `equals`.

The Apache Commons Configuration software library provides a generic configuration interface that enables a Java application to read configuration data from a variety of sources. The Commons Configuration library provides typed access to single and multi-valued configuration parameters.

The Apache Commons Lang and Commons Configuration libraries are available in the Cúram Development Environment for Java (CDEJ), and the Cúram Server Development Environment for Java (SDEJ), in the following locations:

- `CuramSDEJ/lib/commons-lang-3.12.0.jar`
- `CuramCDEJ/lib/ext/jar/commons-lang-3.12.0.jar`
- `CuramCDEJ/lib/ext/jar/commons-configuration-2.8.0.jar`

## Additional updates

Two additional JAR files have been updated that have a dependency on the Apache Commons Lang library. A new JAR file, Apache Commons Text, has also been added because some APIs from Apache Commons Lang have been moved to the JAR file.

- The `ical4j.jar` file in `EJBServer/components/core/lib` has been updated from 1.0-beta4 to 3.2.1.
- The `json-schema-validator.jar` file in `EJBServer/components/Rest/graphql_lib` has been updated from 0.1.5 to 1.0.72.
- The `commons-text-1.9.jar` file has been added to `CuramCDEJ/lib/ext/jar/`.

## How to upgrade

1. The new JAR files are provided in the CDEJ and SDEJ, as well as the Rest and core EJBServer components. Remove any existing related files.
2. Regenerate the project class paths by building the server and the client.
3. Replace imports in the implementation code to refer to the appropriate classes and packages in the new JAR file versions. See the next section for examples of the updated package and method names.
4. Implement the required code changes across all classes that reference the affected JAR files, as appropriate. For examples, see the following *Potential required code changes* sections.

## Potential required code changes for Apache Commons Lang

The main change to note is the change of the `org.apache.commons.lang` package to `org.apache.commons.lang3`.

See the official Apache Commons Lang documentation for guidance on how to update implementation code: [https://commons.apache.org/proper/commons-lang/article3\\_0.html](https://commons.apache.org/proper/commons-lang/article3_0.html)

**Table 1. Differences between Apache Commons Lang 2.6 and 3.12.0 package names**

Apache Commons Lang 2.6	Apache Commons Lang 3.12.0
org.apache.commons.lang.SerializationUtils	org.apache.commons.lang3.SerializationUtils
org.apache.commons.lang.StringUtils	org.apache.commons.lang3.StringUtils
org.apache.commons.lang.ArrayUtils	org.apache.commons.lang3.ArrayUtils
org.apache.commons.lang.builder.EqualsBuilder	org.apache.commons.lang3.builder.EqualsBuilder
org.apache.commons.lang.builder.HashCodeBuilder	org.apache.commons.lang3.builder.HashCodeBuilder
org.apache.commons.lang.ObjectUtils	org.apache.commons.lang3.ObjectUtils
org.apache.commons.lang.math.NumberUtils	org.apache.commons.lang3.math.NumberUtils
org.apache.commons.lang.builder.CompareToBuilder	org.apache.commons.lang3.builder.CompareToBuilder
org.apache.commons.lang.time.DurationFormatUtils	org.apache.commons.lang3.time.DurationFormatUtils
org.apache.commons.lang.LocaleUtils	org.apache.commons.lang3.LocaleUtils

**Table 1. Differences between Apache Commons Lang 2.6 and 3.12.0 package names (continued)**

Apache Commons Lang 2.6	Apache Commons Lang 3.12.0
org.apache.commons.lang.reflect.FieldUtils	org.apache.commons.lang3.reflect.FieldUtils
org.apache.commons.lang.BooleanUtils	org.apache.commons.lang3.BooleanUtils

The following table shows examples of differences between Apache Commons Lang 2.6 and 3.12.0, where code changes in addition to the package name updates were required.

**Table 2. Differences between Apache Commons Lang 2.6 and 3.12.0 that require code changes in addition to package name updates**

Apache Commons Lang 2.6	Apache Commons Lang 3.12.0
org.apache.commons.lang.exception.ExceptionUtils:  <ul style="list-style-type: none"> <li>ExceptionUtils.getFullStackTrace(Throwable th);</li> </ul>	org.apache.commons.lang3.exception.ExceptionUtils:  <ul style="list-style-type: none"> <li>ExceptionUtils.getStackTrace(Throwable TB);</li> </ul>
org.apache.commons.lang.math.RandomUtils:  <ul style="list-style-type: none"> <li>RandomUtils.nextInt(Integer.MAX_VALUE);</li> </ul>	org.apache.commons.lang3.RandomUtils  <ul style="list-style-type: none"> <li>RandomUtils.nextInt(0,Integer.MAX_VALUE);</li> </ul>
org.apache.commons.lang.math.DoubleRange:  <ul style="list-style-type: none"> <li>DoubleRange doubleRange = new DoubleRange(Double lowerRange, Double higherRange);</li> <li>doubleRange.containsDouble(Double value);</li> </ul>	org.apache.commons.lang3.Range:  <ul style="list-style-type: none"> <li>Range&lt;Double&gt; doubleRange = Range.between(Double lowerRange, Double higherRange);</li> <li>doubleRange.contains(Double value);</li> </ul>
org.apache.commons.lang.math.NumberUtils:  <ul style="list-style-type: none"> <li>NumberUtils.isNumber(String)</li> </ul>	org.apache.commons.lang3.math.NumberUtils  <ul style="list-style-type: none"> <li>NumberUtils.isCreatable(String)</li> </ul>
org.apache.commons.lang.ObjectUtils	org.apache.commons.lang3.ObjectUtils

**Table 2. Differences between Apache Commons Lang 2.6 and 3.12.0 that require code changes in addition to package name updates (continued)**

Apache Commons Lang 2.6	Apache Commons Lang 3.12.0
<ul style="list-style-type: none"><li>• <code>ObjectUtils.equals(Object, Object)</code></li></ul>	<ul style="list-style-type: none"><li>• <code>java.util.Objects.equals(Object, Object)</code></li></ul>

The following table shows some examples where a class existed in Apache Commons Lang 2.6 but can now be referenced from the Apache Commons Text 1.9 library.

**Table 3. Apache Commons Lang 2.6 classes that can now be referenced from Apache Commons Text 1.9 library**

Apache Commons Lang 2.6	Apache Commons Text 1.9
<code>org.apache.commons.lang.StringEscapeUtils</code> <ul style="list-style-type: none"><li>• <code>StringEscapeUtils.unescapeHtml(String input);</code></li><li>• <code>StringEscapeUtils.escapeHtml(String input);</code></li><li>• <code>StringEscapeUtils.escapeJavaScript(String input);</code></li></ul>	<code>org.apache.commons.text.StringEscapeUtils</code> <ul style="list-style-type: none"><li>• <code>StringEscapeUtils.unescapeHtml4(String input);</code></li><li>• <code>StringEscapeUtils.escapeHtml4(String input);</code></li><li>• <code>StringEscapeUtils.escapeEcmaScript(String input);</code></li></ul>
<code>org.apache.commons.lang.text.StrBuilder</code>	<code>org.apache.commons.text.TextStringBuilder</code>
<code>org.apache.commons.lang.text.StrSubstitutor</code>	<code>org.apache.commons.text.StringSubstitutor</code>
<code>org.apache.commons.lang.WordUtils</code>	<code>org.apache.commons.text.WordUtils</code>

See the official Apache Commons Lang and Commons Text documentation for guidance on how to update implementation code: [https://commons.apache.org/proper/commons-lang/article3\\_0.html](https://commons.apache.org/proper/commons-lang/article3_0.html)<https://commons.apache.org/proper/commons-text/index.html>

## Potential required code changes for Apache Commons Configuration

The following table shows some examples of the package name differences between Apache Commons Configuration 1.3 and 2.8.0. The main change to note is the change of the `org.apache.commons.configuration` package to `org.apache.commons.configuration2`.

**Table 4. Package name differences between Apache Commons Configuration 1.3 and 2.8.0**

Apache Commons Configuration 1.3	Apache Commons Configuration 2.8.0
org.apache.commons.configuration.ConfigurationException	org.apache.commons.configuration2.ex.ConfigurationException;
org.apache.commons.configuration.SubnodeConfiguration;	org.apache.commons.configuration2.SubnodeConfiguration;
org.apache.commons.configuration.XMLConfiguration;	org.apache.commons.configuration2.XMLConfiguration;
org.apache.commons.configuration.tree.xpath.XPathExpressionEngine;	org.apache.commons.configuration2.tree.xpath.XPathExpressionEngine;

The following code snippet for retrieving XML configuration data, and for retrieving nodes and attributes, shows an example of code changes that might be required when you upgrade. See the official Apache Commons Configuration documentation for guidance on how to update implementation code: <https://commons.apache.org/proper/commons-configuration/index.html>

### 1.3

```
import org.apache.commons.configuration.SubnodeConfiguration;

import org.apache.commons.configuration.XMLConfiguration;

import org.apache.commons.configuration.tree.ConfigurationNode;

import org.apache.commons.configuration.tree.xpath.XPathExpressionEngine;

...

// Get config file

XMLConfiguration xmlConfig =

    new XMLConfiguration(java.net.URL url_for_xml_config_file);

xmlConfig.setExpressionEngine(new XPathExpressionEngine());

...

// Get XML node attribute values

SubnodeConfiguration xmlNode = xmlConfig.configurationAt("XML_NODE");

List<?> xmlNodeAttribs xmlNodeAttribs = xmlNode.getRootNode().getAttributes("XML_ATTRIB");

if (xmlNodeAttribs.size() > 0) {
```



```
String xmlNodeAttrib = (String) ((ConfigurationNode) xmlNodeAttribs.get(0)).getValue();

....
}

Iterator<?> i = curamConfig.configurationsAt("POPUP_PAGES/POPUP_PAGE").iterator();
while (i.hasNext()) {
    ....

    (SubnodeConfiguration) i.next();

    ....
}
```

## 2.8.0

```
import org.apache.commons.configuration2.HierarchicalConfiguration;
import org.apache.commons.configuration2.SubnodeConfiguration;
import org.apache.commons.configuration2.XMLConfiguration;
import org.apache.commons.configuration2.builder.fluent.Configurations;
import org.apache.commons.configuration2.tree.ImmutableNode;
import org.apache.commons.configuration2.tree.xpath.XPathExpressionEngine;

...

// Get config file
Configurations configurations = new Configurations();

XMLConfiguration xmlConfig =

    configurations.xml(java.net.URL url_for_xml_config_file);
xmlConfig.setExpressionEngine(new XPathExpressionEngine());

...

// Get XML node attribute values
if (!xmlConfig.configurationsAt("XML_NODE").isEmpty()) {
    HierarchicalConfiguration<ImmutableNode> xmlNode = xmlConfig.configurationsAt("XML_NODE").get(0);
} else {
    ....
}

Map<String, Object> xmlNodeAttribs = xmlNode.getNodeModel().getInMemoryRepresentation().getAttributes();
```

```
if (xmlNodeAttribs.size() > 0) {  
    String xmlNodeAttrib = (String) xmlNodeAttribs.get("XML_ATTRIB");  
    ....  
}  
  
while (i.hasNext()) {  
    ....  
    (SubnodeConfiguration) i.next();  
    ....  
}
```

## Potential required code changes for the **ical4j** JAR file

The following example shows a code change that might be required because of the update of the **ical4j** JAR file from the 1.0-beta4 JAR file to 3.2.1. See the official **ical4j** documentation for guidance on how to update the implementation code: <https://github.com/ical4j/ical4j>

### 1.0-beta4

```
import net.fortuna.ical4j.model.Dur;  
import net.fortuna.ical4j.model.Period;  
  
Period p = new Period(net.fortuna.ical4j.model.DateTime startDate, net.fortuna.ical4j.model.DateTime endDate);  
Dur duration = p.getDuration();  
int days = duration.getDays();  
int hours = duration.getHours();  
int minutes = duration.getMinutes();  
int totalMinutes = minutes + hours * 60 + days * hours * 60;
```

### 3.2.1

```
import java.time.Duration;  
import java.time.temporal.TemporalAmount;  
import net.fortuna.ical4j.model.Period;  
  
Period p = new Period(net.fortuna.ical4j.model.DateTime startDate, net.fortuna.ical4j.model.DateTime endDate);  
TemporalAmount d = p.getDuration();  
Duration ds = (Duration) d;  
int totalMinutes = ds.toMinutes();
```

## Potential required code changes for the `json-schema-validator` JAR file

The following example shows a code change that might be required because of the update of the `json-schema-validator` JAR file from 0.1.5 to 1.0.72. See the official `json-schema-validator` documentation for guidance on how to update the implementation code: <https://github.com/networknt/json-schema-validator>

### 0.1.5

```
JsonSchemaFactory factory = new JsonSchemaFactory();
```

### 1.0.72

```
import com.networknt.schema.SpecVersion.VersionFlag;

...

JsonSchemaFactory factory = JsonSchemaFactory.getInstance(VersionFlag.V7);
```

# Notices

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents.

MERATIVE PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information is intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. Merative shall not be responsible for any loss whatsoever sustained by any organization or person who relies on this material.

Any references in this information to non-Merative websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this Merative product and use of those websites is at your own risk.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

Information concerning non-Merative products was obtained from the suppliers of those products, their published announcements or other publicly available sources. Merative has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-Merative products. Questions on the capabilities of non-Merative products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to Merative, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all

conditions. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

## Trademarks

Merative <sup>™</sup> and the Merative <sup>™</sup> logo are trademarks of Merative US L.P. in the United States and other countries. Other product and service names may be trademarks of Merative or other companies. IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Other names may be trademarks of their respective owners. Other company, product, and service names may be trademarks or service marks of others.