Developers Guide

This page explains how to create a patch, modify the core JOSM application or existing plugins.

At least basic Java programming skills are necessary. See WikiStart how you could improve JOSM without Java skills.

Source code and JOSM builds

- Installing and Running JOSM
- Source code#Getthesource
- Source code#CompilingusingAnt
- Compiling JOSM using IDE
  - How to create a JOSM build

Developer guidelines

- Development Guidelines (Coding style and Internationalization)
- Default Presets (about new osm-tags and icons in JOSM)
- Patch Guidelines (How to submit a patch)
- Track the state of development (Release Schedule)
- Releasing
- Releases

Eclipse

- Various Eclipse tips
- Video: how to checkout JOSM into Eclipse
- Video: how to checkout a JOSM plugin into Eclipse

YourKit

JOSM core developers are using the YourKit Java Profiler, as we found other tools lacking required functionality for profiling. YourKit supports open source projects by granting developers free licenses.

To enable the YourKit Java agent, add the following JVM parameter:

```
-agentpath:<yourkit_path>/yjpagent.dll=delay=10000
```

VisualVM

VisualVM may be used to find memory leaks or generate heap dumps from a josm instance running concurrently. Heap dumps can be structurally examined. VisualVM organizes the dump data e.g. by number and type of objects, memory used and cpu time consumed in an explorable tree view. Heap dump and garbage collection of the target instance is requestable using button clicks. Be aware that the default configuration of the memory sampler will sample VisualVM generated objects as well, which may be confused as a memory leak. The memory profiler can be set to profile specific classes only, it tracks allocations once started.

- VisualVM
- Step by step guide to memory profiling with VisualVM

If an autonomous instance of josm should be debuggable from within eclipse at the same time, command line parameters must be supplied to open a transport socket that the debugger can connect to:

```
# run JOSM with remote debug options; connect a debugger to
127.0.0.1 and port 9988
java
-agentlib:jdwp=transport=dt_socket,address=127.0.0.1:9988,server=y,su
-Xmx1200m -jar josm-latest.jar
```

This is not necessary if the internals of JOSM are to be explored by VisualVM exclusively.

JOSM core

- JavaDoc class documentation (for JOSM and JMapViewer)
- JavaBugs - you may encounter one of them

JOSM plugins

- Develop Plugins
  - Plugin publication steps
- Plugin installation without restart

SVN

- How to use your JOSM SVN account

JOSM Help System

- Context-sensitive help
- Current list of context-sensitive help topics

Shortcuts

- Current list of used shortcuts

Translation

- see Translations, Translations/Wiki and TaggingPresets#Translation

Developers

Dirk Stöcker is currently the lead maintainer of JOSM. Developers of JOSM with SVN commit privileges active in the past several months include:

Trac username	Name
akks	Alexei Kasatkin
bastiK	Paul Hartmann
Don-vip	Vincent Privat
Klumbumbus	Stefan Volke
michael2402	Michael Zangl
simon04	Simon Legner
stoecker	Dirk Stöcker
wiktorn	Wiktor Niesiobędzki

Immanuel Scholz was the original creator of JOSM, and previous maintainers include Frederik Ramm.

Previous developers include Frederik Ramm (framm) and Jiri Klement (jttt).

Many others have contributed through bug reports, patches, translations, and documentation work.

Last modified on 2018-06-26T23:04:01+02:00

Attachments (1)