

Memo

Subject: Building Whydah

From: Greg Sandell

To: Tim O'Brien

Date: Mar 15, 2005

The following is an email that was sent to **ies-dev** on 2004-10-22 which describes the use of moped and kestral's ant (`build.xml`) scripts.

[Begin email]

I've made some revisions in Whydah and Nightjar that makes building now a no-muss-no-fuss affair. Basically, the ant scripts take care of doing the right build for development, preview or production machines.

Whydah is of course really two source trees, **moped** (JSP, Java and XML files) and **kestral** (gif, jpg, flash & pdf files).

Kestral has nothing fancy about it. Just say

ant buildZip

Moped is a more ornate affair because there are various possibilities. You can say:

ant development build

ant preview build

ant production build

...each of these runs two scripts. The first does the configuration (manages some properties files) and the second does the build based on the config. You could also say **ant development** followed by **ant build** but that does the same thing.

When you are developing code in moped, there are several other helpful ant targets:

ant bounceJava	Recompile Java code and copy it to webapps
ant bouncePages	Refresh all the JSPs in webapps
ant bounceXml	Refresh all the xml and xsl in webapps
ant dist	Rebuild the war and put it in webapps
ant cleanDist	Wipe app clean, rebuild war and put in webapps

What you have to do:

Of course, you have to checkout/update moped.

In moped's root directory a file called `domain_user_name.build.properties` (e.g. `smeghji.build.properties`) to customize the build for your machine, and add/commit it to moped. If you are developing on a PC, all this file needs to manage is the choice of location on your computer where files are built (a temporary scratch space). Copy `gsandell.build.properties` and customize it as you wish. Under linux machine using jPackage, you have to use this file to customize other things; see

`publish.build.properties` for example. (*publish* is the name of a user on sheng, alex and beaker.)

Special Details for Nightjar

There are none. It follows all the same conventions as moped.

Underneath the Hood:

With Tim's help I learned to use Jakarta commons' utilities for properties files. In moped (in `/WEB-INF/classes`) there are the following files:

<code>whydah.preview.properties</code>	Sets production database, caching turned on
<code>whydah.development.properties</code>	Sets localhost database, caching turned off
<code>whydah.production.properties</code>	Sets production database, caching turned on
<code>whydah globals.properties</code>	Database host, port, usernames, password, etc.
<code>whydah.properties.xml</code>	The "boss" that configures which properties files will be used.

For example, when you do an **ant preview build**, the ant script overwrites `whydah.properties.xml` with instructions to use `whydah globals.properties` as the main properties, and layers over `whydah.preview.properties` with further refinements.

If your platform was really different, you could create a personal properties files, such as `dsummer globals.properties` that would address your needs. But then you'd also need to add a `dsummers` target to `build.xml`.

If you want to dig into greater detail, look at the class `org.iesabroad.strutsLocal.WhydahConfigSingleton`.

- Greg

[end email]