

CREATE AND USE AN ICON FONT FOR MAPPING: THE USE CASE OF 3W MAPS

CREATION OF A CUSTOM ICON FONT ON IcoMOON APP (TUTORIAL 1) AND ITS USE ON QGIS AND ARCGIS PRO (TUTORIAL 2)

The 3W (Who does What, Where) or 4W (Who does What, Where, When) maps are well-known maps in the development and humanitarian sector. They are often produced by coordination or governmental stakeholders on a national or provincial scale and they aim to provide a global view of development and humanitarian actors and/or projects in a given territory. Some organizations also turn to this type of maps for less operational purposes, such as better communicating about their projects in the field.

Considering their GIS expectations and capacities, such organizations usually use the QGIS software in the field. This is why the use of advanced labels and icon fonts as presented in this technical guidance guarantees an adapted workflow for CSOs/NGOs. The workflow is also explained with ArcGIS pro software for organizations which are already equipped.

This resource compiled in the second half of 2020 is composed of 2 distinct tutorials:

1. A first tutorial that one can use to learn how to create an Icon Font using the IcoMoon App
2. A second one that will help the reader create 3W maps with icon font using either QGIS or ArcGIS Pro

I. Tutorial 1: Create an Icon Font using IcoMoon App	3
I.1. Presentation of the interface	4
I.2. Select icons for your new font	5
I.3. Personalize your icons and font	6
I.4. Export and install the font	7
II. Tutorial 2: Create 3W maps with icon font	9
II.1. Manage data for 3W maps: how to shape your data?	10
II.2. Use the icon font on QGIS	11
II.2.A. Import and join CSV file	11
II.2.B. Adding the labels.....	12
II.2.C. Advanced customization	17
II.3. Use the icon font on ArcGIS Pro	18
II.3.A. Import and join CSV file	18
II.3.B. Placing the labels	19
II.4. Conclusion	21
II.5. Appendices	22
II.5.A. Appendix 1: ArcGIS text formatting tags	22

This publication is supported by the French Development Agency (AFD). Nevertheless, the ideas and opinions presented in this document do not necessarily represent those of AFD.

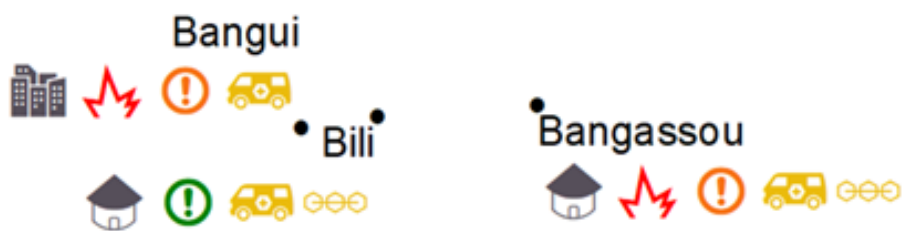
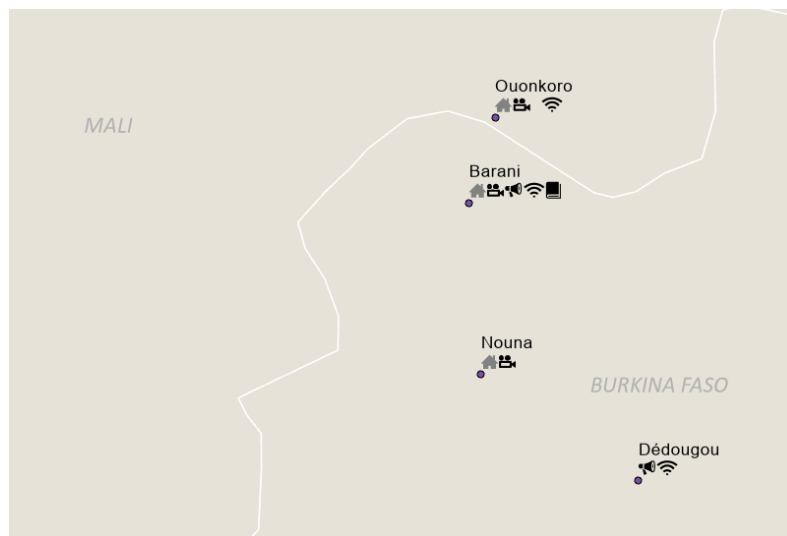


I. Tutorial 1: Create an Icon Font using IcoMoon App

Customized icon fonts ease implementation of different icons in GIS software.

Icon font can be useful in GIS for symbology but also to create activity mapping and 3W maps (Who does What, Where). It allows to automatically display icons on GIS Software (QGIS, ArcMap or ArcGISPro) and make easy updated maps.

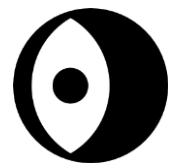
Here are examples of icon font in maps.



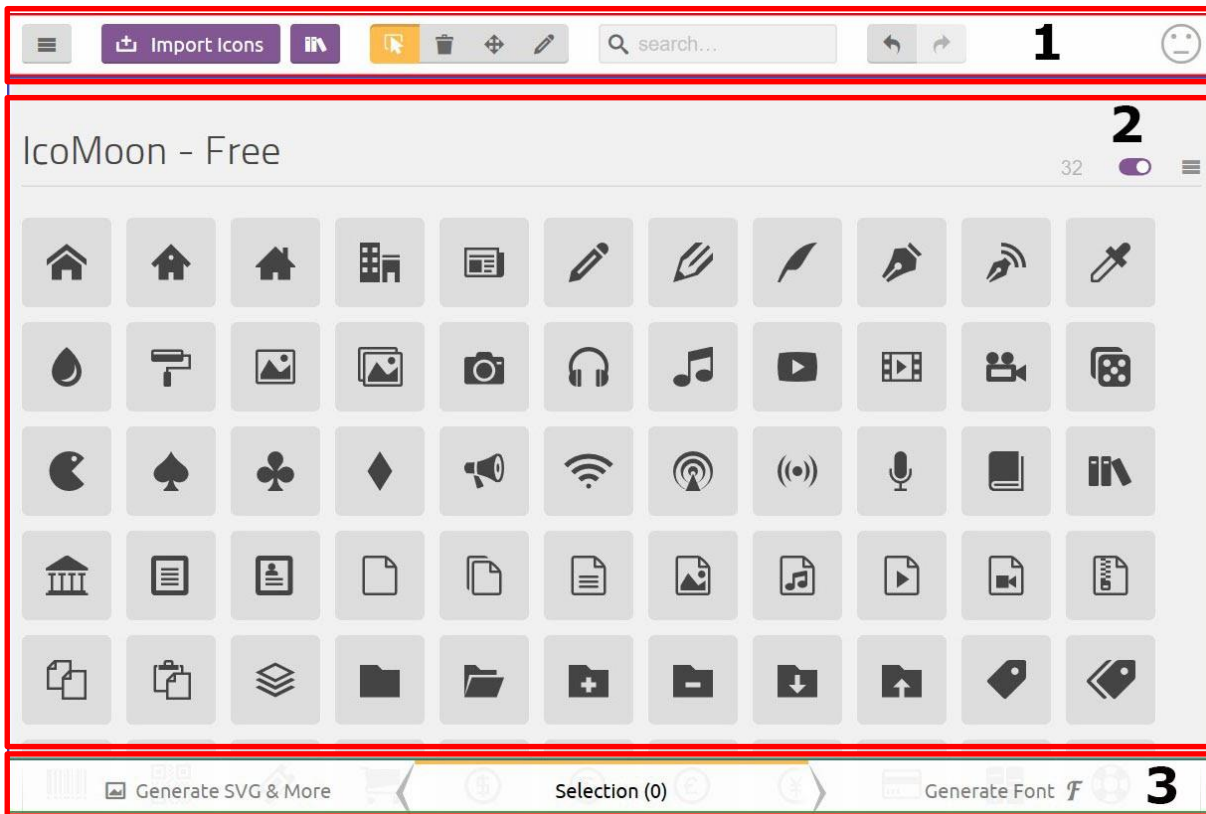
We will use the IcoMoon App website to set up a custom icon font:

<https://icomoon.io/app/#/select>

It is a free icon font generator, documentation available [here](#) for further info.



I.1. Presentation of the interface



1. The top-toolbar:

- This is where you can have access to the menu, import icons, see icons library, search for icons, etc.

2. Centre:

- This is where the icons are displayed, where you can navigate and select icons.

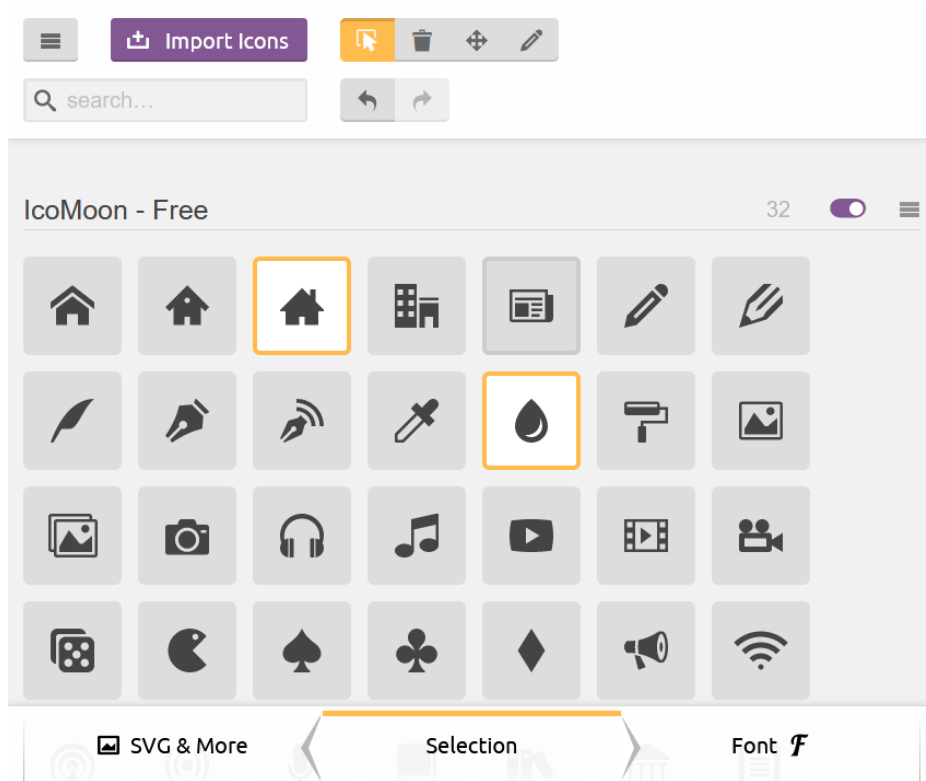
3. Footer:

- You can switch here between 3 panels:
 - On the left, the import and modification of svg icons,
 - On the centre, the selection panel,
 - On the right, the panel to generate your font.

I.2. Select icons for your new font

- **Select icons**

To select icons, you simply have to click on icons, and a yellow stroke will indicate the selected icons. You can use the search bar to see other icons, from different icons library available on the website.



Note: if you use the research bar, the app will automatically search in all packs, including buyable packs.

To improve icon numbers, you can select another open library, such as material design icons with the 'icon library' button on the top bar.

- **Import icons**

If you miss some icons, you can try external resources to find svg format, and then import it. You can find icons on the following websites:

- <https://www.flaticon.com/>
- <https://www.freeicons.io>
- <https://thenounproject.com/>

Note: Always check licences and credits needed to use the icon, and import icons with one colour only. Also don't forget to add the related credits on the maps whenever applicable.

I.3. Personalize your icons and font

Once you have selected or imported all the icons you need, you will need to be sure that your font is harmonized to avoid major differences between your icons. Such as size or forms for example.

IcoMoon allow advanced customization of the icons.

- **Modify icons**

You can use the edit button on the top bar to edit icons directly on the selection panel.



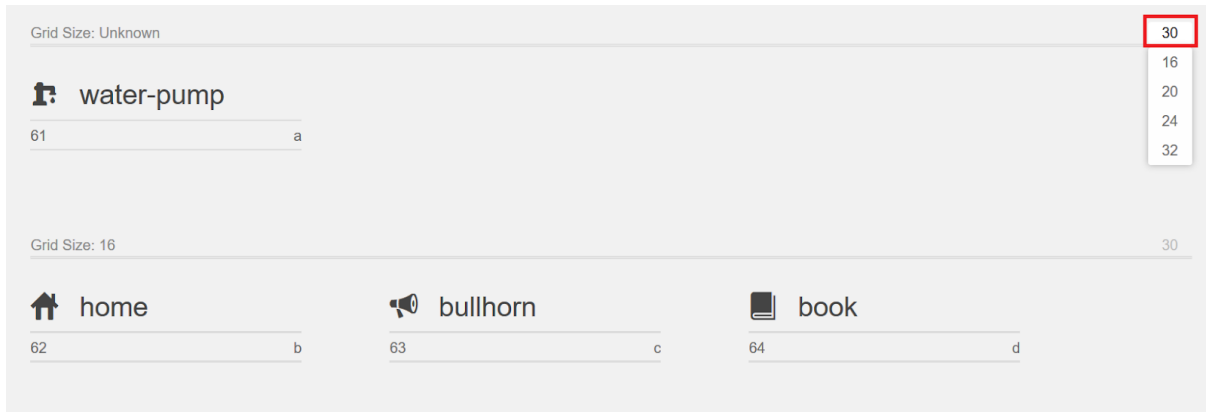
Once you select an icon, you will be able to change alignment of the icon, its size, or duplicate the icon.

> **Note:** All the modifications you do will be saved on the selection panel, but you can duplicate the icon before any changes to avoid any problems.

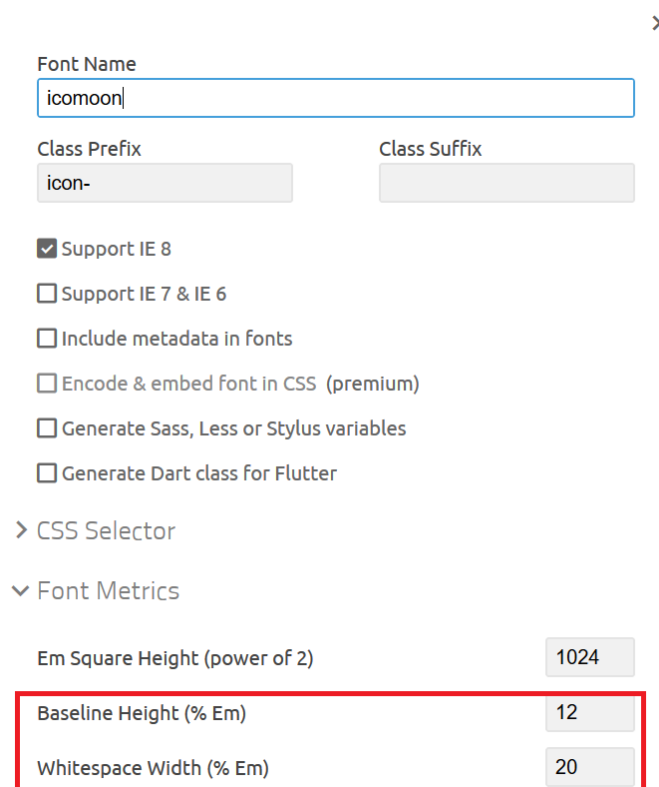
- **Harmonize fonts**

When all your icons are selected, you can go on the right panel '*Generate Font*'

You can see all the icons selected and manage the size of each of them. In order to have a harmonized size among your selected icons, put the same grid size, you can change it on the top right of each group of label.



Finally, to check text alignment, go on the 'preferences' button on the top bar



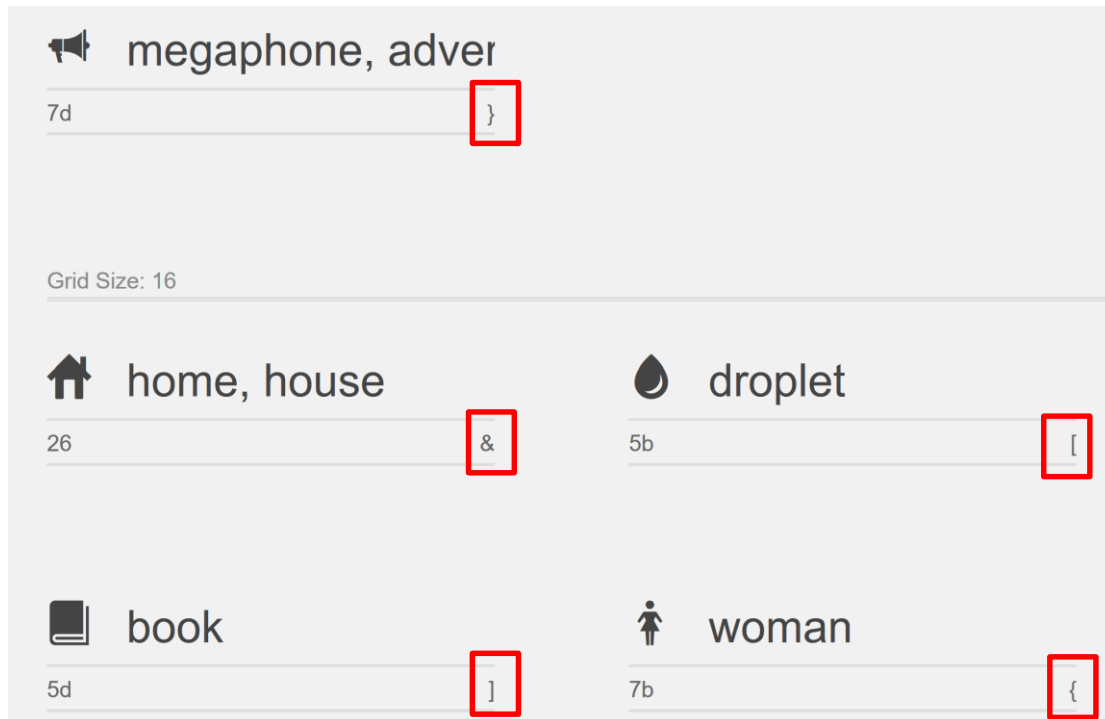
You can play with the Baseline Height and the Whitespace Width to change alignment with the text, which you can see with the text following each icon.

I.4. Export and install the font

- **Assign keyboard key to icons**

You can assign any keyboard keys to each icons clicking on the bottom-right letter of each icon.

Note: Remember that you may need some letters for you labelling, you can use special characters to be sure that you will not have any problems using common letters.



`&[{}]` = `&[{}]`

- **Name your font**

Remember to name your font in the preferences on the export panel.

- **Download the font**

Now that your font is ready, you only need to download it with the button on the bottom right of your screen.

You can now share it and install it on your computer.

Note: Be sure to install the font as the administrator on your computer to allow the font for all the applications (e.g. QGIS, ArcGIS, etc.).

You are ready to use your customized icon-font!

II. Tutorial 2: Create 3W maps with icon font

The “**Who does What, Where**” map, or **3W Map**, is a type of maps often used in the development and humanitarian sector. Similarly, **4W Maps: “Who does What Where and When**” may also be found. These maps provide information on WHO (which organizations) are doing WHAT (which activities), WHERE (in which locations) to enable national and local authorities, donors, NGOs and other organization to improve the implementation of projects so as to ensure that humanitarian, development and peace-focused needs are met.

What can we find on these maps?

- Partners name
- Project Name / Type (i.e. sector of intervention)
- Start/End Date of projects
- Number of projects, dedicated amount of money, number of people reached, etc.

These maps are great use cases for icon-fonts and advanced labelling in GIS software. We are going to implement these maps on both QGIS and ArcGis Pro.

Labels and advanced labelling use a different language in each software, implicating some differences in the making of a 3W Map.

Criteria	QGIS	ArcGis Pro
Multicolour labels	No	Yes
Many labelling languages	No	Yes
Easy updates	Yes	Yes
Stack labels	Yes	Yes
Managing <i>Null / Empty data</i>	Yes	No

II.1. Manage data for 3W maps: how to shape your data?

&[{}]} = &[{}]

With the example of the icon font created in the first tutorial concerning icon fonts (link), we can imagine data to create our 3W Map.

For each locality:

- Name of projects
- Number of projects for each sector (e.g. water, shelter, communication, etc.)
- The icons related to the type of project, if there is such project in the locality, linked with our icon font (water_binary, shelter_binary, communication_binary).

ID	Name	Project_name_1	Project_name_2	Project_name_3	water	shelter
1	Place 1	AAA	CCD		12	4
2	Place 2	BBB	ABC	CCC		10
3	Place 3	BBC			8	

communication	water_binary	shelter_binary	communication_binary
9	[&	}
		&	
4	[}


water_nb	shelter_nb	communication_nb
5.532134	7.891432	2.42435
	1.02498	
1.72197		2.81453

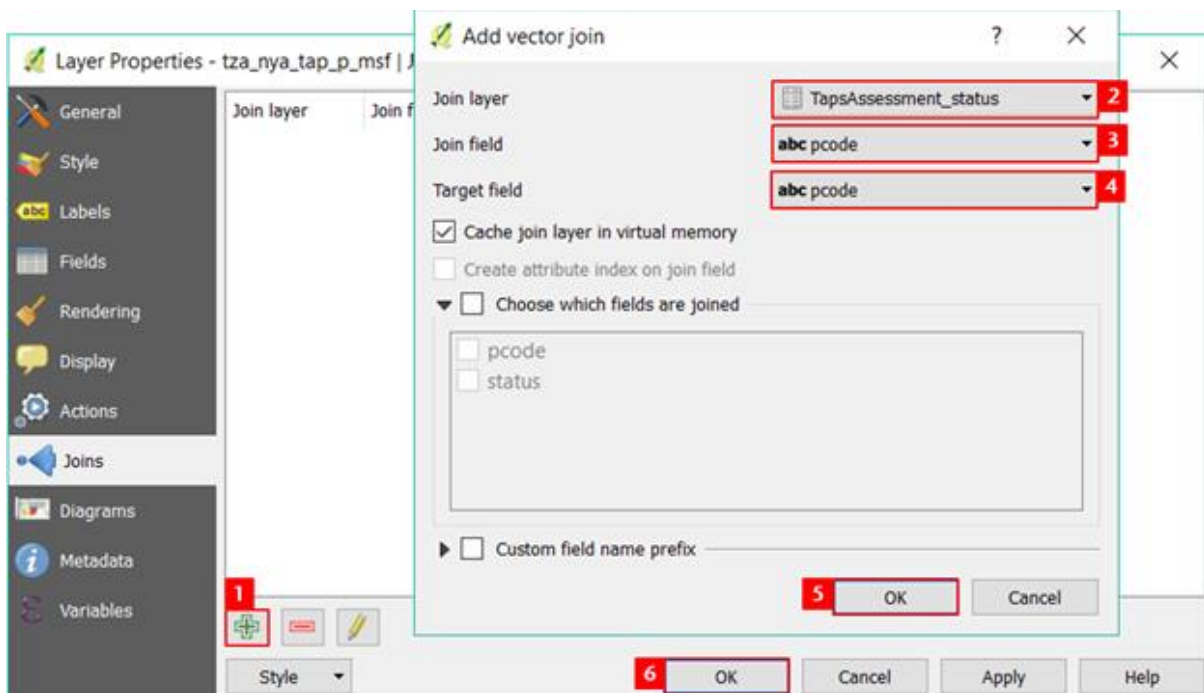
II.2. Use the icon font on QGIS

⚠️ QGIS 3.10 was used for this demo, be sure to have a version higher than 3.0.

II.2.A. Import and join CSV file

Import the data table shaped in the previous part in csv and join it with the associated layers having geometries (shapefile, geojson). Here, we will use an admin layer.

In order to join the csv file, you need to go into the **Layer Properties window and go into the Joins tab.** 



Then follow these 6 steps:

1. **Click on the green + button** > a window will open
2. **Indicate the table you want to link to this layer**
3. **Indicate the field of the table which will be used as merging column**
4. **Indicate the field of the layer which will be used as merging column**
5. **Click on OK** in *both* windows

II.2.B. Adding the labels

1- Understand NULL vs empty values management

As any software, QGIS can both have NULL or empty values in each cells.

NULL means that the value does not exist.

Empty means that the value exists but is empty.

Sous-colline	nb_tot	nb_1
Kinyeregete	139	10
	NULL	NULL
Kirimba	467	10

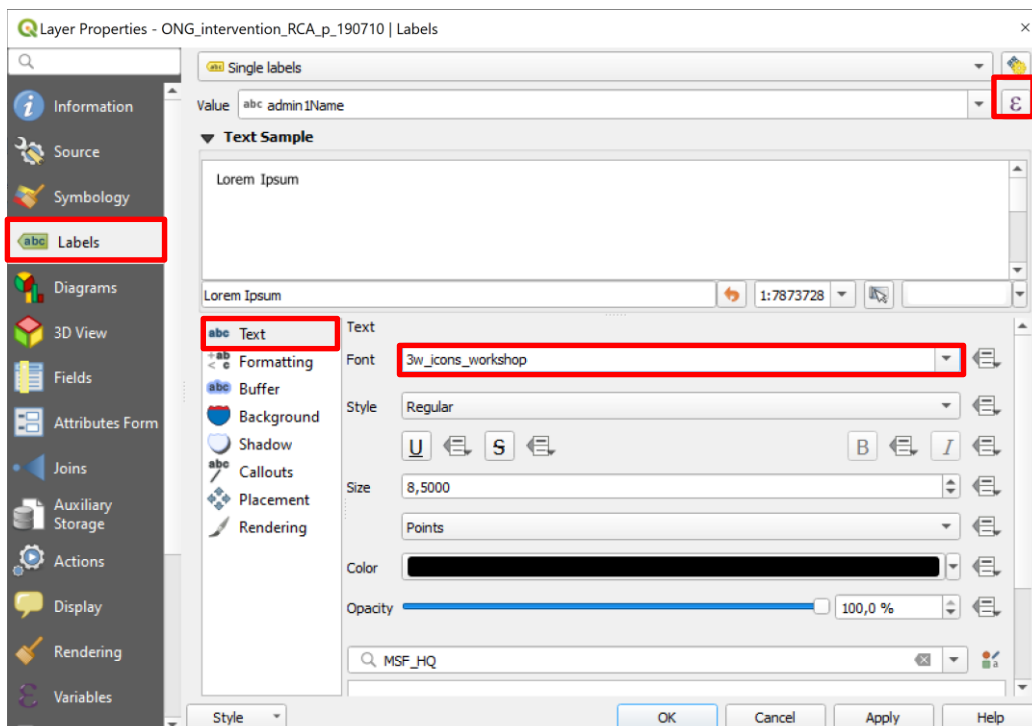
Here we have the difference in displaying between an *empty cell* for the "**Sous-colline**" column and *NULL* values for the **other columns** in the layer attribute table.

2- Display advanced labels using expressions

Reminder: Set an expression for a label

In the **Properties window** go to the **Labels tab** > **select Show labels for this layer from the drop-down menu** > **select the field that will be used as a label** > **change all the parameters you want** to achieve the desired result.

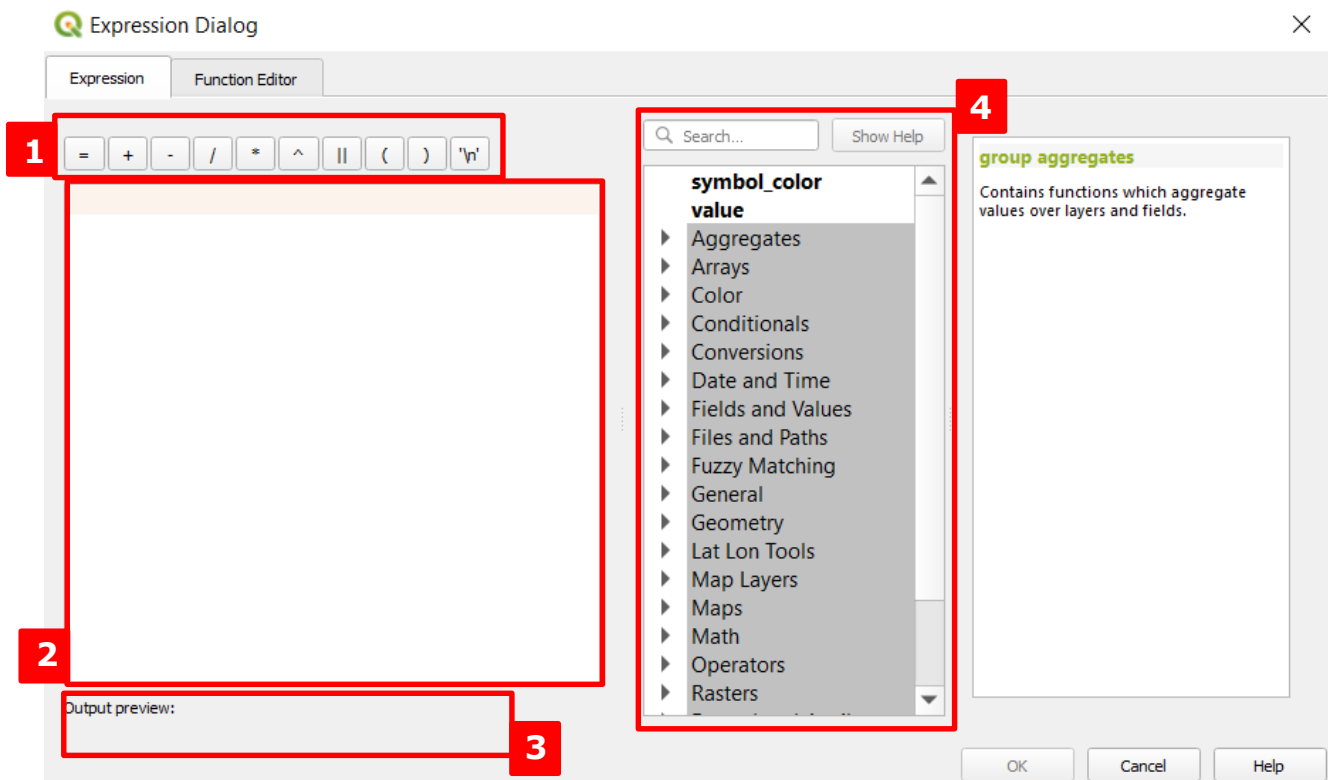
 Do not forget to select your icon font in the Text parameters



Modify the label expression to integrate the icons and the associated values.

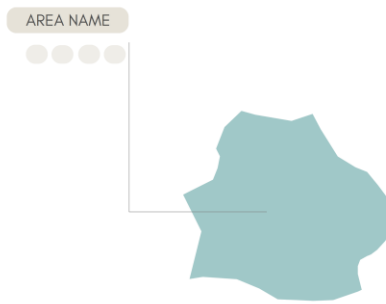
Basic Vocabulary:

- `\n`: Line break
- `||`: Concatenation
- `+` or `*` or `/` : arithmetic operators
- `""` : Insert a fields "fieldname"
- `'` : insert a string 'string'



1. Basics operators
2. Expression edition part
3. Preview
4. Function management

3- Icon labels

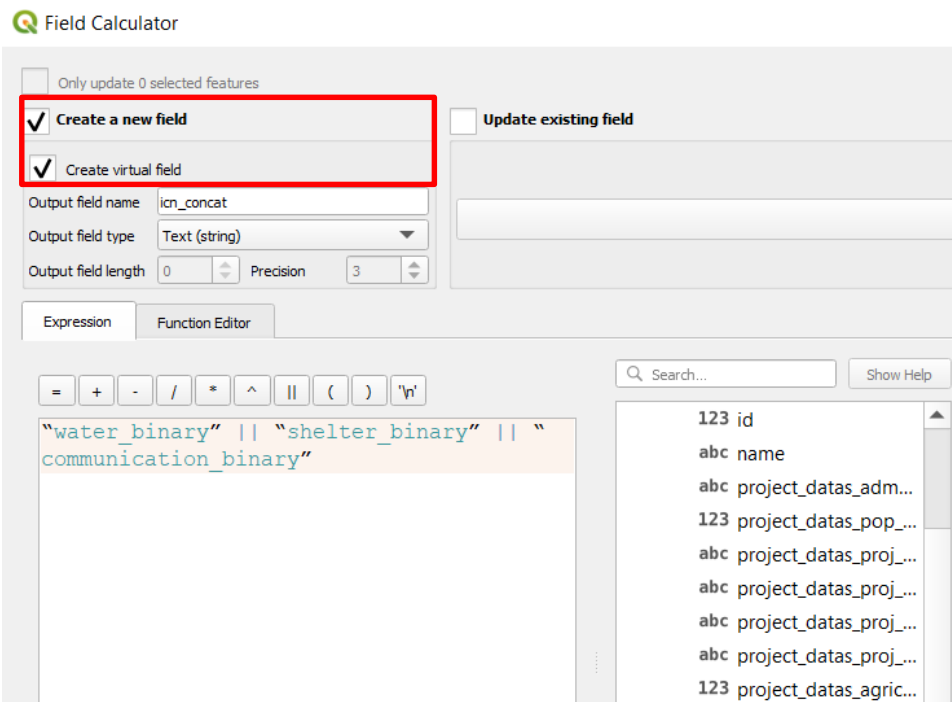


We would like to have the following display on our labels, with name of the area and icons of each projects in the area.

a. *Create a virtual field to concatenate icon value (symbols of our fonts) into one string only*



1. Use the open field calculator
2. Create a virtual field in the field calculator with expression to concatenate icons values



Our expression would be used to concatenate each icon value (a,b,c, etc.):

Expression: "water_binary" || "shelter_binary" || "communication_binary"

Result: [&}

b. Insert the field inside the label expression

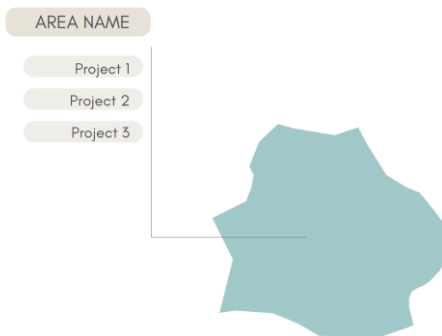
Do not forget to select your font

Expression: "name" || '\n' || 'icon_concatenate'

Result:



4- Text Labels



We would like to have the following display on our labels, with the name of the area and each project/partners name on line.

a. Display it with the expression

Expression: "name" || '\n' || "project_name_1" || '\n' || "project_name_2" || '\n' || "project_name_3"

Result:



⚠ Problem: If any null value is encountered, the string result of the expression will be null.

b. Manage null values

We can manage null values with the **coalesce()** function in QGIS

This function returns the first non-NULL value from the expression list, which allows null values management.

Example: `coalesce('null' , 'null', 'string3')` results: `'string3'`

This way by adding an empty string in our expression, it will return an empty value and not a null value, allowing to display our label.

Expression: `"name" || coalesce ('\n' || "project_name_1", '') || coalesce ('\n' || "project_name_3" , '') || coalesce ('\n' || "project_name_3" , '')`

Result:

```
SANTA YEOGAT
AAA
BBB
CCC
DDD
```

Knowing this, we can now easily display advanced labels, even if some values are null, and it will automatically display it as we want.

5- Text & Icons labels

Using the `coalesce` function we can have **advanced labels** like the following example.

Example:

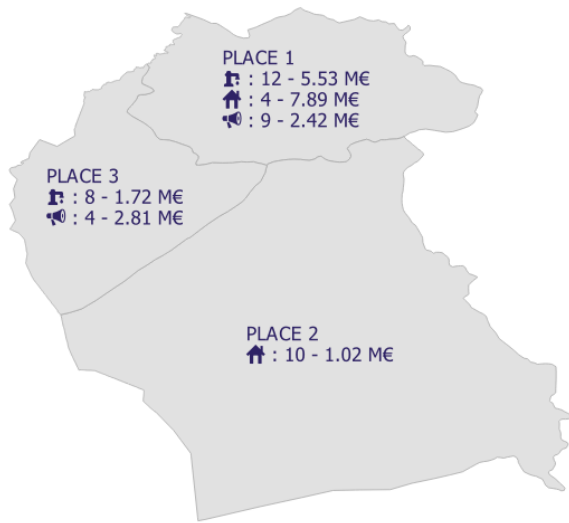
```
coalesce( "Fichier_Carto_Nom" ,'' ) ||
```

```
coalesce( '\n' || "Fichier_Carto_water_binary", "" ) || coalesce( ' : ' || "Fichier_Carto_water", "" ) ||
coalesce( ' - ' || (round( "Fichier_Carto_water_nb",-4 )/1000000) || ' M€', "" ) ||
```

```
coalesce( '\n' || "Fichier_Carto_shelter_binary" ,'' ) || coalesce( ' : ' ||
"Fichier_Carto_shelter", "" ) || coalesce( ' - ' || (round( "Fichier_Carto_shelter_nb",-4
)/1000000) || ' M€', "" ) ||
```

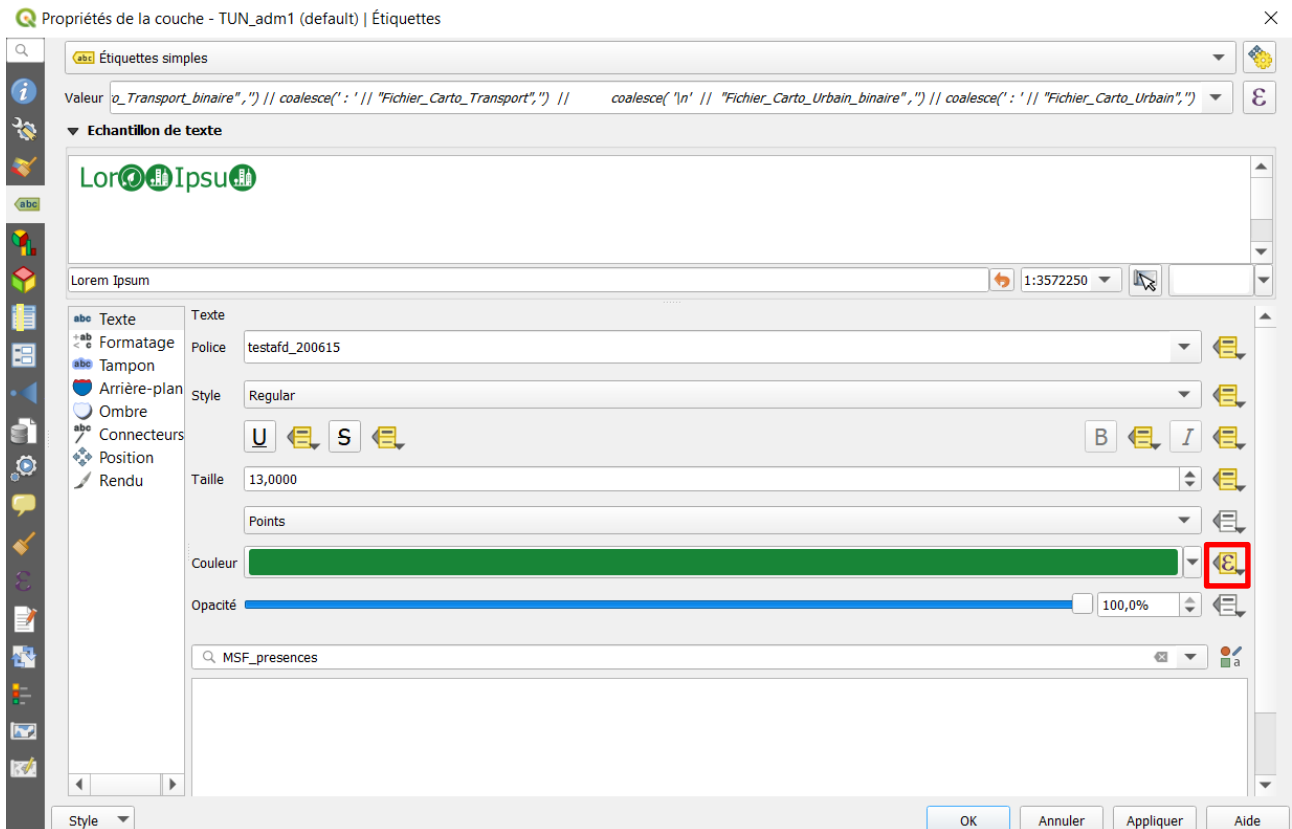
```
coalesce( '\n' || "Fichier_Carto_communication_binary" ,'' ) || coalesce( ' : ' ||
"Fichier_Carto_communication", "" ) || coalesce( ' - ' || (round(
"Fichier_Carto_communication_nb",-4 )/1000000) || ' M€', "" ) ||
```


Result:



II.2.C. Advanced customization

As explained in the table above, QGIS field expression language does not allow to modify colour for a part of the label (e.g. a colour for each icon), a big limitation for graphic styling of the label. You can change the colour of label using an expression in the colour of the label. This allow to display different colours for the all label, but this is not adapted to our data structure of 3W Map.



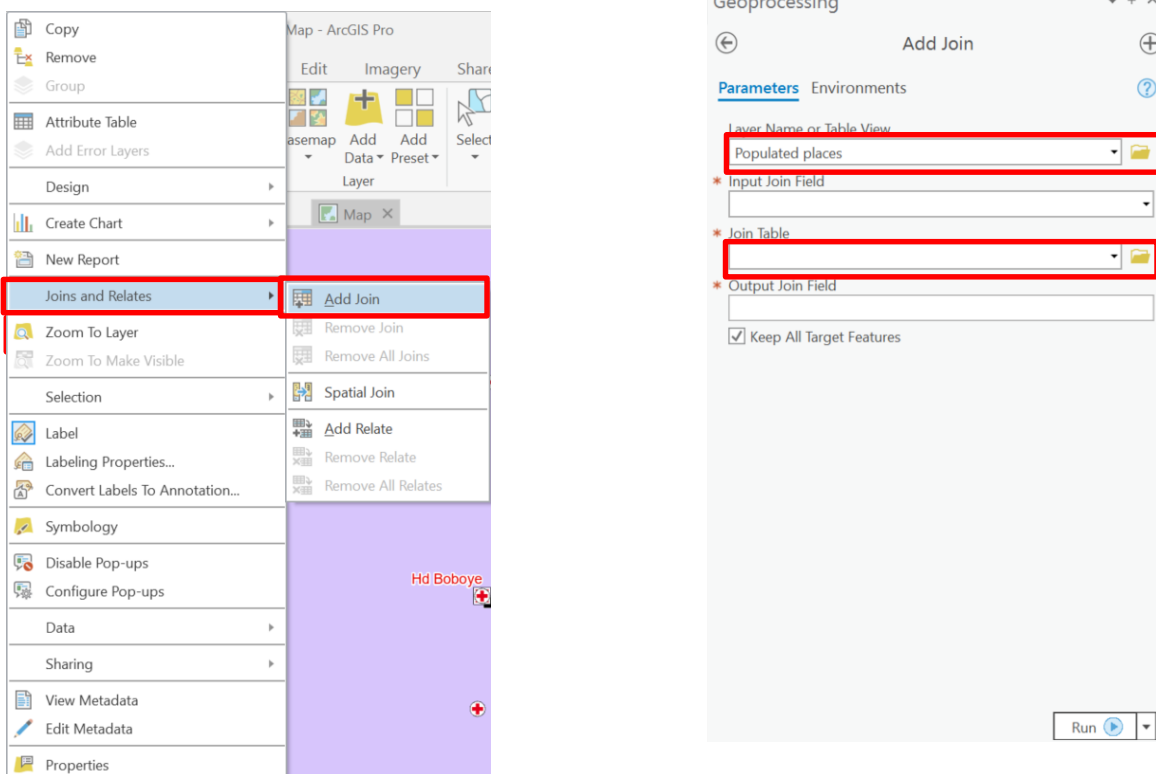
II.3. Use the icon font on ArcGIS Pro

 ArcGIS Pro 2.6.1 was used for this part, any version of ArcGIS Pro will work.

II.3.A. Import and join CSV file

After importing the geodata layer and the table (csv file), right click on the geodata layer and select **Joins and Relates > Add Join**.

Select your geodata layer, and click on Label, and then open the Expression manager.



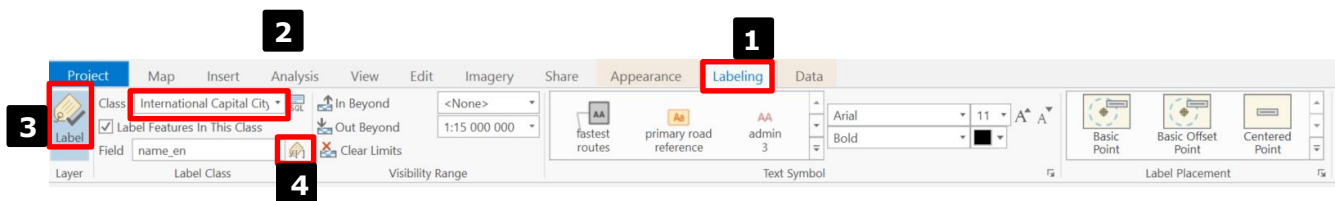
1. Right click on your geodata layer > **Join and relates > Add Join**
2. Fill the parameter with the **table to join** and the **ID field to join data**
3. **Click on "Run"**

II.3.B. Placing the labels

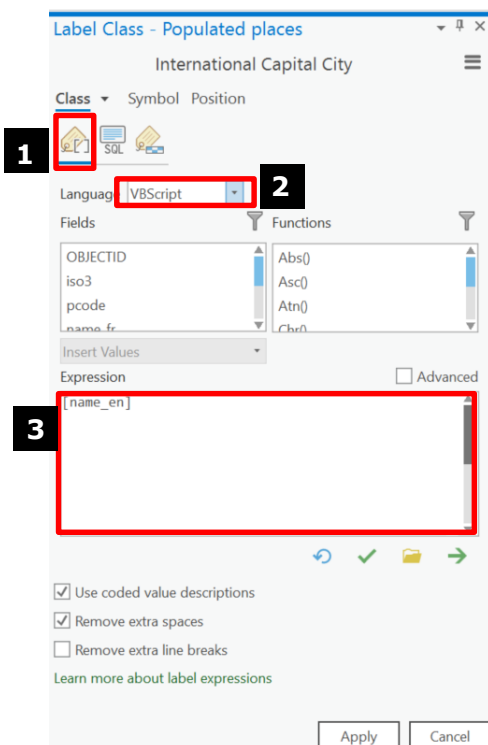
Now that we have our data ready, we can create our labels with custom expression in ArcGIS Pro. ArcGIS Pro propose **4 languages** for field expressions:

- **Arcade**: designed by Esri specifically for creating custom visualizations and labeling expressions in the ArcGIS Platform
- **VBScript**: Visual Basic Scripting Edition, created by Microsoft
- **JScript**: Script related to JavaScript
- **Python**: Python language

We will use **VBScript** for our labels, but each language has its own advantages. VBScript allows advanced modification on labels (font, colours, etc.).



1. Go on the **labelling panel** while your layer is selected
2. Choose or create a **label class**
3. Click on the label icon to **render the labels**
4. Click on the **Expression icon** to open the advanced expression panel



1. Go on the **expression panel**
2. Choose the **VBScript language**
3. Fill the area with **your expression**

VBScript Vocabulary:

[Field]: display the data of the field

vbCrLf: line break

"<CLR>" "</CLR>": tag to manage colours

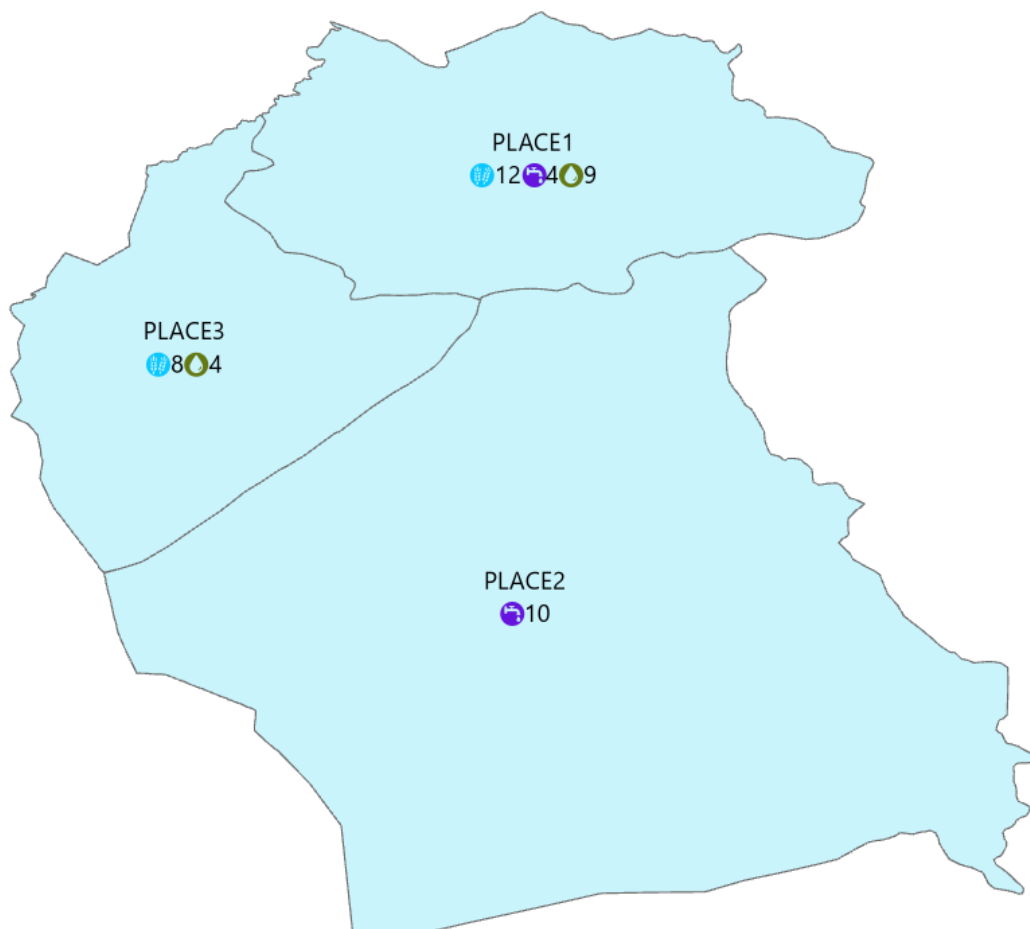
You will find more possibility of customization using VBScript on the Appendices n°1.

Example:

```
[Fichier_Carto2.csv.Gouvernorats] & vbCrLf & "<CLR red='10' green='200' blue='255'>" &
[Fichier_Carto2.csv.Water_binary] & "</CLR>" & [Fichier_Carto2.csv.Water] & "<CLR
red='100' green='20' blue='220'>" & [Fichier_Carto2.csv.Shelter_binary] & "</CLR>" &
[Fichier_Carto2.csv.Shelter]& "<CLR red='100' green='120' blue='20'>" &
[Fichier_Carto2.csv.Communication_binary] & "</CLR>" & [Fichier_Carto2.csv.Communication]
```

```
[Fichier_Carto2.csv.Gouvernorats] + vbnewline & "<CLR red='10' green='200' blue='255'>" &
[Fichier_Carto2.csv.Water_binary] & "</CLR>" & ":" & [Fichier_Carto2.csv.Water] + vbnewline
& "<CLR red='100' green='20' blue='220'>" & [Fichier_Carto2.csv.Shelter_binary] & "</CLR>"
& ":" & [Fichier_Carto2.csv.Shelter] & vbnewline & "<CLR red='100' green='120' blue='20'>"
& [Fichier_Carto2.csv.Communication_binary] & "</CLR>" & ":" &
[Fichier_Carto2.csv.Communication]
```

Result:



II.4. Conclusion

Icon font creation can easily be implemented in QGIS or ArcGIS Pro, allowing you to have original graphical rendering. With a data structure adapted upstream, updating the map can be very simple, with a simple update of the spreadsheet data (xls, csv).

However, one must be aware that each software has its advantages and disadvantages in terms of label expressions. In fact, we sometimes have to adapt to the language and our ability to manipulate it for the creation of our maps.

II.5. Appendices

II.5.A. Appendix 1: ArcGIS text formatting tags

Labels are drawn using the symbol specified in the *Label Manager* or on the **Labels** tab of the *Layer Properties* dialog box. You can modify or override the appearance of this symbol for particular portions of the expression by inserting ArcGIS text formatting tags into the expression as text strings. This lets you create mixed-format labels where, for example, one field in a label is underlined.

The tags that you can use are listed in the table below. Acceptable values for Colour (RGB) are red, green, blue = 0–255, and acceptable values for Colour (CMYK) are cyan, magenta, yellow, black = 0–100; missing colour attributes are assumed to be 0.

Font

"<FNT name='Arial' size='18'>" & [LABELFIELD] & "</FNT>"

"<FNT name='Arial' scale='200'>" & [LABELFIELD] & "</FNT>"

Colour

"<CLR red='255' green='255' blue='255'>" & [LABELFIELD] & "</CLR>"

"<CLR cyan='100' magenta='100' yellow='100' black='100'>" & [LABELFIELD] & "</CLR>"

Bold

"<BOL>" & [LABELFIELD] & "</BOL>"

Italic

"<ITA>" & [LABELFIELD] & "</ITA>"

Underline

"<UND>" & [LABELFIELD] & "</UND>"

All capitals

"<ACP>" & [LABELFIELD] & "</ACP>"

Small capitals

"<SCP>" & [LABELFIELD] & "</SCP>"

Superscript

"^{" & [LABELFIELD] & "}"

Subscript

"_{" & [LABELFIELD] & "}"

Character spacing

(0%=regular) "<CHR spacing='25'>" & [LABELFIELD] & "</CHR>"

Character width

(100%=regular) "<CHR width='150'>" & [LABELFIELD] & "</CHR>"

Word spacing

(100%=regular) "<WRD spacing='150'>" & [LABELFIELD] & "</WRD>"

Line leading (pts)

"<LIN leading='12'>" & [LABELFIELD] & "</LIN>"

Un-Bold

"<_BOL>" & [LABELFIELD] & "</_BOL>"

Un-Italic

"<_ITA>" & [LABELFIELD] & "</_ITA>"

Un-Underline

"<_UND>" & [LABELFIELD] & "</_UND>"

Un-Superscript

"<_SUP>" & [LABELFIELD] & "</_SUP>"

Un-Subscript

"<_SUB>" & [LABELFIELD] & "</_SUB>"

The following syntax rules apply to tags in label expressions:

Formatting tags must be surrounded by double quotes and concatenated to other parts of the expression using the & operator:

"<BOL>" & [LABELFIELD] & "</BOL>"

Tags are not interpreted by VBScript/JScript. Instead, they are passed to the ArcMap

framework as plain text to be dynamically formatted as they are drawn. You don't need to quote tags included inside quoted strings:

```
"Current <BOL>status</BOL> of parcel: " & [LABELFIELD]
```

The ArcMap text formatting tags follow XML syntax rules. Each start tag must be accompanied by an end tag. Tags can be nested, but you must close the inner tag before closing an outer tag:

```
"<BOL><UND>" & [LABELFIELD] & "</UND></BOL>"
```

The case of tag pairs must match exactly. So <BOL>...</BOL> is valid, as is

<bol>..</bol>, but <Bol>...</bol> is invalid.

In label expressions, tag attributes must be surrounded either by single quotes or by two sets of double quotes. The following expressions are equivalent:

```
"<FNT name=""Arial"" size=""18"">" & [LABELFIELD] & "</FNT>"  
"<FNT name='Arial' size='18'>" & [LABELFIELD] & "</FNT>"
```

The special characters & and < are not valid in your text if formatting tags are used. Use the equivalent character codes & and < instead. For example, this expression displays the values of the label field inside < > characters:

```
"<ITA>&lt;" & [LABELFIELD] & "></ITA>"
```

If you have special characters embedded in the values of the label field, you can replace them dynamically using a simple label script:

```
Function FindLabel ([LABELFIELD])  
NewString = Replace([LABELFIELD], "&", "&amp;")  
FindLabel = "<ITA>" & NewString & "</ITA>"  
End Function
```

Formatting tags can be embedded in the values of the field you use to label a layer's features whether or not you use a label expression. In this way, you can change the format of any portion of a particular value in a label field. To embed formatting tags, the label field must be of string type. Tags and tag attributes used in field values do not need to be surrounded by quotes, so the following are valid values for a label field:

```
<ITA>Rochester</ITA>  
<FNT size='18'>C</FNT>Colourado
```

Tags aren't resolved by the ArcMap **table of contents**, **Table** window, or **Identify Results** window, so tags added to field values are visible as tags in those windows.