Conversion of Specifications to Markdown

Documentation

**The task**

The Common Specification for Information Packages and the CITS SIARD specification was at first created in Microsoft Word and Google Docs and is now converted to markdown in order to have the specification in a github native format that among other benefits will enable us to:

* manage issue handling with direct referencing
* manage version control and branch future developments
* easily publish the specification on websites

The conversion of the specifications is not an easy task, and there are several methods and tools to use. This document is written by the authors of the CITS SIARD specification. We are new to markdown and this paper is intended to document the method we used for conversion of the CITS SIARD Specification. In best case, it might be useful to authors who wish to let their specifications become github native and in worst case it might mislead them and create the same issues that we dealt with.

**What is markdown?**

[Markdown](https://en.wikipedia.org/wiki/Markdown) is a lightweight markup language with plain-text-formatting syntax. Markdown is often used for formatting readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor.

**Tools**

There are many tools to use when converting files into markdown format. In this instance, the applications “Pandoc”, “Docs to Markdown”, and “Markdown Monster” have been used. Pandoc is a free and open-source document converter, primarily used as a writing tool and as a basis for publishing workflows. “Docs to Markdown” is an add-on tool and converts Google Docs to readable Markdown or HTML.

**Usage**

First, download and install Pandoc here: <https://pandoc.org/installing.html>

To use pandoc: 1. open the command prompt, 2. go to the folder the files are stored in (to get to the root enter "cd /"). Then use the command "dir" to see what is on the level, and "cd xx" to go to xx folder.

In order to run pandoc on a specific document/file, run the command: *pandoc -s example.docx -t markdown -o example.md*

* *Explanation*: The command means, that the application (Pandoc) takes the (-s) docx document "example" and converts it (-t) markdown-language (markdown) and creates a new file (-o) named "example.md". Alternatively, it is possible to convert to Github Flavored Markdown (GFM). Then type the command: pandoc -s example.docx -t gft -o example.md

**Issues locating the right file**

It is not possible for pandoc to identify the document when spaces are used in the name.

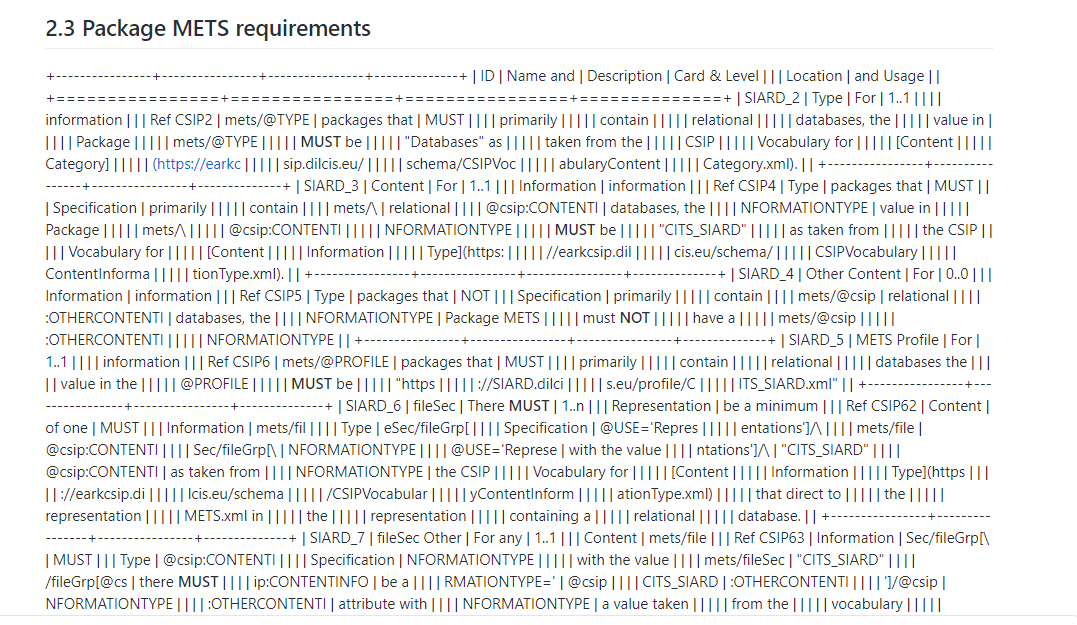
* *Solution*: One solution is to put ”” around the name of the file being converted. Example: pandoc -s “example with spaces.docx” -t markdown -o “example with spaces.md”
* Otherwise, swap the spaces with underscores (or other characters) and edit the name of the document you want to convert.

**Issues with tables**

When converting the file to normal Markdown the tables cannot be read on Github. Luckily, Pandoc can also convert files to Github Flavored Markdown (GFM). If for some reason, it is not possible to convert to GFM, the program "Docs to Markdown" can be used. Its table conversion is readable on Github. The MD file can then manually be updated by inserting the "Docs to Markdown" tables.

There is another issue with tables in relation to **Github Pages**. Tables (created either in GFM or normal Markdown) are created in a version that cannot be read/viewed on Github Pages.

* *Solution*: Use the application "[Markdown Monster](https://markdownmonster.west-wind.com/)" program to manually alter the tables from grid- or pipetable to html. Paste these into MD file. Then it can read / view on Github Pages. This was the only solution I could come up with.

 Illustration 1: Example of table-error on Github

**Issues with images**

In order to display images in markdown, the images must be uploaded to the repo and placed in a folder. In the MD-file, you must refer to where the image is be "taken" from. This is done by the following command: ! [] (../ figures / example.png)

* *Explanation*:”! []” Means it is an image. “(../figures/example.png)” means that the image "example.png" is located in the "figures" folder. REMEMBER to name the images correctly when adding them to the repo

**Other/Overall/general issues**

* There may be problems with **headlines** not being the same. The reason for this error is not known.
* *Solution*: The fastest solution is/was to correct these by hand.
* Problems with **division of words**. Several times in the file, wrongly created spaces were created during conversion – sometimes splitting words in two. The reason(s) for this is/are not known.
  + *Solution*: The errors were corrected by: 1. reviewing the MD file in Github, 2. noting errors, and 3. "Find and replace" function in Notepad ++. (I used GitHub Desktop).