Ant task

An *Ant* task is provided to compile the annotations.

Usage

First you need to activate the custom task in your *Ant* `build.xml` file with the following: (refer to *Ant* documentation on *"taskdef"* for more details)

```
<!-- we assume we defined a classpath with
the id="backport175.class.path" for
backport175 jars -->
<path id="backport175.class.path">
    ...
    <pathelement
path="pathToBackport175.jar"/>
    ...
</path>

<!-- define the custom task (annotationc can
be changed to what you prefer)
<taskdef name="annotationc"

classname="org.codehaus.backport175.compiler
.task.AnnotationCTask"
         classpathref="aw.class.path"/>
<!-- Note: the <taskdef> element can be
nested within a <target> element at your
convenience -->

<!-- invoke the annotationc defined task -->
<target name="samples:task:annotationc"
depends="init, compile:all">
    <annotationc
```

```
        verbose="true"

destdir="${basedir}/target/samples-classes"

properties="${basedir}/src/samples/annotatio
n.properties"
        copytodest="**/*.dtd">

        <src path="${basedir}/src/samples"/>
        <src path="${basedir}/src/test"/>
        <classpath
path="${basedir}/target/samples-classes"/>
        <classpath
path="${basedir}/target/test-classes"/>
        <classpath
path="${basedir}/target/classes"/>
        <fileset dir="other">
            <include name="**/BAZ.java"/>
```

```
        </fileset>
    </annotationc>
</target>
```

Reference

The `AnnotationCTask` task accepts the following:

- verbose: (optional) flag marking the task verbosity (true/false)
- ignoreUnkown: (optional) flag marking if the task should ignore unknown annotations (like f.e. @author if not associated to an annotation interface) (true / false)
- properties: (optional) path to a properties file when user-defined annoations are to be used
- destdir: (optional unless input classes are in more than one path) directory where to put annnotated class files
- copytodest: (optional) filename pattern to copy extra resources like DTD, XML, or properties files that were found in the input classes path(s). By defaults, only `.class` files will be handled. It is ignored if `destdir` is not set.

Use the following parameters to configure the classpath to point to the classes to be weaved. Those can be specified with nested elements as well/instead:

- properties path=..: extra path to a properties file where user-defined annotations are to be used
- classpath: classpath of classes to be annotated, as well as classpath to discover user-defined annotations if any
- classpathref: classpath reference of classes to be annotated, as well as classpath to discover user-defined annotations if any
- srcdir: directory where to find annotated java source files
- sourcepath: path where to find annotated java source files
- sourcepathref: path reference where to find annotated java source files

Nested elements are similar to the `javac` task when you configure a classpath and a sourcepath:

- classpath: Path-like structure of classes to annotated, as well as classpath to discover user-defined annotations if any
- src: single path entry of annotated java source files
- sourcepath: Path-like structure of annotated java source files
- fileset: fileset to contain annotated java source files