

Clearspace Source Build

The Clearspace source code is available for licensees of the software. **Clearspace is not Open Source software.** Use of the source code is governed by the license you agreed to when you purchased the software. If you do not have a commercial or non-commercial license to the product, the Evaluation License applies. **In no event can the source code be embedded into a separate application without an OEM license from Jive Software.** Some of the source code in this package is Open Source software developed by third parties. Wherever this is the case, it is specifically noted in the source, and that source is governed by the license given to it by its author.

The Clearspace source build can:

- Help you manage modifications to Clearspace while making it easy to upgrade to future versions
- Help you to develop your own custom extensions or modifications
- Help you integrate Clearspace in your environment, especially for user management or authentication

For additional developer resources, please visit: <http://www.jivesoftware.com/developer/>. The Jive Clearspace build process is based on [Ant](#). There is no need to download and install Ant - a version of it is included in this distribution.

This document includes the following sections:

- [Files in the Source Distribution](#) — What you get in the source distribution.
- [Setting Up](#) — Getting your environment ready to build.
- [Additional Build Targets](#) — Ant targets you might find useful.

Files in the Source Distribution

You'll find the following in your source distribution (sub-directories omitted for brevity):

```
jive_clearspace_source_x_x_x/  
| - build/  
| - custom/  
| - database/  
| - jiveHome/  
| - lib/  
| - plugins/  
| - resources/  
| - README.html  
| - source  
| - target  
| - webapp
```

Setting Up

Getting your machine ready for Jive Clearspace development requires a few steps. Wherever possible, instructions are provided for both Unix/Linux and Windows users.

Minimal Requirements

- An application server that supports the Servlet 2.3 and JSP 1.2 specs.
- A Database with a JDBC driver (an embedded database is available for evaluation purposes).

Pre-Compile Tasks

Set \$JAVA_HOME

UNIX/Linux

1. Edit the ".profile" file in your home directory (or the corresponding file for your shell).
2. Set the JAVA_HOME environment variable by adding the following line:
 - export JAVA_HOME=/path/to/java/directory
 - Be sure that there are no spaces after the end of the directory name.
 - Do not add an extra slash after the directory name.
3. Save changes to the file and then "source" it: source .profile

WindowsNT/2000/XP

1. Navigate to your desktop, right click on **My Computer**, then choose **Properties**.
2. Click the **Advanced** tab and click **Environment Variables**.
3. Under **System variables**, click **New**.
4. Enter the variable name "JAVA_HOME" and set the variable value to the full path of your Java installation. For example, "c:\jdk1.5". Be sure to not add an extra slash to the end of the directory name.
5. Click **OK** in all dialogs to accept the changes.
6. Close any open command prompt windows. The next time you open a command prompt, the "JAVA_HOME" variable will be set correctly.

Set \$ANT_OPTS

- If you have not done so already, you will need to allocate more memory to Ant in order to compile. To do so, follow the same instructions above for setting JAVA_HOME environment variable: ANT_OPTS=-Xmx256m (don't forget to source your .profile in on Linux/UNIX).

Testing Your Build Scripts

1. In a command prompt, change directory to JIVE_ROOT/build.
2. Issue the build command to compile the build:
 - **In UNIX/Linux**
./ant.sh
 - **In Windows**
ant
3. If everything compiles and completes with BUILD SUCCESSFUL you're ready to build your deployable WAR

Additional Build Targets

The list of ant targets is below. All build commands should be run from the "build" directory of your Clearspace distribution. You might also want to edit the build/build.xml file, where you can specify a number of build properties such as the directory that JAR files should be copied to after building. For a list of the commands and a brief description from the command line, type:

```
ant -projecthelp
```

To execute a build target, type `ant options <targetname>` where "targetname" is one of the targets listed below:

- [all](#)
- [clean](#)
- [compile](#)
- [ear](#)
- [init](#)
- [jar \(default\)](#)
- [jar-custom](#)
- [libraries.prepare](#)
- [runner](#)
- [validate](#)
- [war](#)

Each task is documented with a syntax guide and description. Optional parameters for each task are also listed. If you would like to permanently set the value of a property, add it to build/build.xml file.

all

Equivalent of calling `ant jar`.

Syntax

```
ant all
```

clean

Cleans your Jive Clearspace distribution directory by deleting compiled class files, the `clearspace-version.jar`, `clearspace.war` files. If you specified alternate destination directories for any of those files, you'll need to delete them manually.

Syntax

```
ant clean
```

compile

Compiles all the Clearspace source code into <clearspace-distribution>/classes/

Syntax

```
ant compile
```

ear

Creates a deployable EAR from the clearspace.war file.

Syntax

```
ant ear
```

init

Initializes the build and creates the temp directory.

Syntax

```
ant init
```

jar (default)

Creates a JAR of Clearspace classes called clearspace-x.x.x.jar, where x.x.x is the distribution version number.

Syntax

```
ant jar
```

jar-custom

Creates a JAR of custom code from the overlay/custom/ directory. This is a useful place to put your own integration code. By default the name of the JAR will be "custom.jar". To override this, pass in a "jarname" parameter.

Syntax

```
ant [-Djarname=<jar_name>] jar-custom
```

Optional Parameters

-Djarname=<jar_name>
The name of the JAR file to create.

Example

Creates foo.jar:

```
ant \-Djarname=foo jar-custom
```

libraries.prepare

Converts a standard build of a library to one that can be bundled.

Syntax

```
ant libraries.prepare
```

runner

Builds a preconfigured Tomcat based Clearspace server in target folder, launch with 'target/start-clearspace'.

Syntax

```
ant runner
```

war

Bundles the clearspace-1.0.0.jar, dependency jars, skins (FTL files) and config files into a single WAR for deployment. The WAR (clearspace.war, by default) is copied into the base directory of this distribution. You can specify a different name using the warname parameter.

Syntax

```
ant [-Dwarname=<war_name>] war
```

Optional Parameters

-Dwarname=<war_name>
The name of the WAR file to create.

Example

Creates foo.war:

```
ant.sh \-Dwarnname=foo war
```

validate

Validates a given XML file. The filename must be specified via the file parameter.

Syntax

```
ant -Dfile=file.xml validate
```