

Beehive Ant Macros

Table of contents

1 Overview.....	2
2 The build-controls Ant Macro.....	2
2.1 Sample.....	2
3 The build-pageflows Ant Macro.....	3
3.1 Samples.....	3
4 The build-schemas Ant Macro.....	4
4.1 Sample.....	4

1. Overview

The Beehive distribution includes several Ant macros to assist developers creating Ant build files for NetUI Page Flow, Controls and Web Services. These macros are located in the `<BeehiveHome>/ant/beehive-tools.xml` file.

To use any of these macros import the beehive-tools file into your Ant script:

```
<import file="${beehive.home}/ant/beehive-tools.xml"/>
```

2. The build-controls Ant Macro

This macro uses the Java 5 Annotation Processing Tool (apt) for control generation and compilation. Any .java files generated by this macro can be found in the directory specified by the `tempdir` parameter.

`build-controls` accepts the following parameters:

Parameter Name	Required	Description
<code>srcdir</code>	Yes	The directory containing the controls to build.
<code>destdir</code>	Yes	The destination directory for the compiled controls files.
<code>tempdir</code>	Yes	A temporary directory for any generated java files.
<code>classpathref</code>	Yes	A classpath reference for building the controls. Required.

2.1. Sample

The following project has a source directory, a destination directory, and a temporary directory for generated files.

```
project
  build
    classes
  src
  tempsrc
```

For this project, the `build-controls` call would look like this:

```
<build-controls srcdir="project/src"
```

```

destdir="project/build/classes"
tempdir="project/tempsrc"
classpathref="build.classpath"/>

```

3. The build-pageflows Ant Macro

This macro is intended for the compilation of the **Page Flow** portions of a web application. This macro will **not** compile controls inside of a web application. If the web application contains controls, they must be compiled first using the `build-controls` macro (see above). Once the compilation of the controls is complete, the page flows within the web app can be compiled.

`build-pageflows` accepts the following parameters:

Parameter Name	Required	Description
<code>srcdir</code>	Yes	The root directory which will be scanned for source files.
<code>classpathref</code>	Yes	The classpath reference for building page flows.
<code>sourcepathref</code>	No	A reference to a path that contains all the source roots. Defaults to a path that contains <code>\${srcdir}</code> and <code>\${srcdir}/WEB-INF/src</code> .
<code>webcontentdir</code>	No	The root location for web content (e.g., JSPs, <code>web.xml</code> , etc.). Defaults to <code>\${srcdir}</code> .
<code>destdir</code>	No	The directory for compiled classes and generated resources. Defaults to <code>\${srcdir}/WEB-INF/classes</code> .
<code>tempdir</code>	No	The directory for temporary .java files, copied from page flows (etc.) with non-.java extensions. Defaults to <code>\${srcdir}/WEB-INF/.tmpbeansrc</code> .

3.1. Samples

Consider a simple project with the following structure:

```
project
```

```

WEB-INF
  classes
  lib
  src
  web.xml

```

For this project, the build-pageflows call would look like this:

```
<build-pageflows srcdir="project" classpathref="webapp.build.classpath"/>
```

In a more complex project, web content, source, and the target (build) directory may be in different places:

```

project
  src
  web
    WEB-INF
      lib
      web.xml
build
  webapp

```

In this case, the build-pageflows call would be:

```
<build-pageflows srcdir="project/src"
  webcontentdir="project/web"
  destdir="project/build/webapp/WEB-INF/classes"
  classpathref="webapp.build.classpath"/>
```

4. The build-schemas Ant Macro

This macro can be used to parse an XML Schema or Apache XMLBeans xsdconfig file into Apache XMLBeans. It is really just a wrapper for the XMLBean schema compiler which is part of the Apache XMLBeans distribution.

build-schemas accepts the following parameters:

Parameter Name	Required	Description
srcdir	Yes	The directory containing XML Schemas or XMLBeans xsdconfig files to build.
destdir	Yes	The directory to use for files generated during an XSD build.

4.1. Sample

In this example, schemas are being built from a webapp's WEB-INF/schemas to WEB-INF/classes.

```
<build-schemas srcdir="WEB-INF/schemas" destdir="WEB-INF/classes"/>
```