Highway Data Collection

I recently was updating map data to clear up any inaccuracies and improve the tagging. To do this I used some custom XLSXForms to collect the data, as ODK Collect is often used for ground data collection. Part of the complication is that XForm metadata doesn’t follow the same standards as OSM, although with care, it can be close.

To start with, much of this trip was validating data on remote roads in rural Colorado. As I was pretty sure some of the data was good, and some bad, but you have to stand there to be really sure. This project (other than some nice camping) was to validate improvements to the metadata to

I was not planning on importing all of this data luckily, maybe only a tiny handful that met the criteria I was looking for. Part of the goal was cataloging water sources and parking areas for wildland firefighting. Often what looks like a nice meadow in satellite imagery is full of Aspens trees. And springs dry up, or are inaccessible or lack sufficient flow to be used for refilling trucks,

The other thing I was focusing on was adding detail to the roads. Much of rural Colorado suffers from the TIGER import in 2008. To add complexity, in rural Colorado a road may have multiple names, which OSM can handle. For example, what looks like a jeep track might have a county designation, and Forest Service reference, and a local name. Often map data from multiple sources doesn’t agree, so the only way to validate is to go there. What the sign says on the intersection is the correct version. All this becomes important, as you’ll get any one of the names depending who called 911, and our job is to find that location regardless. For wildland fires, many responders are from out of state, so they too benefit by better map detail. The other critical detail is the road condition, which is used to determine which apparatus to respond in. That can’t be determined from satellite imagery.

While I do find Street Complete or Vespucci useful, the goal of this trip was to only use ODK Collect. That presented a few challenges, which I’ll cover.

The office, sometimes you can find AC power

Creating The XForm

An XLSXForm is edited via any spreadsheet program. This forces some weird syntax at times, but it can be very flexible once you wrap your head around it. An XLSXForm has two primary sheets, survey and choices. Survey is the top level questions, and choices are the values that get displayed for selection via menu pulldowns. On the survey sheet, you get to choose a variable to hold the data. This variable is then referenced by the choices. If the variable name used for the survey question matches an OSM keyword, then that can be converted easily into OSM format. Same thing for the choices, If each choice is also an official value for the top keyword, then that becomes a 1 to 1 conversion, so easy.

This applies to all survey questions that aren't a selection, and to menus that only have one value returned. Multiple selection menus return a string of all the values concatenated with spaces as delimiters. Those can be cleaned up later using an OSM editor like JOSM, but I prefer a conversion program.

Data collected using ODK Collect is stored in an XML format. The only tool that can read this format is QGIS. QGIS can convert the XMLdata file to GeoJson, which can be edited using JOSM. And of course the data will be edited and validated before uploading to OSM.

The other format for the collected data is when you download a CSV file from ODK Central. Every question is a separate field in the file. Each of the keywords are used as the field headers. If you used groups in the XLSXForm, the are prefixed to the keyword using an underbar as a delimiter, as multiple levels of grouping is supported.

JOSM can't read the CSV files, so conversion is required, There's several manual ways to do the conversion, but it still requires editing the tags. I use a Python script to do the conversion, as it generates the best OSM formatted data, requiring the least editing.

The Conversion Process

I wrote a conversion program that works with the XLSXForms I've been using. Several of my XLSXForms use grouping and conditional display, so in that case I have a top level keyword that doesn't relate to an OSM keyword, its value does. For example, one of my XForms for highways the first question is this a highway, a path, or a barrier. That selection enables questions specific to that feature type. That keeps my XForms efficient, and reduces clutter. The simpler way to handle this is to have one XForm for each feature type. I have a lot of Xforms for various features, so prefer to combine related data in a single form. Many of my XForms start with asking what the top level feature is, or enabling more detailed survey questions.

As ODK Collect adds multiple fields OSM doesn't care about, the first thing is to just ignore those. All keywords are forced to lower case for the comparison. Any of the grouping variables are stripped off, leaving the final part, which is the actual keyword. Some of the standard ones ODK adds are converted to their OSM equivalent, ie,,, "Latitude" becomes "lat".

The conversion is relatively simple for most XForms. Conversion gets more interesting when you have values for your questions that don't fit under the keyword's official value, it fits under a different one. The possible values for a campsite might want "leisure=fire-pit", and also "tourism=camp_pitch", and "amenity=parking", but these might all be values for the questions in the XForm. To handle that conversion a simple config file is used. Some OSM editing apps will let you choose a value knowing the keyword, but for new users it's easier the just have the values for the same question.

Collecting Data

The technical goal of this trip was to experiment with the data flow of getting data collected with ODK Collect into OSM, since they use incompatible file formats.

Validating Highways

The goal wasn't to drive every jeep track in the mountains, but sometimes data is lacking from any source for the really obscure tracks. These are used heavily for wildland fire or backcountry rescues, and often haven't been updated since the TIGER

import in 2008. It is possible to make basemaps for ODK Collect that work offline, but it doesn't allow you to edit any existing data in OSM. Often you are driving up one jeep track, and another branches off that you want to record but not travel on.

For my highway XForms, I use a GeoPoint. I'll stop at an intersection, and as quickly as possible record the data, and offset the location to be up the side road. Later in JOSM I can copy any of the tags from ODK data to the existing OSM data, since the GeoPoint will be on the highway. The highway XForm collects data on smoothness, surface, track type, width, etc… While not accurate for the entire road (conditions change) it's usually more data than is already available.