Bringing Validation to Users
Integrating Quality Assurance Checks into Map Editors

Clarisse Abalos @map2save
Matthew Gibb @giblet22
The Problem

• Abundance of data issues waiting to be cleaned up
• There are more mappers than validators
• Mappers need more guidance while mapping
Quality Assurance Approach

Stop bad or incomplete data from being created
Solutions

Validation rules in JOSM and iD

Custom Presets and Validation Rules

Validation-centric Overpass Queries
Validation in JOSM

Validation on dataset while editing features

Warnings before upload

Validation Results

Errors (1)
- route=bus on a node. Should be used in a relation (1)

Warnings (3)
- Crossing highways (1)
  - 2 ways: highway (2 nodes), highway (2 nodes)
- Unnamed ways (2)

Suspicious data found. Upload anyway?

The following are results of automatic validation. Try fixing these, but be careful (don't destroy valid data). When in doubt ignore them.
When you cancel this dialog, you can find the entries in the validator side panel to inspect them.

Errors
- Usually this should be fixed.

Warnings
- Fix these when possible.
- Informational warnings, expect many false entries.

Errors (1)
- route=bus on a node. Should be used in a relation (1)
- node

Warnings (3)
- Crossing highways (1)
  - 2 ways: highway (2 nodes), highway (2 nodes)
- Unnamed ways (2)
  - highway (2 nodes)
JOSM Data Validator
Types

Tag Checker Rules (MapCSS)
- Great for basic topology and tag checks

Tests (Java)
- Able to handle more complex geometry checks
What is MapCSS?

- CSS-like language for map stylesheets
- Used in JOSM for map styling and data validation

Each color represents a building with a different name.
Basic MapCSS Validation Checks

```
geometry[key=value] {
    throwWarning: "Fix this";
}
```

Selector: what feature the rule is looking for

Instructions on what to do with the feature

```
*[amenity=hospital][!name] {
    throwWarning: "Hospital without a name tag";
}
```

Selector: * = all geometries (node, way, relation), Features tagged as **amenity=hospital** that does not have a **name** tag.

Warning in JOSM validator window

Features that meet the rule criteria appear in the JOSM Validation Results window
Create your own MapCSS checks locally

1. Write the rule - save as .mapcss or .validator.mapcss file
2. Add as an Active Rule in JOSM
3. Click “Validation”
Converting other QA tools to MapCSS

```sql
/* Keep Right motorways connected directly 270 */
/* Message: This node is a junction of a motorway and a highway other than motorway, motorway_link, trunk, rest_area or construction. Service or unclassified is only valid if it has access=no/private or if it is a service=parking_aisle. */

way[hhighway=motorway] >[index=-1] node:connection,
    set kr_270_end_motorway;
}

way[hhighway][highway!=motorway] node:connection {
    set hwycon270;
}

/* allowed junctions */

way[hhighway=service][access=no] node:connection,
    way[hhighway=service][access-private] node:connection,
    way[hhighway=service][service=parking_aisle] node:connection,
    way[hhighway=unclassified][access=no] node:connection,
    way[hhighway=unclassified][access-private] node:connection,
    way[hhighway=unclassified][service=parking_aisle] node:connection,
    way[hhighway=motorway_link] node:connection,
    way[hhighway=motorway_junction] node:connection,
    way[hhighway=construction] node:connection {
        set kr_270_allowed;
    }

way[hhighway=motorway] node:connection.hwycon270!..kr_270_allowed!..kr_270_end_motorway {
    throwWarning: "270: motorways connected directly";
    group: "KeepRight";
}

/* KeepRight missing maxspeed 300 */
/* Message: missing maxspeed tag */

way[hhighway =~/\^\(motorway|trunk\|primary|secondary\)\$/][junction|roundabout][!bridge][!tunnel][!maxspeed] {
    throwOther: "300: missing maxspeed on major road";
    group: "KeepRight";
}
```
Available JOSM Tag Checker Rules

- Osmose (QA Tool Inspired Validations)
- KeepRight
- OSM Lint
Create new rules

If you want to add your rules to this list use one of the two options:

- Simply create a page in this wiki with your rule, e.g. Rules/MyRuleName. This is similar to Styles and Presets. See there for more information about this topic.
- Externally hosted rule URLs can be entered when editing the RulesSource page.

If your rule applies to a limited area, use Territory selectors, to limit the execution to the area.

Rules managed in JOSM wiki

Josm wiki based Rules is the preferred method.

- Rules/AtmsSlovakia
- Rules/CzechRepublicAddressSystem
- Rules/FRFranceSpecificRules
- Rules/GermanySpecific
- Rules/KeepRight
- Rules/MissingStreetnumberOrConscriptionnumber
- Rules/MissingWeekdays
- Rules/OSMLint
- Rules/Portuguese-Errors-PT2
- Rules/Portuguese-Specific
- Rules/QAToolInspiredValidations
- Rules/SuspiciousSwimming_Pool
Osmose-QA : MapCSS, OpenData and Mapillary

Focus on MapCSS rules set as common rules validation. Explore integration of OpenData and Mapillary AI vision extracted data.

Room: Großer Hörsaal

Frédéric Rodrigo (Makina Corpus)

Monday, 09:30

At the beginning, Osmose-QA is a Quality Assurance tool pointing issues on map, but also able to compare external data set to OpenStreetMap.

Back to schedule

First talk part is about the new support of MapCSS rules in Osmose-QA. MapCSS language was primarily designed as way to style map objects, it look like the web styling language CSS. The base of MapCSS is to use selectors to filter objects. JOSM already use it for map styling and validator. This new implementation of MapCSS in Osmose-QA allows to write rules more easily. It also allow to reuse JOSM validator rules set. Next step would be to run Osmose rules in JOSM. It's a great opportunity to share rules across tools and projects. But come the question on how to share and manage this rules while considering tools particularities.

What's behind JOSM?

Presentation of the JOSM development model: how is it made, by who? How can I help?

Room: Großer Hörsaal

Vincent Privat

Sunday, 14:00

JOSM is almost as old as OSM but few people really know what it takes to maintain your preferred editor. We'll present the development model of JOSM and who's part of its active community: developers, translators, testers, plugin authors, end users, sponsors, etc. We'll talk about the project difficulties, the major achievements made in the past years, what work is currently in progress and what will happen in the near future!
JOSM Validator – Water Structures

Inspired by Osmose analyser: #1221 “Far from water”

(Developed by: Milla Zagorski)
JOSM Validator - Waterways

Inspired by Osmose analyser: #1220 “Waterway”

(Developed by: Milla Zagorski)
JOSM Validator – Public Transport

Inspired by Osmose analyser: #1260 “Public Transport”

(Developed by: Milla Zagorski)
What's new in JOSM?

58 new validators!
- Osmose – 41
- KeepRight – 7
- OSM Lint – 9
- Atlas – 1

(Developed by: David Manzer & Milla Zagorski)
Mainly addressing...

highway issues

tagging issues

tree-trimming issues
A really sweet bar chart
Custom Presets and Validation Rules

Simple UI to create guidelines that can be tied to campaigns and generate presets and validation rules that can tailor the editors to your campaign/organization focus specific needs.

Requirements for mapping features become
• Simplified
• Standardized
• Streamlined
Creating Attribution Guidelines

Assisted with taginfo popular combinations
### Campaign Focused Presets

<table>
<thead>
<tr>
<th>Search</th>
<th>Feature Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>![Area Icon]</td>
</tr>
<tr>
<td>hospital</td>
<td>![Hospital Icon]</td>
</tr>
<tr>
<td>sports field</td>
<td>![Sports Field Icon]</td>
</tr>
<tr>
<td>school</td>
<td>![School Icon]</td>
</tr>
</tbody>
</table>

- **Tools**
  - Point
  - Line
  - Area

- **Select feature type**: Use the dropdown menu to select the type of feature you want to add to your map.
Realtime Validation

• Only show fields and values relevant to campaign
• Throw validation errors or suggestions for further information while editing
In Progress

https://github.com/radiant-maxar/maprules

USER PRESETS

SHARING AND REUSING PRESETS

INTEGRATING INTO TASKING MANAGER 4
Validation Based Overpass Queries

Why?
• Stale errors in QA databases
• Pulling in validation targeted data over larger areas
• Bulk operations
MapRoulette Challenges

GeoJSON
- Local File
- Remote Url
- Overpass Query

Add layer to bridges/tunnels in Portugal

INSTRUCTIONS
- add a layer to the bridge: minimum layer=1 ascending
- add a layer to the tunnel: minimum layer=-1 descending

COMPLETION
Add MP Tags
Current Editor: Edit in ID (web editor)

LOCATION

Properties
- osmid: 243199808
- bridge: yes
- source: bing
- highway: residential

Warning: Rebuilding can lead to task duplication if your feature ids are not setup properly or if matching up old data with new data is unsuccessful. This operation cannot be undone!

Learn More

First remove incomplete tasks
<table>
<thead>
<tr>
<th>Validation Query</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary Missing Source?</td>
<td>An administrative boundary does not contain tag source*^; sourcing the origin</td>
<td>OSMOSE 2040: Source</td>
</tr>
<tr>
<td>Building Road Intersection?</td>
<td>Flags buildings that intersect/touch center lines of roads. This does not address cases where buildings get really close to roads, but do not overlap them.</td>
<td>Atlas: Building Road Intersection</td>
</tr>
<tr>
<td>Combinations</td>
<td>Missing tags on a given combination, suspicious tag combinations</td>
<td>JOSM: combinations?</td>
</tr>
<tr>
<td>Deprecated Tags?</td>
<td>Deprecated tagging that needs to be updated</td>
<td>JOSM: deprecated?</td>
</tr>
<tr>
<td>Elevation?</td>
<td>Some elements, including the peak (natural:peak) and mountain_pass (mountain_pass=yes), has an evolution. This is shown in OSM with tag ele=* in meters.</td>
<td>OSMOSE 2020: Elevation</td>
</tr>
<tr>
<td>Tumble?</td>
<td>Items tagged with tumble</td>
<td>KeepRight 170: tumble tagged items</td>
</tr>
<tr>
<td>Geometry?</td>
<td>Incorrect geometry used or topological issues</td>
<td>JOSM: geometry?</td>
</tr>
<tr>
<td>Highway?</td>
<td>Highway validations</td>
<td>JOSM: highway?</td>
</tr>
<tr>
<td>Missing Highway?</td>
<td>The way seems to be a highway^#.</td>
<td>OSMOSE 2080: Missing Highway,CITY 20800, 20801, 20803, 20804</td>
</tr>
<tr>
<td>Missing OneWay on Motorway Link?</td>
<td>Detecting motorway links without oneway tag, which is connected to a highway:motorway with an oneway</td>
<td>OSMLint: missing oneway on motorway link?</td>
</tr>
<tr>
<td>Orphan Nodes?</td>
<td>Untagged nodes that do not belong to a way</td>
<td>OSMOSE 6030: place without name</td>
</tr>
<tr>
<td>Place without Name?</td>
<td>The tag place* must always be used in combination with the tag name*.</td>
<td>OSMOSE 7040: power lines Class 1</td>
</tr>
<tr>
<td>Powerlines?</td>
<td>This analysis mainly verifies the continuity of powerlines</td>
<td>JOSM: power lines</td>
</tr>
<tr>
<td>Relation?</td>
<td>Relation validations</td>
<td>JOSM: relation?</td>
</tr>
<tr>
<td>Tunnel and Bridge without Layer?</td>
<td>This validator detects all the bridges that does not have the tag layer*.</td>
<td>OSMLint: tunnel and bridge without layer?</td>
</tr>
<tr>
<td>Tunnel/Bridge Max Height?</td>
<td>Missing maxheight* or maxheight * for a tunnel or a way under a bridge.</td>
<td>OSMOSE 7130: Tunnel/Bridge</td>
</tr>
<tr>
<td>Two Names?</td>
<td>The tag name* contains two street names, separated by a semicolon, a &quot;or a &quot;'. This issue was probably produced by the fusion of two way and the concatenation of the names of streets.</td>
<td>OSMOSE 5030 - Two names</td>
</tr>
<tr>
<td>Unnecessary?</td>
<td>Features with unnecessary tags that can be removed</td>
<td>JOSM: unnecessary?</td>
</tr>
<tr>
<td>Wrong Highway on Roundabout?</td>
<td>Class 1 &quot;Wrong highway on roundabout&quot;: it must match the highest level of connected routes, except highway and trunk.</td>
<td>OSMOSE 3010: Class 1 wrong highway on roundabout</td>
</tr>
<tr>
<td>Wrong Railway Crossing?</td>
<td>Checks level crossings if all ways have same layer* and any way has tag bridge* or tunnel*.</td>
<td>KeepRight 160: Wrongly Used Railway Crossing Tag</td>
</tr>
</tbody>
</table>
4 Quality Assurance

4.1 Checking boundary relations for correct tagging
4.2 Find all bus stops which are not included in a relation
4.3 Bus relations, which are not part of a master relation
4.4 Counties without fire station
4.5 Buildings crossed by roads that are not tagged as tunnel or covered
4.6 Find duplicate nodes (since 0.7.54)
   4.6.1 Find duplicate nearby nodes with same tag
4.7 Search for untagged ways
4.8 Orphaned nodes (nodes without tags, not part of any way or relation)
4.9 Find broken associated Street-relations
4.10 Find house numbers without street
4.11 Non-matching addr:postcode=XXXXX tags inside a boundary relation with postal code ≠ XXXXX
4.12 Non-matching addr:postcode=XXXXX tags inside multiple boundary relation with postal code ≠ XXXXX (since 0.7.54)
4.13 Ways that are part of a "Bundesstraßenrelation", but do not have a ref-tag
4.14 POIs with websites but with incorrect opening hours

Contents [hide]

1 Complete route relation
2 Part of a (route) relation
3 Power substations without proper voltage tag
4 Living streets with a potentially wrong speed limit
5 Highways with lanes=1 but no other explaining tag
6 Nodes that only have a name tag
7 Bridges without maxwidth=...
8 Tunnels without maxheight=...
9 More
   9.1 Overpass API by Example / DE:Overpass API/Beispielsammlung
   9.2 Quality assurance of postal codes
   9.3 Soldier Boy's Examples
   9.4 Parking areas example
   9.5 More various Overpass examples
   9.6 Public transport Quality Assurance
   9.7 Overpass Queries based on Quality Assurance Tools
Automatically Fix Validation Errors
Why does this matter?

• Custom validations give users and groups the directed focus they need.
  • If you're only interested in mapping railways, why not share your `railway=switch` validation MapCSS checker?
• Thematic validations can assist with directed editing and cleanup
Create and Share Validation Rules

ADD TO MAPCSS VALIDATION RULES
ADD TO JOSM VALIDATORS
CREATE MAPRULES
Questions

clarisse.abalos@maxar.com
@map2save

matthew.gibb@maxar.com
@giblet22