

# asmtools

## AsmTools

The AsmTools open source project is used to develop tools for the production of proper and improper Java '.class' files. AsmTools are being opened in order to facilitate a community of Java .class file production for various testing and other OpenJDK development applications.

AsmTools consist of a set of (Java class file) assembler/disassemblers:

***Jasm/Jdis*** - an assembler language that provides a Java-like declaration of member signatures, while providing Java VM specification compliant mnemonics for byte-code instructions. Jasm also provides high-level syntax for constructs often found within classfile attributes. Jasm encoded tests are useful for sequencing byte codes in a way that Javac compiled code might not normally sequence byte-codes.

***JCod/JDec*** - an assembler language that provides byte-code containers of class-file constructs. JCod encoded tests are useful for testing the well-formedness of class-files, as well as creating collections within a class-file construct that might be size-bounded by a normal Java compiler. JCod can also be used to 'fuzz' class files in a methodical way that respects class-file constructs.

AsmTools are completely reflexive - Java binary (.class) files may be disassembled into textual representations, which in turn can be assembled back to the same binary file.

AsmTools are developed to support the latest class file formats, in lock-step with JDK development.

Other open source Java assembler tools and binary classfile frameworks exist. They can be used for the purpose of synthesizing classfiles, however:

- they typically are designed to enforce the limits imposed by the VM specification of the class file format. They are not designed to produce classes that violate those limits.
- other assembler tools may not necessarily follow strict Java mnemonics as defined in the Java VM spec.
- other assembler tools may not stay in lock-step with the current generation of the JDK and VM specifications.
- class file libraries are harder to use for simple manipulations of any given class file. Typically, one has to create a program in that framework to parse and modify a class for a specific change to a given class.

The AsmTools open source project is part of the [Code Tools Project](#). It exists to promote a community that will improve it, further its development, and use it to develop test suites. We encourage you to browse, download, contribute, and get involved.

## News and Announcements

**The AsmTools project opened in the CodeTools Project of OpenJDK (Dec 11, 2014)**

## Related Projects and Links

This section contains links to technologies and blogs that are related to the AsmTools project.

### Development Tools

<a href="#">JCov</a>	A code-coverage tool, useful for describing test coverage of an API
<a href="#">SigTest</a>	A static analyzer used to describe platform compatibility and API Changes

### JT Harness Testing Harness and Extensions

<a href="#">JT Harness</a>	The JT harness is a flexible test harness that can be used to control and execute test suites.
----------------------------	--



- [Contribute](#)
- [Browse](#)

### Quick Links

#### Downloads

(Provided by [Adopt OpenJDK](#))

- [Tip](#) (latest <dev>)
- [Released](#)

#### Repository

<http://hg.openjdk.java.net/code-tools/asmtools/>

#### Usage and development

- [User's Guide](#)
- [FAQ](#)
- [Open issues](#)
- [Mailing List: asmtools-dev](#)
  
- [Build instructions](#)
- [Repository and code structure](#)
- [Source Layout](#)
- [Developer's guide](#)

#### Related projects

- [JT Harness](#)
- [Jtreg](#)
- [SigTest](#)
- [JCov](#)

#### Blogs

- [Java Compatibility Tools](#)

<a href="#">ME Framework</a>	The ME Framework is an open source set of JT harness plugins that supports the Java ME platform. Test suite architects use the JT harness and the ME Framework to construct test suites for Java ME technologies.
<a href="#">jreg</a>	The jreg test harness is used by the JDK test framework. This framework is intended primarily for regression tests. It can also be used for unit tests, functional tests, and even simple product tests.

## Blogs

<a href="#">Java Compatibility Tools Blog</a>	This blog provides topics of general interest for people writing tools to test Java platform compatibility and conformance.
---	---