

# Beehive and Ant 1.7

## Overview

Out of the box, it's not possible to build a Beehive 1.0.x application with Ant 1.7. The problem is a circular dependency that is created in the Ant build files provided with the Controls or NetUI project templates. Fortunately, with a little bit of surgery to just an application's Ant file, it's possible to fix this problem and successfully use Ant 1.7 with existing Beehive versions. A message from Ant like:

```
/tmp/apache-beehive-1.0.2/samples/fooWeb/build.xml:84: The following error occurred while executing this line:
/tmp/beehive-1.0.2/ant/beehive-tools.xml:75: This data type contains a circular reference.
```

is a symptom of this problem. There are two ways to solve this that are covered below.

## Build Controls and Page Flows Together

This option process annotations and compiles Control and Page Flow source files at the same time and is sufficient for most applications. This requires, however, that there be no references from a Page Flow to any source files generated when processing annotations on Controls; if such references – for example to a generated [ControlBean](#) – are required, use the option below. To use this option with Ant 1.7, replace the `<build-controls>` and `<build-pageflows>` calls in an application's `/build.xml` file with the following:

```
<!-- Declare a classpath used to resolve the APT task -->
<path id="apt.task.classpath">
  <path refid="controls.dependency.path"/>
</path>

<!-- Declare a classpath that extends the webapp's usual classpath to add
     JARs needed in this classpath in order to process annotations -->
<path id="webapp.classpath.ext">
  <path refid="webapp.classpath"/>
  <path refid="velocity.dependency.path"/>
  <path refid="netui-compiler.dependency.path"/>
</path>

<!-- Declare the APT task -->
<taskdef name="apt"
  classname="org.apache.beehive.controls.runtime.generator.AptTask"
  classpathref="apt.task.classpath"
  onerror="fail"/>

<!-- Process annotations and compile source files -->
<echo>Building application source files</echo>
<apt sourcepath="${basedir}/src"
  srcdir="${basedir}/src"
  destdir="${build.dir}/WEB-INF/classes"
  gendir="${build.dir}/WEB-INF/${tmp.sourcegen.dir}"
  classpathref="webapp.classpath.ext"
  processorOptions="web.content.root=${web.dir}"/>
```

This replaces use of the `<build-controls>` and `<build-pageflows>` Ant macros with similar task and classpath declarations and `<apt>` call with tweaks to avoid creating the circular dependency that fails in Ant 1.7.

## Build Controls and Page Flows Separately

This option uses two passes to build an application by first processing Control annotations without compiling the source files; then, Page Flow annotations are processed and the application is compiled. This scheme allows Page Flows to have references to classes that are generated when processing Control annotations. To use this option with Ant 1.7, replace the `<build-controls>` and `<build-pageflows>` calls in the applications `/build.xml` file with the following:

```

<!-- Declare a classpath used to resolve the APT task -->
<path id="apt.task.classpath">
    <path refid="controls.dependency.path"/>
</path>

<!-- Declare classpaths needed to process annotations -->
<path id="webapp.classpath.controls">
    <path refid="webapp.classpath"/>
    <path refid="velocity.dependency.path"/>
</path>

<path id="webapp.classpath.netui">
    <path refid="webapp.classpath"/>
    <path refid="velocity.dependency.path"/>
    <path refid="netui-compiler.dependency.path"/>
</path>

<!-- Declare the APT task -->
<taskdef name="apt"
    classname="org.apache.beehive.controls.runtime.generator.AptTask"
    classpathref="apt.task.classpath"
    onerror="fail"/>

<!-- Compile the source files. -->
<echo>Processing Control annotations</echo>
<apt sourcepath="${basedir}/src"
    srcdir="${basedir}/src"
    destdir="${build.dir}/WEB-INF/classes"
    gendir="${build.dir}/WEB-INF/${tmp.sourcegen.dir}"
    classpathref="webapp.classpath.controls"
    nocompile="true"/>

<echo>Process Page Flow annotations and compile all source files</echo>
<apt sourcepath="${basedir}/src"
    srcdir="${basedir}/src"
    destdir="${build.dir}/WEB-INF/classes"
    gendir="${build.dir}/WEB-INF/${tmp.sourcegen.dir}"
    classpathref="webapp.classpath.netui"
    processorOptions="web.content.root=${web.dir}"/>

```

This Ant snippet is more complicated because it makes two `<apt>` calls with two different classpath references. This is required in order to avoid processing Page Flow annotations in the first `<apt>` call thus forcing it to simply process Control annotations and generate Control source files. Yes, this is a hack, but hey, it works. 😊