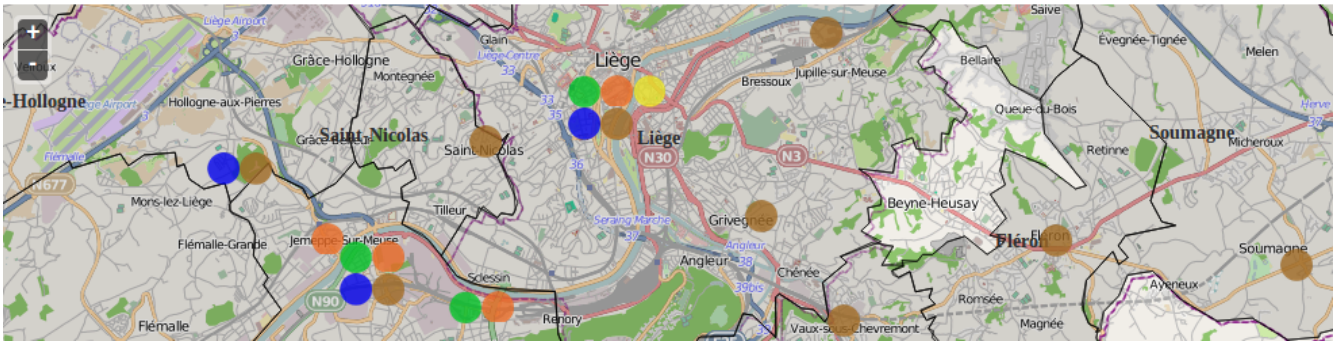
Réseau des Consommateurs Responsables ASBL

Cartographie web des alternatives

Conception de la carte et guide de la mise à jour

version 2.0 – Mars 2014



Plan du document:

Minet Julien
julien_wa@yahoo.fr
0496 24 17 60

Juin 2011 – Mars 2014

1. Introduction

Ce document a pour but d'expliquer la conception et la mise à jour de la carte du site Internet (webmap) de l'ASBL Réseau de Consommateurs Responsables (RCR). Toutes les solutions présentées ici se font avec des logiciels libres, et de préférence sous Linux (Ubuntu), bien qu'une bonne partie des méthodes peuvent se faire sous d'autres systèmes d'exploitation (Windows).

La première partie explique la conception et l'architecture de la carte. Le but de cette partie est de conserver une trace de la réflexion technique qui a prévalu lors de la conception de la carte. La deuxième partie explique en détail les étapes de mise à jour de la carte, lorsque de nouvelles alternatives doivent y être intégrée. Le nombre de pages de cette deuxième partie ne doivent pas effrayer : la plupart des étapes de mise à jour ont été relativement automatisées.

2. Conception de la carte

2.1. Logiciels utilisés

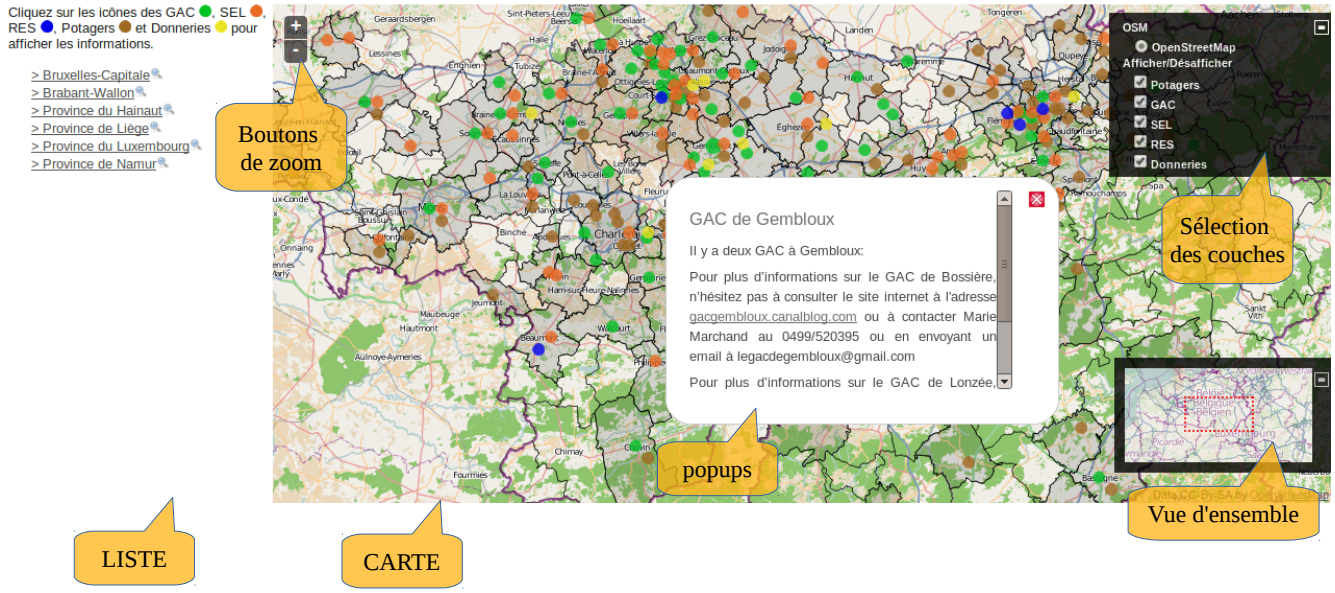
Afin de créer cette carte, un certain nombre de logiciels ou librairies sont utilisées. Cette section décrit brièvement ces outils:

1 OpenLayers: OpenLayers est une librairie écrite en JavaScript qui fournit un environnement de navigation, d'affichage d'informations, etc. pour des cartes dynamiques sur internet (webmap), incluses dans des sites web. Ce n'est donc pas un logiciel à installer sur un ordinateur ou un serveur, mais simplement une librairie à télécharger. Infos & téléchargement : OpenLayers.org, Info sur Wikipedia. Cette librairie permet de construire et de personnaliser la carte des alternatives RCR.

2 QuantumGIS (ou QGIS) est un logiciel de système d'information géographique (SIG) libre. C'est le projet libre le plus avancé et abordable en matière de SIG, avec des fonctionnalités d'édition vectorielle, de requête spatiale, etc. Il permet de manipuler les données géographiques lors de la mise à jour des informations.

3 Python : Environnement de programmation. Il permet d'automatiser la mise à jour des informations dans les popups à partir d'une seule table (fichier Excel).

2.2. Création de la carte



La figure ci-dessus montre une vue d'ensemble de la carte telle qu'elle se présente sur le site. Il y a deux parties principales:la CARTE à droite (carte où s'affichent les alternatives et leurs popups) et la LISTE à gauche (liste des alternatives par communes de Wallonie-Bruxelles).

La carte a été conçue avec OpenLayers pour la Wallonie et Bruxelles. Techniquement, la carte est codée avec des fonctions d'OpenLayers dans le fichier « map.js », qui est ensuite appelé dans un fichier map.html, décliné en « map_list.html » et en « map_iframe.html » pour la carte plein écran et intégrée dans le site, respectivement.

Le fichier map.html appelle les ressources nécessaires pour créer la carte :

1. La librairie OpenLayers.js : La version de la librairie OpenLayers est la 2.12. Cette version a été stockée dans le dossier *lib*.

   ```
   <script type="text/javascript" src="lib/OpenLayers-2.12/OpenLayers.js">
   </script>
   ```

2. Le fichier map.js

   ```
   <script type="text/javascript" src="map.js"></script>
   ```

3. Une feuille de style CSS pour le style des contrôles OpenLayers. Attention, cette feuille code de nombreuses choses importantes pour le design de la carte, dont le style et la taille des popups, la position des contrôles, etc.

   ```
   <link rel="stylesheet" href="css/style-ol.css" type="text/css" />
   ```

4. Une feuille de style CSS pour les autres styles de la page

   ```
   <link rel="stylesheet" href="css/style.css" type="text/css" />
   ```

Le fichier *map.js* contient plusieurs lignes de codes importantes qui sont décrites ici :

1. Projections : La carte est entièrement projeté en WebMercator (EPSG : 900913 ou EPSG :

3857). Cette projection est nécessaire pour les couches OpenStreetMap. La projection affichée (par ex. dans le coin inférieur droit) est par contre la projection la plus répandue, la WGS1984 (EPSG : 4326), dont les coordonnées sont exprimées en longitude et latitude.

```
var proj = new OpenLayers.Projection("EPSG:900913");
var dispproj = new OpenLayers.Projection("EPSG:4326") ;
```

2. Les limites géographiques de la carte pour la Wallonie sont exprimées dans la variable « bounds » en WGS1984:

```
var bounds = new OpenLayers.Bounds(4,49.8,6,50.8)
```

3. Ajout de la couche de base (ou de fond) : Une couche OpenStreetMap a été ajoutée comme fond de la carte. Cette carte libre du monde est appellée depuis les serveurs d'OpenStreetMap:

```
osm = new OpenLayers.Layer.OSM("OpenStreetMap");
map.addLayers([osm]);
```

4. Ajout des couches thématiques (GAC, SEL, RES, …) : Des couches sont ajoutées au-dessus de la couche de fond. Ces couches sont ajoutées par des fichiers vectoriels au format GML. Leur création est expliquée plus loin dans ce document (partie Mise à jour). La stylisation (taille, icône, etc.) de ces couches est codée ici dans les variables StyleMap.
5. Ajout du sélectionneur de couches : Permet de (dé)sélectionner les couches thématiques. Il s'ouvre avec un clic sur la petite croix en haut à droite de la carte.

```
map.addControl(new OpenLayers.Control.LayerSwitcher());
```

6. Ajout de la vue d'ensemble :

```
map.addControl(new OpenLayers.Control.OverviewMap({mapOptions:
ovvwoptions}));
```

2.3. Liste des alternatives

Une liste des alternatives (GAC, RES, SEL, etc.) classées par provinces et par communes a été constituée à gauche de la carte. Toute la liste est écrite dans les fichiers map_list.html et map_iframe.html et plus d'information peur être trouvée sur la conception de la liste dans la section3.4 (Mise à jour de la liste).

Un clic sur les provinces permet de dérouler le contenu (i.e., les communes de la province) avec une fonction JavaScript « toggleVisibility » basée sur les fonctionnalités CSS « display » et « visibility » (pour Internet Explorer uniquement). Cette fonction est écrite au début des fichiers map_list.html et map_iframe.html.

Un clic sur les icônes des alternatives dans la liste ouvre automatiquement le popup correspondant dans la carte, grâce à la fonction « selectlist » écrite dans map.js et qui est décrite dans la section suivante (2.4 Popups).

2.4. Popups

Les popups contiennent de l'information sur chaque alternative dans les communes. Cette information est encodée dans le fichier RCR_carto.xls (voir 3.3 Mise à jour des popups).

Lorsqu'on clique sur les icônes des couches thématiques, le popup s'ouvre. Ce comportement est codé avec les lignes suivantes :

```
	var selectControl = new OpenLayers.Control.SelectFeature([potager, gac, sel,
	res, donnerie], {onSelect: onFeatureSelect, onUnselect: onFeatureUnselect});
	map.addControl(selectControl);
	selectControl.activate();

	function onFeatureSelect(feature) {
	selectedFeature = feature;
	var popupcontent = "<iframe height='190px' width='320px' src='./popups/" +
feature.layer.name + "/" + feature.layer.name + "_" + feature.attributes.NAME +
".html'></iframe>";
	var popup = new OpenLayers.Popup.FramedCloud("chicken",
feature.geometry.getBounds().getCenterLonLat(),
                  new OpenLayers.Size(200,380),
                  popupcontent,
                  null, true,
                  onPopupClose);
                  feature.popup = popup;
                  map.addPopup(popup)     }

	function onPopupClose(evt) {
	selectControl.unselect(selectedFeature);  }

	function onFeatureUnselect(feature) {
	map.removePopup(feature.popup);
	feature.popup.destroy();
	feature.popup = null;   }
```

Les popups peuvent être aussi ouvert depuis la liste des alternatives à gauche de la carte. Une fonction spécifique « selectlist » a été codée pour appeler les popups depuis la liste :

```
var selectedFeature=0;
function selectlist(couche,localite){
	if (selectedFeature!=0){
	try{map.controls[0].unselect(selectedFeature);}
	catch(err){};
	map.controls[0].select(map.layers[couche].features[localite]);}
	else{
	map.controls[0].select(map.layers[couche].features[localite]);}};
```

Cette fonction repose sur la fonction générale d'ouverture (et de fermeture) des popups décrite au début de cette section. Elle a juste pour but de sélectionner un élément thématique de la carte, ce qui provoque l'ouverture du popup correspondant. La fonction est appelée dans la liste à gauche de la carte.

2.5. Graphisme

Le graphisme de la carte dépend de beaucoup d'élément différents codés à différents endroits :

1. La structure générale de la carte, càd la taille respective de la liste et de la carte, l'overflow automatique, la structure de la liste, etc. sont codés dans map_list.html et css/style.css.
2. La plupart des éléments de la carte proprement dite sont codés dans le fichier style d'OpenLayers css/style-ol.css, qui a été modifié pour personnaliser la carte. Ce fichier permet

de styliser les éléments des boutons de zoom, de sélection des couches et de la vue d'ensemble aux points de vue de leur position, leur couleurs, etc. Le fichier code également quelques aspects des popups de la carte. Attention, pour que les propriétés css de ce fichier soient bien prises en compte, il faut souvent ajouter «!important » après la propriété afin de faire primer ces propriétés sur d'autres par défaut qui sont toujours codées dans les dossiers sources d'OpenLayers.

3. Les éléments de popups (forme du cadre) sont codés dans les dossiers sources d'OpenLayers dans le dossier « img » à l'intérieur de la librairie OpenLayers. Ces formes ont été modifiées pour les besoins de la carte.
4. Le fond cartographique utilisé est le rendu Mapnik d'OpenStreetMap. On ne peut pas modifier son graphisme mais il est important de noter que d'autres fonds sont possibles à partir des données OpenStreetMap ou d'autres fonds cartographiques (par ex. GoogleMaps, BingMaps, etc.) !

3. Mise à jour de la carte

3.1. Mise à jour des fichiers RCR_carto.xls et Localites.shp

La première étape de mise à jour consiste à mettre à jour les fichiers de base qui contiennent toute l'information sur les alternatives (GAC, RES, SEL, …) pour la cartographie. Ces fichiers sont donc cruciaux et doivent être mis à jour avec le plus grand soin. Il est utile aussi de garder les anciennes versions comme archives, en les nommant par la date du jour comme « RCR_carto_YYYYMMDD.xls ». En effet, une alternative pourrait disparaître momentanément puis réapparaitre.

Par « alternatives », nous entendons les 6 catégories qui s'affichent sur la carte, à savoir :

1. GAC
2. RES
3. SEL
4. Donneries
5. RepairCafe
6. Potagers communautaires

Mais de nouvelles alternatives pourraient être ajoutés à l'avenir.

Il y a deux fichiers de base :

1. RCR_carto.xls : le tableau de toutes les alternatives avec toutes les informations. Ce fichier sert de référence pour l'ASBL RCR et, pour la carto, il sert à stocker toutes les informations affichées dans les popups.
2. Localites.shp : Fichier d'information géographique (shapefile) qui contient les coordonnées de toutes les localités de Wallonie-Bruxelles ainsi que la présence ou non des 6 alternatives. Ce fichier sert pour la carto à localiser les alternatives sur la carte, ainsi qu'à générer la liste à gauche de la carte.

a) Mise à jour du fichier RCR_carto.xls

Ce fichier contient 6 feuilles (GAC, RES, SEL, ...) pour chaque alternative. Dans chaque feuille, la colonne 'F' intitulée « Popups » contient tous les commentaires pour chaque alternative, située dans une localité. Les autres colonnes contiennent l'information sur la localités, le nom du groupement, les personnes de contacts, e-mails, numéros de téléphone, etc.

Voilà la marche à suivre pour la mise à jour :

1. Faire une copie de sauvegarde du fichier RCR_carto.xls comme archive (par ex. RCR_carto12_09_2012.xls
2. Ouvrir le fichier RCR_carto.xls
3. Modification du texte (par ex. pour les popups) : modifier le texte dans les colonnes adéquates.
4. Ajout d'une alternative (rouge → jaune → blanc) : Si la localité de l'alternative existe dans la couche Localites.shp, l'ajouter simplement avec le même nom dans le fichier RCR_carto.xls et remplir les colonnes du tableau. Il n'est pas nécessaire de remplir chaque colonne, sauf celles de

'Localites sans accent', 'Nom' et 'Popups'. Si la localité de l'alternative n'existe pas dans la couche Localites.shp, il faut d'abord l'éditer dans cette couche (voir la section suivante)

5. Suppression d'une alternative (brun → bleu → blanc) : Supprimer la ligne dans le fichier RCR_carto.xls
6. Sauvegarder le fichier RCR_carto.xls

Que faire si une localité comprend plusieurs alternatives ?

Dans une version précédente, lorsqu'il y avait plusieurs alternatives par localités, il fallait reprendre le texte du champ « Popups » de toutes les alternatives de la localité dans une nouvelle ligne. **Ce n'est plus nécessaire depuis janvier 2014.** Cela facilite donc le comptage des alternatives, qui est simplement le numéro de ligne de la dernière alternative du fichier RCR_carto.xls.

Code des couleurs :

A chaque modification du fichier RCR_carto.xls, on prend le code de couleur suivant :

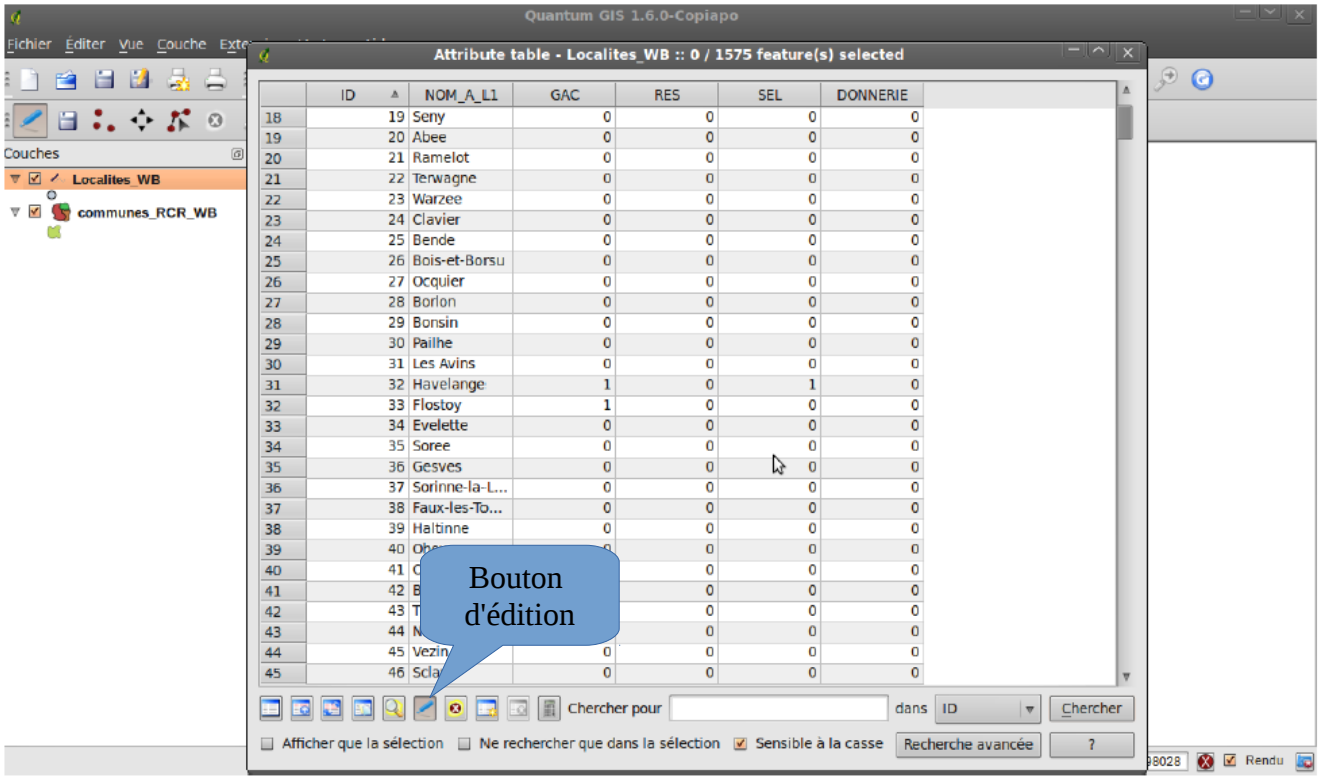
- Ajout d'une alternative : **rouge → jaune → blanc** : un ajout d'une alternative se fait sur un fond **rouge**. Lorsque cette alternative est mise sur la carte web, elle est mise en **jaune**. Lorsque cette alternative est mise sur la carte papier, elle est remise en **blanc**.
- Suppression d'une alternative :**brun → bleu** : une suppression d'une alternative se fait en la mettant en **brun**. Lorsque cette alternative est supprimée sur la carte web, elle est mise en **bleu**. Lorsque cette alternative est mise sur la carte papier, elle est **totalement supprimée**.

b) Mise à jour du fichier Localites.shp

Les informations géographiques sont codées dans un fichier « shapefile» avec l'extension .SHP. Ce type de fichier peut être créé et modifié dans un SIG, par ex : QGIS. Le fichiers Localites.shp contient les coordonnées de toutes les localités de Wallonie-Bruxelles et 6 colonnes correpondant à chaque alternative. Pour chaque localité qui « possède » une alternative, il faut mettre un « 1 » dans la colonne correspondante.

Voici la marche à suivre pour la mise à jour avec QGIS :

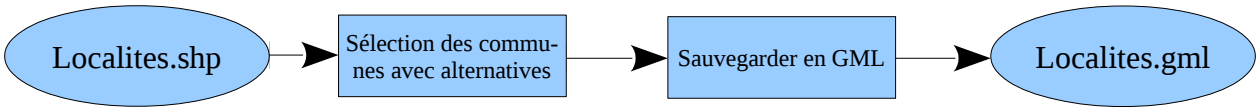
1. Ouvrir QGIS
2. Ajouter le fichier « Localites.shp » en spécifiant UTF-8 comme codage. (Couche > Ajouter une couche vecteur)
3. Ouvrir la table d'attributs (clic droit sur la couche > Ouvrir la table d'attributs)
4. Ouvrir le mode « Edition » en cliquant sur l'icône d'édition en bas de la table (voir figure)
5. Mettre un « 1 » dans la colonne correspondante lorsqu'une alternative est présente dans la localité.
6. Pour sauvegarder les données, re-cliquer sur l'icône « Edition » et sauvegarder les modifications.



NB : Si jamais la localité n'existe pas dans la couche (hameau, …), il y a moyen de créer un nouveau point avec le mode édition et un outil d'édition de couche vectorielle. Cette étape n'est pas expliquée dans ce tutoriel, mais l'édition sous QGIS est assez intuitive et de nombreux tutoriels existent sur le web !

3.2. Mise à jour des fichiers GML

a) Mise à jour du fichier des localités (Localites.gml)



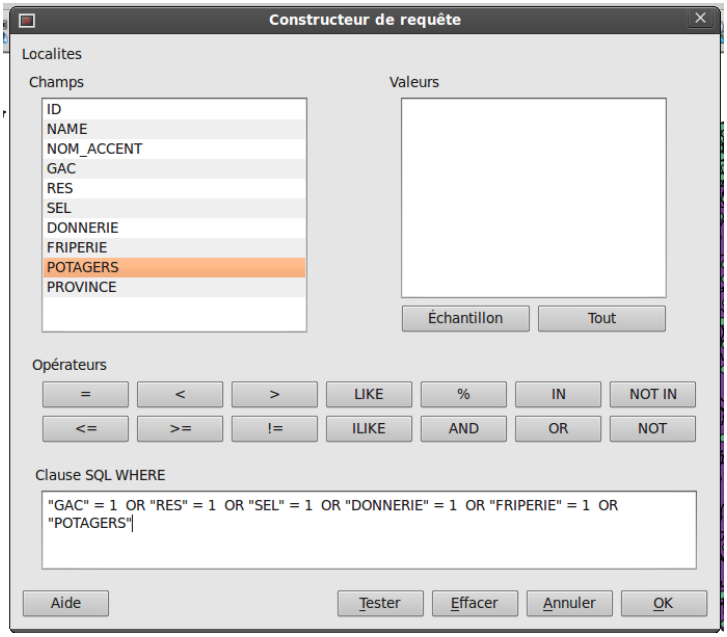
L'étape suivante est d'exporter le fichier « Localites.shp » en « Localites.gml » pour pouvoir l'afficher dans la carte dynamique. Il faut transformer les shapefiles (SHP) dans un format qui est lisible par OpenLayers, le GML, car le SHP n'est pas un format lisible pour des webmaps. Il faut en outre sélectionner uniquement les localités où il y a une alternative afin d'alléger le fichier « Localites.gml » pour la carte[1].

Cette transformation se fait dans QGIS:

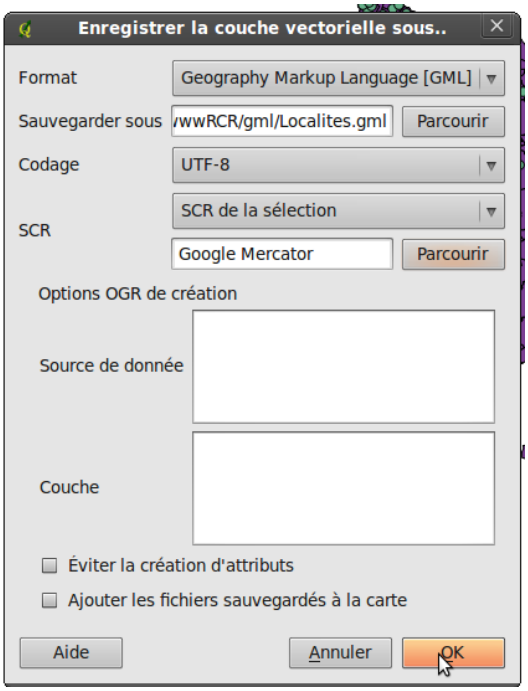
1. Faire un clic droit sur le nom de la couche dans le menu des couches et cliquer sur « Requête »

1 La carte fonctionne tout à fait bien sans cette étape de sélection, mais alors le fichier Localites.gml est très lourd à charger et cela ralentit l'affichage de la carte.

2. Sélectionner les localités avec la requête suivante (copier-coller la requête) :
"GAC" = 1 OR "RES" = 1 OR "SEL" = 1 OR "DONNERIE" = 1 OR "REPAIRCAFE" = 1 OR "POTAGERS" = 1



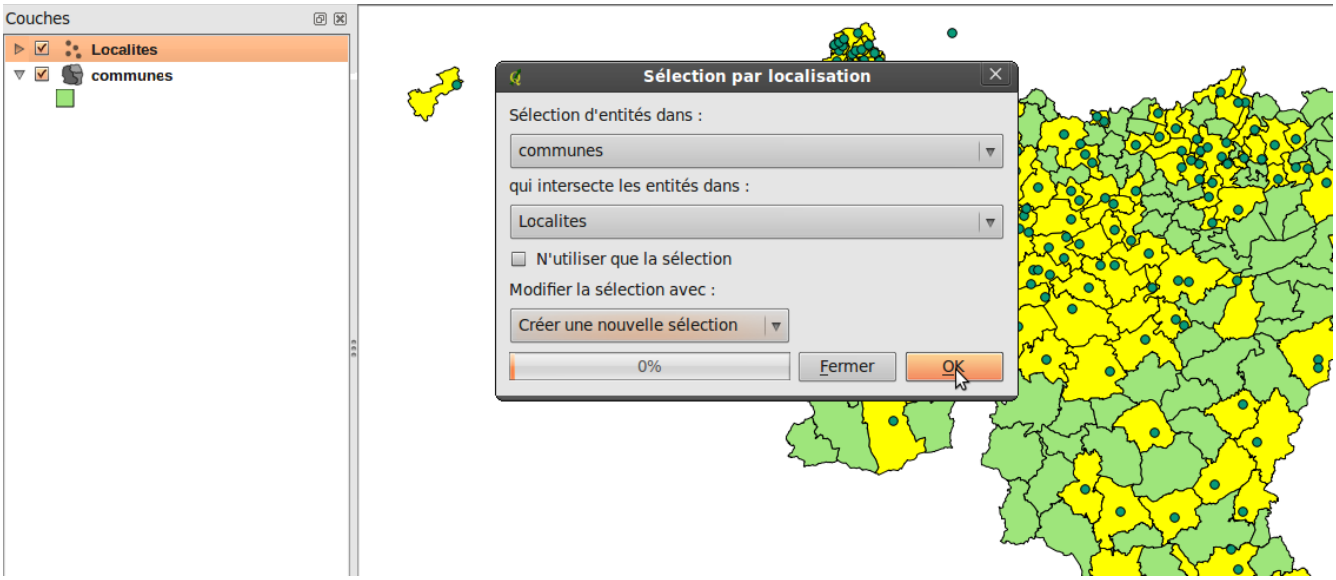
3. Faire un clic droit sur le nom de la couche et choisir « Sauvegarder sous... »
4. Dans la boite de dialogue qui s'affiche, choisissez GML comme format de fichier, UTF-8 comme codage et Google Mercator comme SCR (Système de coordonnées). En changeant le SCR, on va projeter directement la couche dans le même système de coordonnées que la carte (à savoir, Web Mercator, EPSG : 900913 ou 3857).
5. Choisir comme nom de fichier « Localites.gml » et sauver cette couche dans le sous-dossier « gml » dans le dossier « wwwRCR ».



b) Mise à jour du fichier des communes RCR (communes_RCR.gml)

Le but de cette étape est d'afficher les communes qui ont des alternatives (GAC, RES, SEL, …) en grisé par rapport aux autres communes. Pour cela, il faut sélectionner les communes à partir du fichier de base « communes.shp » qui représente toutes les communes de Wallonie-Bruxelles, puis d'exporter le fichier en GML. Cela se fait dans QGIS de la manière suivante :

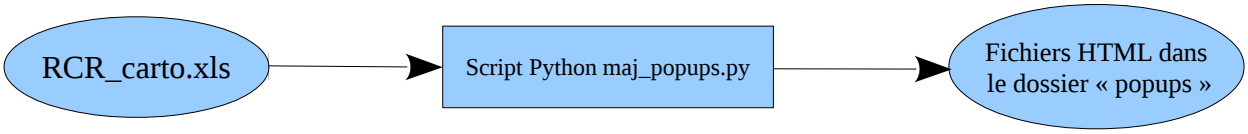
1. Ouvrir QGIS
2. Ajouter les fichiers « Localites.shp » et « communes.shp » en spécifiant UTF-8 comme codage. (Couche > Ajouter une couche vecteur)
3. Si cela n'a pas été fait dans l'étape précédente, sélectionner d'abord les localités qui ont des alternatives en faisant une requête, comme lors de la mise à jour du fichier Localites.gml (voir ci-dessus) :
    1. Faire un clic droit sur la couche « Localites » et cliquer sur « Requête »
    2. Sélectionner les localités avec la requête suivante (copier-coller la requête) :
       "GAC" = 1 OR "RES" = 1 OR "SEL" = 1 OR "DONNERIE" = 1 OR "REPAIRCAFE" = 1 OR "POTAGERS" = 1
4. Les localités sont maintenant sélectionnées. On va sélectionner maintenant les communes qui ont des alternatives grâce à la sélection des localités. Aller dans « Vecteur > Outils de recherche > Sélection par localisation ».
5. Dans la boîte de dialogue qui s'affiche, on sélectionne nos communes qui intersectent les localités sélectionnées.



Il suffit ensuite d'enregistrer la couche en GML, en spécifiant bien qu'on enregistre la sélection uniquement :

1. Faire un clic droit sur la couche « communes.shp »
2. Choisir « Sauvegarder la sélection sous... »
3. Dans la boite de dialogue qui s'affiche, choisir GML comme format de fichier, UTF-8 comme codage et Google Mercator comme SCR (Système de coordonnées). En changeant le SCR, on va projeter directement la couche dans le même système de coordonnées que la carte (à savoir, Google Mercator, EPSG : 900913 ou 3857).
4. Choisir comme nom de fichier communes_RCR.gml et sauver cette couche dans le dossier « gml ».

3.3. Mise à jour des popups



Les popups sont les petites bulles remplies par du texte qui s'ouvrent lorsqu'on clique sur une icône de la carte ou sur une icône de la liste. Le contenu de ces popups, càd le texte, est écrit dans de petits fichiers HTML présents dans le dossier 'popups'. Ces fichiers sont classés dans des sous-dossiers par alternatives (GAC, RES, SEL...). Le nom de ces fichiers HTML doit toujours être le nom de la localité où est présente l'alternative, mais sans accents ni espaces[2]. Les noms de fichier HTML doivent être en fait strictement identique à ceux de la colonne « Localites sans accents » du fichier RCR_carto.xls et du fichier Localites.dbf (présents dans le dossier « Données »)

2 Par exemple, Liège devient Liege, La Louvière devient La_Louviere, Vaux-sur-Sûre devient Vaux-sur-Sure...

Pour changer le texte de ces popups, il suffirait donc d'éditer ces fichiers HTML à l'aide d'un éditeur de texte. Mais pour faciliter la mise à jour, un programme Python permet de générer automatiquement ces petits fichiers HTML à partir du fichier RCR_carto.xls. Il suffit donc de mettre à jour le contenu du fichier RCR_carto.xls et d'exécuter le programme Python pour mettre à jour automatiquement tous les fichiers HTML des popups. Attention, pour que les adresses Internet soient converties en liens dans les popups, il est impératif de les écrire en commençant par « http://www » ou « http:// ». Par exemple, écrire dans RCR_carto.xls « gasap.be » ne rendra pas le lien cliquable, tandis qu'écrire http://www.gasap.be ou http://gasap.be rend les liens effectivement cliquable.

En clair :

1. Exécuter le programme Python map_popup.py en double-cliquant dessus.

Pour que le programme Python fonctionne bien, ne pas modifier la structure du fichier RCR_carto.xls (déplacement de colonnes, ajout de feuilles).

Et pour qu'il fonctionne tout court, il faut bien sûr avoir installé Python[3] sur son ordinateur. Pour cela, télécharger Python sous http://www.python.org/getit/ et suivre les instructions d'installation.

Bug connu : Souvent, le script ne fonctionne pas complètement car certaines cellules sont lues comme des nombres au lieu de texte. Il faut affecter à toutes les cellules du fichier RCR_carto.xls le format « texte ».

3.4. Mise à jour de la liste

Pour faciliter la navigation, une liste des localités comprenant une alternative (GAC, RES, SEL, …) a été constituée à gauche de la carte. Un lien dynamique existe entre cette liste et la carte, par la fonction 'selectlist(couche,localite)'. Cette fonction javascript sélectionne l'élément affiché sur la carte et ouvre automatiquement son popup.

Les lignes HTML qui appellent cette fonction (codées dans map.html) sont générées automatiquement à partir de la table de la couche « Localites.shp » avec OpenOffice Calc (ou MS Office Excel) à l'aide du fichier « maj_liste.ods ». Dans le fichier maj_liste.ods, ces lignes HTML ont été construites en fonction des informations pour chaque localité. S'il y a une modification apporté dans le fichier « Localites.shp » (par ex. nouveau GAC, nouvelle localité avec quelque chose, etc.), il faut mettre à jour également le fichier « maj_liste.ods ».

Voilà la démarche à suivre pour la mise à jour :

1. Dans QGIS, ajouter le fichier « Localites.shp »
2. Sélectionner uniquement les localités où il y a une alternative (voir PLUS HAUT)
3. Après la sélection, enregistrer le fichier (clic droit sur la couche > Sauvegarder sous...) en choisissant « Valeurs séparés par une virgule » comme format et « UTF-8 » comme codage.
4. Le fichier enregistré peut ensuite être ouvert avec OpenOffice Calc (ou MS Office Excel)
5. Trier les lignes du fichier par ordre alphabétique pour les noms des localités, puis pour les provinces .

3 Voir « Logiciels utilisés » pour une description sommaire de Python et son installation.

6. Copier tout le contenu du fichier .
7. Ouvrir le fichier « maj_liste.ods » .
8. Coller ce contenu à la place des 10 premières colonnes du fichier « maj_liste.ods » .
9. La dernière colonne de « maj_liste.ods » intitulée « TexteFinal » contient alors toutes les lignes HTML générées automatiquement. Copier-coller ces lignes de code HTML aux bons endroits (classement par province) dans le fichier « map_list.html » et « map_iframe.html ».

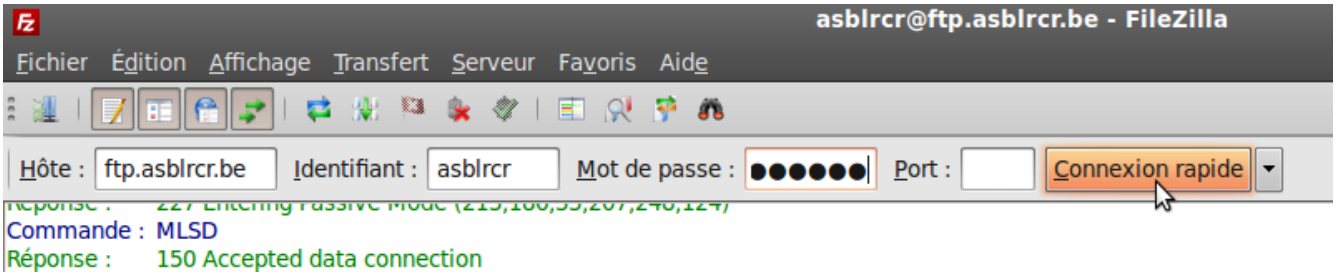
[Python pour automatiser la procédure? : voir http://stackoverflow.com/questions/9125465/how-to-sort-xls-file-column-wise-and-write-it-to-another-file-with-entire-row-us]

3.5. Transfert des fichiers sur le site web

Lorsque toutes les modifications sont terminées, il faut mettre à jour les fichiers (fichiers HTML, dossier gml, dossier popups) sur le serveur web du site www.asblrcr.be. Pour transférer les fichiers mis à jour vers le serveur, il faut utiliser un programme de transfert de fichiers FTP, comme par exemple FileZilla (http://filezilla.fr/).

Pour pouvoir modifier le site, ouvrir FileZilla et entrer les informations de connexion, à savoir :

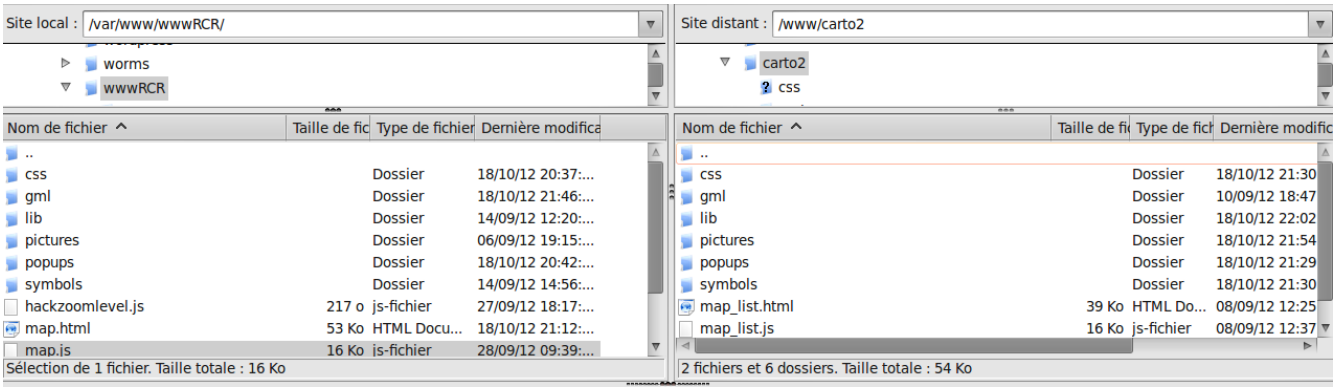
- Hôte : ftp.asblrcr.be
- Identifiant: asblrcr
- Mot de passe : *********



Les fichiers mis à jour doivent être tous transférés dans le dossier de la carte sur le serveur. Ces fichiers sont donc :

- localites.gml (mise à jour des points)
- communes.gml (mise à jour des communes)
- map.html (mise à jour de la liste)
- dossier des popups (mise à jour des popups)

Il suffit de les glisser depuis l'ordinateur vers le serveur comme indiqué dans la figure ci-dessous. Tout ce qui concerne la carte se trouve dans le dossier «carto» (dans le dossier www). Attention de ne pas faire de mauvaise manipulation qui pourrait endommager le site ! Bien vérifier avant tout transfert de fichier que la page HTML de la cartographie fonctionne bien en local.



ATTENTION, NE PAS OUBLIER aussi de mettre la dernière version du fichier RCR_carto.xls sur le Drive du RCR et de le nommer avec la date du jour ! Lors de cette manipulation, les anciens fichiers RCR_carto.xls sont précieusement conservés dans un dossier Archives du Drive.

Enfin, la date de la mise à jour est également notée en bas de la carte sur la page « Alternatives près de chez vous » du site asblrcr.be

Généralement, la mise à jour de la carto est communiquée sur les réseaux sociaux et/ou sur la page d'accueil du site asblrcr.be.