

Apache UIMA PEAR Packaging Ant Task Documentation

Authors: The Apache UIMA Development Community

Version 2.2.2

Incubation Notice and Disclaimer. Apache UIMA is an effort undergoing incubation at the Apache Software Foundation (ASF). Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

License and Disclaimer. The ASF licenses this documentation to you under the Apache License, Version 2.0 (the "License"); you may not use this documentation except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, this documentation and its contents are distributed under the License on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Trademarks. All terms mentioned in the text that are known to be trademarks or service marks have been appropriately capitalized. Use of such terms in this book should not be regarded as affecting the validity of the the trademark or service mark.

Table of Contents

Overview	v
1. Using The PEAR Packaging Ant Task	1

Overview

To make UIMA analysis components easily consumable for other users, UIMA defines the PEAR (Processing Engine ARchive) file format. This allows applications and tools to manage UIMA analysis components automatically for verification, deployment, invocation and testing. So far the PEAR file format is the desired output format for an UIMA analysis component. With the PEAR packaging Ant Task it is possible to create such a PEAR package for the analysis component automatically.

Chapter 1. Using The PEAR Packaging Ant Task

To use the PEAR packaging Ant task it is necessary to define an additional classpath setting for the ant build. The two necessary jars for this setting are the `uima-core.jar` that contains the PEAR packaging code and the `uima-pear-ant-task.jar` that contains the code for the PEAR packaging Ant task. The additional classpath setting can look like:

```
<path id="pearTask.classpath">
  <fileset dir="/home/apache-uima/lib/" includes="uima-pear-ant-task.jar"/>
  <fileset dir="/home/apache-uima/lib/" includes="uima-core.jar"/>
</path>
```

Update the classpath fileset with your local path settings.

Next we have to define a custom Ant task for the PEAR packaging. The definition looks like:

```
<taskdef
  name="packagePear"
  classname="org.apache.uima.pear.tools.PearPackagingAntTask"
  classpathref="pearTask.classpath"/>
```

The custom PEAR packaging Ant task is called `packagePear`. This task name is used later to package a PEAR file. The `classname` attribute contains the main class of the PEAR packaging Ant task component and should not be changed. The `classpathref` attribute refers the defined `pearTask.classpath` that is necessary to execute the PEAR packaging task.

Now we are ready to use the PEAR packaging Ant task within our build target. To call the PEAR packaging just integrate the custom defined task `<packagePear>` as shown in the fragment below:

```
<packagePear
  componentID="SampleAnnotator"
  mainComponentDesc="desc/mainComponentDesc.xml"
  classpath="$main_root/pearClasspahtEntry;$main_root/anotherPearClasspahtEntry"
  datapath="$main_root/resources"
  mainComponentDir="/home/user/workspace/SampeAnntotator"
  targetDir="/home/user/pearArchive">
  <envVar name="ENV_VAR_NO1" value="value1"/>
  <envVar name="ENV_VAR_NO2" value="value2"/>
</packagePear>
```

The `<packagePear>` task has the following mandatory properties. If a property is not necessary, provide it with an empty content.

-
- `componentID` - The main component ID of the UIMA component that should be packaged
 - `mainComponentDesc` - The main component descriptor that should be used to run the PEAR package content
 - `classpath` - The classpath settings that must be used to run the PEAR package content - use `$main_root` macros
 - `datapath` - The datapath settings that must be used to run the PEAR package content - use `$main_root` macros
 - `mainComponentDir` - The main component directory that contains the PEAR package content
 - `targetDir` - The target directory where the created PEAR package is written to. The filename for the created PEAR package is `<componentID>.pear`

Additionally it is possible to specify environment variable settings that are necessary to run the PEAR package. This is done with the `<envVar>` element as shown above. The `name` specifies the variable name and the `value` specifies the variable value.