

Main

- [Valhalla](#)
 - [Resources](#)
 - [Additional Resources](#)
 - [How you can help!](#)
 - [Pre-requisites](#)
 - [Levels of expertise / suggested participation:](#)
 - [Early Prototypes](#)
 - [Source Code and Building Valhalla](#)
 - [Sub-project Branches](#)
 - [Recent space activity](#)
 - [Space contributors](#)

Valhalla

Welcome to the Valhalla Project!

The [Goals](#) of this [Project](#) is to provide a venue to explore and incubate advanced Java VM and Language feature candidates such as:

- "Inline types" (originally [JEP 169 "Value Objects"](#))
- [Generic Specialization](#)
- [Enhanced volatiles](#)
- And possibly other related topics, such as reified generics.

This Project is sponsored by the [HotSpot Group](#).

Resources

- [Members](#)
- [Goals](#)
- [Valhalla Project](#)
- [Repository](#)
- Mailing lists
 - [valhalla-dev](#) is the primary list used for all technical discussion. ([archives](#))
 - [valhalla-spec-comments](#) is for sending comments, suggestions, and other feedback directly to the EG. Only EG members may subscribe to this list, but anyone may post, although posts are moderated in order to prevent spam. The EG will read all messages sent to this list. ([archives](#))
 - [valhalla-spec-experts](#) is the moderated EG mailing list for discussion of specification only. Implementation discussions should take place on [valhalla-dev](#). ([archives](#))
 - [valhalla-spec-observers](#) is for those who wish to monitor, and perhaps discuss the EG's progress. All messages sent to the [valhalla-spec-experts](#) list are automatically forwarded to this list. Anyone may subscribe to this list, and any subscriber may post. EG members are under no obligation to follow the traffic on this list. ([archives](#))

Additional Resources

The following provide a sketch of **possible** features and implementation approaches.

- [JVM Languages Summit 2018](#)
- [JVM Languages Summit 2017](#)
- [JVM Languages Summit 2016](#)
- [JVM Languages Summit 2015](#) (August 2015)
- [State of the Specialization](#) (July 2014)
- [JVM Language Summit: Evolving the JVM](#) (video, July 2014)
- [ClassDynamic](#) (June 2014)
- [State of the Values](#) (April 2014)

The following is a list of community presentations

- [Primitives in Generics - Presentation from Bulgarian JUG](#) (29 Jan 2015)
- [Results of Bulgarian JUG Hackathon](#) (29 Jan 2015)
- [Bulgarian JUG Hackathon background: parts 1, 2 and 3](#) (07 March 2015)

Working Documents:

- [Brian Goetz](#) (Java Language Architect)
- [John Rose](#) (VM Architect)

How you can help!

If you're new to OpenJDK then it is strongly advised that you join the [Adoption Group](#) and learn about OpenJDK before diving into advanced Valhalla topics.

Pre-requisites

Most theoretical ideas have been well explored by the core [Members!](#) The greatest help can therefore be given by those who:

- Have read the [Additional Resources](#)
- Gone through the [valhalla-dev Archives](#)
- Provide concrete prototypes of their ideas and/or
- Try out concrete prototypes and then give input into discussions of prototype semantics.

Levels of expertise / suggested participation:

1. *Full Time JVM internals expert / implementer*
Please join the mailing list and produce concrete prototypes, try out the nightly builds and give feedback on existing prototypes.
2. *Java/JVM Language, Framework, Library authors/experts*
Please join the mailing list, try out the nightly builds and give feedback on existing prototypes.
3. *Day to Day Java Developers (or developers using other JVM languages)*
Please join the mailing list and join the [Adoption Group](#) to participate in hack days on stable prototypes.

Early Prototypes

- [Minimal Value Types](#)
 - ~~Early not stable prototype for Value Types with a subset of functionality for expert experimentation~~
- [L-World](#)
 - Moving inline types into the existing type system

Source Code and Building Valhalla

Valhalla is built the same as most OpenJDK projects, although Valhalla uses branches for various sub-projects, e.g. "lworld", "lw1", "nestmates" etc. At the command line execute:

```
hg clone http://hg.openjdk.java.net/valhalla/valhalla valhalla
cd valhalla
# Update to the sub-project of interest, e.g. "lworld" (use "hg branches" to list)
hg update -r lworld
bash configure
make
```

Once you've done this, when you want to update to the latest version: `hg pull && hg update -r <branch>`

There is a general OpenJDK [build guide](#) if you need help.

Sub-project Branches

Valhalla contains a number of sub-projects, and as such the source repository separates these via named branches:

- default: periodically updated copy of "jdk/jdk" (i.e. JDK mainline, <http://hg.openjdk.java.net/jdk>, contains no project changes)
- exp: ad-hoc experimental branch for various collaboration efforts (not currently used)
- lworld: [LWorld](#) series of prototypes, receives updates (merges) from "default" (Current prototyping work)
 - lw1: LW1 minimal LWorld prototype, build-0 at tag lw1_0 (Dead release branch, branched from "lworld")
- mvt: [Minimal Value Types](#) prototype, (Dead release branch)
- nestmates: Nest-Based access control work (JEP 181)

Recent space activity



[Mandy Chung](#)

LW2 updated Aug 16, 2019 • [view change](#)



[David Simms](#)

LW2 updated Jul 12, 2019 • [view change](#)

[Karen Kinnear](#)

LW2 updated Jul 01, 2019 • [view change](#)

Space contributors

- [Mandy Chung](#) (709 days ago)
- [David Simms](#) (744 days ago)
- [Karen Kinnear](#) (755 days ago)
- [Frederic Parain](#) (1083 days ago)
- [Paul Sandoz](#) (1479 days ago)
- ...



David Simms



L-World updated Jul 01, 2019 • [view change](#)

LW2 Command Line Options updated Jun 27, 2019 • [view change](#)