GettingStartedMapWriter
The user documentation for the map-writer Osmosis plugin.
Featured

Updated Dec 1, 2012 by thilo.mu...@gmail.com

User Documentation: Mapsforge Map-Writer (0.3.0)

Introduction

This document describes the mapsforge map-writer plugin. It allows to convert OpenStreetMap data into the .map format which is needed to display maps with mapsforge-based applications. The tool is implemented as a plugin to the Osmosis software. To use the tool, you are required to have a working installation of Osmosis and the map-writer plugin copied to the plugins directory of Osmosis. You should also be familiar with the Osmosis tool.

Plugin Usage

- Activate the plugin with the Osmosis parameter '–map-writer', or short '–mw'.
- The plugin requires an input stream of valid OSM data. Use for example the MySQL, PostGIS, XML or PBF-Binary tasks of !Osmosis to read OSM data.
- Use the following optional parameters to configure the process of map creation:

Basic Options

Option	Description	Valid Values	Default Value
file	path to the output file, the file will be overwritten if existent		mapsforge.map
type	switch for main memory or hd mode	ram, hd	ram
bbox	bounding box definition as comma-separated list of coordinates in the form: minLat,minLon,maxLat,maxLon (be aware that osmosis does not allow **white space** in its command line parameters)	minLat, minLon, maxLat, maxLon in exactly this order as degrees or microdegrees	(blank)
map-start-position	write a start position to the file which is used, when the file is first opened in the MapViewer	latitude, longitude in degrees or microdegrees	(blank)
map-start-zoom	write a start zoom level to the file which is used, when the file is first opened in the MapViewer	zoom level as integer in [0;21]	(blank)
preferred-language	will write names of geo objects in the preferred language to the file, this only works for objects which have been tagged for the preferred language	language code as as defined in ISO 639-1/2	(blank)
comment	writes a comment to the file		(blank)

Advanced Options (only use when you know what you are doing)

Option	Description	Valid Values	Default Value

tag-conf-file	path to an XML file that customizes the definition which OSM-tags are recognized	path to an XML file, please read section 'Defining a Custom Tag Mapping via XML' carefully before using thsi parameter	(blank) internal default tag mapping is used
polygon-clipping	use polygon clipping to reduce map file size (minimal performance overhead)	true/false	true
way-clipping	use way clipping to reduce map file size (minimal performance overhead)	true/false	true
label-position	compute label position for polygons that cover multiple tiles (minimal performance overhead)	true/false	true
simplification-factor	simplifies ways and polygons with a topology preserving algorithm similar to the Douglas Peucker algorithm, using as the maximum distance difference value the given simplification factor (evaluated in pixels on max zoom level of a base zoom level); on base zoom levels higher than 12, no simplification is computed	positive real number	5
bbox-enlargement	amount of pixels used for enlarging bounding boxes in computations	positive integer	20
zoom-interval-conf	configure the zoom intervals used in this file, configuration is given in the form: baseZoomA, minZoomA, maxZoomA, baseZoomB, minZoomB, maxZoomB,..., baseZoomN, minZoomN, maxZoomN, in most cases you do **not** need to alter the standard configuration	intervals must not overlap and must not contain gaps	5,0,7,10,8,11,14,12,21
debug-file	switch for writing debug information to the file, *do **not activate** this option unless you know what you are doing*	true/false	false

Examples

- Write map file for Berlin using Binary-PBF format and writing into file /tmp/berlin.map:
  `$ bin/osmosis --rb file=../data/berlin.osm.pbf --mapfile-writer file=/tmp/berlin.map`
- Write map file for Germany using Binary-PBF format and writing into file /tmp/germany.map, setting the processing mode to hard disk:
  `$ bin/osmosis --rb file=../data/germany.osm.pbf --mapfile-writer file=/tmp/germany.map type=hd`
- Write map file for Bremen using XML format and writing into file /tmp/bremen.map, setting map start position to Bremen HBF:
  `$ bin/osmosis --rx file=../data/bremen.osm --mapfile-writer file=/tmp/bremen.map map-start-position=53.083418,8.81376`
- Write map file for Berlin-Dahlem. Data has been exported as XML from OSM website, so that we must use a bounding box definition:
  `$ bin/osmosis --rx file=../data/dahlem.osm --mw file=/tmp/dahlem-high.map bbox=52.4477300,13.2756600,52.4588200,13.2986600`

Known Pitfalls

- The plugin requires a bounding box definition, which is either included in the data or is given via the command line parameter `bbox`. Take note that the XML export functionality of the OSM website currently produces invalid bounding box definitions, so that the `bbox` parameter must be used in this case.
- If you installed the plugin into the user home, please make sure that you run osmosis with exactly this user and not with another user (e.g. the root user).

Plugin Installation

- Download the map-writer plugin from the download section and read the Osmosis documentation of how to install a plugin. There are several alternatives:
  1. Copy the downloaded plugin to $USER_HOME/.openstreetmap/osmosis/plugins (Linux) or Application Data\Openstreetmap\Osmosis\Plugins (Windows).
  2. Copy the downloaded plugin to subdirectory plugins in the current directory.
  3. Use the Osmosis parameter '-plugin

     <plugin_qualified_classname>

     ' to load the plugin. The qualified classname of the map-writer is 'org.mapsforge.map.writer.osmosis.MapFileWriterPluginLoader'.
- If you want to use a SNAPSHOT version of the plugin, you can build it yourself by following the instructions given in the article GettingStartedDevelopers. The produced plugin will be automatically copied to the Osmosis plugin directory in your home directory.
- You may want to increase the Java heap space that may be allocated for osmosis. You can do so by editing the script $OSMOSIS_HOME/bin/osmosis(.bat). Insert a line with 'JAVACMD_OPTIONS=-Xmx800m'. This sets the maximum available Java heap space to 800M. Of course you can set this parameter to a value which ever fits best for your purpose.
- See http://wiki.openstreetmap.org/wiki/Osmosis/Installation for further information about Osmosis usage.

Installation Example

- Let the home directory of the user ($USER_HOME) be "/home/prometheus/".
- Let the directory where osmosis has been extracted to ($OSMOSIS_HOME) be "/home/prometheus/development/osm/osmosis/osmosis-0.40.1/".
- Create the directory "/home/prometheus/.openstreetmap/osmosis/plugins/" and copy the mapsforge-map-writer-x.x.x.jar to this directory.

Software Requirements

- Osmosis 0.36 or higher (tested with Osmosis 0.36 - 0.40.1)
- Java 6 Runtime (tested with Sun JDK and OpenJDK)
- Tested on Linux platform (tested with Ubuntu 10.04) and MS Windows platform (tested on Windows Vista and Windows 7)

Hardware Requirements

The currently realeased version of the map-writer gives you the choice of either processing the input data completely in main memory or using the

hard disk for storing temporary data structures. You can switch between these processing modes by using the 'type' parameter, see Plugin Usage below.

We recommend using the main memory mode only for small input files (< 200 MB in PBF format) as it requires quite a huge amount of memory (about ten times the size of the input file).

Defining a Custom Tag Mapping via XML

This section describes how to configure the ***known tag set*** via an XML file. The *known tag set* comprises all OSM tags (for ways and POIs) that are known to the renderer. You can use the XML configuration to define which subset of the known tag set should be included in the map file and to configure the zoom levels on which map objects first appear.

The default internal tag mapping defined in http://code.google.com/p/mapsforge/source/browse/mapsforge-map-writer/src/main/config/tag-mapping.xml is kept in sync with the known tag set of the renderer. So, if you want to define your own custom mapping use the internal mapping as a reference. Please consult the XML-Schmea documentation of http://code.google.com/p/mapsforge/source/browse/mapsforge-map-writer/src/main/resources/tag-mapping.xsd to learn about how to implement your custom mapping. The XML-Schema is very easy to understand. We recommend to use Eclipse as an editor for XML as it allows for auto-completion if you provide an XML-Schema.

You need to be aware that this configuration only defines what data is to be included in the map file. How the data is eventually rendered is specified by a rule-set that is attached to the renderer. So if you add any tag to the writer's tag configuration that is not recognized by the renderer, it will not be displayed in the map. In this case, you have to make sure that you also define in which way the new tag is to be rendered. How to configure the rendering is described in the article RenderThemeAPI.

Feedback

Please report any bugs and feature requests via the mapsforge issue tracking.

Changelog

0.3.0

- implements new binary format version 3
- improved handling of multi-polygons and relations
- improved clipping of polygons and lines (ways)
- more efficient encoding of coordinates
- supports definition of preferred language
- many other improvements and bug fixes

0.2.4

- minor improvements and bugfixes

0.2.3

- minor improvements and bugfixes

0.2.2

- improved coast line handling: detects tiles that are completely covered by water and marks them as such in the binary file
- the tag list of the map-writer can now be configured via an XML file, please read section 'Defining a Custom Tag Mapping via XML' carefully and see the new option 'tag-conf-file' below
- computes the distribution of tags in the input file and based on this assigns internal ids in an optimized fashion, reduces file size
- polygon clipping is more restrictive due to a conceptual problem with clipping outlined (instead of filled) polygons, increases file size (admittedly overcompensates file size reduction mentioned above)
- bug fixes

0.2.1

- supports hard disk as temporary storage device, reduces required amount of main memory significantly, see Hardware Requirements below
- support for MS Windows platform
- removed dependency on JTS library
- writes new file format v2, reduces file sizes about 20%, see SpecificationBinaryMapFile
- bug fixes