The AntTask

Table of contents

The `org.apache.lenya.cms.task.AntTask` class can be used to invoke targets of an Ant project.

1. Task Parameters

The task parameters are:

- **`publication-id`**: The publication ID
- **`buildfile (optional)`**: The location of the build file relative to the publication directory. If this parameter is not provided, the file is loaded from the default location (see section File Locations).
- **`target (optional)`**: The build target. If this parameter is not provided, the default target is executed.
- **`properties.*`**: The project properties.
- **`ant.*`**: The command-line parameters for Ant **(not implemented yet!)**

2. Logging

Every time an `AntTask` is invoked, a log file is created unsing the `XmlLogger` (manual entry (http://ant.apache.org/manual/listeners.html) , JavaDoc (http://issues.apache.org/gump/javadoc/ant/build/javadocs/org/apache/tools/ant/XmlLogger.html)). For the location of the log files, see section File Locations. The log history can be viewed at the URI

```
http://.../<publication>/logs/tasks/index.html
```

3. Writing AntTask Buildfiles

Any Ant project file can be used as a buildfile for the `AntTask`. There is one implicit property that is always set when an `AntTask` is executed:

- **`pub.dir`**: The absolute path of publication directory.

The runtime properties of the target can be set using task parameters with the prefix `properties`, e. g. `properties.filename` for a buildfile property named `filename`.

4. Using custom Ant Tasks

The implementation of custom Ant tasks is described in the Ant User Manual (http://ant.apache.org/manual/index.html) . If you want to write a general Lenya task, put it into the package `org.lenya.cms.ant.` If you want to write a task that is only suited for your publication, put it in the `<publication>/java/src/` directory.

5. File Locations

Default buildfile location:

```
<publication>/config/tasks/targets.xml
```

Log files:

```
<publication>/logs/tasks/*.xml
```

Log file presentation stylesheets:

```
<webapp>/lenya/xslt/logs/*.xsl
```

6. Example

The following buildfile contains the target `publish` that can be invoked using the `AntTask`:

```
<project name="Example Project" default="publish" basedir=".">
```

```
   <!-- implicit properties (set by the AntTask) -->
   <property name="pub.dir" value=""/>

   <!-- publishing properties -->
   <property name="authoring.dir" value="content/authoring"/>
   <property name="live.dir" value="content/live"/>
   <property name="publish.sources" value=""/>

   <target name="publish">
     <echo>Publish: Copying files from ${authoring.dir} to ${live.dir}</echo>
     <copy todir="${pub.dir}/${live.dir}">
       <fileset dir="${pub.dir}/${authoring.dir}">
         <include name="${publish.sources}"/>
       </fileset>
     </copy>
   </target>

</project>
```

You define the task in your `tasks.xconf` file:

```
<task id="ant" class="org.lenya.cms.task.AntTask"/>
```

To invoke the task from your sitemap, you have to define an appropriate `TaskAction` instance:

```
<map:action name="publish"
    src="org.lenya.cms.cocoon.acting.TaskAction">
  <task id="ant"/>
</map:action>
```

You call the action in a pipeline:

```
<map:match pattern="publish.html">
  <map:act type="publish">
    ...
  </map:act>
</map:match>
```

And finally, go to your browser and call the URI with the appropriate parameter(s):

```
http://.../publish.html?properties.publish.sources=test.xml
```