Beehive Ant Macros

Table of contents

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 specifed by the `tempdir` parameter.

`build-controls` accepts the following parameters:

Parameter Name	Required	Description
srcdir	Yes	The directory containing the controls to build.
destdir	Yes	The destination directory for the compiled controls files.
tempdir	Yes	A temporary directory for any generated java files.
classpathref	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). 0nce the compilation of the controls is complete, the page flows within the web app can be compiled.

`build-pageflows` accepts the following pararmeters:

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

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"/>
```

5. The build-webservices Ant Macro

This macro is intended for the compilation of the **web service** 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). 0nce the compilation of the controls is complete, the web servcies within the web app can be compiled.

`build-webservices` accepts the following pararmeters:

Parameter Name	Required	Description
srcdir	Yes	The directory containing web services to build.
destdir	Yes	The destination directory for the compiled web service files.
tempdir	Yes	A temporary directory for any generated java files.
classpathref	Yes	A classpath reference for building the web service files. Required.

5.1. 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-webservices` call would look like this:

```
<build-webservices srcdir="project/src"
                destdir="project/build/classes"
                tempdir="project/tempsrc"
                classpathref="build.classpath"/>
```