

## Gisaf - Feature #6100

### Shapefile basket: organize with projects

26/06/2018 11:40 - Philippe May

<b>Status:</b>	Closed	<b>Start date:</b>	26/06/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Philippe May	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>Description</b>			
<p>This issue was actually pending for a long period.</p> <p>It came to surface in a simple use case: importing a shapefile that doesn't have guaranteed unique ids (eg. the paved roads in the MM) generated duplicate features in the database.</p> <p>Until now, Gisaf could manage with multiple of shapefiles because they all had a unique id, so Gisaf could <b>update</b> the features already in the database.</p> <p>Another aspect of the same problem is: the features in a shapefile like the one for "the paved roads in the MM area" has its ids (presumably starting from 1). A shapefile of paved roads for another project would have another set of features, sharing the same ids.</p> <p>The proposed solution involves:</p> <ol style="list-style-type: none"><li>1. organize the basket of shapefiles the same way than the one for raw survey points: shapefiles are in subfolders named after a project (as defined in the "project" table)</li><li>2. add a column (project_id) to all the geometries generated from shapefiles</li><li>3. modify the "shapefile import" function:<ul style="list-style-type: none"><li>◦ before adding the features to the database, it will first <b>delete all the features in that layer (table) for the given project</b></li><li>◦ assign the given project for all the features in the shapefile</li></ul></li></ol>			

#### Associated revisions

##### Revision bd1fcae7 - 26/06/2018 12:30 - Philippe May

Refs #6100: add project\_id field and project relation to BaseSurveyModel; organize shapefile admin in project folders; import shapefile delete the features in the given project and assigns the project to each "new" feature

##### Revision 2759da02 - 27/06/2018 12:55 - Philippe May

Refs #6100: add index to project and refactored the notebook

#### History

##### #1 - 28/06/2018 14:38 - Philippe May

The import of "arbitrary" shapefile (with no id for the features) is a challenge for keeping track of the elements in the database.

So, the import function is now refined, together with the notion of project (which was introduced in relation with the geo survey points, and also the requirement of generating unique ids).

The new version is smarter than the proposed solution.

For each feature (something in the shapefile that has a geometry and eventually attributes) found in the shapefile, search in the database, in the same table (category) and project:

- if a feature already exists. If yes, do not touch it and only update the attributes, eventually.
- add new geometries not found in the database: the unique id is generated by the database
- delete geometries which are not in the shapefile

In other words, concerning the unique id: a feature is given its unique id when it's created OR modified.

I think that it's as good as we can do, as long as we allow importing shapefiles which do not have ids given by Gisaf or the database.

In order to be able to trace the origin of the items in the database and on the map, there's now a new table (called `feature_import_data`). The shapefile import function stores, for each feature, the file name, the date, the table, the "original id" (as in the shapefile) and the id of that feature in the database.

Actually, i'm just thinking of it while writing this, it could be used for keeping track of the changes in geometries...

## #2 - 28/06/2018 15:06 - Philippe May

For now, i'll update the production with the changes described above.

This implies:

- move the shapefiles in the basket to folders named according to their project. I choose the convention: "Misc" is a project for shapefiles not originated from survey data. I have written a small script "mv\_basket.py" for moving files in the baskets to folders
- make changes to the database schema, adding the project, indexes, constraints to all existing tables. I have written a notebook for that: "Add project\_id columns.ipynb"
- tag all the existing features not originating from the survey data, with the project "Misc" (also in "Add project\_id columns.ipynb")

Then, we should empty the tables of lines and polygons from survey data, and reimport from the shapefiles, in order to be able trace the origin of all the features in the database.

## #3 - 29/06/2018 13:06 - Philippe May

- Status changed from *New* to *Resolved*

- % Done changed from 0 to 100

Seems to be OK:

- moved the shapefiles in their respected (project) folders
- re-imported the shapefiles for project MM (using the Add project\_id columns.ipynb notebook, which now has an option to force\_delete all the items in the database
- also improved the output of the import function in the basket, making it more clear the number of features added, updated, deleted, etc.

The table which tracks the origins of the features now contains all the references. For completeness about how to track where the items on the map come from:

- points: they keep themselves the original\_id, project, surveyor and equipment, as Gisaf has complete control of the process
- lines and polygons: the table `feature_import_data` keeps track of the imports done through the basket (not used as of today, only for further reference)

The only visible change is that the items on the map now show their "project".

I didn't test if all these changes affect the shapefiles in the Misc project (not made from survey data, ie. the wells, zones, etc).

It was a bit of an headache, with so many edge cases to take in account...

**#4 - 19/12/2018 15:45 - Philippe May**

- *Status changed from Resolved to Closed*