Creating Open Data with Excel

The problem

A large majority of Open Government Data is published as Comma Separated Value (CSV) files. Much of this data sourced from Microsoft Excel. There are number challenges in preparing Excel data for publication on Open Data Portals, including:

1. Staff are not confident in structuring Excel spreadsheets to make it easy to publish.
2. Discovery metadata has to be manually prepared.
3. Metadata about the structure of the data and its validation rules are not published.
4. Data is not comprehensively checked for completeness and consistency before being published.
5. Excel's "Save As CSV" function writes files using ANSI rather the UTF-8 encoding (the dominant character encoding for the World Wide Web).

The opportunity

What if someone could push a button in Excel and prepare their data for publishing on an Open Data Portal and address the challenges above?

Imagine this process:

1. After reading a guideline on how to structure spreadsheets for Open Data, a spreadsheet is created.
2. Each columns is given appropriate Data Type (e.g. Date, Currency, Text, etc.).
3. Columns are optionally given Data Validation rules (e.g. a list of valid values).
4. Data is entered into the spreadsheet.
5. A "Save as CSV Data Package" option is selected to create a web friendly CSV file and an associated metadata file.
6. These files then go through the normal organisational processes to approve publishing the resources on an Open Data Portal.

What are the benefits?

The solution is generic and can be applied to all open data initiatives around the world. The benefits include:

1. Reduced effort in creating metadata saves time for open data publishers.
2. Increased accuracy in metadata saves time for data re-users, helping them quickly determine if it is fit for their purpose and understand its structure and meaning.
3. Improved metadata supports the automatic awarding of Open Data Institute Open Data Certificates – a way to quickly show your data is easy to find, use and share (see Figure 1).
4. For organisations and community members without the ability to publish open data, their CSV Data Package could be easily published to community portals such as Open Knowledge's data portal where not only can the data be accessed but the metadata can be viewed in a user-friendly format (see Figure 2).

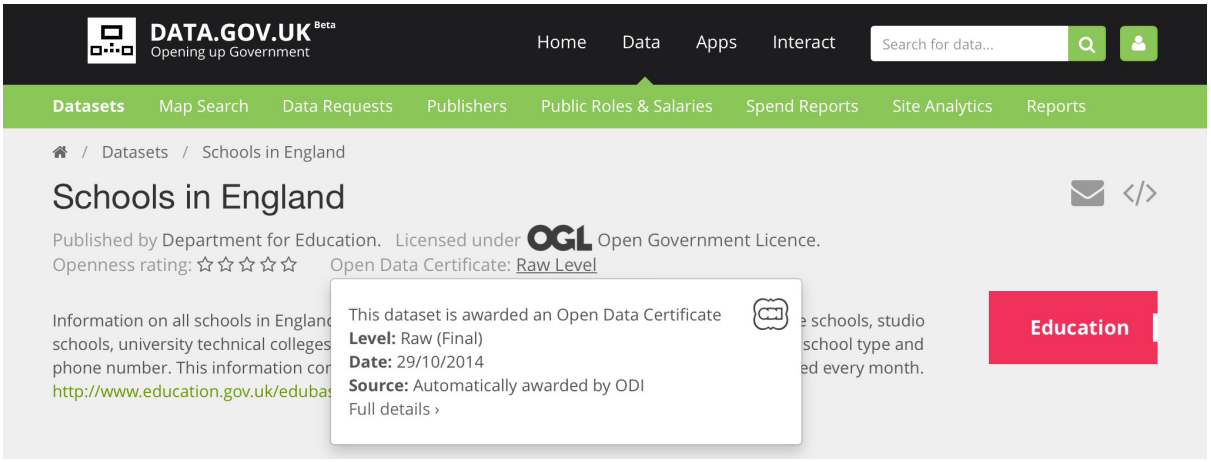

Figure 1 Open Data Certificate displayed in the United Kingdom's data portal, http://data.gov.uk/

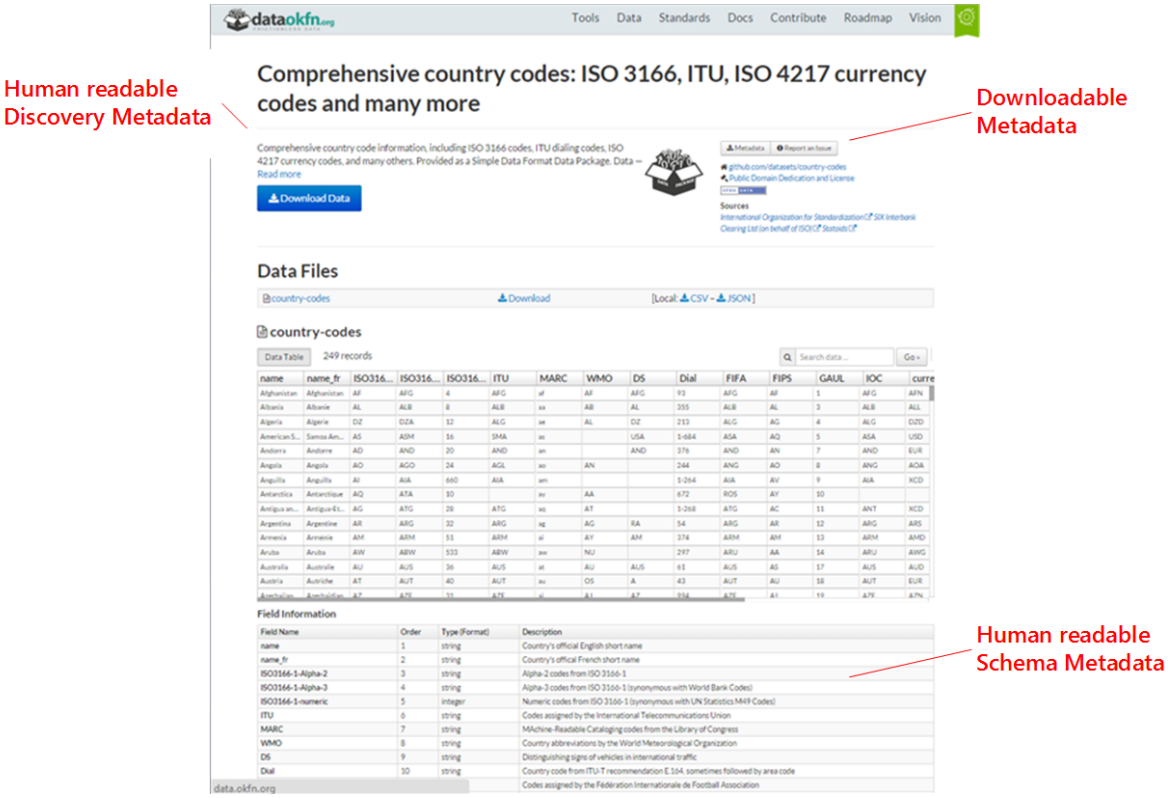

Figure 2 CSV Data Package displayed in Open Knowledge's Data Package Portal, http://data.okfn.org/data

How would it work?

Create a spreadsheet, define column data types and data validation

- This is achieved using standard Excel functionality and resource formatting guidelines (e.g. the Queensland Government's Resource Formatting Guide or the Guide from CSVLint).
- The spreadsheet is saved as an Excel file.
- The user then selects the "Save as CSV Data Package" option that triggers the background process steps below.

Generate discovery metadata

- Read the spreadsheet properties to pre-fill the discovery metadata attributes.
- If any required metadata attributes are missing, prompt the user to provide them.
- Update the spreadsheet properties.
- Store the metadata attributes.

Generate schema metadata

- Read the spreadsheet column data types to derive the schema metadata that describes the structure of the data.
- Where no data types where provided, scan the data to determine if a stronger data type than "general" can be derived.
- Update the spreadsheet column with the stronger data type.
- Store the schema metadata.

Generate schema constraints

- Read the column data validation rules and store the schema constraint.
- If there are no data validation rules, read each column of data to determine if a data constraint can be derived. E.g. If the column only contains, "yes", "no", "unsure", then ask the user if that is a valid constraint and if so:
    - Add that data validation to the column.
    - Store the schema constraint.
- Propose a constraint. E.g. If the majority of values in a column contains, "yes", "no", "unsure", but there are a few exceptions:
    - Ask the user if the exceptions are valid or if they'd like to return to the spreadsheet and correct them.
    - (A nice visualisation of this is CSV fingerprints).

Things to work out

- Metadata standard(s) to follow e.g. DCAT, Tabular Data Schema, W3C work on CSV.
- File format of metadata RDF/XML, JSON, etc.
- Viability of making the code open source so it can be extended (e.g. add new metadata standards and formats, integration to publishing workflows).
- Options to make code easily accessible and integrated into Excel, avoiding the constraints often applied by large corporations with controlled operating environments.