Main

Da Vinci Machine Project Wiki

The Da Vinci Machine Project is a set of modifications of the JVM to support implementations of non-Java languages at a quality level (performance, compactness) near that of Java. The project is a part of the OpenJDK. Developers can write notes here.

In order to be able to edit pages in this wiki, please read the blog post explaining how to become an editor.

Intentions and non-intentions

This wiki contains:

- Instructions and advice on downloading and building this project’s code.
- Notes about the ongoing design and development of the various subprojects.
- A place to document likely future work.
- Quick links to similar information in other places on the web.
  - [http://openjdk.java.net/projects/mlvm/](http://openjdk.java.net/projects/mlvm/) (Da Vinci Machine Project home)
  - [http://wikis.oracle.com/display/HotSpotInternals/Home](http://wikis.oracle.com/display/HotSpotInternals/Home) (HotSpot internals)

Look for non-technical content on the static project pages: [http://openjdk.java.net/projects/mlvm](http://openjdk.java.net/projects/mlvm)

Relation to JDK 7

As of 10/2010, JDK 7 includes method handles and invokedynamic, but not language support for invokedynamic.

- Released builds of this technology can be found here.
- The source repositories are with the rest of JDK7.
  For example some of the JDK code is here and some of the JVM code is here.
- An advance copy of the javadoc (derived from the mlvm patch queue) is posted here.
- Here’s an example of how to run a warm-and-breathing test for method handles in JDK 7.

Contents

- Building
  - BootClassPath
  - Building on Windows
- JDK7TestExample
- MetaobjectProtocol
  - DynamicJava
  - Nomenclature
- PatchRepository
- Presentations — Condensed notes on the Da Vinci Machine Project.
  - PyCon 2009 VM Summit
- SubProjects
  - AnonymousClasses
  - CoroutineJSR
  - FastInterfaces
  - HotSwap
  - ImmediateWrappers
  - InterfaceInjection
  - InvokeDynamic
    - JavaLanguageSupport
      - InterfaceDynamic
      - ProjectCoinProposal
      - TemplatedConstructorExpressions
  - MethodHandles
  - NewArrays
  - StackContinuations
    - Coroutines
    - On-demand implementation
  - TailCalls
    - TailCalls GLS1998
  - TupleSignatures

Interesting tasks to do

- Document the process of pulling and building on various platforms.
  - Building on Windows
- List relevant RFES and subprojects. Link or enumerate architectural issues.
- Extract links and info. from messages and blog postings, especially via hotspot-dev@openjdk.

Who writes this wiki?
We are using the Community Write pattern.

There is no public write access, except for leaving page comments. You must be registered and logged in to leave comments.

Please do add page comments when a page should answer your question but doesn't. (That means, potentially on any page.)

The most effective page comments will include links to relevant email discussion threads, e.g., on the OpenJDK mlvm-dev list.

Any member of the OpenJDK Hotspot Group or a related group should feel free to contribute.

Commiters, contributors, and users of the Da Vinci Machine Project code base are welcome to contribute.

Write permissions are not automatic until and unless the DBs (Sun wiki vs. OpenJDK) get on speaking terms. If you think you should have write permission and don't ask jrose