

ClassLibrary

Harmony class library code

Componentization

We have split the JSE class library into a number of 'modules'. You can see them laid out in [the classlib area of the Subversion repository](#). Modules contain discrete sets of functionality and have well-defined dependencies between one another. The module interface is defined by an [OSGi](#) manifest (here is [an example](#) taken from the LUNI module). A separate page details the package-level [componentization](#) of the class library code.

Development status

Each class library module has a status page describing the [Harmony development mailing list](#). Development discussion is conducted on the [\[http://incubator.apache.org/harmony/mailling.html\]](http://incubator.apache.org/harmony/mailling.html).

Obtaining the code

The Harmony website describes [how to obtain and run the class library code](#).

What can it do?

Here are some applications that have been tried with the class library code. Please download the code, try running some applications, and report your findings here. It will be helpful if you record the code revision you are using, any special steps to run the application, and the results of your experiments!

- [Apache_Ant](#) build tool (tbd)
- [Azureus](#) Java [BitTorrent](#) Client
- [Eclipse JDT batch compiler](#) for Java (tbd)
- [Apache_Velocity](#) template tool (looks good)
- [Apache_Derby](#) database (partially working)
- [Apache_Tomcat](#) servlet container (works)
- [Apache_Axis](#) web services platform (partially working)
- [Apache_Maven_Continuum](#) continuous integration server (fails)
- [Apache_Geronimo](#) J2EE server (fails)
- [Apache_Commons](#) reusable Java components
- [Apache_Struts](#) servlet framework (fails)
- [Apache_Cocoon](#) web development framework (fails)
- [Apache_MINA](#) network application framework (looks good)
- [ArgoUML](#) UML modelling tool (fails)
- [Apache_Log4j](#) Apache logging tool for Java (looks good)

[Coverage_information](#)

[Excluded_tests](#)