Working with Data

The City of White Rock aims to publish data in an open format and provide people with easy and reliable access to data, all while reducing barriers to entry, promoting innovation and eliminating unnecessary costs for businesses using the data.

Data is presented in different formats. This document contains information, tips and suggestions on how to use some of the formats presented in the Open Data Portal.

Structured Data

Structured data is usually in the form of text but follows a specific format giving it machine readability. Some more complex formats may be in binary format such as maps and GIS related files. You will find many different formats in the Open Data Portal. This section covers some of the popular formats and provides some tips on how to work with them.

CSV

The easiest method for working with a **Comma-Separated Values (CSV)** file is to use a spreadsheet application such as OpenOffice Calc or Microsoft Excel. Simply launch the application and open the .csv file to launch the application's CSV import wizard.

Character Encoding

One important detail is to pay attention to **Character Encoding** which can affect how the data, mainly the text is parsed during the import. This will be more noticeable when dealing with languages such as French which contain extended characters.

OpenOffice provides easy selection of the character encoding to be used. Figure 1 and Figure 2 illustrate an example where the downloaded CSV dataset file was encoded using a different format than the default UTF–8 for OpenOffice (illustration using NeoOffice).

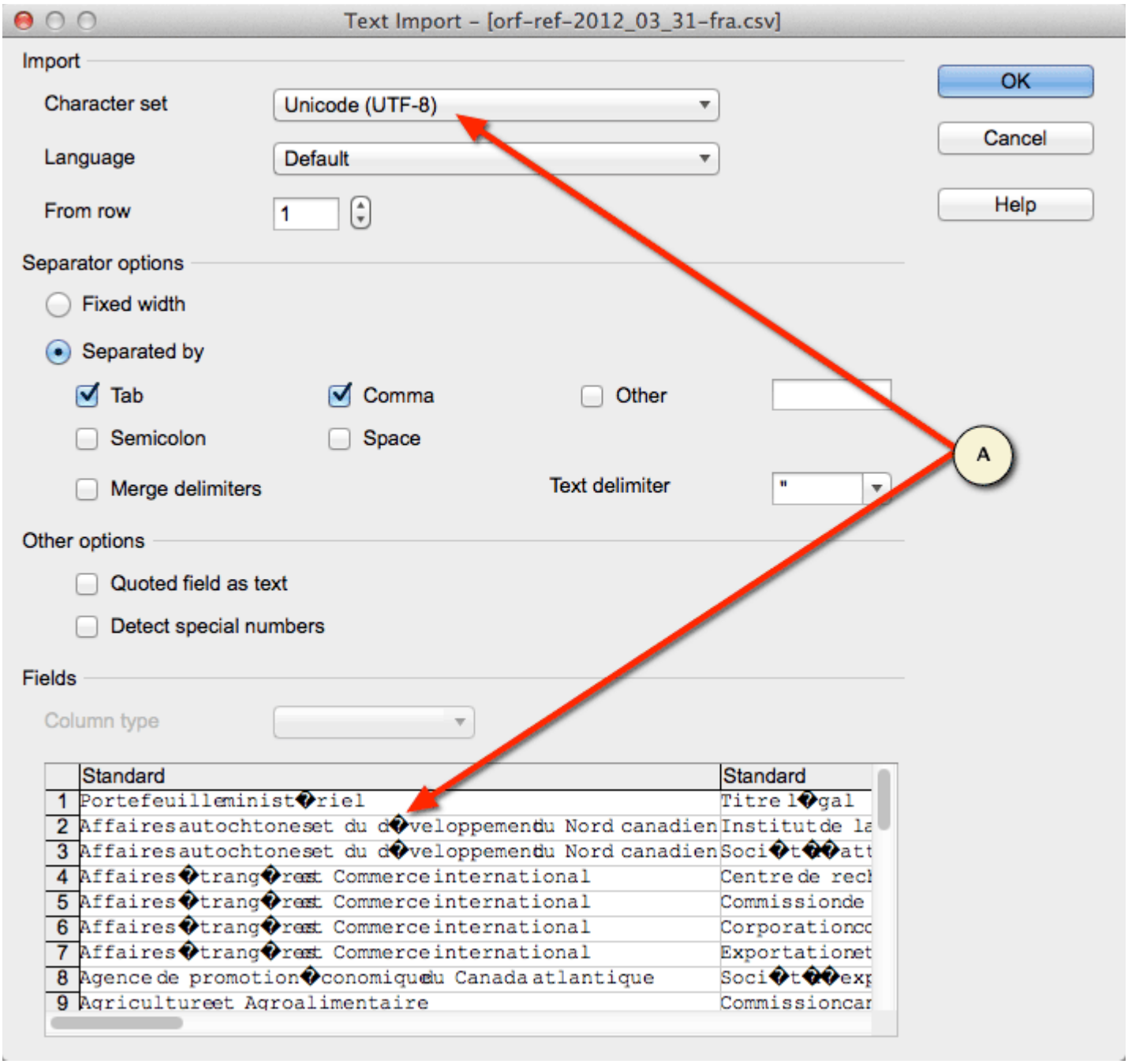


Figure 1: Example of a CSV import wizard showing incorrect encoding

Accessible description of Figure 1

Screenshot showing the comma-delimited file import wizard in Neo Office. The file being imported contains French extended characters. The user has selected UTF-8 as the character encoding. The preview area is displaying the characters incorrectly as it is not the proper character encoding for this file.

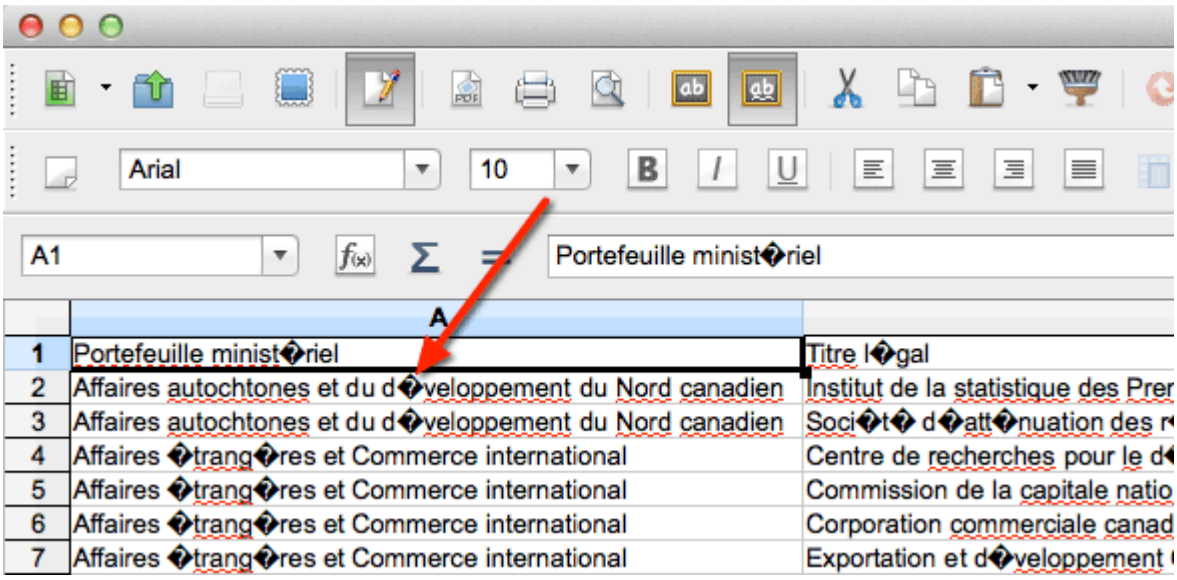


Figure 2: A CSV file imported using the incorrect character encoding

Accessible description of Figure 2

A comma-delimited file displayed in the Neo Office spreadsheet editor. The file contains French extended characters which were not properly imported due to using the incorrect character encoding.

Another common 8-bit character encoding in Canada is ISO–8859–1. In this scenario, selecting the ISO–8859–1 character encoding resolves the extended character issue as shown in Figure 3 and Figure 4.

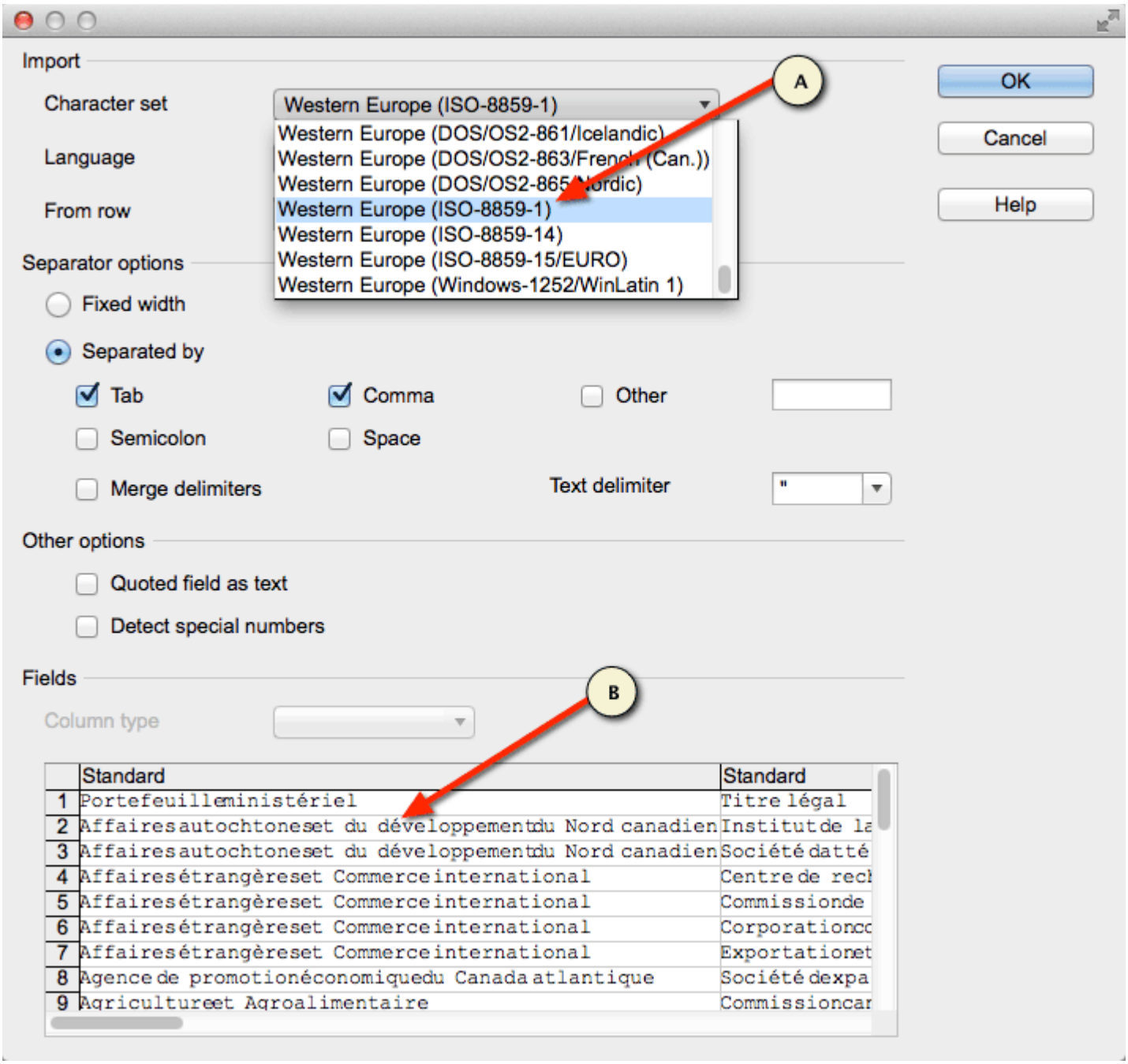


Figure 3: A CSV import wizard showing proper character encoding

Accessible description of Figure 3

Screenshot showing the comma-delimited file import wizard in Neo Office. The file being imported contains French extended characters. The user has selected the appropriate ISO-8859-1 encoding and the preview area is displaying the characters properly.

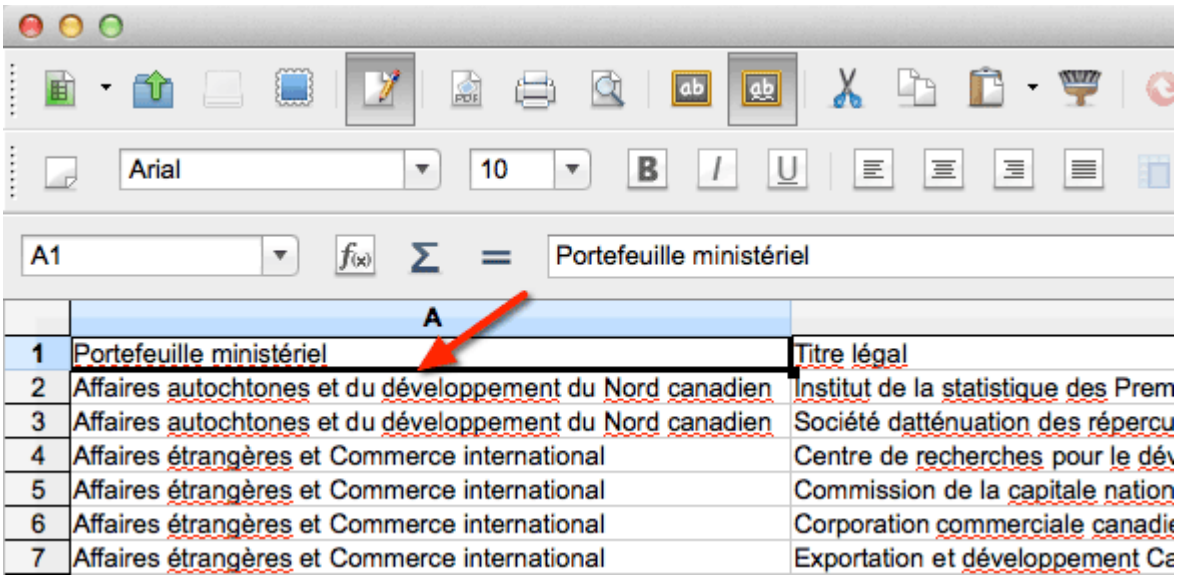


Figure 4: A CSV file imported into NeoOffice using the proper character encoding

Accessible description of Figure 4

A comma-delimited file displayed in the Neo Office spreadsheet editor. The file contains French extended characters which were properly imported using the correct character encoding.

You may notice character encoding issues when opening a CSV file in Microsoft Excel as shown in Figure 5. For this example, we will be using Microsoft Excel in Microsoft Office Professional Plus 2010.

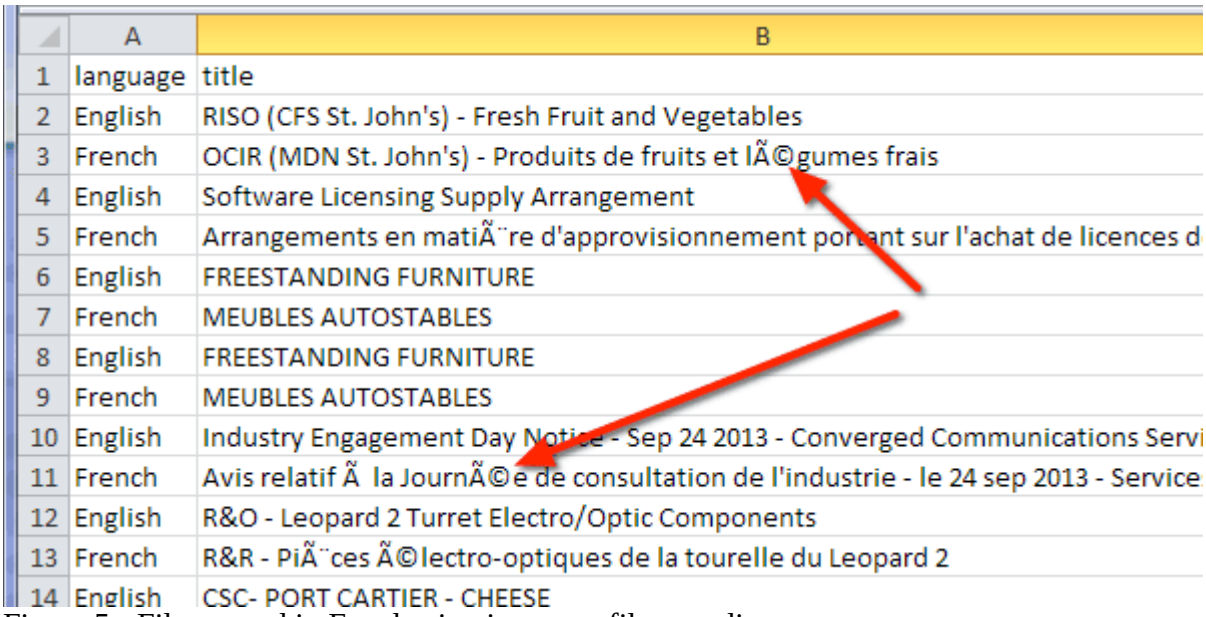


Figure 5 - File opened in Excel using incorrect file encoding

Accessible description of Figure 5

A screenshot of a comma-delimited file opened in Excel showing incorrectly decoded French characters.

The character encoding can be manually selected using the data import wizard. The first step is to start with a new Excel workbook. Go to the Data tab and click on From Text which will launch the import wizard (Figure 6).

The first step in the wizard is to select the appropriate encoding. In this example, we are selecting the UTF-8 encoding which is located near the bottom of the list. Once selected, you can preview the character decoding in the data preview pane of the wizard.

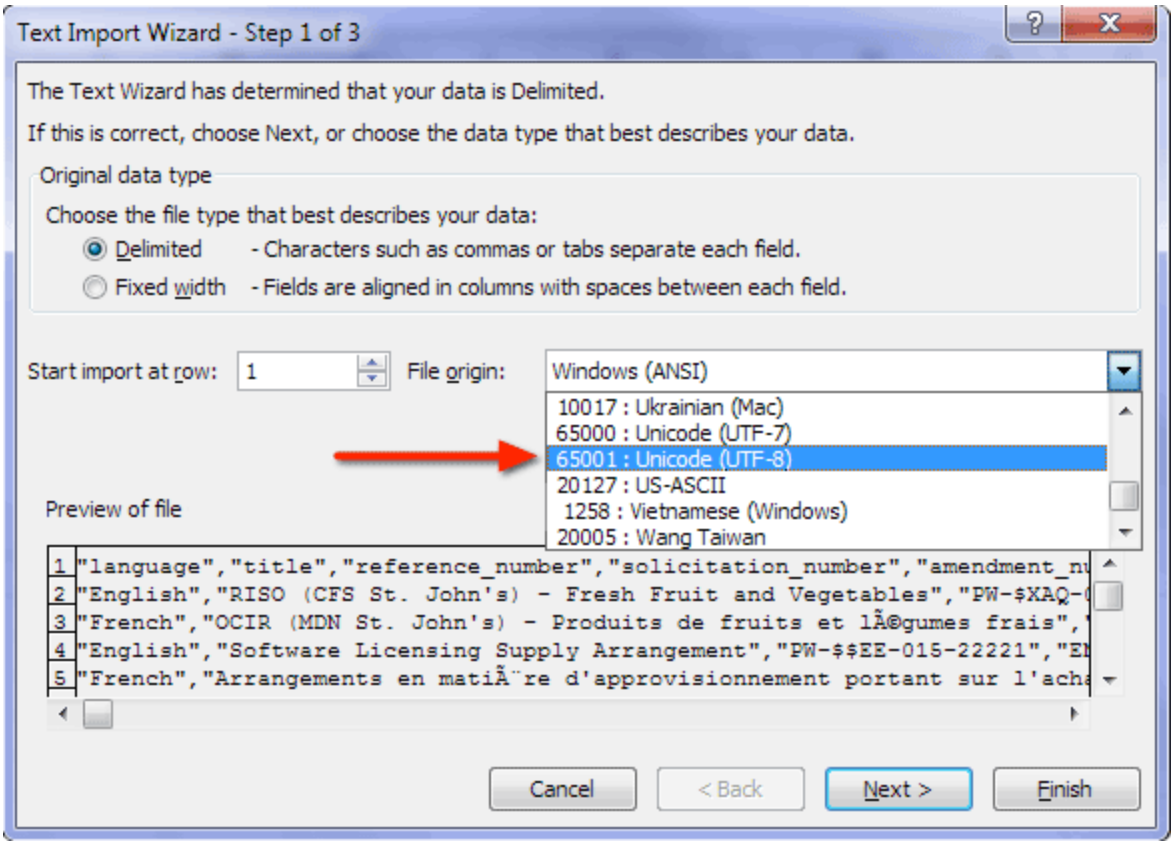


Figure 6 - Selecting UTF-8 in Text Import Wizard

Accessible description of Figure 6

A screenshot showing Step 1 of 3 of the Text Import Wizard in Microsoft Excel. The file origin list box is open with Unicode UTF-8 selected.

The next step (Figure 7) allows you to specify the proper delimiter(s) used in the data file. Once again, the Data preview pane allows you to preview your data and see if it is parsing the data correctly.

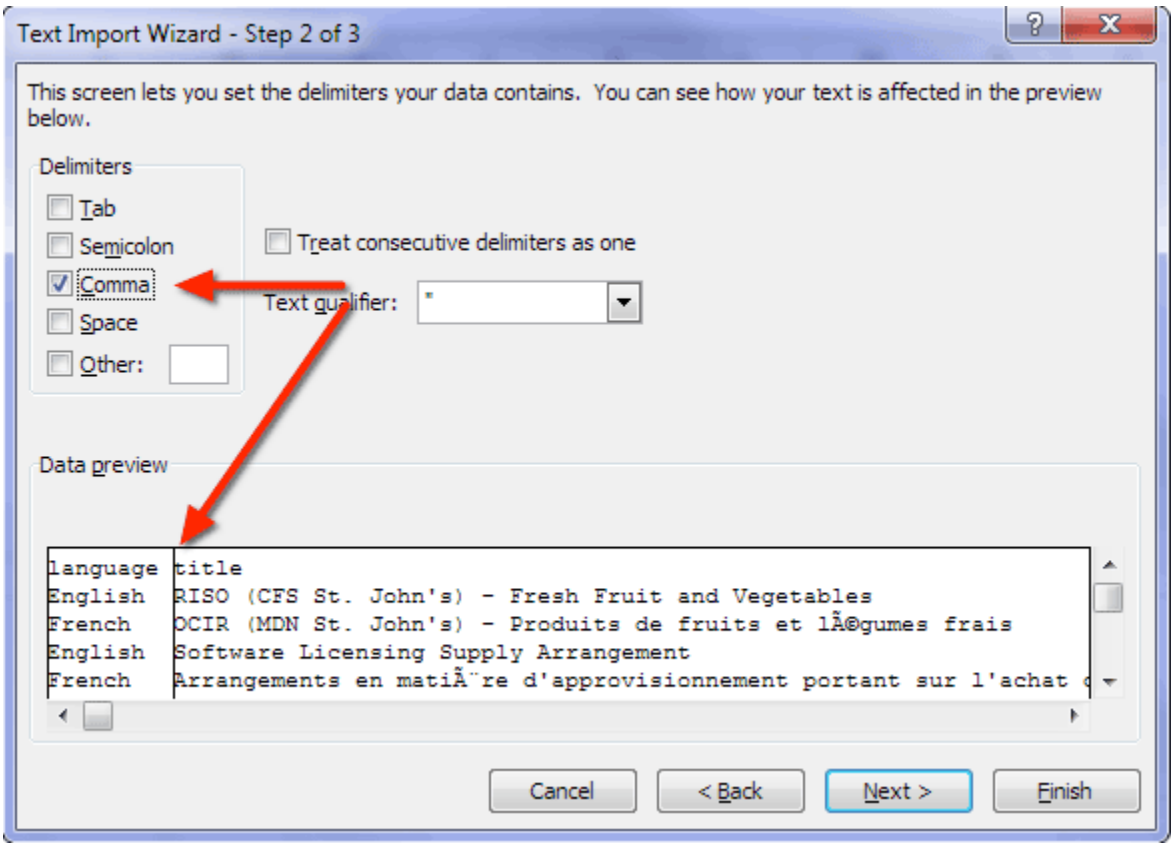


Figure 7 - Selecting delimiters in Text Import Wizard

Accessible description of Figure 7

A screenshot showing Step 2 of 3 of the Text Import Wizard in Microsoft Excel. The Comma delimiter is checked and the data preview pane is showing the data separated into columns.

Optionally, you can specify the data types for each of the columns (Figure 8). In most cases, simply clicking Finish will suffice.

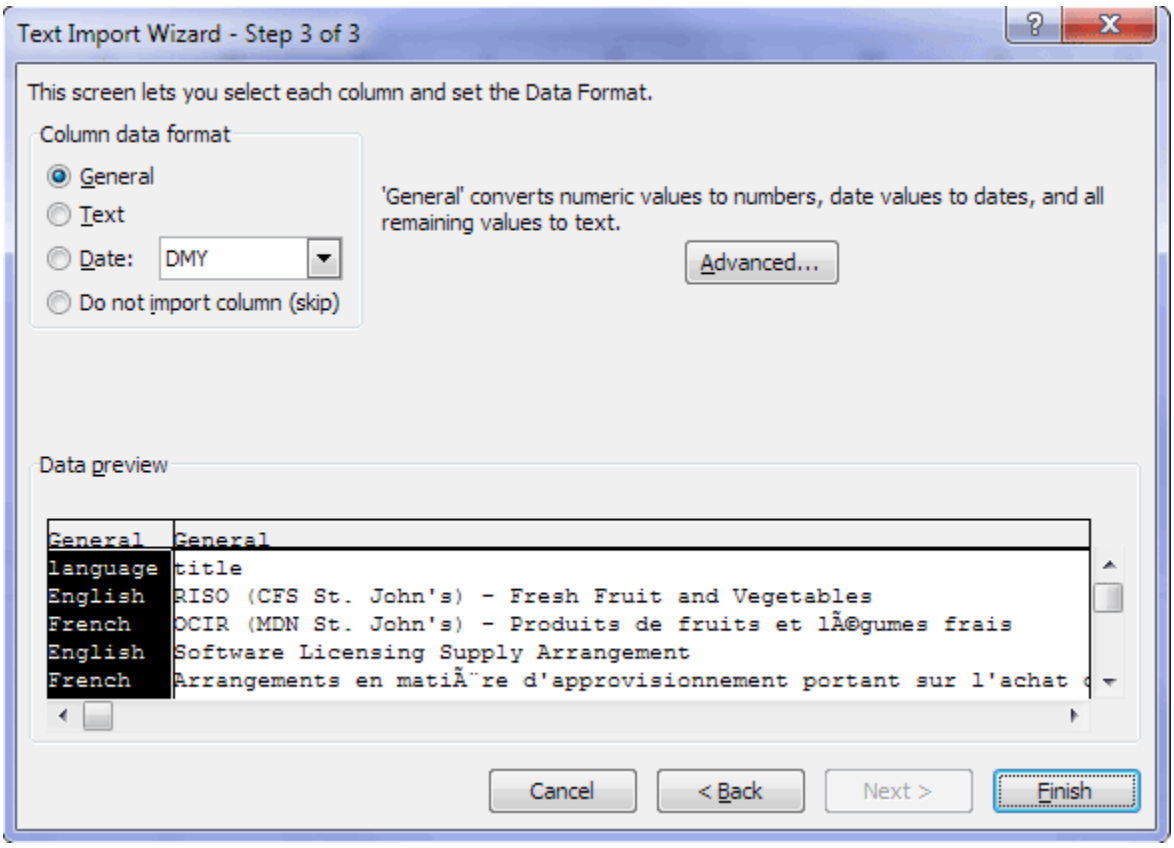
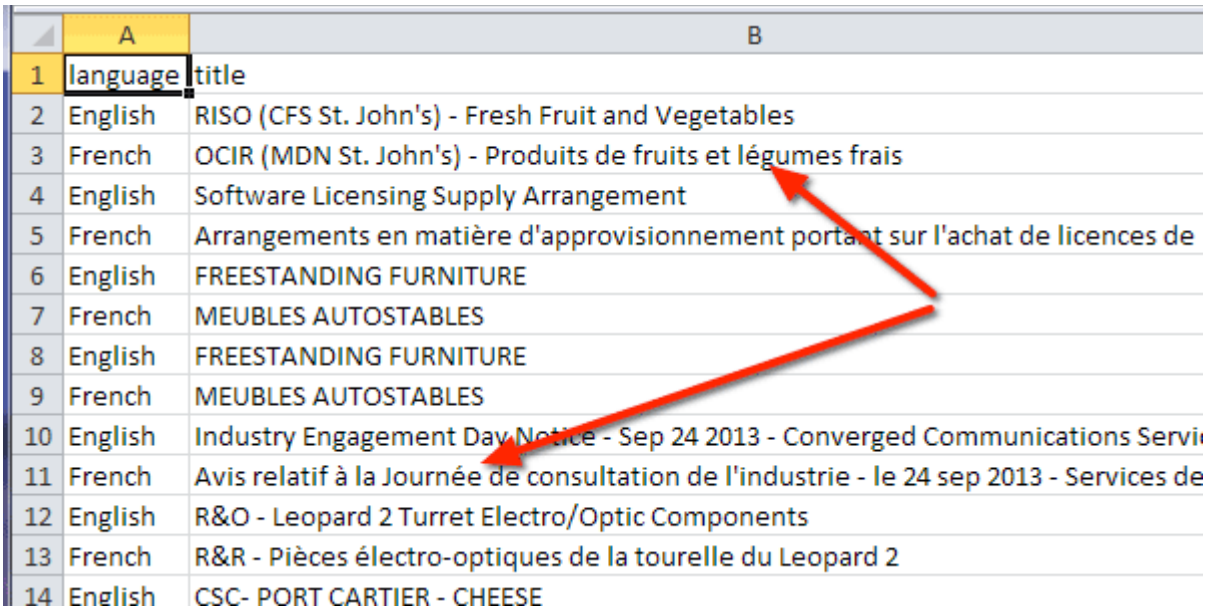


Figure 8 - Selecting data formats in Text Import Wizard

Accessible description of Figure 8

A screenshot showing Step 3 of 3 of the Text Import Wizard in Excel. This step has settings allows the user to set the data formats for each of the columns. The defaults are used for this example.

Figure 9 shows the characters properly decoded in Excel having used the Text Import Wizard.



	A	B
1	language	title
2	English	RISO (CFS St. John's) - Fresh Fruit and Vegetables
3	French	OCIR (MDN St. John's) - Produits de fruits et légumes frais
4	English	Software Licensing Supply Arrangement
5	French	Arrangements en matière d'approvisionnement portant sur l'achat de licences de
6	English	FREESTANDING FURNITURE
7	French	MEUBLES AUTOSTABLES
8	English	FREESTANDING FURNITURE
9	French	MEUBLES AUTOSTABLES
10	English	Industry Engagement Day Notice - Sep 24 2013 - Converged Communications Servi
11	French	Avis relatif à la Journée de consultation de l'industrie - le 24 sep 2013 - Services de
12	English	R&O - Leopard 2 Turret Electro/Optic Components
13	French	R&R - Pièces électro-optiques de la tourelle du Leopard 2
14	English	CSC- PORT CARTIER - CHEESE

Figure 9 - Data imported into Excel using proper character encoding

Accessible description of Figure 9

A screenshot showing a comma-delimited file imported into Excel using the Text Import Wizard. The French characters are shown properly having applied the appropriate character encoding.

Byte Order Mark (BOM)

You may discover an interesting special byte at the beginning of a file. This byte is called the Byte Order Mark or BOM. This marker is used to signal the size of the character encoding used in the UTF encoded file. For example, UTF–16 will render as `U+FEFF` (Figure 5) or þÿ in a text editor capable of displaying this value. In general, you do not need to concern yourself with the BOM as it is used by the application you are using. However, it can be a useful byte to help you select the correct file encoding during an import if you are having issues.

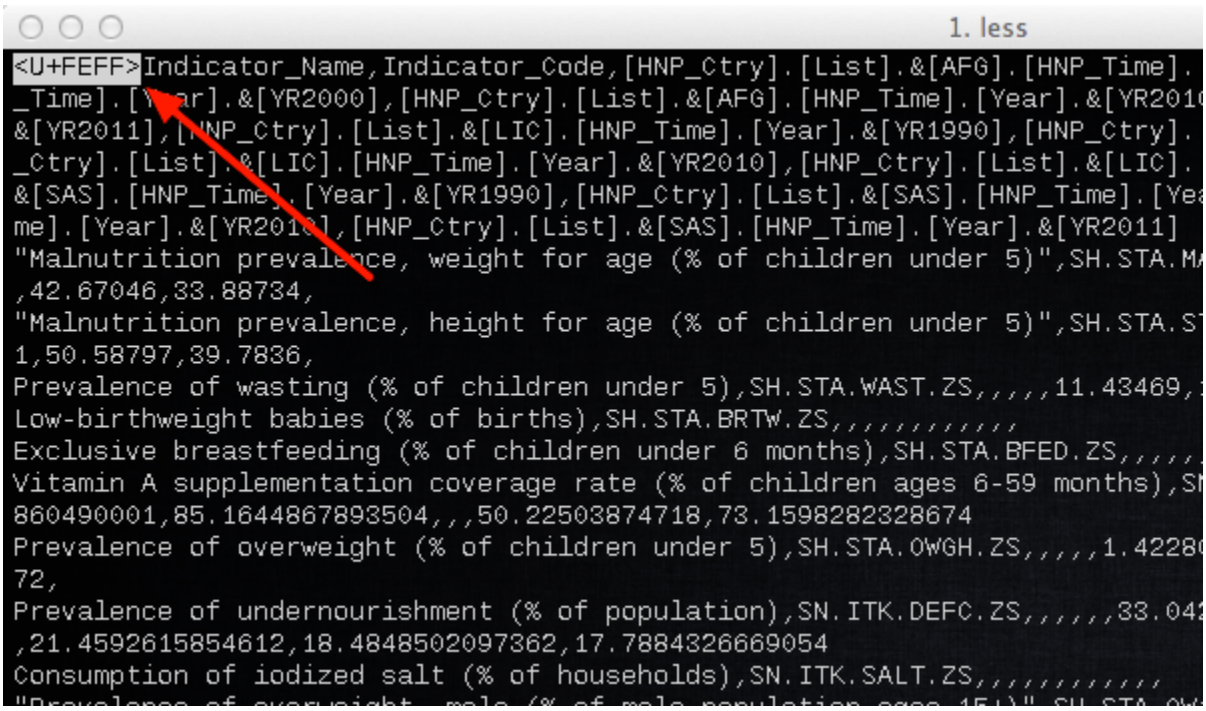


Figure 10: A Byte Order Mark in a data file

Accessible description of Figure 10

A Byte Order Mark displayed in a text editor.

JSON

Java**S**cript **O**bject **N**otation (JSON) is a structured open data format that is lightweight and very well suited for mobile applications. It is used either natively or as a library available in most web and mobile development languages. Detailed information on JSON can be found on the official JSON web site. Here is a simple example of JSON data:

```
1. "metadata":{
2.   "request":{
3.     "dateCreated":"2014-01-01T00:00:00+00:00",
4.     "name": "Example JSON Dataset" },
5.   "items":[
6.     { "name":"Item 1", "colour":"red" },
7.     { "name":"Item 2", "colour":"blue" }
8.   ]
9. }
```

We have lots of great examples of working with JSON data and RESTful APIs in various programing languages.

XML

XML (e**X**tensible **M**arkup **L**anguage) defines a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification.

Geo Formats

GeoTIFF

GeoTIFF files are similar to standard graphical TIFF files; they are viewable using a regular image viewer that supports the TIFF format. However, GeoTIFF files contain extra metadata related to the geographical nature of the image. Refer to the official sites on GeoTIFF for additional information on the format:

- Official GeoTIFF web site
- GeoTIFF Specification

An open source library exists to enable your program the ability to extract the metadata information from a GeoTIFF. Below is an example on how to install and run a simple C program to extract metadata information.

Here we are using the Homebrew package manager on OSX to install the GeoTIFF library and use a utility to read a GeoTIFF:

```
1. $ brew install libgeotiff
2. $ listgeo <filename>.tif
```

Once the GeoTIFF library is installed, you should be able to compile a simple C program to read the metadata:

```
1. /**
2.  * Simple example of a C program to read metadata from a GeoTIFF.
3.  *
4.  * NOTE: This is a simple example with very little error handling.
5.  *
6.  * @license http://data.gc.ca/eng/open-government-licence-canada
7.  */
8. #include <stdio.h>
9. #include <stdlib.h>
10.  #include <xtiffio.h>
11.  #include <geotiffio.h>
12.  #include <geokeys.h>
13.
14.  int main(void) {
15.
16.      TIFF *tif  = (TIFF*)0; // TIFF-level descriptor
17.      GTIF *gtif = (GTIF*)0; // GeoKey-level descriptor
18.      double major_axis_key = 0;
19.
20.      // Open TIFF file
```

```
21.      tif = XTIFFOpen("geo.tif", "r");
22.      if (!tif) {
23.          printf("Unable to open GeoTIFF file.\n");
24.          exit(-1);
25.      }
26.
27.      // Open GTIF Key parser; keys will be read at this time.
28.      gtif = GTIFNew(tif);
29.      if (!gtif) {
30.          printf("Unable to access GTIF metadata.\n");
31.          exit(-1);
32.       }
33.
34.      // Read a GTIF Key pair
35.      if (GTIFKeyGet(gtif, GeogSemiMajorAxisGeoKey, &major_axis_key, 0,
   1)) {
36.          printf("GeogSemiMajorAxisGeoKey: %f\n", major_axis_key);
37.      } else {
38.          printf("Unable to locate GeogSemiMajorAxisGeoKey\n");
39.      }
40.
41.      // Clean up and exit.
42.      GTIFFree(gtif);
43.      XTIFFClose(tif);
44.      return 1;
45.  }
```

Output:

```
1. $ ./geotiffmeta
2. GeogSemiMajorAxisGeoKey: 6378137.000000
```

Shapefiles

Browse Shapefile Datasets

ESRI Shapefiles (SHP) are popular geospatial vector data files which are accompanied by a series of files necessary to use the SHP file. As listed on Wikipedia, the following is usually packaged into a compressed ZIP file:

Mandatory files:

- .shp — shape format; the feature geometry itself
- .shx — shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly
- .dbf — attribute format; columnar attributes for each shape, in dBase IV format

Optional files:

- .prj — projection format; the coordinate system and projection information, a plain text file describing the projection using well-known text format

- .sbn and .sbx — a spatial index of the features
- .fbn and .fbx — a spatial index of the features for shapefiles that are read-only
- .ain and .aih — an attribute index of the active fields in a table
- .ixs — a geocoding index for read-write shapefiles
- .mxs — a geocoding index for read-write shapefiles (ODB format)
- .atx — an attribute index for the .dbf file in the form of shapefile.columnname.atx (ArcGIS 8 and later)
- .shp.xml — geospatial metadata in XML format, such as ISO 19115 or other XML schema
- .cpg — used to specify the code page (only for .dbf) for identifying the character encoding to be used

Source: Shapefile - Wikipedia.org

Links to useful tools:

- Online Shapefile Dump Utility
- Perl Shapefile Extension
- Python Shapefile Library
- C Shapefile Library

Converting Shapefiles to GeoJSON

There is a great write-up by Ben Balter on converting Shapefiles to GeoJSON using gdal's ogr2ogr utility as follows:

```
1. ogr2ogr -f GeoJSON -t_srs crs:84 [name].geojson [name].shp
```