

FAQ (AsmTools)

This page contains recommendations on solving various problems related to AsmTools

- [AsmTools OS Community](#)
 - [What is AsmTools?](#)
 - [What are the licensing terms for OS AsmTools?](#)
 - [What is the classpath exception?](#)
 - [Why do you need the classpath exception?](#)
 - [What is the relationship between the AsmTools project and the OpenJDK - CodeTools community?](#)
 - [How can I submit or suggest changes to the AsmTools project?](#)
 - [What can we expect in next AsmTools version?](#)
- [AsmTools product](#)
 - [What documentation is available for developers?](#)
 - [Why are AsmTools the desirable choice for OpenJDK developers?](#)
 - [What is the difference between Jasm assembly and JCod assembly?](#)
 - [How does one work with AsmTools?](#)
 - [What are the unique advantages of AsmTools?](#)

AsmTools OS Community

What is AsmTools?

AsmTools is a set of Java **Assembler Tools** which provides a means to assemble textual assembly files into Java .class (object) files, and to disassemble .class files back into assembly files. This can be used to test the VM for proper acceptance/rejection of Java objects, but also for any tool that works with Java objects.

What are the licensing terms for OS AsmTools?

The Jcov project is licensed under the GNU Public License, version 2, with the Classpath Exception. See the [license page](#) for more information.

What is the classpath exception?

The classpath exception was developed by the Free Software Foundation's GNU/Classpath Project (see <http://www.gnu.org/software/classpath/license.html>). It allows you to link an application available under any license to a library that is part of software licensed under GPL v2, without that application being subject to the GPL's requirement to be itself offered to the public under the GPL.

Why do you need the classpath exception?

If Jcov was distributed under GPL v2, that application could be subject to the requirements of the GPL that all code that is shipped as part of a "work based on the [GPL] program" also be GPL licensed. Accordingly, a GPL license exception is needed that specifically excludes from this licensing requirement any application that links to the GPL implementation. The classpath exception accomplishes this.

What is the relationship between the AsmTools project and the OpenJDK - CodeTools community?

The OpenJDK - CodeTools project is the Java open source home for technologies used to test a Java Platform.

Similar to other CodeTools projects, the AsmTools project started out as a commercial test technology, and has been released to the Java open source community. The AsmTools project also shares the same governance model and licensing requirements as the OpenJDK CodeTools Project.

How can I submit or suggest changes to the AsmTools project?

We encourage you to contribute source code to the AsmTools project. In accordance with the OpenJDK project governance process, you can submit code to the project leader either as a user, a developer, or as a committer. If your code is accepted to be integrated into the source tree, you will be asked to sign a Contributor's Agreement similar to what Apache Software Foundation requires. For more information about the governance process and the various roles for members in the community, please review [OpenJDK Governance](#).

What can we expect in next AsmTools version?

The next AsmTools milestone will be connected with JDK9. Specifically, this means that AsmTools will support changes in JDK infrastructure - as proposed for project Jigsaw (<http://openjdk.java.net/projects/jigsaw/>). Other future AsmTools directions include: more specific high-level syntax for class-file attributes.

AsmTools product

What documentation is available for developers?

There is a [user guide](#) which describes the syntax of both the Jasm and JCod assembler languages.

Why are AsmTools the desirable choice for OpenJDK developers?

AsmTools is interesting because it is developed in lock-step with the development of the Java platform (JDK). AsmTools always maintains the version of Java which is currently under development. Also, AsmTools is well integrated with Oracle's current test infrastructure, such as JavaTest and the JTReg test harnesses.

What is the difference between Jasm assembly and JCod assembly?

The JAsm assembler language tries to do two things: Keep a member declaration syntax that is relatively similar to Java member signatures, and represent all byte-code instructions as mnemonics. This is useful for concentrating on issues such as sequencing byte-code instructions in a way that is different than how a compiler sequences instructions.

The JCod assembler language focuses on the structure of a classfile. It typically sees each classfile construct as a structure or collection of bytes. Byte-code instructions are seen as collections of hexadecimal bytes. JCod provides a syntax where bytes can be inserted around constructs in ways that are un-natural to a compiler. Attributes for classes/members are visible as structs in JCod, where they are typically not directly visible in Jasm.

Which assembler you should use is mostly dependent on the type of class issue you wish to synthesize. Typically, if you wish to construct structurally malformed classes, you might choose JCod. If you wish to concentrate on instruction formatting or sequencing issues, you might choose JAsm.

The syntax for each of these assemblers is described in detail in the appendices of the [User Guide](#).

How does one work with AsmTools?

The best way to synthesize classes with AsmTools is to first start by creating a template class in Java. This should be a compilable class that is similar to the class you wish to create.

After creating the template, and compiling into a Java .class file, you can disassemble the object (.class) into an assembly file - using either the JDis (for the Jasm assembly language) or JDec (for the JCod assembly language) disassemblers. This will create a textual assembly, which is conventionally stored in a .jasm or .jcod file (respectively).

These assembly files can be edited in any text-editor. Typically, engineers alter specific areas within this assembly to create a place where a class can become malformed.

What are the unique advantages of AsmTools?

- TBD