

WOProject-Ant

WOProject Ant Tasks is a suite of extensions for ant to build WOApplications, WOFrameworks and native launchers for the standalone Java application.

Introduction

Ant build files are broken into what are called 'targets', similar to a [Makefile](#). These targets are a group of, what are called, 'tasks'. Tasks are used for getting work done; they make something happen directly, e.g. [move files](#) on the filesystem, [compile java](#) source files into classes, [transfer files](#) from one server to another over ssh, etc.

While Ant is distributed with around fifty built-in tasks, any number of customized tasks can easily be added by developers outside of the core Ant team.

WOProject is such a project. WOProject adds a half dozen tasks to Ant. These tasks allow WebObject developers the freedom to use Ant in their build process, instead of the using Apple's XCode suite. A few of these tasks are: [WOPcompile](#), [WOApplication](#), [WOFramework](#). By using the -lib command line option to refer to woproject.jar, these tasks can be used directly in the typical Ant `build.xml` file in building your WebObjects projects outside of the Eclipse/WOLips environment. It is not recommended to drop woproject.jar into your ant lib directory since woproject is under continuous development and different versions can trigger problems such as errors when build WOrder from source.

See the link above to the old site for in-depth documentation on each of these new tasks.

Reasons for using Ant to build WebObjects frameworks and applications:

- Ant is highly configurable and flexible, with constant additions from the open source community, and is considered a defacto standard by the Java community
- Builds of projects with a very large number of files without dependency failures
- Allows easy automation of daily builds for teams of developers, i.e. nightly builds for smoke testing, etc.

Download and Install

You can download the ant tasks ~~from the sourceforge page~~ from the automated [Hudson build](#) page for WOProject. If you have installed the wolips material for using WebObjects with Eclipse then you can also find the ant tasks there. The path for this will look something like `../eclipse/plugins/org.objectstyle.wolips.woproject.ant_2.0.0.3458/lib/woproject.jar`. To use the ant plugins outside of Eclipse, either by extracting the jar file from the Eclipse distribution or by downloading it separately, you will need to install them into your computers' ant lib directory. On MacOS-X this would be `/Developer/Java/Ant/lib/woproject.jar` (on Leopard w/ XCode 3.0, this would be `/usr/share/ant/lib/woproject.jar`). You can also put this jar in `USER_HOME/.ant/lib/woproject.jar`. A quick way to see where all of these lib directories are and what is loaded is to type `'ant -diagnostics'`.

Ant Tasks

(for tasks not documented here see [the old site](#)).

- Java Launchers
 - [JApplication](#): A task for assembling native launchers for any Java applications for a number of OS platforms.
- Core WebObjects Tasks
 - [WOPcompile](#): A Javac task extension that allows including WOFrameworks in CLASSPATH.
 - [WOApplication](#): A task for assembling WebObjects applications.
 - [WOFramework](#): A task for assembling WebObjects frameworks.
 - [WOGen](#): A task for generating custom Java EO classes from EOModel.
 - [EOModel](#): A task for configuring custom connection dictionary for EOModels.
- WebObjects Tools Integration Tasks:
 - [PBIndex](#): Maintains PB.project file for integration with WOBuilder on Windows.
 - [PBXIndex](#): Maintains PBX.pbproj file for integration with WOBuilder and EOModeler on older Mac OS X.
 - [XcodeIndex](#): Maintains XCode project files for integration with WOBuilder on the latest Mac OS X.
- [Embedding WOFrameworks](#)
- [Servlet Builds](#)

FAQ

I am making an EO adaptor framework and so need to manipulate the `Info.plist`

Edit the `build.properties` file and modify the `customInfoPListContent` which will be appended near the end of the `Info.plist` file that is constructed when the framework is build.